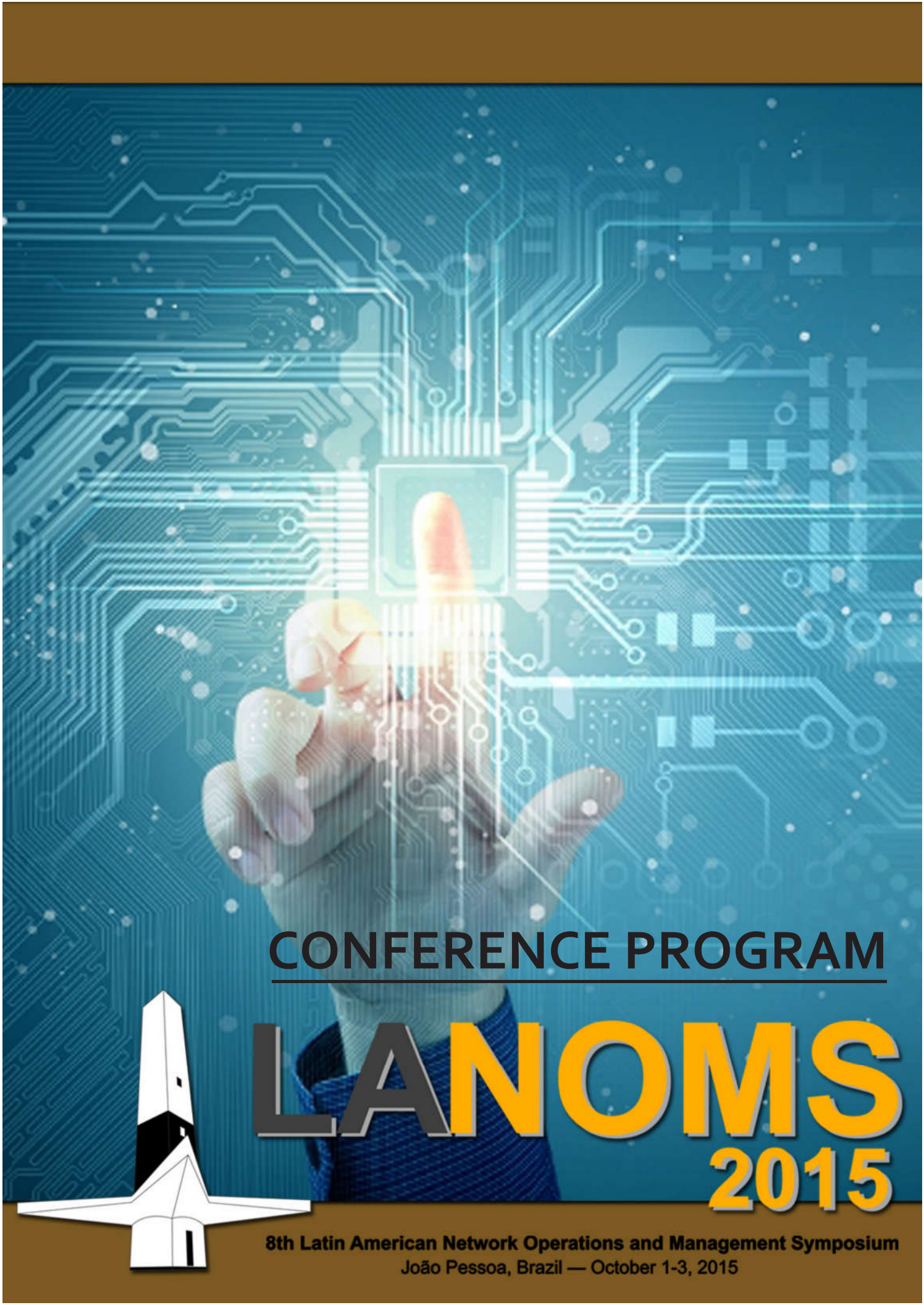


CONFERENCE PROGRAM

LANOMS 2015

8th Latin American Network Operations and Management Symposium

João Pessoa, Brazil — October 1-3, 2015

It is our great pleasure to welcome you to the 8th Latin American Network and Operations Management Symposium (LANOMS 2015), sponsored by IEEE, IFIP and Computing Brazilian Society (SBC). LANOMS is a premier conference in the field of network and services management, and for the first time being held in the highly touristic Northeast region of Brazil, in the city beautiful city of João Pessoa. We hope this conference will be a great opportunity to share current and future hot research trends amongst researchers from around the world.

We are proud to highlight that LANOMS 2015's organizing committee put together a rich and varied program, including 4 technical sessions, one application sessions and one poster session. The program also includes two high-level keynote speakers, with vast expertise in their respective fields. Finally, it also promotes a social agenda to offer participants the opportunity to visit the beautiful city of João Pessoa and its wonderful coastline.

We are most grateful to the authors who submitted their work, the technical program committee members who have timely supported the peer-review process, the organizing committee members who have all worked hard on the details and important aspects of the conference, and finally, but no least, our sponsors who helped us to cope with the costs for the LANOMS 2015 organization, seeking to ensure a high-level event for all participants.

We are sure that all you attendees will enjoy the beautiful and pleasing city of João Pessoa, notable for its beaches and natural environment.

ISBN: 978-1-4673-9408-6

Table of Contents

Session 5 – Poster Session

Session 6 – Application Session

HotOM: A SDN based Network Virtualization for Datacenters

Lucas R. B. Brasilino, Kelvin L. Dias
{lrbbs,kld}@cin.ufpe.br
Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Anibal Fernandes, s/n – Recife – PE – Brazil

***Abstract*—Datacenters are in a process of a huge shift by the deep adoption of virtualization. To be economically doable, virtualized datacenters must reach a high level of resource utilization by simultaneously hosting as many tenants as possible. Traditional network virtualization techniques do not scale to accomplish those requirements. Furthermore, new demands are arising like programmable networks leveraged by the Software-Defined Networking (SDN) paradigm. This paper introduces HotOM, a network multi-tenancy approach to overcome traditional limitations through redefining a L2 header purpose *and* introducing a new L2.5 shim protocol, preserving legacy network core's device. Functional explanation and evaluations were done to support the proposal.**

I. Introduction

Datacenters and networking, despite been treated distinctly for years, now have reached a need for joint advance in pursuance of addressing the huge and complex demands in virtualized environments.

In order to circumvent the hardness of traditional network technologies, a new paradigm has attracted attention in recent years: *Software-Defined Networking* (SDN). SDN introduces an architecture where network's control plane is separated from data forwarding plane and placed in a centralized controller. With the introduction of OpenFlow[1], the most prominent SDN architecture nowadays, a myriad of cutting-edge applications can be built by changing network's behavior through *programmability*.

Additionally, virtualized datacenters are being challenged on how to effectively step forward and push virtualization into network field while properly managing the related resources. Many solutions have arisen, some of them are widely supported by industry but not always benefit from SDN paradigm. *Virtual Extensible LAN* (VXLAN) and *Network Virtualization using Generic Routing Encapsulation* (NVGRE), for instance, are based on multicast and GRE, but they are not SDN applications *per se*. Both work by encapsulating L2 frames into L3 packets, thus expanding protocol overhead, squashing data payload size and increasing fragmentation.

Investigations on how to better lead a SDN adoption in datacenters were done recently. Some of them argue that the best way is by deploying edge SDN-aware access switches, while preserving legacy devices in core's network[2]. In an already established and operational datacenter, we believe that this is the better approach because physical topology was already designed and (legacy) network devices were bought, configured and in operation. Demanding datacenters to discard their switches and replace with new SDN-aware ones will surely postpone SDN deployment to the next investment's cycle, and it can take years.

978-1-4673-9408-6/15/$31.00

With those facts in mind, this paper presents **HotOM**[1]: a proposal that redefines the purpose of a L2 field, employs address translation mechanisms and introduces a new L2.5 protocol header to implement network virtualization on datacenters by leveraging SDN. It was designed to allow datacenters to host a large number of virtual networks (VN) and virtual machines (VM), while keeping legacy switches and their managing systems, so preserving OPEX and lowering CAPEX.

At this time of writing, HotOM is a *prototype*, designed to prove its usefulness in terms of *VN instantiation*, *VM connectivity* and *network isolation*. It uses OpenFlow as its enabling technology but, looking ahead to Sections III-F and III-G, it does not meet all HotOM's requirements. To circumvent this OpenFlow's deficit, a non performance prone design was taken, introducing throughput penalty. This issue is discussed later on, as well as some feasible solutions.

This paper is organized as follows. Section II discuss the background about virtualized datacenters and networks. Section III describes HotOM and its mechanisms. Section IV lists hardware and software resources used in an early implementation. Evaluation results are demonstrated in Section V. Section VI discuss some related proposals. Finally, Section VII discussm HotOM results and concludes the paper pointing some new potential investigations.

II. Background

Modern datacenters are buildings that houses a range of computing, connectivity and storage resources. In early days, services were available by allocating dedicated servers and disks to a service and sharing network without proper traffic isolation. As result, datacenters have terrible server utilization, large disk-stored data fragmentation, high power consumption and, thus, huge operational costs.

In order to lower the wasting of resources, a *virtualized datacenter*[3] paradigm was introduced by leveraging benefits from virtualization. With it, slices of resource's providers like servers, storages, switches, routers and links could be allocated for many customers/users. Moreover, the more resource's slices

[1] **HotO**at**M**eal is a popular saying in Brazil alluding that complex problems should be attacked first by the edges.

are hosted simultaneously, the better resource utilization, lower operational cost and higher profitability are achieved.

A virtual network (VN) is a instance (share) of network resources such as links, switching, routing and bandwidth that lays over a physical network. Virtual switches (VS) and virtual routers (VR) typically maintain a number of isolated forwarding and routing decision tables, as well as a set of virtual interfaces that are mapped to physical ports. Although datacenter server's hosts are well supporting virtualization, network core's forwarding devices are not. Networks are frequently built upon legacy switches that support simplistics 802.1q[4] VLANs to instantiate VNs to tenants, thus suffering scalability constraints. Those switches forwards Ethernet frames merely based on both destination MAC and VLAN tags.

New network's requirements are driven by modern virtualized datacenter. Virtualization at L2 is highly desirable, allowing tenants to deploy their own MAC address' namespace, hence choosing their own routing protocols and schemes. Furthermore, VNs must span beyond server's boundaries, connecting VMs placed in different hosts that can be geographically spreaded around the globe *and* the 4K VN's limit imposed by VLAN must be overcome, once this VN's quantity isn't feasible anymore.

With those requirements in mind and to provide a network service that encompass them all, two possible options must be considered[2] *(a)* network's core should support a given (new) VN technology; or *(b)* the VS should somehow bypass the core's limitations by hiding the (new) VN technology through its *encapsulation in a well-known protocol*[5]. HotOM employs the latter technique.

III. HotOM

The proposal presented by this paper, HotOM, leverages network virtualization and programmability on datacenter's network in a pragmatic way. HotOM main objective is fairly simple: achieve high network resources utilization while allowing flexibility and diminishing costs, thus maximizing datacenter's profitability.

The main objective was divided in goals, designated as **G*x***. They are - **G1**: work on network core's legacy switches; **G2**: lay on a small but effective L2.5 shim protocol header; **G3**: provide complete L2 and therefore L3 network isolation to VMs; **G4**: be L3 routing agnostic; **G5**: provide scalability far beyond the 4K VLAN's constraint. The rationale behind each goal will be discussed along with HotOM's aspects.

The architectural aspect of HotOM was designed to fit flawlessly into traditional datacenter's organization. This means that a migration process from former technologies to HotOM is made in a smooth way.

In traditional datacenters, hosts are connected to network's core through a physical switch called *access switch*. In turn, the access switches are connected to a upper layer called *aggregation switches* (some literature calls them *distribution switches*). In HotOM, access switches are pushed down to virtualization layer and are called *access virtual switches* (AVS). It is AVS' responsibility to provide VN abstraction and VM connectivity by applying HotOM mechanisms. The aggregation layer is

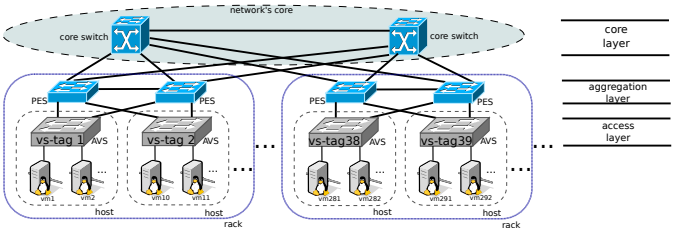


Fig. 1: HotOM datacenter topology

provided by *physical edge switches* (PES), which are in charge of connecting AVS' (and hosts) to network's core (Figure 1). Each key aspect of HotOM architecture is discussed in further subsections.

A. VLAN ID based forwarding

On the datacenter's network core point of view, HotOM was designed to take advantage of fast switching capabilities provided by L2 technologies. So, HotOM uses frame's VLAN ID tag as an unique key (an index) in a forwarding decision. To each AVS is assigned an unique VLAN ID tag called *vs-tag*.

The reason for using VLAN ID for routing is to accomplish goals **G1**, **G4** and **G5**. This forwarding mechanism demands simpler L2 devices. **G4**, in particular, avoids the employment of expensive L3 switches. Moreover, **G4** also permits tenants to use arbitrary L3 protocols other than IP, turning HotOM into an open technology for new protocol stacks above L2.

Current switches forwards frames based on VLAN ID tag *and* destination MAC address. HotOM creates a possibility for developing a switch that forwards frame *only* based on VLAN ID tag. This kind of switch would be much cheaper than current ones, because it might use a very small Content-Addressable Memory (CAM) portion, just to store [`tag` $\rightarrow$ `port`] entries.

Using VLAN ID tags as an index to identify AVS has a drawback. They are a 12-bit long header field, so it is possible to address up to 4096 AVS'. HotOM architecture overcome this problem by reserving two *special purpose* vs-tags and the remaining 4094 AVS' are grouped in a logical arrange called *cluster*. Moreover, supposing that each rack has 40 hosts, each one running its own AVS, a cluster would be a set of about 102 racks.

The first special purpose vs-tag reserved, number 1, is used to tenants reach public IP address space, i.e., to send traffic to the Internet. This means that the border router, the one that interconnects the datacenter to the Internet, is connected to the AVS which vs-tag is 1.

The second special purpose vs-tag, number 2, is used to reach a VM placed in another cluster. For instance, if AVS' vs-tag 2 is a dedicated switch with 48 ports, the entire datacenter would host 196.512 AVS' (48 times 4094). Special purpose AVS' and clusters provides a high level of scalability in compliance with goal **G5**.

B. MAC address translation

As discussed in Section III-A, HotOM implements routing based on VLAN tags. But actual legacy L2 switches forwards

frames based on tag *and* destination MAC address. For this reason, a MAC address translation must be performed. When a local VM sends a frame to a remote one (i.e. to a VM connected to another AVS), the frame's source address is translated to the local AVS' MAC address, while the destination address is translated to the destination AVS' MAC address. In addition, the destination vs-tag is inserted into 802.1q header.

There is a key advantage on using MAC address translation mechanism: physical switches (both PES and core) are not aware that a number of VNs are in place. They just deal with the AVS' vs-tag and MAC addresses. For example, if there are 48 VMs in a traditional Ethernet network connected to a virtual switch, each physical network device would populate 48 similar forwarding table's entries as `[(VM_MAC) → port]`. When using HotOM, only *one* entry as `[(vs-tag,AVS_MAC) → port]` would be created. This is an important factor to maintain a low pressure over physical switches' forwarding table (CAM) size and allows the employment of simpler and cheaper network devices, as goal **G1** requires, while providing a fast table lookup and, thus, switching speed.

C. Gratuitous ARP

As discussed in Sections III-A and III-B, an ingress frame from a VM has its destination MAC translated to the destination AVS address plus an added vs-tag. In order to correctly deliver a frame, physical switches must *learn* through which port an AVS is reachable by adding entries to their forwarding table. To accomplish this, HotOM uses Gratuitous ARP (GARP) broadcasts to force switches to set L2 "routes" (forwarding table's entries).

GARP is a special crafted ARP broadcast packet sent, from time-to-time, by the AVS to *announce* its vs-tag and MAC address. Every physical switch on the network, starting by the PES, receives that GARP packet on a port, adds an `[(vs-tag,AVS_MAC) → port]` entry in their forwarding table and then floods the packet to others physical switches throughout their ports. Using this mechanism, all switches learn paths to every single AVS.

Being a broadcast, GARP tends not to be scalable. It should be done less frequently as possible. Typical physical switches hold for 300 seconds forwarding table entries in their CAM. This time is configurable and should be increased when using HotOM. So, sending GARP frames a little less than this time frame is sufficient to populate the desired entries while avoiding network congestion.

D. HotOM Protocol Header

To achieve high scalability in terms of VN's quantity, some mechanism to add a layer of indirection is needed. This indirection defines orthogonal address *namespaces*. So, it is possible to run many different VNs over the same physical network using the same L2 and L3 addresses without overlapping. In turn, another desired property is to identify a huge amount of VMs on each VN. These are the main purposes of the *HotOM Protocol Header*.

The HotOM protocol header is a L2.5 shim one placed as a VLAN payload in a frame that transverses the network's core. It is composed by four fields, three of them are 24-bit length and one is 8-bit length. They have a Ethernet-like semantics.

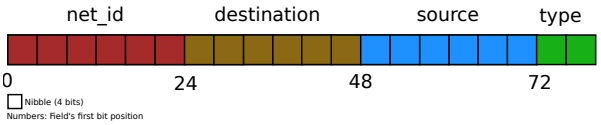


Fig. 2: HotOM Protocol Header's fields

Figure 2 depicts field's placement within the header. The first field is the *net_id*, being an *index* that identifies which VN a given flow belongs to. It is represented as a hexadecimal value. For instance, a value like `0x00FFBA` defines a VN. Since this field's length is 24 bits, a datacenter can instantiate up to 16.8M VNs, far beyond the 4K VLAN's limit. Summarizing, *net_id* applies the desired layer of indirection *and* identify the VN.

The following two fields are *HotOM **addresses*** and they accomplish the other HotOM Protocol Header's purpose: VM addressing. The second field is the HotOM *destination* address, associated with the destination VM MAC address, and the third is the HotOM *source* address, also associated with the source VM MAC address. The field's values are the last 24 bits from VM MAC addresses. Thus, each VN can address up to 16.8M of VMs. Similar to Ethernet, HotOM represents these addresses as three colon-separated hexadecimal bytes. For example, values like `00:00:01` or `AD:DA:FF` are valid HotOM addresses.

The fourth field is the *HotOM **type*** field. This field is a simple mapping from original EtherType. Since this field is 8-bits length, it is possible to map 256 different types of Ethernet frames. This number is sufficient for a real datacenter's deployment because only a few L2 payloads, like IPv4, IPv6, AoE, HyperSCSI and FCoE, are effectively used.

HotOM assigns VM's MAC addresses in a predefined way. The first 24 bits, known as *Organization Unique Identifier* (OUI), *should be* equal to `00:00:00`. This definition eases VN's migration from another virtualization platform. For instance, VMWare uses `00:50:56` as OUI, Hyper-V uses `00:15:5D`, Xen uses `00:16:3E` and VirtualBox uses `08:00:27`. Using a simple script it is possible to convert this 24-bit value to `00:00:00`. The last 24 bits of VM's MAC address, just like any other platform, are freely chosen as desired. This policy is enforced twice by (*i*) a VM managing system in charge of creating VNs and VMs and (*ii*) the Local Agent Service (later discussed) when it replies ARP request. The VM managing system is not discussed in this paper, since it isn't its focus. Finally, HotOM addresses are designed by the tenant, achieving goal **G3**.

HotOM protocol header field's length were chosen to fulfill two aspects. First, some widely accepted network virtualization technologies, like VXLAN and NVGRE, uses a 24-bit field as a VN identifier. So, the authors of this work realize that 16.8M VNs are a extensive number enough, even for the largest datacenters. Second, understanding that Ethernet address are 48-bit long and desiring a fast translation between it and HotOM address, it is doable to use the remaining 24 bits for VM addressing. These decisions were taken to meet goals **G2**, **G3** and **G5**.

Network forwarding is performed in a switch, no matter if it is physical or virtual, by executing many repetitive tasks

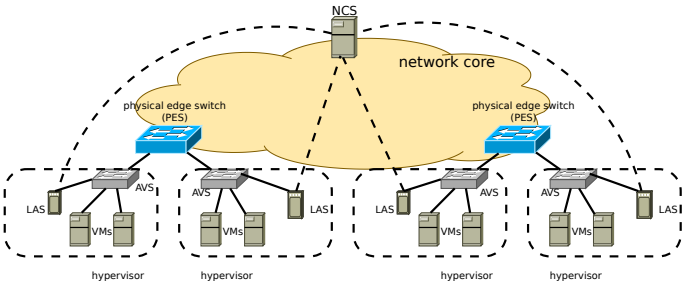


Fig. 3: HotOM architecture

on each packet in a flow. These tasks might be designed to be as fast as possible or data movement would have a poor throughput. With this in mind, the address translation between VM MAC address and HotOM address can be done quickly. The task is simple. Considering `Ethernet[0:95]` as the bit array of destination and origin VM's addresses in an Ethernet header, destination and origin HotOM addresses would be `Ethernet[24:47]` and `Ethernet[72:95]`, respectively. These operations are easy to do in software, by a virtual switch, as in hardware, by a dedicated ASIC or FPGA.

Finally, the concept of locality of a VM into the network topology is defined by the vs-tag, since it identifies which AVS a VM is connected to. The concept of identity, in turn, is defined by the virtual network identifier (net_id) and the HotOM addresses, i.e., the net_id *and* the HotOM address when grouped together turns into a tuple that are, by definition, *unique across the entire datacenter*.

E. Network Coordinator Service

The HotOM Network Coordinator Service (NCS) is a central service that is in charge of tracking, computing and maintaining the VNs topologies, managing which VMs are connected in a particular VN and their resource usage. So NCS has a "broad view" of the physical *and* virtual networks.

Additionally, NCS maintains the primary network database, applies AVS' and vs-tag relationship, stores VMs data (MAC and IP addresses, AVS port), VNs data (net_id, VMs, AVS'), and so forth. Also, it is in charge of replying Local Agent Service (LAS) queries about VN and VM's information. This service will expose an API to install forwarding rules to AVS' through LAS. Those rules can define, for instance, a load-balancing, firewall, router and others network services. Figure 3 depicts the relationship between HotOM NCS, LAS, AVS' and VMs running in a hypervisor.

F. Local Agent Service

The Local Agent Service (LAS) is indeed the main HotOM's OpenFlow application. It typically runs on hypervisor's host. Since in HotOM architecture AVS' are OpenFlow-enabled switches, they have a related controller executing LAS code.

Further, LAS manages local, i.e. in-host, information of VMs and VNs. A local cache table stores the VM name, AVS port, MAC and IP addresses, and the VN it belongs to. Another table caches remote VM information like destination AVS' vs-tag and MAC address. All those data are provided by the NCS.

Finally, due OpenFlow limitations, looking ahead Section III-G, the current LAS implementation are in charge of applying/removing vs-tags (Section III-A) and adding/removing L2.5 shim protocol header (Section III-D) to each packet in a flow that are transmitted from/to VMs. Also, LAS demands AVS to broadcast GARP packets from time to time (Section III-C). Since every packet must be transmitted to an OpenFlow controller (a userspace process) that executes the LAS code, there is a associated performance penalty. Again, HotOM is in a *prototyping* stage to investigate its usefulness on VN instantiation, VM connectivity and network isolation.

G. OpenFlow applied to HotOM

Although OpenFlow introduces flexibility through programmability to network field, it fails badly on providing a way to introduce arbitrary protocol headers in flows while accomplishing performance. It basically works by creating a set of matching-action table's entries in dataplane (switch), where the available basic actions are[6]: (*i*) DROP, to drop a flow; (*ii*) FORWARD, to forward a flow to controller, output port or all ports (flood); (*iii*) ENQUEUE, to associate a flow to a QoS queue; and (*iv*) MODIFY-FIELD, to add/remove VLAN tags and to translate L2 and/or L3 addresses.

OpenFlow can add a rule in AVS matching frames from a VM, adding a VLAN ID tag and translating MAC addresses. But it *cannot* add or remove a HotOM Protocol Header. So, the current HotOM implementation faced a trade-off between functionality and performance: the LAS had to be in charge of inserting/removing the cited header (function), but with a drawback of inserting delays on every frame (performance penalty). The authors of this paper foresee some possible solutions for this shortcoming that are discussed later on in Section VII.

H. ARP query

In traditional virtualized datacenter network, VM's ARP queries are sent throughout the entire network, populating a huge amount of forwarding table's entries among switches. Physical switches can typically hold up to 8K MAC addresses in their tables. If they are exposed to a number higher than that, they have to constantly *learn* new MACs by flooding the ARP query out through all ports, wait for response, search and evict older entries, and insert the new one. All these operations imposes scalability problems into the network.

In a HotOM-enabled datacenter, the LAS is responsible for capturing, parsing and replying ARP queries from VM based on information retrieved from NCS and available in its local cache table. This means that no VM's ARP queries are flooded to network's core, helping to maintain low resource's allocation (CAM) and processing power in physical switches.

I. Physical infrastructure

HotOM does not stick to any particular topology. It can be used in an already designed datacenter network. HotOM focus its functionality on network edges by the employment of OpenFlow-enabled VS'. Nevertheless HotOM being topology agnostic, it is recommended to use a hierarchical network one due its ability on structurally distribution and assignment of vs-tags to AVS', easing maintenance and management.

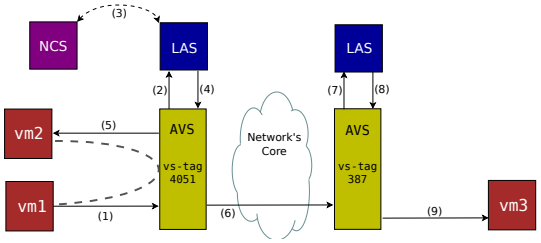


Fig. 4: Packet's life

Figure 1 depicts a network topology and a vs-tags distribution's example.

HotOM requires only an easy physical switches' configuration: every link must be configured as *trunk*. This simple on-time configuration is necessary since HotOM uses vs-tags for frame "routing" inside network's core, and it can be done using switch's CLI, scripting or even via SNMP.

J. Packet's life

For a better understanding how HotOM works, Figure 4 shows every performed steps on a unicast communication within a given VN. A frame originated from VM_1 has two possible destinations: VM_2 or VM_3. During the entire process, all actions are done over L2 protocol headers (Ethernet and 802.1q).

Suppose VM_1 wants to send an IP packet to other VM. First, it broadcasts an ARP query packet on *(1)* asking for a MAC address of who have that IP address. The AVS captures this query and send it to LAS on *(2)*. The LAS makes a lookup in its cached data searching for an association between MAC address and IP. If there's a cache miss, the LAS retrieves that association from the NCS on *(3)*, caching it. Then, it creates an ARP reply and demands AVS to send it back to the VM (reverse direction of *(1)*).

After the ARP address resolution, VM_1 sends a frame in *(1)* to destination VM. The AVS captures this frame and send it to LAS through an OpenFlow message *(2)*. At this point, LAS can perform two actions. If the destination VM is local, i.e., it is connected to the same AVS, *(4)* is an OpenFlow message installing a forwarding table entry in AVS direct connecting VM_1 and VM_2 *(5)* (dashed line). If the destination VM is remote, LAS labels the frame with destination AVS' vs-tag, changes EtherType to 0x080A, translates the source and destination MAC addresses to its MAC and remote AVS' MAC, adds the L2.5 shim protocol (by adding *net_id*, *HotOM addresses* and *type*) and demands by OpenFlow message *(4)* to AVS to forward the frame *(6)* to network's core through PES.

Once the frame arrives the destination AVS, it sends the frame to its LAS by the OpenFlow message *(7)*. The receiving LAS then removes the vs-tag and reads the L2.5 shim protocol header. With the information available within that header (net_-id field), the LAS knows to which VN that frames belongs to and reading the HotOM source and destination addresses, it translates the MAC addresses back to the original one, also translating the original EtherType back using the *type* field. Finally, LAS demands the AVS to deliver the frame by issuing the OpenFlow message *(8)*. A frame identically to the original one, reaches the destination VM_3 in *(9)*.

IV. Implementation

An early implementation of a HotOM prototype was developed using a collection of softwares distributed with an Open Source Initiative compliant license. The physical infrastructure was made by two Dell PowerEdge R310 1U servers with one Intel Xeon Quad-Core Processor X3470 running at 2.93Ghz, 2GB of RAM, 250GB 7.2K RPM SATA hard drive and two 1Gbps Ethernet port. The network switches were two ExtremeNetworks Summit 430-48t as PES, connected to a single server each, and Summit 440-48t as core, interconnecting the two PES.

The operating system of choice was Linux Fedora 19. AVS' were implemented using Open vSwitch version 2.1.2 because it is OpenFlow compliant and is ready to be used in production environments. KVM was used as hypervisor, along with LibVirt 1.0.5 for VM's administration purposes.

Finally, the OpenFlow controller was a modified POX version 0.2 (codename carp). The modifications added support to HotOM Protocol Header. The NCS was written in Python version 2.7.5. In turn, LAS was also written in Python as a component of the adapted POX controller.

V. Evaluation

To support practical experiments, a testbed was setup with the available physical components discussed in Section IV. The chosen topology was identical to Figure 1, where the access switches were the ExtremeNetworks Summit 430-48t and the core was the ExtremeNetworks Summit 440-48t. Every physical link was configured as *trunk*, allowing all VLAN-tagged frames to be forwarded.

Sixteen VMs running CentOS 6.5 were instantiated and equally distributed on both hosts. Since hosts have 2GB of RAM, it was necessary to limit VM's number on each to eight. This limitation was imperative considering that a VM needs about 256MB of RAM to boot. Moreover, host's CPUs allow the Linux Kernel to deliver four virtual processors to hypervisor (KVM), which means that it is highly advisable to run tests with up to four VMs simultaneously, since each VM allocates one virtual processor.

A. Isolation

Isolation is a "must-have" feature in a multi-tenant datacenter. This property assures that a data traffic will not be accessible or "sniffed" from an unauthorized entity (tenant). HotOM builds its isolation towards upper layers from L2.5. Different tenants can use any IP address they want to, so the same IP could be assigned to many VMs. Tenants also have freedom on choosing any HotOM addresses to their VMs. Ethernet addresses will then be accordingly assigned by the VM managing system, as discussed in Section III-D.

In this experiment, two virtual network were instantiated. The first VN was VN_1, net_id `0xAABBCC`, that connected two VMs VM_{11} and VM_{12}. The second VN, VN_2, had its net_id `0xCCBBAA` and connected VMs VM_{21} and VM_{22}. To VMs were assigned, respectively, HotOM addresses `00:00:01` and `00:00:02`, and IP addresses 10.0.0.1 and 10.0.0.2.

At the same time, a ping was initiated from VM_{11} to VM_{12} and from VM_{21} to VM_{22}. The results of VM_{11} and VM_{21}

```
# ping -c2 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=11.87 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.338 ms
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 0.339/6.10/11.87/5.76 ms

# arp -n
Address    HWtype  HWaddress           Flags Mask Iface
10.0.0.2   ether   00:00:00:00:00:02   C          eth0
```

Fig. 5: Ping and ARP commands for isolation tests

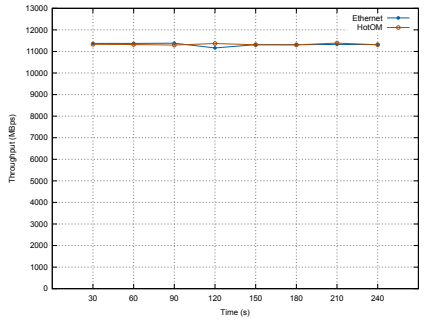


Fig. 6: Local throughput

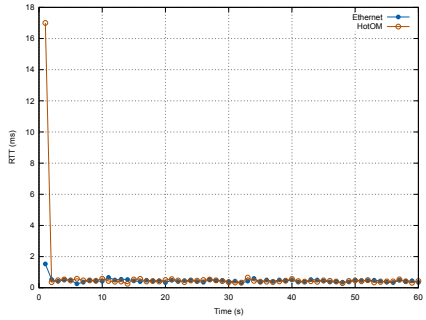


Fig. 7: Local RTT

are similar. Figure 5 demonstrates the ping output and the populated ARP table, showing that there is connectivity.

After the initial connectivity test, a packet sniffing test were performed. `tcpdump` was executed in VM_{21} and VM_{22} trying to capture frames from VN_1. No frames were captured. Then, the symmetric test was done, pinging VM_{12} from VM_{11} and trying to capture these frames in VN_2. Again, no frames were captured, demonstrating the complete isolation between both VNs.

B. Performance

A typical VN connects VMs placed in both same or different hosts. For example, suppose a VN connecting 2 VMs that can be allowed to run on 2 hosts, say $host_1$ and $host_2$. There are two possible different scenarios (*a*) all VMs running on the same host or (*b*) one VM running on each host. These two options are boundaries in the space of possible throughput results, since in (*a*) no frame is address rewritten and vs-tag labeled neither forwarded to network, while in (*b*) all frames are rewritten, vs-tag labeled, forwarded to network, then in destination are again address rewritten, removed the vs-tag and forwarded to VM. Thus, throughput and Round-Trip Time (RTT) experiments were done using those both scenarios, named respectively "all *local* communication" and "all *remote* communication".

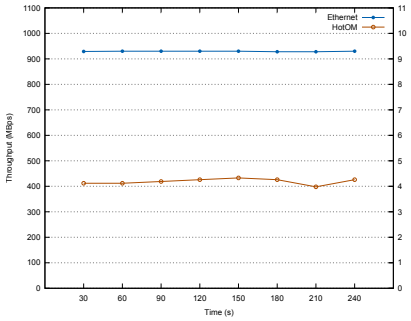


Fig. 8: Remote throughput

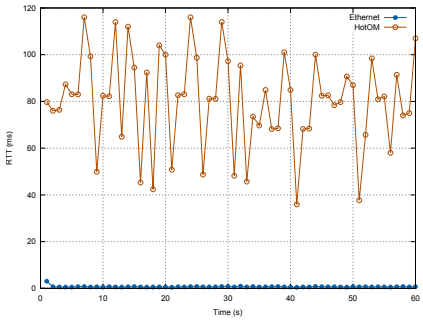


Fig. 9: Remote RTT

The throughput experiments were done using the wide known `iperf` tool for 240 seconds, using TCP protocol and *default* options. RTT tests were performed using `ping` command on the first 60 seconds of throughput experiments. Due the available physical resources constraints, the VNs were made up with 2 VMs.

The evaluations were done facing HotOM against "plain" VLAN. On VLAN setup, virtual switches were not ruled by an OpenFlow controller: they were working just like a traditional *learning switch*.

In order to ease experiments, the testbed was dedicated to them. No other traffic was applied to switches neither processing load to servers other than generated by HotOM. This means that repetitions would have similar results. For this reason these initial evaluations were done in a single time.

The first evaluations were the "all *local* communication". Figure 6 depicts the throughput results. The plotted levels (dots) were output from `iperf` in intervals of 30 seconds. Both VLAN and HotOM's results are similar, around 11500Mbps and quite constant. In turn, Figure 7 shows the RTT results. HotOM's first RTT is higher due the time spent on AVS sending the ARP request to LAS, which process and crafts an ARP reply, then sending it back to VM. After that, RTT are analogous to each other. Worth to denote that performance results are strictly tied to available hardware resources.

The second evaluations were done upon "all *remote* communication". Figure 8 demonstrates the throughput results. There is a gap in performance between VLAN and HotOM. The former has a throughput of around 930Mbps, which is near the nominal 1Gbps bandwidth, the latter has a low result of 4.3Mbps. The difference is also observable in terms of RTT, as seen in Figure 9. While the average VLAN's RTT is 0.66ms, average HotOM's RTT is 81.14ms. These differences are due OpenFlow's shortcomings as discussed in Section III-G.

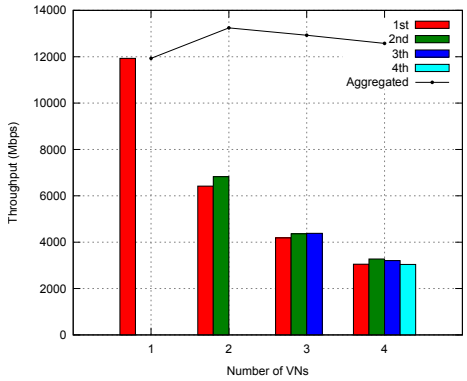


Fig. 10: Multiple local VNs throughput

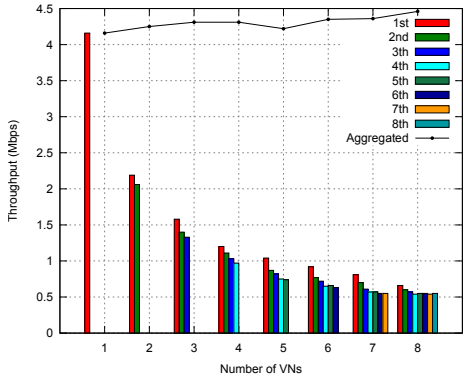


Fig. 11: Multiple remote VNs throughput

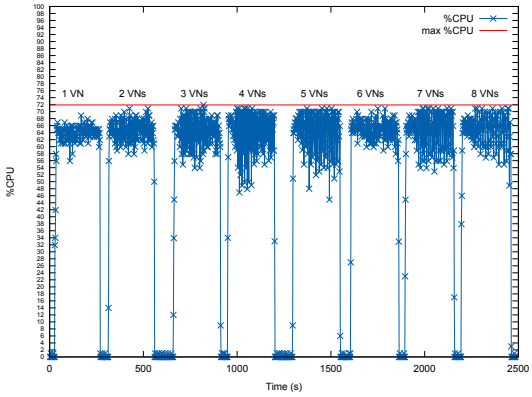


Fig. 12: Controller's CPU usage

Scalability evaluations were also performed, ranging the VN's number from one to the maximum viable in testbed (4 for *local*, and 8 for *remote*). The resulting graphs, depicted in Figure 10 and 11, are a composition of the data rate maintained per VN, in bars, and the aggregated throughput among all VNs in a given experiment, in a black line-dot plot. The aggregate throughput are fairly constant, evidenced by a hyperbolic decay on each VN when they scale up. At same time of this evaluation, controller's CPU usage was measured and plotted at Figure 12. It shows that no matter how many VNs are instantiated, the controller uses about the same high level of CPU. This fact is discussed furthermore in Section VII.

VI. Related Work

Datacenters network virtualization have attracted a lot of attention from academic and professional community in last years. A number of architectures were proposed to address its needs.

Trellis[7] deals directly with virtual topology by creating a mesh (trellis) of point-to-point virtual links, using Ethernet over GRE (EGRE), interconnecting every single host. Some paths are created by a set of virtual links transversing hosts. Interconnecting hosts by a point-to-point virtual link does not scale very well, because its quantity increases by the power of two of the host's number. Moreover, letting hosts to forward packets are not ideal. VM's frame is entirely encapsulated in a GRE packet, what significantly increases protocol overhead. HotOM, in turn, uses a L2 VLAN ID for routing, which demands less resources on maintaining paths to reach destination and leaves the forwarding job to physical switches. Finally, HotOM uses a small L2.5 header for lowering protocol overhead.

VL2[8] was proposed to create a illusion for a service that it is laying over a single Ethernet switch - a ***Virtual Layer 2*** switch. Additionally, VL2 focus on providing scalability and flexibility, by interconnecting switches based on Clos topology. Each service has an IP address, called application-specific address (AA) while switches has an location-specific address (LA). VL2 works on encapsulating AA in LA, in a IP-in-IP scheme, then routing the packet by one of the core L3 switches through a link chosen by Variant Load Balancing (VLB) algorithm. A local agent is in charge of encapsulation/decapsulation while all information is provided by a central Directory System. VL2 has some drawbacks. First, it is an application-driven architecture, not a VM-driven one as HotOM. Moreover, VL2 increases protocol header's overhead, shrinking payload size. Finally, VL2 requires L3 routing inside network's core raising forwarding table size pressure over devices, i.e., datacenter must use expensive network switches. On contrary, HotOM forces a small set of forwarding table entries by encapsulating the payload in a L2.5 protocol header.

Portland[9] focus in network scalability using a Fat-Tree[10] topology and encoding a host position within data-center in its MAC address, called Pseud-MAC (PMAC). So, Portland uses a central OpenFlow application, Fabric Manager, to answer ARP queries and to install OpenFlow rules in the access switches translating back and forth the actual MAC (AMAC) of a host or VM to/from its PMAC. Inside the core's network, frames are routed based on a long prefix matching over the PMAC address. This forces that *all* switches in the topology must be OpenFlow-enabled, in contrast to HotOM that needs only the virtual switch within the hypervisor to be OpenFlow compliant. Furthermore, Portland do not guarantee total isolation on different virtual network.

NetLord[11] architecture virtualizes network by encapsulating VM's frames into a specially crated L3 header by the NetLord Agent (NLA). The outer L2 header carries the source and destination's switch and a VLAN tag that determines through which link the frame must be sent. L3 headers encodes tenant information, like tenant ID. NetLord is much more complex than HotOM. The path is chosen using a heavy algorithm before the VLAN tag being added. Also, NetLord encapsulates the entire L2 frame from VM, creating even higher protocol overhead. Finally, NetLord employs L3 switches on network's edge, increasing deployment costs.

The NVP[12] was proposed to provide network virtualization using programmable switches and allocating tenant-specific logical set of tunnels, called datapath. The VM frame are encapsulated in protocols such as STT, VXLAN or NVGRE before being sent through L3 tunnel. A central SDN controller's cluster takes care of tracking logical datapaths, computing their establishment and ending and configuring virtual switches. But, it calculates the entire network's state and tenant's virtual networks before programming switches. This calculation may take hours. HotOM uses a NCS to maintain a lightweight database storing VMs and AVS' location an data. NVP, as other proposals, uses a encapsulation scheme that strongly increases overhead. It demands that datacenter uses expensive L3 switches in their network's core.

Finally, HotOM in-network routing scheme has some resemblance with MPLS. MPLS is a L2.5 protocol that uses a 20-bit field to set which path the packet must be routed. Only a few specialized physical network's devices support MPLS based routing. HotOM pushes the data used for routing to the L2 header, more precisely into the VLAN tag. Doing so, any network device, even the most ordinary, can route HotOM frames. In addition, MPLS do not support addressing of VMs.

VII. Discussion and future works

This paper introduces HotOM, a new datacenter network virtualization approach with focus in network core legacy devices reuse, programmability, complete L2 isolation and scalability. Its architecture was described in details by disclosing each aspect. To prove its usefulness, an initial evaluation were conducted with successful results.

At this time and stage of research, HotOM is focused in being *functional*, not in being *performatic*. The "all *local* communication" proved to be doable, but the "all *remote* communication" throughput is not ideal because encapsulation/decapsulation VM's payloads in/from HotOM Protocol Header are made by the controller (LAS). This means that for every *remote* frame sent/received has associated delay, what imposes a performance penalty.

One possible future research on achieving high *remote* throughput is to add HotOM L2.5 shim header support directly into virtual switch datapath, running as a kernel module. This approach was adopted in others proposals, like NVP, and certainly will push *remote* throughput and RTT to be close to the measured on VLAN's evaluations (Figures 8 and 9). Other research can consider implementing HotOM support in hardware, where NetFPGA[13] is a natural choice. With it, the OpenFlow controller would offload frame's processing to hardware, saving host's CPU cycles and time on answering NIC's interrupts.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[2] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, N. Gude, P. Ingram *et al.*, "Network virtualization in multi-tenant datacenters," in *USENIX NSDI*, 2014.

[3] V. Soundararajan and K. Govil, "Challenges in building scalable virtualized datacenter management," *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 4, pp. 95–102, Dec. 2010.

[4] IEEE Computer Society, "IEEE Standard for Local anda Metropolitan Area Networks, Media Access Control (MAC) Bridges and Virtual Bridge Local Area Networks," 2011.

[5] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 2, pp. 909–928, 2013.

[6] Open Networking Foundation, "OpenFlow Switch Specification 1.3.4," 2014. [Online]. Available: https://goo.gl/RWyV1R

[7] S. Bhatia, M. Motiwala, W. Muhlbauer, V. Valancius, A. Bavier, N. Feamster, L. Peterson, and J. Rexford, "Hosting Virtual Networks on Commodity Hardware," *Georgia Tech. University., Tech. Rep. GT-CS-07-10*, 2008.

[8] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A Scalable and Flexible Data Center Network," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 51–62, Aug. 2009.

[9] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric." in *SIGCOMM*, vol. 9, 2009, pp. 39–50.

[10] C. E. Leiserson, "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing," *Computers, IEEE Transactions on*, vol. 100, no. 10, pp. 892–901, 1985.

[11] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "NetLord: A Scalable Multi-tenant Network Architecture for Virtualized Datacenters," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 62–73, Aug. 2011.

[12] T. Koponen, K. Amidon, P. Balland, M. Casado, A. Chanda, B. Fulton, I. Ganichev, J. Gross, N. Gude, P. Ingram *et al.*, "Network Virtualization in Multi-tenant Datacenters."

[13] J. Naous, G. Gibb, S. Bolouki, and N. McKeown, "NetFPGA: Reusable Router Architecture for Experimental Research," in *Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow*, ser. PRESTO '08. New York, NY, USA: ACM, 2008, pp. 1–7.

A Network Monitor and Controller using Only OpenFlow

Renato B. Santos[1], Thiago R. Ribeiro[1], Cecília de A. C. César[1]
[1] Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brazil
{rebasantos,thiagorramos}@gmail.com, cecilia@ita.br

Abstract— Software-Defined Networking (SDN) Architecture is relatively well defined, but the standards still need further development. Currently, working groups are developing Application Programming Interfaces (APIs) in which network monitoring is a priority, because it is the first step towards controlling a network. We present a network monitor that provides basic Quality of Service (QoS) parameters without the addition of a probe in the network equipment, making the system independent of suppliers. Our monitor provides percentage of utilization, delay, jitter and loss by only exploring OpenFlow capabilities. Moreover, we present a Traffic Engineering application that selects the route according to flow requirements, using the monitor developed. The results indicate an accurate monitor which is useful for controlling the network.

Keywords: SDN, Openflow, Network Monitoring, Traffic Engineering, QoS

I. INTRODUCTION

Network Monitoring is crucial for network administrators, Internet Service Providers (ISPs) and users, since knowing the network status helps to decide, plan, and use the resources with adequate quality. In the short, medium, or long term, administrators collect data regarding the conditions of the network.

Currently, Software-Defined Networking (SDN) is part of a scenario of Software Defined Infrastructure that controls both computing and networking resources. With this approach, the details of configuration and management of resources are hidden from upper layers through software modules composition. SDN offers the required information for high-level applications thus providing greater flexibility[1].

The network status changes continually and this dynamic behavior presents challenges. Monitor to keep updated status of the resources is essential, since this knowledge is the first step towards the subsequent task of controlling the network. Besides other benefits [2], SDN helps monitor and control the network because it provides a central software controller with a general view of the network. The majority of the SDN solutions, proprietary or not, add a probe inside the network device, giving the controller specific information [4,7,11]. Inserting a probe in the switch contradicts the idea that SDN favors commodity hardware, since each vendor has added its own software agent, increasing the cost of the switch. So, the question becomes: what can be done regarding monitoring without adding more software to the switch?

Figure 1 shows a general architecture of SDN with three layers. The top layer, called the Application Layer, contains different applications such as an Intrusion Detection System, Routing, Load Balancer and Network Monitoring. The Application Layer via a Northbound API accesses the Control Layer, that hosts a Network Operating System, or a central controller which commands the bottom layer via a Southbound API. The bottom layer, called the Data-Plane Layer, contains physical or virtual network equipment. The OpenFlow (OF) protocol is the most widespread Southbound API. The Northbound API is under discussion within the network community.

The traditional approach of monitoring with SNMP is not fully used in this new context because some SNMP objects are already constructed by the OF and other objects no longer make sense [21].

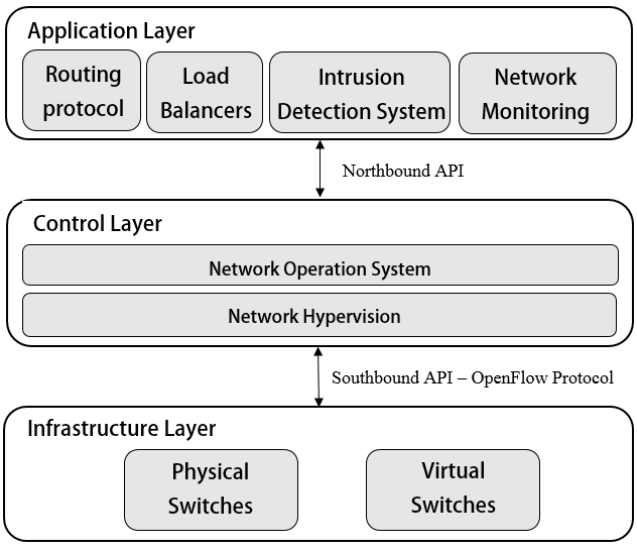


Figure 1 - SDN Architecture

Our research hypothesis was that OF provides sufficient information to achieve basic Quality of Service (QoS) parameters without insertion of additional software in the switches. A software module running only in the Control Layer, using a standard Southbound API, offers a more flexible

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

situation. This module could monitor cheaper bare-metal switches, and it could also be applied to the common strategy of virtual network overlays on a legacy infrastructure.

At this moment, the research community seeks standards to build more independent and interoperable systems. The Open Networking Foundation (ONF), a non-profit industry consortium, has created a specific working group seeking contributions from the network community to deal with the Northbound Interface [3]. They voted to prioritize SDN use cases for network monitoring.

With regard to controlling, short-term information can help to make decisions about the QoS requirements of each different application. For example, a file transfer with a requirement of high bandwidth can be reached via a route with known bandwidth, or at least, the capacity measured in the last seconds can indicate that a specific route is more indicated. The switches can be pre-loaded with the best route for a specific flow.

Our first contribution to the problem is a network monitor that provides four basic QoS parameters without the addition of a probe in the switches. Our monitor is called the OOFMonitor, the Only OpenFlow Monitor, emphasizing the potential of the OpenFlow standard. We provide Percentage of Utilization, Delay, Jitter, and Percentage of Packet Loss. The monitor application asks the Control Layer to obtain the parameters from the network. This communication may be part of a Northbound API proposition, actually, we have developed a JSON interface providing these parameters.

The monitoring may be integrated with controlling. With the central view of the SDN approach, better decisions can be taken, favoring Traffic Engineering. To develop this idea, we implemented a QoS Router that obtains monitored parameters and decides the best route according to a given requirement. Thus, the second contribution of this paper is a Traffic Engineering application.

This paper is structured as follows: Section 2 contains an overview of related work regarding Network Monitoring with SDN; Section 3 presents the OOFMonitor, detailing the strategy to build each parameter; Section 4 presents the QoS Router, using the API developed for the OOFMonitor; Section 5 shows validation of each parameter built and the results of applying these parameters in practical use. Section 6 concludes the paper.

II. Related Work

Traditionally, monitoring systems have used distributed protocols like the SNMP standard, free tools, like Netmon, or proprietary ones. In both, proprietary protocols and open standards, the best tools available are expensive and attached to vendors. This scenario is expected to change with SDN, in which a central point provides secure access to the switches. This central point having direct access to all switches, can get all the information needed with a simple protocol like OpenFlow, simplifying the monitoring task greatly. Thus, conceptually, a specific protocol for monitoring is not needed.

More and more SDN tools are emerging with different features to satisfy different purposes. Onix and OpenDaylight are among the most mature. Onix [4] is a complete framework with 150,000 lines of code and a general API. It addresses scalability issues, but as a commercial platform, the Onix API remains closed. OpenDaylight [5] uses standard flow table statistics that can increase, compromising performance due to CPU overload. When using OF, the system must be configured carefully to avoid a large amount of information yet maintain a reasonable volume useful for managing the status.

Bismark [6], a project used in home monitoring in the United States, builds Bandwidth, Delay, Jitter, and Loss by inserting their agent in the user's modem. They recommend not to use a probe in the user's machine, but in a gateway to allow continuous measurements and independence.

Other frameworks [1,7-10] calculate one or more of these four parameters, but not all four together. The main concern of Giotis et al. [7] is to detect anomalies. Thus, they monitor only network usage, specifically packet rate. Lin et al. [1] propose a Resource Management System, but they present only experiments using packet rate. The idea of a QoS API appears in Kim et al. [8]. They start with link utilization and packet loss. PANE [9] has projected a protocol for the application to be informed of expected network behavior. For example, the application tries to reserve bandwidth with a query to the controller that, knowing device status, informs at what time this request can be satisfied. However, the application is not informed about the expected delay, only about bandwidth. Handigol, N. et al. [10] present a load balance system called Aster*x, which measures link utilization and average response time for each request. The article does not detail the method, because the code is closed. No Network Monitor was found that builds these QoS parameters without an agent in the switch.

Monitors must also handle queries, or pulling of information, and receive reports not associated to queries, called pushing. Pushing decreases traffic, since in a configured interval, the information is sent by the switch without queries. Rezende et al. have shown that pulling is more accurate than pushing, as they compared OF mechanism with sflow [22]. In their tests, they varied the pulling frequency but there is no analysis regarding scalability. Giotis et al. [7] argue that sampling at longer intervals generates less information, which alleviates the switch; but they insert an agent in the switch to sample and send collected data to the controller. The query periodicity has no consensus. A short interval promotes accurate information at the cost of overloading the system; on the other hand, a long interval may miss important events. The reported collection interval varies between 15 and 30 seconds. OpenDaylight [5] and Bismark [6] use 15 seconds, Giotis et al. 30 seconds, Heller et al. [11] use the strategy from Phemius and Bouet [12], the same as we used to measure delay, but they do not mention the periodicity of queries.

Several studies have shown promising results from their use of some of these network status parameters to make decisions regarding Traffic Engineering (TE). QoS allocation is highlighted as an important TE issue, which can be exploited

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

by SDN [13]. The QNOX - QoS-aware Network Operating System - [14] developed a central version of OSPF, showing that a new route in a network with 100 switches can be calculated in less than 100ms, suggesting even greater scalability. Silva et al. [15] and Georgopoulos et al. [16] showed optimization of QoE in video applications. A promising study by Cui et al. [17] classified flows according to QoS and found a path that satisfies requirements of both bandwidth and delay. These are still preliminary results in the area but the OOFMonitor and QoS Router confirm the good results of the SDN approach for monitoring and controlling the network.

III. OOFMONITOR

This section details our first contribution, the OOFMonitor with an integrated Control Layer. Fig. 2 shows the complete System Architecture. In the Control Layer, the Controller used was the Ryu [18], a mature module that implements the latest versions of OpenFlow. Part of the OOFMonitor was implemented in the Statistics Manager, since this module obtains the monitoring parameters from the switches and sends to the Application Layer to be consumed. Of course, the statistics refer to the current topology which is maintained by the Topology Manager.

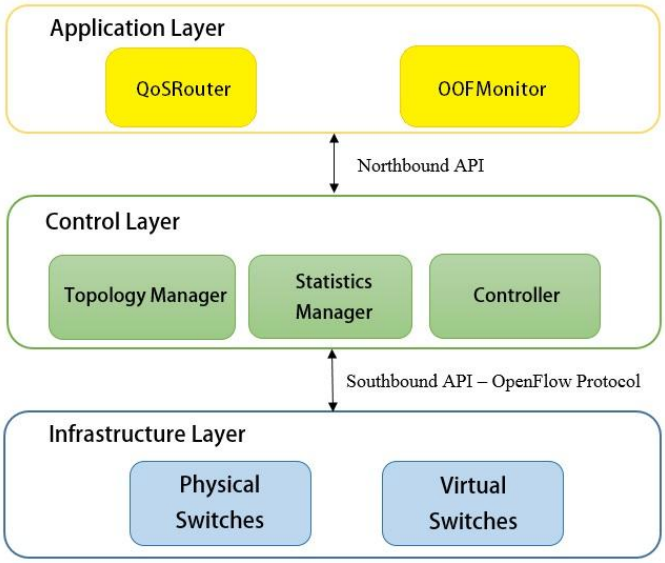


Figure 2 - Projected System Architecture

We suppose that the switches in the Infrastructure Layer implement OF protocol. This is enough for our monitor.

In section IV the QoS Router will be presented. Each of the next subsections will explain how to obtain the four parameters without an agent in the switch, emphasizing which OpenFlow feature was exploited.

III.1 Link Utilization and Loss Rate

We decided to monitor the bandwidth per port, as this is an essential basic information on any monitor. Furthermore, the bandwidth per port is easier to achieve than bandwidth per flow, since a flow can start and end in an interval that could be too brief to monitor. OpenFlow has a pair of messages, Request and Reply, to gather statistics from each port; periodically, the Statistics Manager sends a *OFPT_MULTIPART_REQUEST* with type *OFPMP_PORT_STATS* to ask for Volume of transmission and Volume of dropping. The switch answers with a *OFPT_MULTIPART_REPLY*, containing *tx_bytes* and *tx_dropped*. The transmitted data is compared with the link capacity given by the Topology Manager, thus describing the link utilization. Equation 1 calculates the volume of transmitted data in a polling interval [i, i-1]. Equation 2 is similar for Packet Loss Rate.

$$TxRate(i) = \frac{tx\_bytes(i) - tx\_bytes(i-1)}{Time(i) - Time(i-1)} \quad (1)$$

$$LossRate(i) = \frac{tx\_dropped(i) - tx\_dropped(i-1)}{Time(i) - Time(i-1)} \quad (2)$$

III.2 Delay

Standard OF does not provide delay measures. The average delay between the controller and the switches is easy to calculate. The controller sends requests regularly and receives replies, and this time interval is a good measure of delay between the controller and the switches. The problem is to evaluate the delay between switches.

Figure 3 illustrates the delay measurement process adopted here from Phemius and Bouet [12]. First, the Controller (C) creates a special packet with an unknown *Ethertype* and sends this packet to Switch 1 (*S1*), as Step (1) in Fig. 3. This probe packet is sent via a *OFPT_PACKET_OUT* message with a command to Switch 1 to forward this packet to Switch 2 (*S2*) in the link we want to measure. In Step (2), *S1* sends the packet to *S2*. Since there is no rule matching this packet, when *S2* receives this unknown packet, it forwards it back to the Controller in Step (3) via *OFPT_PACKET_IN* message.

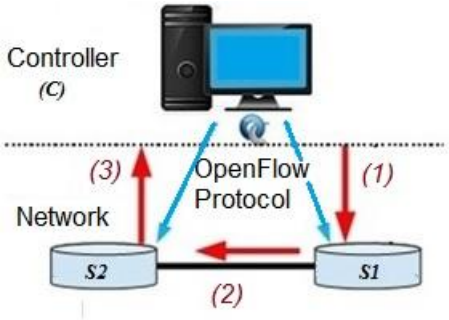


Figure 3 - Delay measurement process

When the controller receives the message with the packet built by it, it can measure the packet travel time. This travel time includes the three steps. Since the delay between the controller and the two switches is known, $D_{c\text{-}s1}$ and $D_{c\text{-}s2}$, the final delay $D_{s1\text{-}s2}$ must subtract these two delays, according to Equation 3.

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

$$D_{s1\text{-}s2} = TravelTime - D_{c\text{-}s1} - D_{c\text{-}s2} \quad (3)$$

In order for the Controller to identify which network section each returned packet from Step 3 refers to, identifying information must be inserted in the probe packet. Thus, the probe packet includes an identification number of the switch that sent the packet and the time the packet was sent.

This strategy to measure delay can be applied to discover the network. In fact, some implementations of the discovery protocol LLDP (Link Layer Discovery Protocol) for OpenFlow also generate a packet with a special *Ethertype* sent via an *OFPT_PACKET_OUT* message directed to all ports. We chose to perform network discovery and maintenance using our own packets because they are smaller than the LLDP packets and, therefore, generate less overhead. Because our inspection packet is light, we decided to monitor the network every 5 seconds. Our measurement scheme generates traffic of only 6Kbps in contrast to the LLDP that would generate 55Kbps just for discovery without measuring delay. Of course, the calculated delay includes propagation and processing delay, but it is a good indication of the current load in the switches.

III.3 Jitter

Jitter is determined from the variation of delay. Depending on the application, a large value of jitter is a problem which greatly compromises the application performance. Our jitter calculation is a measure of the behavior of the network by itself, based on the discussion of RFC 3550 [19] which considers the accumulation of previous measurements.

As we have Equation 3 to calculate delay, we need to store both the previous and current delay to calculate its variation in a polling interval [*i, i-1*]. Equation 4 calculates updated jitter, considering instant *i* and the previous instant (*i-1*). The delay in this interval is *D(i-1,i)*.

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16 \quad (4)$$

As stated by RFC 3550, the change of inter-arrival time is divided by 16 to reduce noise. The division by 16 helps to reduce the influence of large random changes. A change in delay needs to be repeated several times to influence the jitter estimate significantly.

IV. QoS Router

The QoS Router is our second contribution illustrated in Fig. 2. It uses the QoS parameters given by the OOFMonitor to make Traffic Engineering decisions. QoS means providing the appropriate parameters that applications demand. Our hypothesis is that the network layer with SDN improves performance, giving each application the best route currently available, regarding a relevant parameter for the application. For example, if VoIP is an active flow, the delay is critical. If the application is a File Transfer, bandwidth is critical. OF creates a viable environment to implement QoS treatment, because the first packet of each flow goes to the Controller and can be classified and aggregated with others with the same demand for further actions such as finding the best route.

In our approach, we used the classical Dijsktra Shortest Path Algorithm (SPA), assuming that the network is a graph. The cost function is defined according to the relevant parameter for the flow. The QoS parameters of each link given by the OOFMonitor should be transformed into a positive cost parameter to be used for the SPA. However, we are interested in the whole path, so we make some assumptions in calculating a specific link cost in the context of the complete path. Each of the next subsections will explain how to obtain the cost function for each parameter, and why it is important to maintain a view of the accumulated cost.

The cost function has three elements:

- Link Identification;
- Current Cost Value - the total path cost calculated from the source until the current link.
- Current Statistics - used to compute the cost.

The great advantage of using SDN is that in the central controller we have all the information about all switches and we can immediately calculate the associated cost. To define cost functions below, we assume a current link *j* in a current instant *i*.

A. Cost of Link Utilization

A static approach would only consider that the lower the link capacity, the higher the cost. For a better result, we consider the actual status of the network, using Equation 1 to calculate the dynamic transmission rate. If a link *j* is widely used, the bytes transmitted approach LinkCapacity(j) and the cost increases:

$$LinkCost(j) = 1/(Link\ Capacity(j) - TxRate_j).$$

B. Cost of Loss Rate

Some applications, like file transfer are very sensitive to loss of data. Calculation of Loss along the whole path is difficult because it is not simply the addition of the loss of each link. We should find a way to calculate cost which express the fact that the smaller the loss, the lower the cost.

As we have *LossRate* from Equation 2 and *TxRate* from Equation 1, we can calculate the loss percentage in a link *j:*

$$Loss(j) = \frac{Transmitted_{source} - Received_{dest} + Dropped_{dest}}{Transmitted_{source}}$$

We are considering that *(1-Loss(j))* packets cross the link j, and the total Loss in a path with *n* links is:

$$Loss(1,2,\ldots,n) = 1 - \prod_{j=1}^{n} (1 - Loss(j)) \quad . \quad (5)$$

This equation should be transformed such that *LossCost(j)* be a function of *Loss(j)* to be used in the SPA. We apply a logarithm to change the product to sum of the log.

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

$$log\left(1\text{-}Loss(1,2,\ldots,n)\right) = log\left(\prod_{i=1}^{n}\left(1\text{-}Loss(j)\right)\right)$$, then

$$-log\left(1\text{-}Loss(1,2,\ldots,n)\right) = \sum_{j=1}^{n}[-\log\left(1 - Loss(j)\right)].$$

The bigger the Loss(1,2,...n), the bigger the right side becomes; so, we can define the cost of each link as the inside part of the sum:

$$LossCost(j) = -log(1\text{-}Loss(j)).$$

The cumulative cost can now be calculated as

$$\sum_{j=1}^{n} LossCost(j).$$

C. Cost of Delay

The delay of each link is determined by equation 3, and, as positive value, can be directly associated to the cost. Bigger the delay, bigger the cost. Therefore, as we try to have smaller delays, we can define the cost of the link *j* as its delay

$$DelayCost(j) = D_j(i\text{-}1,i)$$

The expected delay of a whole path can be given by the sum of the delays of each link, that can be summed up directly.

D. Cost of Jitter

The total jitter of a path cannot be defined exactly as the sum of the jitter of each link, because doing so assumes that every link has a delay variation of the same sign, i.e., when one delay increases, every other delay increases as well. This is only a worst-case scenario.

However, the sum of jitter values is a good approach to avoid paths with potentially large volumes of jitter. Due to the simplicity of the calculation and the good results, we assumed that the cost of Jitter parameter is as the jitter calculated by Equation 4.

$$JitterCost(j) = J_j(i)$$

V. EVALUATION

We used Mininet [20] a useful and simple tool to develop our experiments with Openflow and SDN in a virtual machine environment.

Our validation experiments are presented in the next two subsections. The first subsection is dedicated to OOFMonitor validation, showing that our measurements are accurate. The second subsection presents the QoS Router results.

V.1- OOFMonitor Validation

OOFMonitor measurements taken in different situations of load and topology were compared to measurements taken by iperf and ping. We evaluated delay, jitter and loss. The link utilization is just about getting switch statistics and this process has already been validated by the community.

A. Delay

When Mininet creates a network topology, in each link creation it allows the definition of delay as a characteristic of the link. We used this initial delay definition to create our scenario. The scenario is shown in Fig. 4, with the link between hosts and the respective switch created with delay=0. After the initial creation, host *h1* pings *h2* and we registered the delay.

With the definition in Mininet of delay=0 in link *h1-s1* and *h2-s2*, the delay reported by the ping reflected only the delay between *s1* and *s2*. The comparison with our measurement was direct. Fig. 5 shows the measurements of ping and the OOFMonitor. On the x axis, we vary the delay parameter in the Mininet creation of the link *s1-s2*.

We ran the test 25 times in order to get statistical volume, activating *iperf* randomly during execution to generate competitor traffic, thus, simulating a real network environment. The delay obtained by the OOFMonitor was very close to the delay obtained by ping, reaching a maximum of 7%.

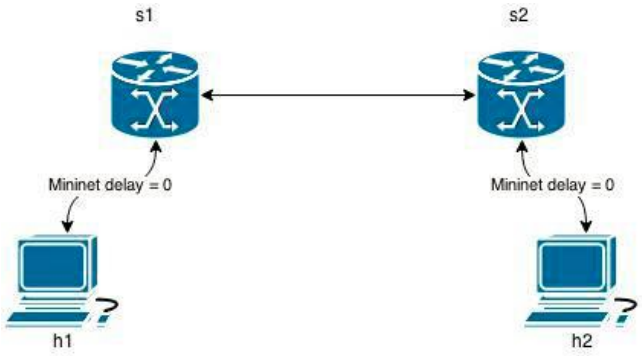


Figure 4 - Scenario for Delay validation

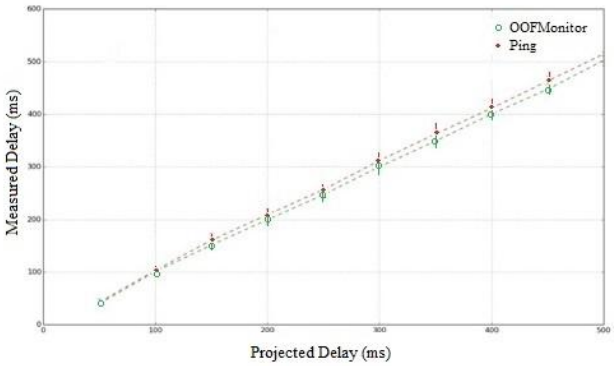


Figure 5 - Delay comparison

B. Jitter

The strategy to measure jitter was the same strategy as delay, since Mininet also allows the definition of jitter as a characteristic of the link in the link creation. Fig. 5 shows the comparison between jitter measured with ping and jitter measured with the OOFMonitor. On the x axis, we vary the jitter parameter in the Mininet creation of the link *s1-s2*. For

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

both calculations we used the delay from Equation 3 and jitter from Equation 4. The results are more dissimilar in low levels of jitter. Due to the effect of the Mininet processing, see that the confidence interval is quite high.

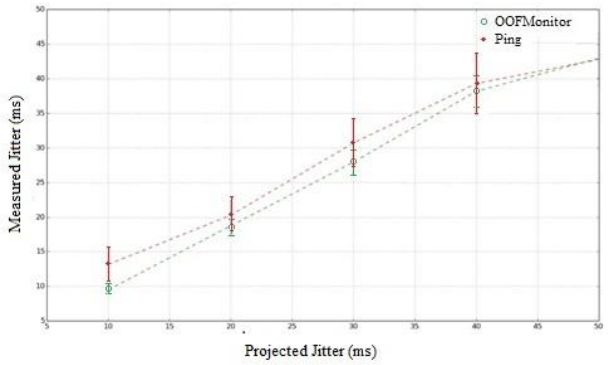


Figure 6 - Jitter comparison

C. Loss

To measure loss, the loss rate was defined in Mininet in the link creation and the real loss was measured with ping and with the OOFMonitor. The peculiarity here was due to loss calculation, because ping measures the round trip loss which is not simply the sum of going and returning as explained in section IV-B.

In order to compare the OOFMonitor and ping, we should consider the loss of going and the loss of return. This cumulative loss is called *RoundTripLoss*. As a result of Equation 5, we have

$$RoundTripLoss= Loss(going, return,)=1-[1-Loss(going)]*[1-Loss(return)]$$

Based on our experiments, we considered that the loss of going is equal to the loss of return, simply called now *Loss*. Consequently,

$$RoundTripLoss=2*Loss - Loss^2.$$

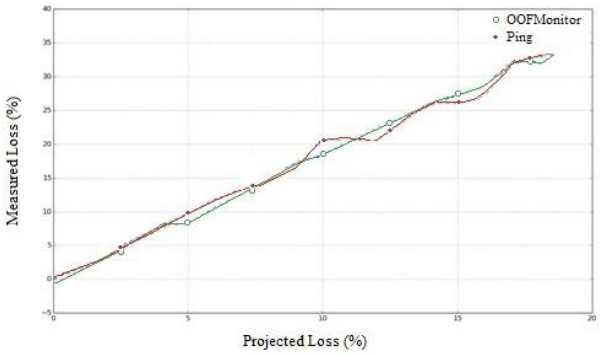


Figure 7 - Loss comparison

V.2- QoS Router results

A JSON interface was built to access the Control Layer making it easy for the Application Layer to use the whole database of topology and statistics. The Topology Manager and the Statistical Manager (Figure 2) made available its data to OOFMonitor and QoSRouter via JSON. The QoS Router used JSON interface to select one of the parameters: link utilization, loss, delay, or jitter.

Fig 8 shows one of the tested scenarios. We created this network in Mininet with the parameters defined for each link according to Table 1. The purpose was to decide the route from the host connected to *s1* to the host connected to *s6*.

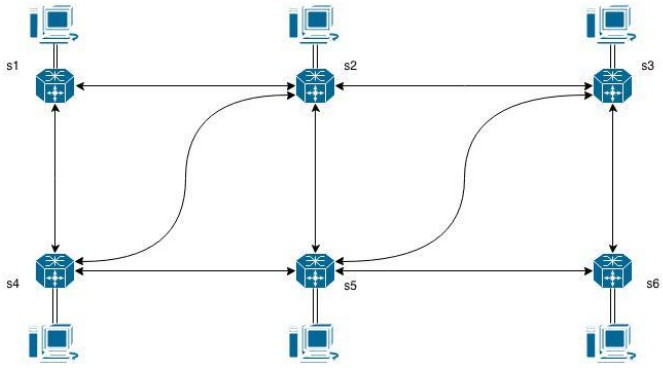


Figure 8 - Scenario for QoS Router validation

From *s1* to *s6* there were several paths, even with the same number of hops, but we wanted to calculate the shortest path with respect to a specific cost function. Applying the cost function from section V, each criteria resulted in a different path.

With an association of Fig 8 and Table 1, the results discussed below can be checked.

TABLE I. LINK CONFIGURATION

Source	Dest	Delay (ms)	Jitter (ms)	Loss (%)	Capacity (Mbps)
s1	s2	50	20	5	10
s1	s4	150	10	10	1
s2	s3	50	20	5	10
s2	s4	100	10	1	1
s2	s5	150	10	10	100
s3	s5	100	10	1	1
s3	s6	150	10	10	1
s4	s5	100	5	5	10
s5	s6	100	5	5	10

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

A. Best Route for Delay

When the flow indicates that a route with minimum cost for delay is needed, the QoS Router takes the column Delay from Table I to configure the switches in the path that produces the minimum delay. The route found was *s1-s2-s3-s6*, which produced a delay of 250ms.

B. Best Route for Jitter

For the jitter parameter, the best route found was *s1-s4-s5-s6*, which produced jitter of 20ms. Note that the best route for delay was not the best for jitter.

C. Best Route for Loss

As stated in section IV, the path chosen must reflect the cumulative packet loss according to Equation 5. There were two best routes found: *s1-s2-s4-s5-s6* and *s1-s2-s3-s5-s6*. The cumulative packet loss was 1-0.95*0.99*0.95*0.95=0.19. Note that both routes had one additional hop, but was chosen for the minimum loss criteria and not number of hops. Our algorithm chose one of the two routes found. Finding two routes is promising for an Equal Cost Multipath (ECMP) implementation.

D. Best Route for Link Capacity

To escape the 1Mbps bottleneck links, the QoS Router chose *s1-s2-s5-s6* as the best way to use the most bandwidth available.

This test used the link capacity to determine routing, but the current utilization could be used in a similar way.

After the initial test, we stressed the system. We activated several TCP flows with iperf and several UDP flows with 10 Mbps of bandwidth randomly. With this heavy traffic running, we did the ping from *s1* to *s6* and collected the measurements, calculating the best route according to each parameter. Table II consolidates the final measurements of the best routes according to each Route Approach.

Each parameter had a considerably better result in its respective optimization area, in relation to other areas. Delay was 200ms lower than the average result. Jitter was less than half of the second best result. Packet loss was 8% better than the average and bandwidth jumped from 1Mbps to 8Mbps.

TABLE II. RESULTS ON DIFFERENT PARAMETER

Parameter / Route Approach	Delay (ms)	Jitter (ms)	Loss (%)	Bandwidth (bps)
Delay	499	25	32	973 Kbps
Jitter	701	7	35	874 Kbps
Loss	835	60	22	860 Kbps
Available Capacity	597	18	25	7.69 Mbps

In the stress test, each parameter produced a different best path. Routing without considering QoS finds the same route for all cases, and would not be expected to produce the best results meeting the demands of the different flows.

VI. CONCLUSION

We have built a monitor to collect delay, jitter, loss rate, and link utilization. Our monitor does not require a software agent inside the switch, despite using only the features available in OpenFlow version 1.3. This approach promotes the use of a cheaper switch without the need for customization. In fact, OF is strong enough for basic QoS monitoring and OOFMonitor encourages its use.

The use of these basic parameters easily produced by the SDN paradigm is of great potential. We developed a QoS Router that takes these parameters as input and produces a best route according each parameter. Our QoS Router found the best route for the four parameters in a sample scenario.

Other applications can make use of QoS parameters as it is built into the Control Layer and available from a NorthBound API.

In the future we want to integrate these parameters, because the QoS Router currently only calculates the best route for one parameters at a time. Some applications require the combination of parameters, such as video that requires low delay and high bandwidth. Applications that demand low jitter frequently demand low delay also. It is a matter of building an adequate cost function by combining the individual costs, since the parameters are already available.

A promising approach might be to implement the use of more than one route when available, or to define an additional criterion for the decision between equal cost routes.

Finally, a detailed investigation comparing the features of SNMP and OF is essential to detect the differences between these two approaches.

REFERENCES

[1] Lin, T.; Joon-Myung Kang; Bannazadeh, H.; Leon-Garcia, A. (2014), "Enabling SDN applications on Software-Defined Infrastructure," Network Operations and Management Symposium (NOMS), 2014 IEEE, vol., no., pp.1,7,. doi: 10.1109/NOMS.2014.6838226.

[2] Levin, D., Canini, M., Schmid, S., & Feldmann, A. (2013, August). Incremental SDN deployment in enterprise networks. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (pp. 473-474). ACM.

[3] ONF, "ONF North Bound Interface Working Group Charter" (2013), http://www.onfsdninterfaces.org/index.php/onf-nbi-leadership-roundtable-materials/6-onf-nbi-work-group-charter.; Last accessed: November, 2014.

[4] Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M. & Shenker, S. (2010, October). Onix: A Distributed Control Platform for Large-scale Production Networks. OSDI'10 Proceedings of the 9th USENIX conference on Operating systems design and implementation. (Vol. 10, pp. 1-6)

[5] http://www.opendaylight.org/software.

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

[6] Sundaresan, S., Burnett, S., Feamster, N., & De Donato, W. (2014, June). BISmark: A testbed for deploying measurements and applications in broadband access networks. In 2014 USENIX Conference on USENIX Annual Technical Conference (USENIX ATC 14) (pp. 383-394).

[7] Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., & Maglaris, V. (2014). Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments.Computer Networks, 62, 122-136.

[8] Kim, W., Sharma, P., Lee, J., Banerjee, S., Tourrilhes, J., Lee, S. J., & Yalagandula, P. (2010). Automated and scalable QoS control for network convergence. Proc. INM/WREN, 10, 1-1.

[9] Ferguson, A. D., Guha, A., Liang, C., Fonseca, R., & Krishnamurthi, S. (2013, August). Participatory networking: An API for application control of SDNs. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (pp. 327-338). ACM.

[10] Handigol, N., Seetharaman, S., Flajslik, M., Gember, A., McKeown, N., Parulkar, G. & Anderson, T. (2011). Aster* x: Load-Balancing Web Traffic over Wide-Area Networks.

[11] Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., & McKeown, N. (2010, April). ElasticTree: Saving Energy in Data Center Networks. In NSDI, 2010. Proceeding of the 7th USENIX Symposium on Networked Systems Design and Implementation (Vol. 10, pp. 249-264).

[12] Phemius, Kévin; Bouet, Mathieu (2013). Monitoring latency with OpenFlow. In: 9th International Conference on Network and Service Management (CNSM). IEEE.

[13] Akyildiz, I. F., Lee, A., Wang, P., Luo, M., & Chou, W. (2014). A roadmap for traffic engineering in SDN-OpenFlow networks. Computer Networks, 71, 1-30.

[14] Jeong, K., Kim, J., & Kim, Y. T. (2012, April). QoS-aware network operating system for software defined networking with generalized OpenFlows. InNetwork Operations and Management Symposium (NOMS), 2012 IEEE (pp. 1167-1174). IEEE.

[15] Silva, D. P., Pontes, A. B., Avelar, E. A. M., & Dias, K. L. (2013). Uma Arquitetura para o Aprovisionamento de QoS Interdomínios em Redes Virtuais baseadas no OpenFlow. 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 893-906.

[16] Georgopoulos, P., Elkhatib, Y., Broadbent, M., Mu, M., & Race, N. (2013, August). Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking (pp. 15-20). ACM.

[17] Cui, H., Zhu, Y., Yao, Y., Yufeng, L., & Liu, Y. (2014, May). Design of intelligent capabilities in SDN. In Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on (pp. 1-5). IEEE.

[18] Ryu homepage: http://osrg.github.io/ryu/

[19] Schulzrinne, H.; Casner, S.; Frederick, R.; Jacobson, V. (2003, July); “RTP: A Transport Protocol for Real-Time Applications”, https://www.ietf.org/rfc/rfc3550.txt.

[20] Mininet homepage: http://mininet.org/.

[21] Van Adrichem, N.L.M.; Doerr, C.; Kuipers, F.A., "OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks," Network Operations and Management Symposium (NOMS), 2014 IEEE, vol., no., pp.1,8, 5-9 May 2014.doi: 10.1109/NOMS.2014.6838228

[22] Rezende, P.H.A., Coelho, P.R.S.L., Faina, L.F.,Camargos, L.,Pasquini, R. "Plataforma para Monitoramento de Métricas de Nível de Serviço em Redes Definidas por Software". XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), May, 2015

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

Performance Evaluation of OpenFlow in Commodity Wireless Routers

Leonidas Lima, Diego Azevedo, and Stenio Fernandes
Federal University of Pernambuco, Center of Informatics
Recife, Brazil
{lflj, dcna, sflf}@cin.ufpe.br

***Abstract*—Software-Defined Networking technologies are rapidly reaching wireless environments and a number of questions arise regarding its efficiency in such scenarios. This work evaluates the OpenFlow performance in scenarios with based commodity wireless router running an Open vSwitch (OVS) Linux kernel module and different SDN controllers. A set of client hosts connected at the commodity wireless router sends data to a sink server host. We collected key performance metrics, such as throughput, delay, jitter, and packets loss, to evaluate the efficiency of using a commodity wireless router in a SDN context. Results show some bottleneck when using UDP protocol in high rates. These results suggest that a commodity wireless router can be only deployed in reduced size networks, as homes and small to medium organizations, contributing to accelerate the pace of SDN adoption based on use of legacy network devices.**

Keywords—OpenFlow; Performance evaluation; Software-defined networking; Wireless Router

I. INTRODUCTION

In recent years, the Internet community has put much research efforts on new network architectures for the Future Internet [1]. Software-Defined Networking (SDN) has emerged as a new paradigm on this scenario. By relying on OpenFlow [2], which is a protocol that provides the southbound standard communication between SDN switches and controllers, it is possible separate data and control planes, through a forwarding paradigm based on multiple fields of packet headers at different protocol layers.

SDN has the potential to offer high flexibility in order to deploy new services, but the adoption process of such new technology has been slow [3]. Although OpenFlow is already present in several commercially available SDN switches, the cost involved to replace all legacy devices by new ones, is slowing down the pace of a wide spread SDN adoption. Therefore, the possibility of use commodity routers with the support of an OpenFlow software module might be a smart strategy to accelerate the deployment of SDN technologies.

This work aims at investigating the performance of a software switch based on Linux kernel module, running in a commodity wireless router, as an OpenFlow switch. We show that there exists some performance bottlenecks to overcome. However, we also show that, at certain limits, the wireless OpenFlow router provides good performance, with the advantage of its associated low cost. Results support the idea that adapted wireless router with Openflow compatibility can be widely utilized in home networks, university campus, and small and medium organizations, thus contributing to stimulate widely deployment of SDN solutions. To the best of our knowledge, this is the first work that provides a performance evaluation of the popular Open vSwitch (OVS)[1], an open code software switch widely used in virtualization platforms, running in a commodity wireless router, as a Linux kernel module.

The remainder of this paper is organized as follows. Section II presents the related work and highlights the contributions of this paper. Section III presents the evaluation methodology and the *testbed* architecture we used for running the experiments. Section IV presents experimental results and analysis. We provide concluding remarks in Section V.

II. RELATED WORK

In [4], the authors present an architecture to improve the lookup performance of the OpenFlow Linux kernel module implementation. They utilized the hardware classification feature available on the network interface Intel 82599 and used multiple CPU cores to perform packet processing concurrently. Despite the better performance obtained, such a solution was limited to a specific hardware set, which do not allow its use on wireless networks.

Fratczak et al. [5] propose HomeVisor, a tool for management and configuration of home networks, based on FlowVisor [6] and OpenFlow. Dely et al. [7] present an architecture to integrate OpenFlow with wireless mesh network, to solve mobility problems on mesh networks. Both solutions use routers running Pantou[2], an OpenFlow software implementation for OpenWRT[3] running on user space. The

[1] http://openvswitch.org
[2] http://archive.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT
[3] http://openwrt.org

978-1-4673-9408-6/15/$31.00

utilization of the user space induces to a low performance and the evaluations were performed using rates below 15 Mbps. Araniti et al. [8], using OMNeT++ [4] network simulator framework, investigate potential advantages introduced by the use of SDN architecture and OpenFlow into wireless networks, and considering different types of multimedia applications, as Interactive Gaming, VoIP, Streaming Audio and IPTV. Their results show that the usage of the SDN architecture and OpenFlow standard introduces benefits in terms of delay, throughput, and jitter.

Oliveira et al. [9], presents the performance evaluation of some OpenFlow implementations, using the Oflops tool [10]. The authors analyse the OpenFlow running in a NetFPGA[5] board and in two machines running the OVS in a Linux kernel module. We emphasize that their results indicate that distinct hardware use can affect a lot the loss percentage result of OVS in high load situations. In [11] the authors implemented a low cost SDN *testbed* using Raspberry Pi[6] and demonstrated that OVS running in this hardware presents a maximum network throughput almost equal to that shown by NetFPGA cards.

The main contribution of our work is evaluate the effective and efficient use of SDN with OpenFlow in a low cost commodity wireless router, running an OVS Linux kernel module. We are trying to have an in-depth knowledge of the prospective SDN adoption in home and small to medium business networks.

III. Evaluation Methodology

In this section we describe the *testbed* architecture, the tools, and the methodology for collecting performance metrics in our experiments.

A. Testbed Architecture

A group of client devices, connected over a commodity wireless router, which transmits data to a sink server, using a traffic generator tool, are the elements of the *testbed* architecture used in this work. The network topology is presented in Fig. 1. There are ten clients (as eight virtual machines connected at a wired network and two laptops using wireless interfaces), one sink server, one SDN controller, and one wireless router that interconnects all the devices. The wireless router selected for this study was the TP-Link TL-WR1043ND v1.0, because it offers a good processing capacity (Atheros AR9132 CPU @ 400MHz), a reasonable amount of memory (32MB of RAM and 8MB of flash), five Gigabit Ethernet interfaces (4 LAN and 1 WAN) and a 802.11n wireless network interface. During the experiments, two notebooks (Intel Dual-Core i5-2450M CPU @ 2.50GHz/4GB RAM/802.11n wireless interface) were used as wireless clients.

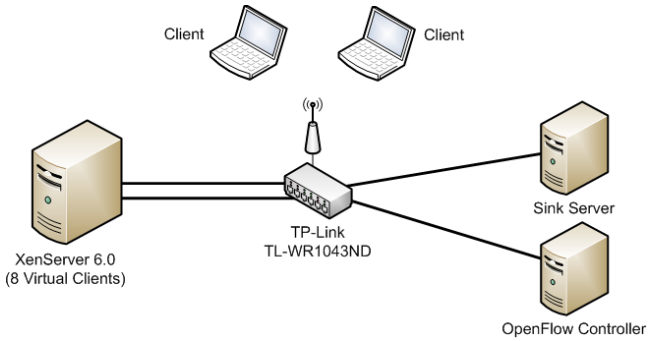


Fig. 1. Network topology utilized in the experiments. Eight virtual clients using wired network and two wireless clients generate traffic to a sink server. The wireless router TL-WR1043ND interconnects all devices and is controlled by the SDN controller.

The wired clients were deployed using eight virtual machines, running on a hypervisor Xen Server 6.0 (Intel Quad-Core i7-3770 CPU @ 3.40GHz/32GB RAM/3 Gigabit Ethernet interfaces). The SDN Controller and the sink server were executed in real machines, equipped with 4GB of RAM and 1 Gigabit Ethernet interface, but the controller has used a Intel Quad-Core i7-3630QM CPU @ 2.40GHz, while the server has utilized a Dual-Core i3 2310M CPU @ 2.1GHz. All hosts have been running the Linux Debian 7 operational system, including the clients.

B. Tools

We have used the Distributed Internet Traffic Generator (D-ITG) tool [12] to generate traffic from the clients to the server. D-ITG is a well-known open source tool capable of generating both IPv4 and IPv6 packets as well as synthetic traffic from a number of Internet applications. By relying on D-ITG, we were able to control the packet size and sending rate in all experiments. At the end of each data transfer session, the D-ITG generates a report with different performance metrics, which was used in our analysis.

We also took into account time synchronization between clients and the server. As we need to measure the packet delay, we used the Network Time Protocol (NTP)[7]. In addition, we have used the PF_RING[8] library to eliminate the possibility of bottlenecks in both clients and server. PF_RING replaces the default Libpcap as a module in the Linux kernel, to ensure the generation and receiving rates up to 10 Gbps. The OpenFlow functionality in the wireless router was deployed using OVS running in a Linux kernel module. The OVS module used was based on the Openvwrt[9], a module OVS to OpenWRT.

C. Methodology

We have used three scenarios to perform the experiments, namely:

[4] http://www.omnetpp.org
[5] http://netfpga.org
[6] http://www.raspberrypi.org
[7] http://www.ntp.org
[8] http://www.ntop.org/products/pf_ring
[9] https://github.com/pichuang/openvwrt

(i) the wireless router running the original equipment manufacturer (OEM) firmware;

(ii) the firmware OEM substituted by the OpenWRT 12.09 without OpenFlow module;

(iii) the OpenFlow 1.0 support in the OpenWRT firmware, using the OVS kernel module, was enabled.

Performance factors for the experiments were: number of traffic generator clients, packet size, packet sending rate, and the transport protocol. In the experiments using OpenFlow, we have used four types of OpenFlow controllers: POX[10], NOX[11], Beacon[12] (running their MAC learning standard application), and the OVS standalone mode. We evaluated throughput, jitter, one-way delay, and packets losses in order to assess the viability of the OpenFlow use in a commodity wireless router. We repeated the experiments 10 times for each level and we calculated the median of all values observed. The confidence interval for all performance metrics was set to 95%.

First, we compared the performance of the OEM firmware as compared to the OpenWRT without OpenFlow. We aim at observing the performance difference between the both firmwares. Then we analysed the impact of the SDN controllers on selected performance metrics. Last, but not least, we have chosen the controller with the best results to perform the feasibility assessment of OpenFlow in the OpenWRT.

IV. Experimental Results

A. *OEM and OpenWRT firmware performance*

We first analyse the general impact of the wireless router firmware OpenWRT on selected performance metrics. In order to do that, we used the wireless router running original manufacturer's firmware and operating with the OpenWRT firmware, which does not use OpenFlow. We generated different sending rates to identify performance issues between the firmwares.

Table I and Table II show the results obtained for the highest transmission rates tested, with clients using either wired or wireless network. The *Clients*, *Packet Size*, and *Rate* columns are the independent parameters. The *Throughput* column presents the aggregate throughput received at the sink server. The other columns names are self-explanatory. Table I shows that there are no large variations between the performance of the OEM firmware and OpenWRT, for data transmission in the wired network. All metrics, namely, throughput, delay, jitter, and packet loss ratio, are similar among the tested scenarios. For the case of wireless clients (cf. Table II), there is a fairly significant change, because the performance of OpenWRT firmware is significantly higher than the OEM firmware in various scenarios.

[10] http://www.noxrepo.org/nox/about-pox
[11] http://www.noxrepo.org/nox/about-nox
[12] https://openflow.stanford.edu/display/Beacon

TABLE I
FIRMWARE OEM X OPENWRT - WIRED CLIENTS

Clients	Packet Size	Rate (kpps)	OEM				OpenWRT			
			Throughput (Mbps)	Delay (us)	Jitter (us)	Loss (%)	Throughput (Mbps)	Delay (us)	Jitter (us)	Loss (%)
1	64	200	82	785	8	0.00	89	289	7	0.00
	1472	55	609	383	29	0.00	611	582	29	0.00
5	64	40	79	242	26	0.00	79	772	27	0.00
	1472	12	559	210	91	0.00	560	412	93	0.00
10	64	40	114	1322	40	0.40	113	1478	40	0.30
	1472	8	658	975	88	0.40	658	704	94	0.00

For test case with 1 client, 64-bytes packets, and sending rate of 40 kpps, we have obtained packets loss over 40% for the OEM firmware and below 0.5% for OpenWRT. Besides, the throughput observed with OpenWRT is over twice that obtained by OEM. Similar results can be observed for test cases with 5 and 10 clients when using 64-bytes packets. For 1472-bytes packets, the low performance of the OEM firmware is not observed in any of the scenarios, regardless of the amount of clients used in the experiment. This suggests that the implementation of OEM firmware do not offer good performance when dealing with a large number of packets (i.e., in the scenario of 64-bytes packets). Based on such preliminary results, we decided to conduct further tests to assess performance of the OpenFlow firmware, using as reference the router running OpenWRT, since it has initially demonstrated superior performance. Therefore, we can make sure that any eventual poor performance of the OpenFlow firmware will not be a consequence of performance issues of the OpenWRT.

B. *OpenFlow firmware performance*

In order to analyse the performance of OpenFlow firmware, we have set some distinct testing scenarios. First, we have varied the type of the OpenFlow controller (i.e., POX, NOX, and Beacon) that is used to manage OpenFlow requests from the wireless router. We have deployed the well-known traffic generator D-ITG that generates traffic from five clients (i.e., four in the wired network and one in wireless one) to the sink server. We set diverse UDP sending rates for 64-bytes packets (4, 5, 6, 7, and 8 kpps) and 1472-bytes packets (2, 3, 4, 5, and 6 kpps), and collected throughput and packet loss for all cases.

TABLE II
FIRMWARE OEM X OPENWRT - WIFI CLIENTS

Clients	Packet Size	Rate (kpps)	OEM				OpenWRT			
			Throughput (Mbps)	Delay (us)	Jitter (us)	Loss (%)	Throughput (Mbps)	Delay (us)	Jitter (us)	Loss (%)
1	64	40	9	65623	97	40.90	20	15236	42	0.30
	1472	10	107	10078	68	0.00	107	16014	184	0.00
5	64	40	10	59786	86	36.70	20	14200	38	0.40
	1472	10	96	17271	164	0.00	113	12196	166	1.50
10	64	20	11	52376	162	22.60	19	18062	90	2.90
	1472	8	101	22149	366	0.30	116	23522	390	0.40

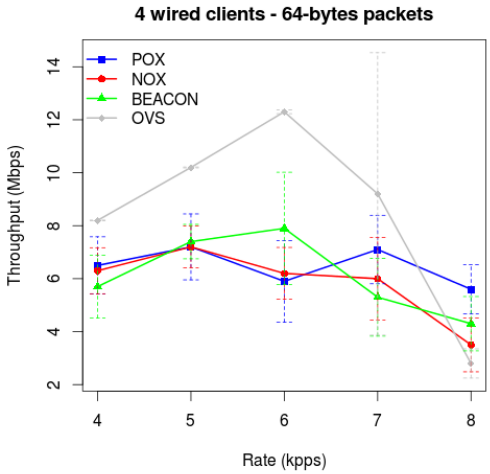


Fig. 2. Throughput for experiments with UDP 64-bytes packets, using different OpenFlow controllers and OVS standalone mode – wired network.

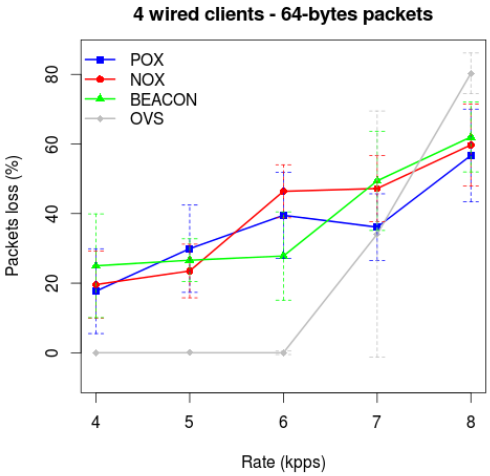


Fig. 3. Packets loss for experiments with UDP 64-bytes packets, using many OpenFlow controllers and OVS standalone mode – wired network.

Fig. 2 and Fig. 3 present the results for clients in the wired network. For 64-bytes packets, we can observe that sending rates above 6 kpps, the receiving rate at the sink server drops dramatically while loss rate increases substantially. From 50% and beyond of loss rate, all controllers reach a saturation point of the routing capacity of the OVS traffic. Similar results were obtained for the cases of UDP packets with size 1472 bytes at the wireless network, when the loss rate greatly increases from 6 kpps.

We realized that regardless the OpenFlow controller type used, the flow management application running on the controller can affect somehow the performance of the data plane. However, the limitations of routing traffic observed in the experiments are directly related to the limitations of implementing OVS, as observed when we used the OVS standalone mode, which is independent of the controller to forward traffic between ports of OVS. Based on that, we are not able to observe no substantial difference in the performance among the SDN controllers. Therefore, we selected the Beacon controller for the other experiments due to other performance results reported in [13] and [14].

Fig. 4 shows the throughput of the OpenWRT firmware with and without the use OpenFlow to send UDP packets of 64 bytes in rates of 4, 5, 6, 7, and 8 kpps, from five client hosts (again, four in the wired network and one in the wireless one). It is straightforward to observe that, for all the sending rates, the use of the OpenFlow firmware shows a much lower performance. The throughput of the OpenWRT firmware is up to 4 times higher than those obtained in experiments with OpenFlow controller using Beacon. In addition, Fig. 5 shows that the OpenWRT without OpenFlow had rates of zero packet loss for all experiments, while using the OpenFlow; these rates reached levels of around 60% when dealing with values of 8 kpps.

These results corroborate previous ones that compare the performance of the SDN controllers (Fig. 2 and Fig. 3), thus highlighting the limitation of implementation to handle a large amount of UDP packets. Based on such results, we have decided to assess whether this 30 kpps (aggregated) bottleneck is at some degree related to the number of network interfaces in the router or it is simply an inner limitation of the processing capacity of the OVS. Therefore, we have used a single wired client traffic generator and we have changed the sending rates, now ranging from 10 to 50 kpps. The wireless router has been configured to operate using the OVS standalone mode, in order to eliminate the influence of any OpenFlow controller in the process.

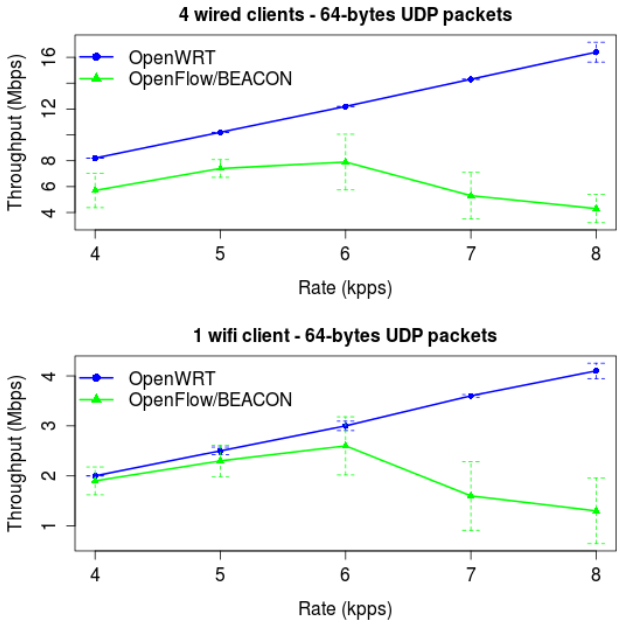


Fig. 4. Throughput for experiments with UDP 64-bytes packets, using OpenWRT and OpenWRT with OpenFlow Beacon Controller – wired and wifi networks.

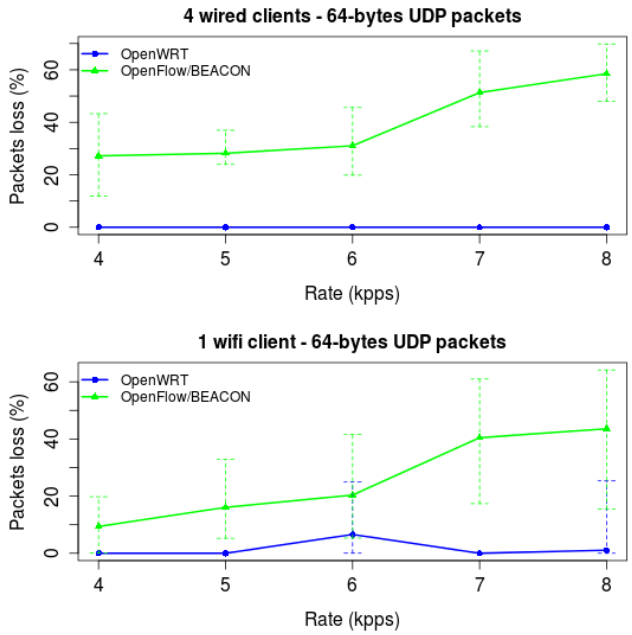


Fig. 5. Packets loss for experiments with UDP 64-bytes packets, using OpenWRT and OpenWRT with OpenFlow Beacon Controller – wired and wifi networks.

The Fig. 6 shows the results obtained in this experiment. We can see that the packet loss is null until 30 kpps, but began to rise at 40 kpps and achieves 60% at 50 kpps. For the throughput, we can observe a growth until 40 kpps and an abruptly decreasing after this value. For the delay and jitter analysis, we observe that they increase after 30 kpps, and the delay exceeds 100 ms to all packets sent, as demonstrated by small jitter value. These results evidence that there is a bottleneck around 30-40 kpps of the OVS module.

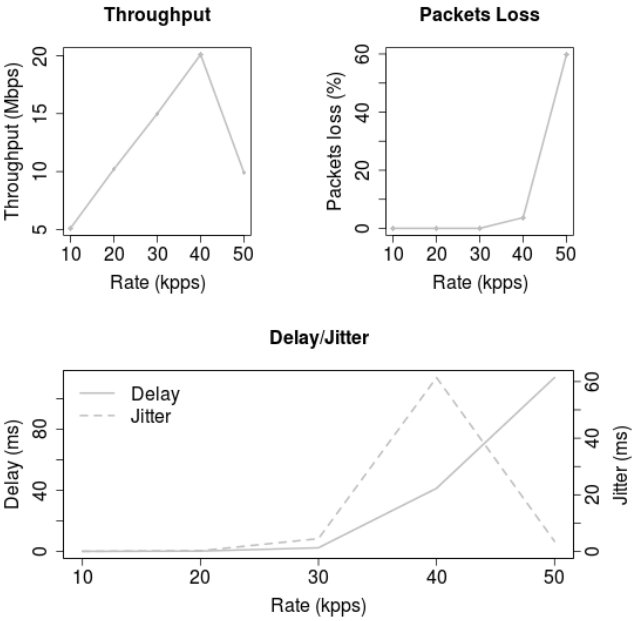


Fig. 6. Throughput, packets loss, delay, and jitter for experiments with UDP 64-bytes packets, using OpenWRT with OpenFlow OVS standalone mode – 1 wired client.

In order to observe the impact of the transport protocol used on the network, we also conducted experiments based on the TCP protocol. The experiment consisted of using five client traffic generators, four in the wired network and one in the wireless one, generating TCP packets with 64 bytes in size at different rates. Rates of 4, 5, 6, 7, 8, 20, 50, 80, 100, and 120 kpps were tested. The rates 4 to 8 kpps are the same as in UDP tests, to facilitate comparison between the results obtained in each experiment. The throughput results are shown in Fig. 7.

We observed that the tests performed with TCP protocol presented emphatically superior results to those obtained with UDP protocol. The throughput curves as a function of generation rate of packets to the router with OpenWRT and using the OpenFlow controller Beacon follow identical to the rate of 80 kpps, per client for the wired network and up to 60 kpps for the wireless client. Furthermore, the differences in throughput values for rates up to 120 kpps are not significant, especially in the wired network.

Apparently, being TCP a stream-oriented protocol, the buffering performed by TCP Nagle's algorithm, causes the merge between the payload of subsequent packets and the payload of already delivered packets, but not yet effectively sent. It reduces the number of TCP segments to be processed by the wireless router, and offers enough that the OpenFlow module can treat adequately, all frames received on network interfaces of the wireless router, a fact that was not observed in the case of UDP protocol when using rates above 40 kpps.

The results for delay and jitter are shown in Fig. 8. We can observe that the jitter values obtained are quite similar, especially in case of the wireless client.

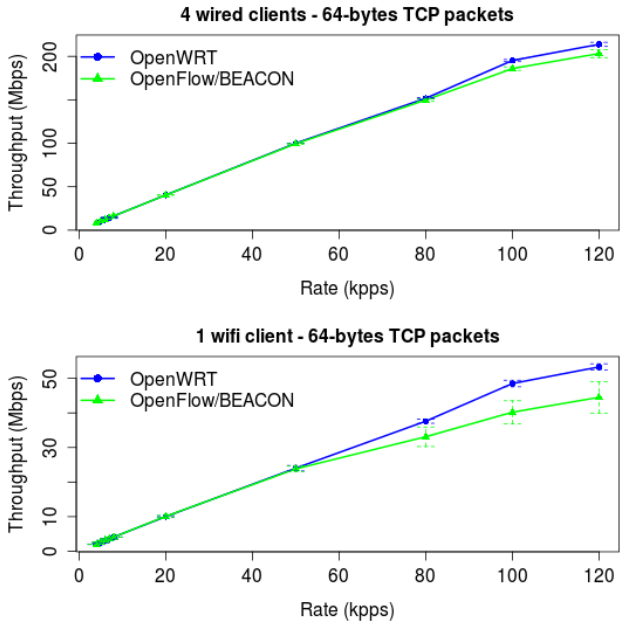


Fig. 7. Throughput for experiments with TCP 64-bytes packets, using OpenWRT and OpenWRT with OpenFlow Beacon Controller – wired and wifi networks.

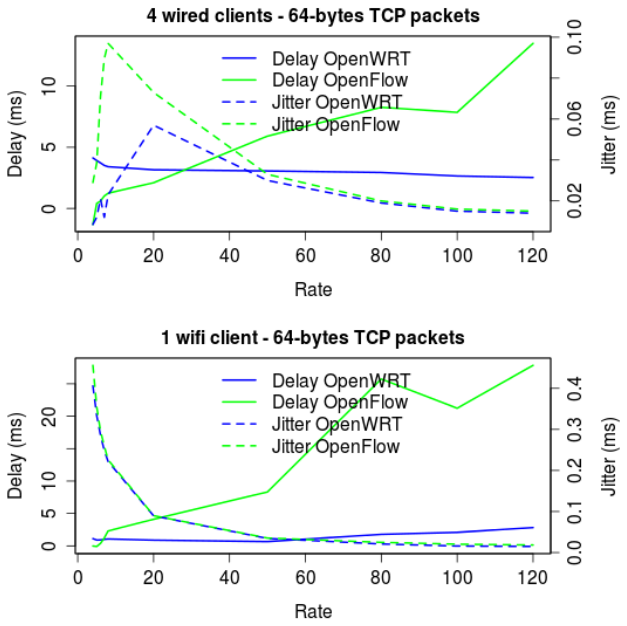


Fig. 8. Delay and jitter for experiments with TCP 64-bytes packets, using OpenWRT and OpenWRT with OpenFlow Beacon Controller – wired and wifi networks.

As far we are concerned to delay, despite tests with OpenFlow have presented values slightly higher than those obtained with the original OpenWRT, values do not exceed 30 ms, even for tests at higher rates.

V. Conclusion

This study evaluated the performance of a commodity wireless router being used as an OpenFlow switch, implemented with an OVS Linux kernel module. Several experiments with distinct scenarios were performed, and we could observe its performance variations when using different OpenFlow controllers, variable packet size, several packets rate generation and traffic source clients, as well as separate transport protocols.

The experiments with the OpenFlow module, using the UDP transport protocol, showed limited results when global traffic rate exceed 30 kpps. We also verified that the performance was similar for the three types of OpenFlow controllers tested, indicating that the bottleneck is at the OVS module used in the implementation of the OpenFlow switch.

Despite such low performance, it is important to observe that bottleneck is related to packets rate processing and can be minimized using transport protocol resources, like TCP Nagle's algorithm, as we could observe in results obtained in experiments using the TCP protocol. In addition, the gain offered by the flexibility provided by the SDN architecture that can be used in the implementation of network applications, justifying the use of the OpenFlow module, although this underperformance presented.

The performance obtained, mainly when using TCP protocol, suggests that a commodity wireless router can be used in scenarios of small networks, as homes and small to medium organizations, contributing to accelerate the pace of SDN adoption using legacy networking devices.

As future work we envisage conducting similar experiments using hardware with high performance, such as NetFPGA boards and native OpenFlow switches, thus allowing us to evaluate the performance differences and any unforeseen bottlenecks.

References

[1] J. Pan, S. Paul, and R. Jain. "A survey of the research on future internet architectures". *Communications Magazine, IEEE*, vol.49, no.7, pp.26-36, July 2011.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. "Openflow: enabling innovation in campus networks," *SIGCOMM ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[3] S. Vissicchio, L. Vanbever, and O. Bonaventure. "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Computer Communication Review*., vol. 44, no. 2, pp. 74–75, 2014.

[4] V. Tanyingyong, M. Hidell, and P. Sjodin. "Using hardware classification to improve PC-based OpenFlow switching". *High Performance Switching and Routing (HPSR), 2011 IEEE 12th International Conference on, IEEE*, 2011. p. 215-221.

[5] T. Fratczak, M. Broadbent, P. Georgopoulos, and N. Race. "HomeVisor: adapting home network environments". *Software Defined Networks (EWSDN), 2013 Second European Workshop on, IEEE*, 2013. p. 32-37.

[6] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. "Flowvisor: A network virtualization layer". OpenFlow Switch Consortium, Tech. Rep, 2009.

[7] P. Dely, A. Kassler, and N. Bayer. "Openflow for wireless mesh networks". Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on, IEEE, 2011. p. 1-6.

[8] G. Araniti, J. Cosmas, A. Iera, A. Molinaro, R. Morabito, and A. Orsino. "OpenFlow over wireless networks: Performance analysis". *Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on IEEE*, 2014. p. 1-5.

[9] R. E. Z. de Oliveira, P. P. Piccoli Filho, M. R. Ribeiro, and M. Martinello. "Parâmetros balizadores para experimentos com comutadores openflow: avaliação experimental baseada em medições de alta precisão". *Simpósio Brasileiro de Telecomunicações*. Brasília, Brazil, 2012.

[10] C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore. "Oflops: An open framework for openflow switch evaluation". *13th International Conference, PAM 2012, Proceedings*. Vienna, Austria, p. 85–95, 2012.

[11] H. Kim, J. Kim, and Y. Ko. "Developing a cost-effective OpenFlow testbed for small-scale Software Defined Networking". *Advanced Communication Technology (ICACT), 2014 16th International Conference on IEEE*, 2014. p. 758-761.

[12] A. Botta, A. Dainotti, and A. Pescapè. "A tool for the generation of realistic network workload for emerging networking scenarios". *Computer Networks* (Elsevier), 2012, Volume 56, Issue 15, p 3531-3547.

[13] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood. "On controller performance in software-defined networks". *USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE)*. 2012.

[14] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky. "Advanced study of SDN/OpenFlow controllers". *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia*. ACM, 2013.

Users facing volume-based and flat-rate-based charging schemes at the same time

Patrick Maillé
Telecom Bretagne, OCIF/IRISA
2, rue de la Châtaigneraie
35576 Cesson-Sévigné Cedex, France
Email: patrick.maille@telecom-bretagne.eu

Bruno Tuffin
Inria Rennes
Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
Email: bruno.tuffin@inria.fr

***Abstract*—In the Internet, the data charging scheme has usually been flat rate. But more recently, especially for mobile data traffic, we have seen more diversity in the pricing offers, such as volume-based ones or cap-based ones. We propose in this paper to study the behavior of heterogeneous users facing two offers: a volume-based one and a flat-rate one. On top of that selection, we investigate 1) the relevance for an ISP to propose the two types of offers, and optimize the corresponding prices, and 2) the existence of a solution to the pricing game when the offers come from competing providers.**

I. Introduction

The common pricing scheme in the Internet is the so-called *flat rate* pricing, where a user pays a fixed subscription fee, and can use the network as much as he wants. But there is currently a trend in telecommunications to move to *usage-based* pricing schemes, where the price you pay depends on your consumption pattern. The reasons invoked for those moves generally involve arguments of congestion and fairness between users. That trend has been observed with the introduction of cap-based pricing in broadband access, and the definition of the charging schemes for wireless networks. For an extensive description of flat-rate versus usage-based pricing, see [1], as well as [2] for a general presentation and discussion about Internet economics.

With the appearance of new wireless technologies (and also the deployment of fiber-to-the-home solutions), users are faced with several options, that are not all priced the same way. In this paper we investigate the co-existence of solutions based on flat-rate pricing with others based on usage-based pricing. Users being heterogeneous in their preferences, their preferred scheme may differ. Focusing on the case of two alternatives (one flat-rate and one usage-based), we analyze the impact of this diversity of users and choices in two settings: in the first one the two solutions are proposed by a monopolist provider trying to maximize revenues, while the second one considers competing providers who set their pricing parameters to attract users in order to make revenue (hence a noncooperative pricing game). Discussions on this can be found in [2] and the references therein, see also [3], [4].

Our model is inspired by [5], where the choice of which scheme to prefer is studied, but not the coexistence of both schemes. The authors of [5] study the case of two schemes jointly proposed to users in [6] to analyze conditions for both schemes to be selected by users, but no revenue-maximization or competition problematics are considered. Our contributions can be summarized as follows:

- for a user with quadratic willingness-to-pay (in terms of the consumed quantity), we express analytically the choice made depending on the user-specific parameters and the parameters of both pricing solutions;
- then, assuming some distribution over user-specific parameters we numerically treat some examples of pricing decisions in the monopoly (revenue-maximization) and competition (non-cooperative game) cases.
- Our results suggest that offering the two options in the case of a monopoly increases very slightly the provider's revenue with limited impact on total demand. Also, when the options are offered by competing providers, the pricing game will end up with a price war, i.e. null revenues, or very limited revenues if we introduce data management costs in the model.

II. Model: pricing schemes and user preferences

We consider two ISPs (possibly controlled by the same entity) offering connectivity services to users. ISP 1 is implementing flat-rate pricing with access price p_1 per user (say, per month, like other values later on), after what the user can consume the amount of volume she wishes at no extra charge. ISP 2 on the other hand implements a volume-based pricing scheme, with still an access charge per user (and per month) p_2, but also a charge per unit of volume q_2, so that the (per month) total charge for a consumed volume v is $p_2 + q_2 v$.

Users are assumed heterogeneous. We index them by a (one-dimensional) *type parameter* θ characterizing their *valuation* (willingness-to-pay) function. We consider without loss of generality a total user mass of 1, and assume a density $f(\theta)$ for value θ with corresponding cdf F. Denote by $V(\theta, v)$ the amount (in monetary units) that a type-θ user is willing to pay for consuming a volume v per month: $V(\theta, v)$ is assumed nondecreasing in v, and constant after some value $v_{\max}(\theta)$ that a type θ user is interested in getting (or that the provider can serve per user; in this case it may not depend on θ) if the service were totally free.

We assume here that the networks are over-provisioned so that there is no QoS issue. This assumption is especially relevant for wired DSL or fiber access networks for instance.

978-1-4673-9408-6/15/$31.00

III. User behavior

We consider in this section that price parameters p_1, p_2 and q_2 are fixed, and study the choice for a particular user, together with her corresponding consumption level.

A user of type θ, if associated to ISP 1 (the ISP implementing flat-rate pricing), will consume a volume $v = v_1^*(\theta)$ such that her net utility $V(\theta, v) - p_1$ is maximized, i.e.,

$$v_1^*(\theta) = \text{argmax}_{v \geq 0}(V(\theta, v) - p_1).$$

But since $V(\theta, \cdot)$ is increasing, we get $v_1^*(\theta) = v_{\max}(\theta)$.

Similarly, if associated to ISP 2 (that implements usage-based pricing), she will consume $v_2^*(\theta)$ with

$$v_2^*(\theta) = \text{argmax}_{v \geq 0} \left(V(\theta, v) - (p_2 + q_2 v)\right),$$

(assuming that this argmax exists).

Overall, a type-θ user will select the option (among {ISP 1, ISP 2, no ISP}) maximizing her utility, considering that the "no ISP" choice yields a null utility. In case of equality between two options (marginal users), we assume that the user chooses the option yielding the largest $V(\theta, \cdot)$, maximizing the generated valuation (like a company trying then to maximize its turnover).

In practice of course, $p_2 < p_1$ otherwise ISP 1 is always preferred to ISP 2. Assume as in our reference paper [5] that valuation functions are quadratic, of the form

$$V(\theta, v) = \begin{cases} \theta v - \frac{v^2}{2a} & 0 \leq v \leq \theta a \\ \frac{a}{2}\theta^2 & v \geq a\theta, \end{cases}$$

with a a constant and θ the (non-negative) user-specific parameter (her type). Such a valuation function corresponds to a demand $D(p, \theta) = a(\theta - p)^+$ with $x^+ := \max(0, x)$ (for more, see [5]), and $v_{\max}(\theta) = a\theta$.

In that case $v_2^*(\theta)$ is obtained from the maximization of $V(\theta, v) - p_2 - q_2 v$, hence from the relation $\frac{\partial V}{\partial v}(\theta, v_2^*(\theta)) - q_2 = 0$, leading to $v_2^*(\theta) = a(\theta - q_2)^+$. In words, $v_2^*(\theta)$ is the level of demand at unit price q_2.

The corresponding utilities for both choices are thus

$$\begin{aligned} V(\theta, v_{\max}(\theta)) - p_1 &= \frac{a\theta^2}{2} - p_1 \\ V(\theta, v_2^*(\theta)) - p_2 - q_2 v_2^*(\theta) &= \frac{a}{2}(\theta - q_2)^2 - p_2. \end{aligned}$$

Summarizing, a type-θ user will choose her service option according to the following threshold rule.

Proposition 1. *Let* $\theta^- := \min\left(\sqrt{\frac{2p_1}{a}}, \; q_2 + \sqrt{\frac{2p_2}{a}}\right)$ *and* $\theta^+ := \max\left(\sqrt{\frac{2p_1}{a}}, \; \frac{q_2}{2} + \frac{p_1 - p_2}{aq_2}\right)$.
Then a type-θ user will

- *i) prefer not to join any provider if $\theta \in [0, \theta^-)$;*
- *ii) choose the volume-based provider if $\theta \in [\theta^-, \theta^+)$;*
- *iii) choose the flat-rate provider if $\theta \in [\theta^+, \infty)$.*

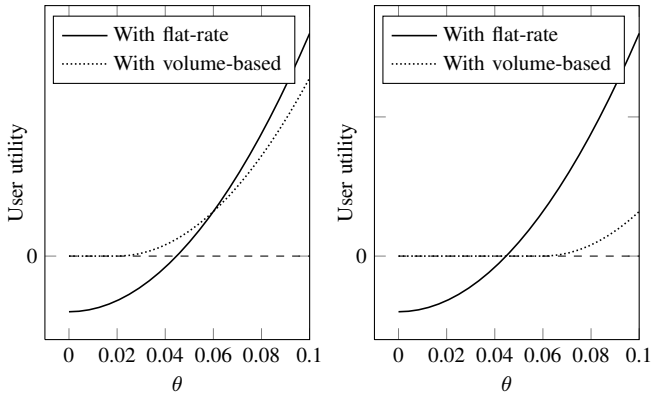
In particular, if prices are such that $\sqrt{\frac{2p_1}{a}} \leq q_2 + \sqrt{\frac{2p_2}{a}}$ then $\theta^+ = \theta^-$ and no user selects the volume-based provider.

Proof: The no-provider option is chosen when we jointly have $\frac{a\theta^2}{2} - p_1 < 0$ and $\frac{a}{2}((\theta - q_2)^+)^2 - p_2 < 0$. Those two inequalities are equivalent to $\theta < \sqrt{\frac{2p_1}{a}}$ and $\theta < q_2 + \sqrt{\frac{2p_2}{a}}$, respectively, giving *i)*. We now assume that $\theta \geq \theta^-$, i.e., one provider is selected.

• If $\sqrt{\frac{2p_1}{a}} \leq q_2 + \sqrt{\frac{2p_2}{a}}$, then the volume-based provider could be preferred to the flat-rate if $\frac{a}{2}((\theta - q_2)^+)^2 - p_2 > \frac{a\theta^2}{2} - p_1$, that is, if $\theta < \frac{p_1 - p_2}{aq_2} + \frac{q_2}{2}$. However, the condition $\sqrt{\frac{2p_1}{a}} \leq q_2 + \sqrt{\frac{2p_2}{a}}$ is equivalent to $p_1 \leq \frac{a}{2}q_2^2 + p_2 + aq_2\sqrt{\frac{2p_2}{a}}$, which yields $\frac{p_1 - p_2}{aq_2} + \frac{q_2}{2} \leq q_2 + \sqrt{\frac{2p_2}{a}}$. Hence $\theta \geq \sqrt{\frac{2p_1}{a}}$ implies that $\theta \geq \frac{p_1 - p_2}{aq_2} + \frac{q_2}{2}$: the volume-based provider is therefore never chosen here.

• Now assume that $q_2 + \sqrt{\frac{2p_2}{a}} < \sqrt{\frac{2p_1}{a}}$. For $\theta \geq \theta^-$ the volume-based provider is selected when $\theta < \frac{p_1 - p_2}{aq_2} + \frac{q_2}{2}$; above that threshold, the flat-rate is preferred. ∎

As an illustration, we draw in Figure 1a the curves of $V(\theta, v_{\max}(\theta)) - p_1$ and $V(\theta, v_2^*(\theta)) - p_2 - q_2 v_2^*(\theta)$, i.e., the utilities that type-θ users would get from choosing Provider 1 or 2, respectively, for the parameter values $a = 2$, $p_2 = 0, p_1 = 0.002$, $q_2 = 0.02$. In this case, $\sqrt{\frac{2p_1}{a}} = 0.0447 >$



(a) $q_2 = 0.02$ (b) $q_2 = 0.06$

Fig. 1: Utility functions for $a = 2$, $p_2 = 0, p_1 = 0.002$.

$q_2 + \sqrt{\frac{2p_2}{a}} = 0.02$, and we indeed observe that under 0.02, no provider is chosen, the volume-based provider is the best option between 0.02 and $\frac{p_1 - p_2}{aq_2} + \frac{q_2}{2} = 0.06$, and the flat-rate one should be chosen above this last value.

We also draw in Figure 1b the case when $q_2 = 0.06$. For those parameter values, $\sqrt{\frac{2p_1}{a}} = 0.0447 < q_2 + \sqrt{\frac{2p_2}{a}} = 0.06$, and we indeed observe that under 0.0447, no provider is chosen, and only the flat-rate one above this value.

In both cases, the curve for the volume-based provider starts at 0 because $p_2 = 0$, but any strictly positive value of p_2 would lead to negative utilities for low θ values. We however say that no ISP is chosen when the user is indifferent between "no ISP" and "ISP 2" here, because the consumed quantity with ISP 2 would be zero.

IV. Determination of prices at the provider level

We now consider the provider point of view, and investigate the pricing decisions that revenue-driven providers would make. To do so we first express the revenue for each offer depending on the pricing parameters, then examine a numerical example of the expected outcome when both options are proposed by the same entity, and when they come from competing provider who may engage in a price war.

A. Expression of the revenue from the two pricing plans

Let m_F (resp., m_V) be the mass of users associated to the flat-rate (resp., volume-based) charging plan. We have

$$\begin{aligned} m_V &= \int_{[\theta^-,\theta^+)} f(\theta)d\theta = F(\theta^+) - F(\theta^-) \\ m_F &= \int_{[\theta^+,\infty)} f(\theta)d\theta = 1 - F(\theta^+) \end{aligned}$$

and the corresponding revenues of providers are

$$\begin{aligned} R_V &= \int_{[\theta^-,\theta^+)} (p_2 + q_2 v_2^*(\theta)) f(\theta)d\theta \\ R_F &= p_1 m_F = p_1(1 - F(\theta^+)). \end{aligned}$$

Since analytical expressions for those revenues become very heavy even for simple distributions of θ, we will present a numerical study in the remainder of this section, to highlight some phenomena that can occur.

B. When the two pricing schemes come from a single provider

We assume in the subsection that the two options are proposed by the same provider. Offering more options may lead to higher revenues because it can allow the provider to segment the market by designing offers targeted at different types of users. The goal is then to find prices maximizing the total gain:

$$\max_{p_1,p_2,q_2\geq 0} R_F + R_V.$$

We consider two different cases, with $a = 2$ in both cases.

1) Log-normal distribution for θ: In a first case, θ is log-normally distributed with $\mu = -1$ and $\sigma = 0.2$. With that type of distribution, the density has a mode, and for those values, we numerically compute the maximum revenue for the provider:

$$\max_{p_1,p_2,q_2\geq 0} R_F + R_V = 8.373 \times 10^{-2},$$

obtained at $p_1 = 0.137, p_2 = 4.43 \times 10^{-2}, q_2 = 9.36 \times 10^{-2}$.

Under those values, we get from Proposition 1 that for $\theta < \theta^- = 0.304$ users do not subscribe to the service, between θ^- and $\theta^+ = 0.542$ they choose the volume-based provider, and the flat-rate one above θ^+. This results in masses $m_V = 0.803$ and $m_F = 0.0264$, and revenues $R_V = 0.0801$ and $R_F = 0.00361$. Hence most revenue comes from the volume-based offer, that is also selected by most users, the flat-rate alternative being used by the biggest users.

For comparison purposes, let us look at the provider revenue, had he offered only one option. If only flat rate was proposed, the optimal price would be $p_1 = 0.1012$ leading to a revenue 7.7547×10^{-2}, while if we only had a volume-based option, the optimal prices would be $p_2 = 4.55 \times 10^{-2}$ and $q_2 = 9.0 \times 10^{-2}$ leading to a revenue 8.3713×10^{-2}. The gain from proposing the two offers is minor here; we do not claim that it will systematically be the case though. The two offers nevertheless allow big users to consume as wished: those with the flat rate option consume a volume $\int_{\theta^+}^{\infty} v_{\max}(\theta)f(\theta)d\theta = \int_{0.54196}^{\infty} a\theta f(\theta)d\theta = 0.0309$, which was only $\int_{\theta^+}^{\infty} v_2^*(\theta)f(\theta)d\theta = 0.02597$ when there was only the volume-based option. Hence a 19% increase of aggregated volume for those "high-θ" users. Remark, to see the proportions, that the total volume for the volume-based option is $\int_{\theta^-}^{\theta^+} v_2^*(\theta)f(\theta)d\theta = 0.4758$.

2) Exponential distribution for θ: If on the other hand, θ is exponentially distributed with rate 1, then we numerically compute the maximum revenue

$$\max_{p_1,p_2,q_2\geq 0} R_F + R_V = 7.358 \times 10^{-2}$$

obtained at $p_1 = 43.35, p_2 = 0, q_2 = 1.0$.

Here, except for a very negligible proportion of the population, only the volume-based scheme is chose. More precisely, $\theta^- = 1$ and $\theta^+ = 22.175$, which results in masses (proportions) $m_V = 0.3679$ and $m_F = 2.34 \times 10^{-10}$, and revenues $R_V = 0.7358$ and $R_F = 1.015 \times 10^{-8}$. We also observe here a negative aspect of monopolies, that is demand reduction with respect to competition situations: the provider sets prices such that less than 37% of users finally subscribe to one plan.

Again, the gain from offering two schemes instead of one is very small: If we had a flat rate offer only, the optimal price would be $p_1 = 3.9998$, leading to a revenue 0.54134, while with volume-based only the optimal prices would have been $p_2 = 0$ and $q_2 = 1.0$ leading to the same revenue as in the two-scheme situation.

C. Pricing game when the two options are offered by providers in competition

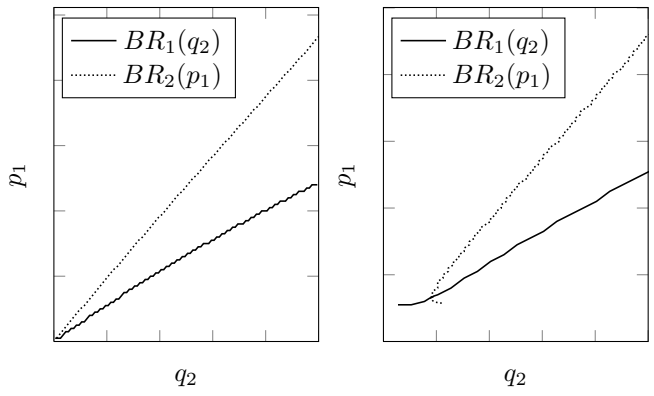
We consider now that the two options are offered by two ISPs in competition, each one trying to maximize its own revenue. We then have a non-cooperative game where both providers play with their price (respectively p_1 and (p_2, q_2) for the flat-rate and volume-based providers).

1) Without any cost: price war: Figure 2a displays the best responses of players when the flat-rate provider plays with p_1 and the volume-based one with q_2 (taking $p_2 = 0$) and F is the cdf of a log-normal distribution with $\mu = -1$ and $\sigma = 0.2$. It can be readily seen that there is no intersection point, so that we end up with a price war: players reduce their price to attract customers, leading to zero prices.

The same result is obtained in the case where θ follows an exponential distribution with rate 1.

2) Game with management costs: Now, let us consider the case when we subtract a cost cm_V and cm_F to the revenues of respectively the volume-based and flat providers, corresponding to a management cost c per unit of mass of customers.

In Figure 2b, we draw the best-response curves of both ISPs when the flat-rate provider plays with p_1 and the volume-based one with q_2 (taking $p_2 = 0$) and the θ parameters are log-normally distributed with $\mu = -1$ and $\sigma = 0.2$. In this



(a) No management cost (b) Management cost $c = 0.01$

Fig. 2: Best-responses when θs are log-normally distributed.

figure, we only draw the best responses leading to a positive revenue, hence there is no best-response when the price set by the opponent is too low.

A Nash equilibrium is then obtained at

$$(q_2 = 0.018104, p_1 = 0.01319)$$

giving revenues (taking into account the costs) $R_F = 0.00150139$ and $R_V = 4.70877 \times 10^{-4}$. Note that these revenues are very small when compared to the monopoly case for which, with this management cost, we obtain an optimal revenue $R_F + R_V = 0.075583$ at $p_1 = 0.143$, $p_2 = 0.049$ and $q_2 = 0.0895$. In this competition case, $m_V = 0.5293$ and $m_F = 0.4707$: all users subscribe to an offer (because $p_2 = 0$), and much more users than in the case of a monopoly choose the flat-rate option.

The decreasing behavior of $BR_2(p_1)$ before starting increasing is intriguing. Figure 3 shows the revenue of the volume-based provider in terms of q_2 for three values of p_1, and confirms that the value of q_2 optimizing R_V is not monotonous in p_1. An explanation for this phenomenon comes from the fact that q_2 affects both θ^- (positively) and θ^+ (negatively). Among the users with parameter close to θ^-, there may be some "costly clients" with small consumption, whose impact on cost exceeds the additional revenue, but ISP 2 cannot deter them from entering the system (by increasing θ^-) without losing "money-making" customers to the competitor (by reducing θ^+). But when p_1 decreases, θ^+ also decreases and the the savings on deterring costly clients from subscribing by increasing q_2 may now exceed the revenue from money-making customers that ISP 2 would lose: in that case ISP 2 is better off increasing q_2. This is possible here because the log-normal distribution has a mode, hence a small change in θ^- may affect a significant proportion of users (the "costly" ones). This phenomenon justifies the introduction of a non-null fixed part p_2 in the volume-based pricing scheme, as a way for ISP 2 to avoid those costly clients.

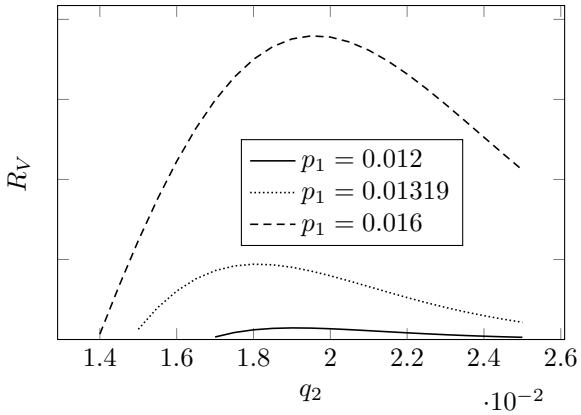


Fig. 3: Revenue for different values of p_1 when when θs are log normally distributed and with a management cost $c = 0.01$.

V. Conclusions and future work

This paper investigates the coexistence of flat-rate and volume-based pricing, where the choice is left to users, but the pricing parameter values are set by revenue-maximizing providers. When both options are controlled by a monopolist provider, our results indicate that the possibility of user segmentation offered by the variety of schemes does not bring large revenue improvements to the provider. On the other hand, if each option is offered by an independent provider, competition drives prices down, questioning the survivability of the providers.

An aspect worth including in future models is the preference of users for flat-rate schemes when the expected price difference with volume-based is low. That preference, apparently contradicting the user rationality assumed in this paper, can be explained by users' aversion to risk and to having to monitor one's consumption. Other directions to investigate include the study of other types of schemes, such as the one proposed in [7] where a flat rate is applied but users can additionally purchase higher-quality service, charged on volume.

References

[1] A. Odlyzko, B. St. Arnaud, E. Stallman, and M. Weinberg, "Know your limits: Considering the role of data caps and usage based billing in internet access service," Public Knowledge, Tech. Rep., May 2012.

[2] P. Maillé and B. Tuffin, *Telecommunication Network Economics: From Theory to Applications*. Cambridge University Press, 2014.

[3] D. Songhurst, Ed., *Charging Communication Networks*. Elsevier, 1999.

[4] J. Mo, W. Kim, and D. Lee, "Impacts of universal service regulation for broadband internet services," in *Economics of Converged, Internet-Based Networks*, ser. Lecture Notes in Computer Science, J. Cohen, P. Maillé, and B. Stiller, Eds. Springer Berlin Heidelberg, 2011, vol. 6995, pp. 14–25.

[5] M. Cho and M. Choi, "Pricing for mobile data services considering service evolution and change of user heterogeneity," *IEICE Transactions on Communications*, vol. E96-B, no. 2, pp. 543–552, 2013.

[6] ——, "Pricing mobile data services of different quality levels," in *Proc. of 3rd International Conference on ICT Convergence (ICTC)*, Jeju, South Korea, 2012.

[7] J. Altmann and K. Chu, "How to charge for network services–flat-rate or usage-based?" *Computer Networks*, vol. 36, no. 5, pp. 519–531, 2001.

Mitigating DoS attacks in Identity Management Systems Through Reorganizations

Ricardo Macedo*, **Yacine Ghamri-Doudane**† and **Michele Nogueira***
* NR2 - Federal University of Paraná, Curitiba, Brazil
† L3I - University of La Rochelle – La Rochelle CEDEX 1 – France
Email: {rtmacedo, michele}@inf.ufpr.br, yacine.ghamri@univ-lr.fr

***Abstract*—Ensuring identity management (IdM) systems availability plays a key role to support networked systems. Denial-of-Service (DoS) attacks can make IdM operations unavailable, preventing the use of computational resources by legitimate users. In the literature, the main countermeasures against DoS over IdM systems are based on either the application of external resources to extend the system lifetime (replication) or on DoS attacks detection. The first approach increases the solutions cost, and in general the second approach is still prone to high rates of false negatives and/or false positives. Hence, this work presents SAMOS, a novel and paradigm-shifting Scheme for DoS Attacks Mitigation by the reOrganization and optimization of the IdM System. SAMOS optimizes the reorganization of the IdM system components founded on optimization techniques, minimizing DoS effects and improving the system lifetime. SAMOS is based on the unavailabilities effects such as the exhaustion of processing and memory resources, eliminating the dependence of attacks detection. Furthermore, SAMOS employs operational IdPs from the IdM system to support the demand of the IdM system, differently from replication approaches. Results considering data from two real IdM systems indicate the scheme viability and improvements. As future works, SAMOS will be prototyped in order to allow performance evaluations in a real testbed.**

I. Introduction

Identity Management (IdM) systems have gained attention from academia and industry due to their potential in integrating different administrative domains, preserving their local policies and technologies [1]. The main advantage of these systems lies in employing authentication authorities, named **Identity Providers (IdPs)**, as guardians of users' critical information, separating resources provisioning (a role of **Service Providers - SP**) of the management of user's critical data [2]. Through IdM systems, users can authenticate at a single domain and access to multiple ones without providing additional information, reducing the management complexity and improving the user's experience [3]. IdM systems can benefit many large-scale systems, such as smart grids, cyber-physical systems [4] and Vehicular Ad-hoc Networks [5].

IdPs are available over the Internet, however they are prone to Denial-of-Services (DoS) attacks [6] that can result in unavailability of IdP operations. IdPs have limited resources, in terms of memory and processing, that can become exhausted under a large number of requests. A DoS attack occurs when a malicious user triggers a large number of requests to overload or exhaust the hardware capacity, generating unavailabilities in resources assignment [7]. Real cases of DoS attacks have been reported even in environments rich in computational resources, for instance in commercial clouds [6], [8], emphasizing the damage of this attack. In IdM systems, a DoS attack can result in unavailability on the identification and authentication of legitimate users, impacting indirectly in services provisioning. In this context, mitigation approaches are promising to minimize the effects of DoS attacks [9].

In the literature, there are different approaches to mitigate DoS attacks effects in networked systems. In a nutshell, they are classified as: (1) migration, (2) replication, (3) resources usage control, and (4) attacker isolation. Migration approaches perform services transferences from one machine to another under attacks [10]. Replication approaches configure many instances of the same services to reply requests in case of failures in replicas. Resources usage control approaches employ access and flow control to servers and networks domains, aiming to reduces the possibility of unavailability in services [11]. However, these approaches are founded on replicas (classes 1 and 2 above), increasing the cost of the solutions; or they are based on intrusion detection systems (classes 3 and 4 above), that in general still yield high rates of false negatives and/or false positives, compromising the detection [12].

This work presents SAMOS, an innovative and paradigm-shifting **S**cheme for DoS **A**ttacks **M**itigation by the re**O**rganization and optimization of the IdM **S**ystem. For the best of our knowledge, this is the first scheme proposing the adaptive reorganization and optimization of the IdM systems' components to mitigate DoS attacks. SAMOS is based on the unavailabilities effects, such as the exhaustion of processing and memory resources, eliminating the dependence of intrusion detection systems. Furthermore, SAMOS employs operational IdPs from the IdM system to support the demand of the IdM system, differently from replication approaches. SAMOS follows three main steps: *Pre-configuration*, *Optimization* and *Utilization*. These steps offer an alternative to reorganize the IdM system in order to mitigate the effects of DoS attacks, balancing the load of authentication operations among IdPs.

Two case studies show the applicability and benefits of

This work has received financial grants from CAPES PDSE program (#99999.007040/2014-08), CNPq, RNP and Araucária Foundation.

978-1-4673-9408-6/15/$31.00 © 2015 IEEE

SAMOS. The first one is conducted using a real dataset of the IdM system from University at Buffalo. The second one uses traces from Federated Academic Community (CAFe), a Brazilian initiative to federate academic identities. In both systems, different probabilities of DoS attacks over IdPs are analyzed.The reorganization speed and the average number of failed IdPs are metrics employed to evaluate the IdM system. Results reveal SAMOS viability as a bottom-up approach, mitigating the attacks effects inside the domains and next in federations.

This paper proceeds as follow. Section II describes the related work. Section III details the system model and the failure model. Section IV presents the proposal of DoS effects mitigation in IdM systems. Section V explains the case study involving a real data set from a IdM system. Finally, Section VI concludes the paper.

II. Related Work

Many initiatives aim to mitigate DoS attacks in the context of Cloud Computing, Content-Centric Networking (CCN) and Software Defined Networks (SDN). In Cloud Computing context, Jia *et al.* presented a mechanism supported by replication to mitigate services prone to distributed DoS attacks [10]. This mechanism use optimization methods to find an alternative in order to separate vulnerable servers from attackers. A case study involving the cloud Amazon EC2 demonstrated the capacity to mitigate distributed DoS attacks in large scale.

Aiming to detect and mitigate DoS attacks in CCN, Compagno *et al.* proposed the Poseidon framework [13]. The authors performed the detection based on anomalies considering distributed and local sources, mitigating the attack through resources usage control. The exploitation of anomalies in network traffic due to DoS attacks also boosted the proposition of a mechanism to detect and mitigate those attacks in SDN architectures [11]. Three modules support the detection and mitigation, data collect, anomalies detection, and anomalies mitigation. The first module collects data in periodic way and transfers this data to next. The second one is responsible to analyze the accuracy of the DoS attack. The third one mitigates the attack through traffic control by SDN settings.

Other works investigated the security and the resilience of operations from IdPs in IdM systems. Barreto *et al.* presented intrusion tolerant IdM infrastructure [2]. Kreutz *et al.* presented an architecture to make IdPs resilient to arbitrary failures through replication techniques. Among then, the authors in [14] discuss the capability of IdPs to tolerate a pre-established number of failures using replication.

In a nutshell, these proposals are founded either on replication approaches or on DoS attacks detection. Initiatives based on attack detection, such as [2], [11] and [13], are prone to false positives and false negatives, compromising the evaluation of attack detection accuracy and then mitigate. Approaches based on replication, as [10], [14], are expensive and can tolerate a pre-established number of failures. Due to these limitations, even employing the existing solutions, a DoS attack can happen, instigating new solutions.

This paper presents SAMOS, a proposal that complements the works in the literature in case where a threshold in the number of failures in exceeded or when the detection of DoS attacks is compromised. SAMOS assumes the use of tools to detect the effects of DoS attacks on IdPs [15], [16], eliminating the dependence of intrusion detection systems. Furthermore, it employs operational IdPs from the IdM system to supply the demand of the IdM system, differently from the traditional approaches based on replication.

III. Models of System and Failure

This section details the system and failures models followed in this work. Subsection III-A presents the main assumptions, the IdM system components, and the system model based on graph theory. Subsection III-B describes the failure model.

A. IdM System

The main components involved in this model are identity, IdM system, IdP, and Service Providers (SP). An identity consists in the digital representation of a real entity in electronic interactions, for example, a person, network services, or devices. We consider an IdM system as a set of software and hardware to process operations related with identities, where IdPs and SPs play important roles. IdPs are responsible to monitor and control identities, providing operations such as authentication in the IdM system. SPs supply specific services to different entities.

In the model, the IdM system lies in a cooperative environment, where a IdP controls and monitors identities requesting services from SPs. It is assumed that IdPs from the same domain trust each other and follow the same security rules, just like occur in big enterprises with many affiliates. Beside this, IdPs employ a secure channel to communicate among themselves, such as Transport Layer Security (TLS). It is also assumed the existence of mechanisms to prevent attackers to insert fake IdPs, such the usage of security certificates.

An IdM system is represented by a graph $G = (V, E)$, where V is the set of vertices originating from the partitioning of three vertices sets, ID, IP and SP. These sets correspond to the identities, IdPs and SPs. Thus, the vertex set of G includes the union of ID, IP and SP. The edges of G are represented by sets $E1$, $E2$, $E3$, $E4$, $E5$, where $E1 = \{(id, ip) | id \in ID, ip \in IP\}$, $E2 = \{(ip, sp) | ip \in IP, sp \in SP\}$, edges among the elements of each set can also exist, $E3 = \{(id_1, id_2) | id_1 \in ID, id_2 \in ID\}$, $E4 = \{(ip_1, ip_2) | ip_1 \in IP, ip_2 \in IP\}$, $E5 = \{(sp_1, sp_2) | sp_1 \in SP, sp_2 \in SP\}$, wherefore $E = \bigcup_{i=1}^{5} E_i$.

Each edges set represents a specific type of relationship. $E1$ represents which IdPs $ip \in IP$ an identity $id \in ID$ can uses to authenticate itself. $E2$ describes which SPs $sp \in SP$ trust in authentications from the IdP $ip \in IP$. $E3$ represents the composition of partial identities of an entity. $E4$ exposes the IdPs affected by DoS attacks. $E5$ details the composition of SPs, i.e., the orchestration of services in service-oriented architectures. In G any type of relationship may exist, except

the association between identities and SPs, because this relationship is not loyal to the sharing of resources in real IdM systems. Fig. 1 illustrates an example.

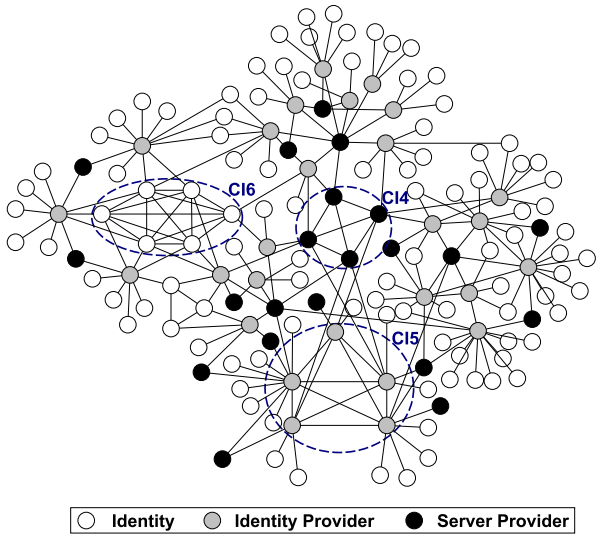


Fig. 1. Graph Example of an Identity Management System

In Fig. 1, the vertices in white represent identities (set ID), in gray represent the IdPs from IP set and, in black, the SPs from SP set. Moreover, there are cycles in G among elements from the same set. Let CIj be a cycle where j vertices are adjacent, the center of the graph shows a $CI4$ among elements from SP, representing a service composed of four SPs. Below the center, there is a $CI5$ among the elements from IP set, characterizing five IdPs under DoS attack. In the center left of the graph, there is a $CI6$ among elements from the A set, describing the union of six partial identities.

A failure is the exploitation of vulnerabilities in the IdM system, that can result in unavailability, disclosure, destruction, or unauthorized modification of the users data [17], [18]. Particularly, it is assumed that a failure is originated by DoS attack in an **IdP**, resulting in unavailability of the IdP operations. In this work, a vulnerability exploitation in an IdP always implies in attempts to exploit the same vulnerability in others IdPs, resulting in the interdependence effect. Consequently, considering the existence of a vulnerability $vul \in IdP_1$ and $vul \in IdP_2$, the exploitation of $vul \in IdP_1$ generates a failure of unavailability in IdP_1 and IdP_2. The sequence of IdPs compromised by the exploitation of a vulnerability interdependently is represented by Pa. Pa also represents the sequence of IdPs with the same vulnerability.

B. Failure Model

This subsection describes a failure model through removal of vertices from G graph. Vertices removed from set $ip \in IP$ represents the unavailability generated by DoS attacks. DoS attack occurrence in an IdM system has as target the IdPs, elements from set IP, impacting in identities ($id \in ID$) and SPs ($sp \in SP$), hence, G become disconnected.

An adjacency matrix represents the IdPs target of a DoS attack. Let G' be a subgraph with that contain the set of edges $E4$, responsible for describing vulnerabilities in common among IdPs. Using an adjacency matrix M, the model describe DoS attacks between an IdP_i, and IdP_j filling an entry $m_{i,j}$ of the M matrix as 1 and 0 otherwise.

Following the related works [19], we denote as $Bin(nt, p)$ the binomial distribution to obtain these values and to represent the probability of DoS occurrence. nt represents the number of attempts from attacks, and p the probability of attack success, where each attempt result in only two possibilities, success (1) or not (0), and p remains constant. In M, the main diagonal of array m is filled with zeros, indicating the absence of edges of a vertex to itself. In order to compose the set of edges $E4$, the values of $Bin(nt, p)$ compose upper and lower triangle, and are symmetrically distributed along the main diagonal. Symmetry of M is guaranteed through the sum of the array with the main diagonal and the lower triangle over your transposed, in other words $M = T(M)$.

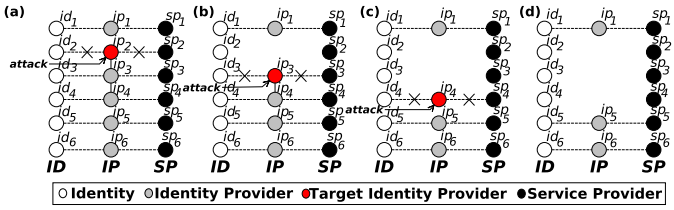


Fig. 2. Iterative Process of a DoS Attack in an IdM System

Through this notation is possible to represent the occurrence of DoS in a set of IdPs. Fig. 2 illustrates the iterative process these attacks in an IdM system. Vertices in white color represent a set of identities stored in an IdP (gray color vertices). Black-colored vertices connected to grays, represent the set of SPs. Edges among ip_2, ip_3 and ip_4 represent that these providers are target of a DoS attack. In (a) instant, the first DoS attack occur on ip_2, removing this vertex. As result the set of identities id_2 and the set of SPs sp_2 become isolated. The second attack occurs in instant (b), where the DoS attack has as target the IdP ip_3, disconnecting id_3 and sp_3. In instant (c) happens the third attack on ip_4, isolating id_4 and sp_4. In instant (d), id_2, id_3 e id_4 represent the set of identities isolated and sp_2, sp_3 e sp_4 describe the isolated SPs.

IV. Proposal to Mitigate DoS Attacks in IdM Systems Through Reorganizations

This section presents SAMOS, the first scheme to mitigate DoS attacks effects in IdM systems through reorganizations of system components. Reorganizations provide resilience to IdPs in face of unavailabilities caused by DoS attacks, enabling the homogeneous distribution of identities and SPs among IdPs. Hence, the mechanism considers the constraints on the reorganizations, making possible to prioritize existing relationships among the components. The reorganizations provide resilience to IdPs in front of unavailability resulted from DoS attacks.

Fig 3 illustrates the three procedures from the proposed mechanism, comprising: *Pre-configuration*, *Optimization* and *Utilization*. *Pre-configuration* adjusts the identities, IdPs and SPs in sets of subsets with the same size, describing possible preferences to the reorganization. *Optimization* finds a solution to the 3DM problem receiving as input an instance of IdM system model. *Utilization* remaps the elements of subsets on IdM system, preserving the relations from the 3DM solution. The execution of the mechanism offers an alternative to reorganize the IdM system to mitigate the effects of DoS attack, balancing the load of authentication operations among IdPs. These procedures are detail next.

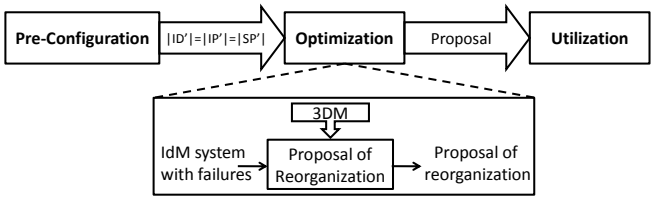


Fig. 3. Procedures of the Proposed Mechanism

A. Pre-Configuration

This procedure receives as input data from IdM system infrastructure and returns a representation of settings to reorganization. The input data describe information about the IdM system condition under the DoS attack, making possible the identification of operational and failure components. This procedure is activated based under the unavailabilities effects such as the exhaustion of processing and memory resources, eliminating the dependence of intrusion detection systems. The procedure describes settings through by triples (id', ip', sp') associated with a weight ben, representing the benefit of each association. The element id' consists in a set of subsets of identities to associate with a set of subsets of IdPs. ip' represents the set of subsets of IdPs, and sp' expresses the set of subset of SPs. Each triple has a value $ben_{id'ip'sp'} \in \mathbb{R}^+$ to express the benefit of the association, where $(1, 1, 1)$ is highest benefit, and $(0, 0, 0)$ the lowest one. This value impact in the way as the system is reorganized, enabling to prioritize some kind of relationships, for instance, to keep in a same set the identities previously stored in an IdP, or the biggest number of association between IdPs and SPs. This procedure returns three sets of subsets with the same size, which are input to the *Optimization* procedure to find a solution of reorganization to IdM system components.

B. Optimization

This procedure finds a set of edges to connect the sets of subsets from identities, IdPs and SPs returned from *Pre-configuration* procedure. The objective is that all sets of subsets from identities and SPs are equally distributed among the operational IdPs. Considering the similarities with the Three-Dimensional Matching (3DM) problem [20], where a optimal configuration must be find to associate the elements from three sets without the repetition of elements in associations, this procedure follow the same steps in regard with IdM system. This procedure has an input three sets, ID', IP' and SP', where $T = |ID'| = |IP'| = |SP'|$. In next, triples (id', ip', sp') are created to represent a probable balanced relationship among the pre-configured sets.

SAMOS solves the 3DM in three steps, the generation of a tridimensional matrix, the reduction to a linear programming problem and the resolution. The tridimensional matrix has a size T^3 and represents the relations and benefits in the reorganization. Each dimension is equal to sets ID', IP' and SP' in 3DM and T describes the size of each sets. The values in this matrix represent the previous relations in IdM system in the moment of DoS attack. The reduction of the matrix to linear programming problem is formalized as:

$$
\begin{array}{lll}
\text{Find:} & x_{id'ip'sp'} \in \{0,1\} & id', ip', sp' = 0, \ldots, T-1 \\
\text{Maximize} & \sum_{id'ip'sp'} ben_{id'ip'sp'} x_{id'ip'sp'} & \\
\text{subject to} & \sum_{id'ip'} x_{id'ip'sp'} \leq 1 & \forall sp = 0, \ldots, T-1 \\
 & \sum_{id'sp'} x_{id'ip'sp'} \leq 1 & \forall ip' = 0, \ldots, T-1 \\
 & \sum_{ip'sp'} x_{id'ip'sp'} \leq 1 & \forall id' = 0, \ldots, T-1
\end{array}
$$

Where, $\sum_{id'ip'sp'} ben_{id'ip'sp'} x_{id'ip'sp'}$ represents the objective function to extract the best relationships. $x_{id'ip'sp'}$ consists in the variable of decision, where $x_{id'ip'sp'} = 1$ describes the selection of the tuple and, $x_{id'ip'sp'} = 0$ otherwise. $\sum_{id'ip'} x_{id'ip'sp'} \leq 1 \ \forall sp' = 0, \ldots, N-1$, $\sum_{id'sp'} x_{id'ip'sp'} \leq 1 \ \forall ip' = 0, \ldots, N-1$ and $\sum_{ip'sp'} x_{id'ip'sp'} \leq 1 \ \forall id' = 0, \ldots, N-1$ consist in limits to constrain the repetition of elements in tuples. The solution to linear programming problem consists in find a optimal combination to relationships considering the benefit ben without the repeat the elements in the tuples, representing a proposal to reorganize the components in IdM system.

C. Utilization

This procedure applies the solution of the reorganization found in *Optimization* procedure in the IdM system. In order to use the 3DM solution, this procedure extracts of the elements from subsets and respecting the relations among the sets. Our mechanism employs the solution of reorganization in the IdM system through migrations of identities among operational IdPs or re-associations between IdPs and SPs. Fig 4 illustrates the procedures of the mechanism in four instants indentified by (a), (b), (c) and (d).

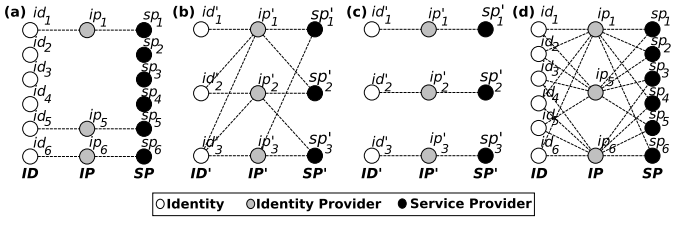


Fig. 4. Operation of Mechanism Proposed

In (a) the graph of a IdM system is under DoS attacks, where the vertices ip_1, ip_5 and ip_6 represent the set of operational IdPs, highlighting the need to reconnect the set of identities id_2, id_3 and id_4 and the SPs sp_2, sp_3 e sp_4 to

operational IdPs ip_1, ip_5 e ip_6. In (b), the *Pre-configuration* procedure forms sets of subsets with size 3, in other words, $ID' = \{id'_1, id'_2, id'_3\}$. Where, $id'_1 = \{id_1, id_2\}$, $id'_2 = \{id_3, id_5\}$, $id'_3 = \{id_4, id_6\}$. $IP' = \{ip'_1, ip'_2, ip'_3\}$. Such as, $ip'_1 = \{ip_1, ip_5\}$, $ip'_2 = \{ip_5, ip_6\}$, $ip'_3 = \{ip_1, ip_6\}$. $SP' = \{sp'_1, sp'_2, sp'_3\}$ and $sp'_1 = \{sp_1, sp_2\}$, $sp'_2 = \{sp_3, sp_5\}$, $sp'_3 = \{sp_4, sp_6\}$. After, *Optimization* procedure finds a possible result to 3DM, as illustrated in c. In (d) the mechanism employs the solution to 3DM to re-associate the elements of IdM system, mitigating the effects of DoS attack. It's important noted that in (d), each set of identities $id \in ID$ and SPs $sp \in SP$ belong at less two IdPs, enabling that the IdM system is not prone to a single point of failure.

V. Case Study of Identity Management Systems

Two case studies are conducted using traces from two real IdM systems: from the University at Buffalo, USA, and from the CAFe Federation, Brazil. Subsection V-A describes the traces. Subsection V-B represents these systems through the system model presented in Section III. Subsection V-C details the results regarding the mitigation of DoS attacks.

A. Description of Traces

Traces collected from a Web Single Sign-on *Shibboleth* system [21] and CAFe federation [22] are used. *Shibboleth* traces are collected from the University of Buffalo IdM system, comprising the period from April 2009 to September 2013. These traces are categorized by month and year, representing the authentications per domain and per service. Authentications per domain account the authentication requests by browsers from internal domains in the network. Authentications per service account and categorize requests by target Web service. In all, 110 files categorized by month are extracted. In order to avoid parse each file, we chose the file referent September 2013 to represent all the others. In this month, the IdM infrastructure has six IdPs and ten SPs, totaling more than two million of authorization requests. These traces can be found on the Website from the University at Buffalo [21].

CAFe comprises a Brazilian initiative for federate identities, in which institutions can act as IdPs and as SPs. This federation permits each user to have a single account in one institution and this account is useful for all services offered. CAFe uses technologies as Security Assertion Markup Language (SAML) [23] protocol and the *Shibboleth* framework in this infrastructure. Statistical information about the adhesion of IdPs and SPs has been collected. The period of the traces comprises from April of 2013 to March of 2014. These traces can be found on the website from *CAFe* [22].

These systems are chosen due to their adoption of technologies used in other academic initiatives, such as Internet2. *Shibboleth* consists of the federated IdM framework more used in academic environments for Web Single Sign-on (WSSO). This technique allows a successful authentication in an IdP from a domain accepted in many SPs of the federation, without the requirement of new authentications for each new session and, consequently, minimizing the need for the users to memorize a large number of passwords [24]. These systems also support shared resources among different domains in decentralized way. Considering scenarios like that, a failure in a point can easily propagate to other domains. These characteristics also make these systems a relevant scenario to investigate DoS attacks.

B. Modeling Real IdMs Systems

In this subsection, the *Shibboleth* IdM system from the University at Buffalo and from the CAFe federation are modeled. Let H be the graph to represent the Shibboleth IdM system from University at Buffalo, keeping the model proprieties, such as $H = (V, E)$. Where, the set of vertices this graph consist in result of the partitioning of three sets, ID, IP, and SP, characterizing identities, IdPs and SPs, respectively, so $V = ID \cup IP \cup SP$, as defined in system model. In this representation, was not possible to extract the exact number of identities to ID set, based on the information available in [21]. This makes difficult to identify the existence of cycles among identities and associations between identities and IdPs. However, this fact does not compromise the objectives of the analysis, since the identities compose a single set and we investigate the Dos occurrence attacks among IdPs.

Differently, the elements from sets IP and SP are available. The IP set, responsible for aggregating the IdPs, presents six elements: *download.acsu.buffalo.edu*, *myub.buffalo.edu*, *ublearns.buffalo.edu*, *ubsis.buffalo.edu*, *myaccount.myubcard.com* and *ubmail.buffalo.edu*. While the set SP, tasked with representing SPs, has ten elements: *messagesystems.com*, *rr.com*, *optonline.net*, *level3.net*, *pavlovmedia.com*, *mycingular.net*, *verizon.net*, *com.sg*, *myvzw.com* and *buffalo.edu*. Fig. 5 illustrates the graph H.

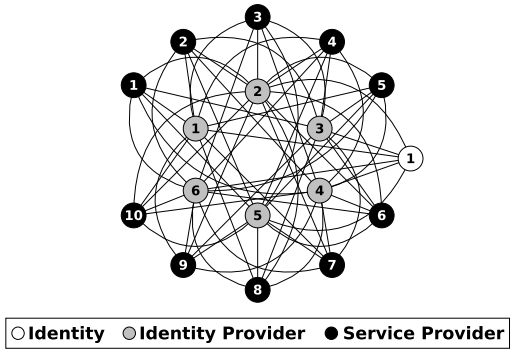


Fig. 5. Graph of *Shibboleth* Identity Management System

In Fig. 5, the vertices in white, black and gray, represent the sets ID, IP and SP. White color vertice is responsible for aggregates all identities of IdM system. Black-colored vertices represent ten SPs from the University at Buffalo, constituting the set SP. Those gray represent the IdPs from set IP. The lines from set ID to each element of IP set, represent the set of edges $E1$. The lines starting from each element of the set IP to all elements of the set SP illustrate the set of edges $E2$. The sets of edges $E3$, $E4$ and $E5$, were not represented in

this figure, because the traces analyzed do not describe these relationships. The subsection V-C describe more details about the set of edges $E4$.

Same procedure is followed to model the *CAFe* federation. We create a graph, separating SPs, IdPs and identities in different sets and then associate them following the definitions presented in Section III. Denote $I = (V, E)$ as a graph to represent the CAFe federation, where V is the parting of three vertex sets, ID, IP and SP, representing, identities, IdPs, and SPs, respectively, so $V = ID \cup IP \cup SP$. The exact number of identities is not available in CAFe traces [22], making hard to identify cycles among identities and the respective associations between identities and IdPs. With this goal, all the identities $id \in ID$ are included in the single vertex. On the contrary, the elements from sets B and C are available.

The set responsible by aggregating the IdPs, denoted as IP, presents a total of 71 elements. Among the IdPs there are institutions related with research and teaching, such as the Brazilian Coordination of Improvement of Higher Education Personnel, Brazilian National Observatory, Brazilian Research and Education Network and also federal and private universities. Whilst the set designed to represent the SPs, set SP, comprise 18 elements. The types of services offered by these SPs cover laboratories environments, video transmissions, grids, medical applications, systems for managing papers submission, applications from Italian academic network and in Hungarian services of video. Fig. 6 illustrates I.

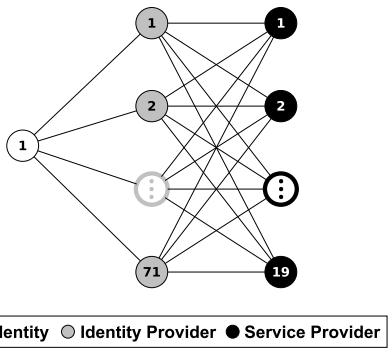


Fig. 6. Graph of CAFe Federation

In Fig. 6, vertex in color black represent the 18 SPs (set SP). Those gray represent the IdPs from set IP. White color one represents the abstraction of identities from *CAFe* (set ID). The edge set $E1$ is characterized by the edges from set ID to set IP. $E2$ has been illustrated by the edges from set IP to set SP. The sets of edges $E3$, $E4$ and $E5$ are not illustrated, because the traces analyzed do not describe these relationships. The subsection V-C describes more details about the set of edges $E4$.

C. Analyze of mitigation of DoS attacks in IdM systems

The objective of this analysis comprises in evaluating the reorganization of IdM systems in face of DoS attacks. The analysis of each system is performed in three stages, in the first maps the subgraph of relations among IdPs through of a matrix of adjacency. The second one uses variations of probabilities of binomial distribution to generate different scenarios of DoS attacks. The third one evaluates the time to reorganize the operational components of IdM system.

DoS attacks occurrence is investigated in each system observing the subgraphs of IdPs. Let H' be the subgraph with six vertices that describes the relationships among the IdPs of IdM system from the University at Buffalo (graph H). We denote as I' the same subgraph of *CAFe* federation. Using this notation, the vertices of H' and I' consist in IdPs and the edges comprise in $E4$ set.

In order to represent these subgraphs adjacency matrices are used. We denote $M1(H') = M1_{6,6}$ as the adjacency matrix to represent the subgraph H', and $M2(I') = M2_{71,71}$ as the adjacency matrix to represent the subgraph I'. In these matrices, an entry $m_{i,j}$ can be $m_{j,i} = 1$, indicating the IdPs i and j are target of DoS attacks, or $m_{j,i} = 0$ otherwise. Main diagonal of each matrix is completed with zeros, indicating the absence of edges of a vertex to itself. Values of the upper and lower triangle are obtained from binomial distribution $Bin(nt, p)$, representing samples of DoS occurrence through edges $E4$. Symmetry of the matrix along the main diagonal is guaranteed through the sum of the array with the main diagonal and the lower triangle over your transposed, in other words $M1 = T(M1)$ and $M2 = T(M2)$.

A binomial distribution was used to obtain the values from the adjacency matrix, generating samples of DoS occurrence. For instance, in order to represent the edges among the six vertices from IdM system of University at Bufallo, fifteen values are required to be symmetrically distributed between the upper and lower triangle, varying between zero and one. Total combinations number of zeros and ones to fifteen values is equal to $2^{15} = 32768$ values. Considering the presence of a large number of IdPs in an IdM system, analyze all the possibilities can be an exhausting task, justifying the analysis of samples.

A range of probabilities p is chosen, comprising the values 0.01, 0.02, 0.03, 0.04, 0.05. Considering each value, we collected 30 samples. As a result of this application each sample of H' and I' subgraphs presents n IdPs with degree 0, representing the operational IdP in a DoS attack.

Two metrics are used: the speed of reorganization and the average number of failed IdPs. We investigate the relation among the probability of success of DoS attack, the number of operational IdP under the attack and the time to find a optimal solution to reorganize the system. In order to measure the speed of reorganization, the metric υ_r is denoted by equation $\upsilon_r = \frac{\Delta_n}{\Delta_t}$. Speed is give by the ratio between the quantity of reorganized IdPs in a interval of time. Where υ_r is the speed of reorganization. Δ_n is the number of operational IdPs under the DoS attack, obtained by subtraction of Number of Failed IdPs (NFIdP) of Number of All IdPs (NAIdP), ($\Delta_n = NFIdP - NAIdP$). Δ_t is the machine time to find a solution with the instance n. Unity of speed of resilience used is n/s, indicating the quantity of IdPs by seconds. Average

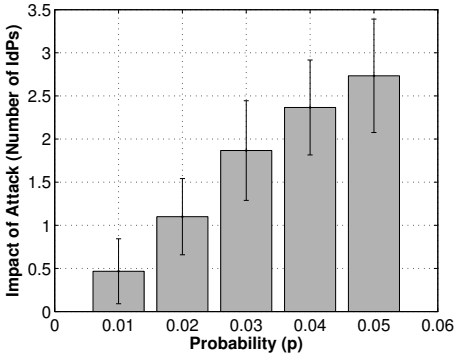


Fig. 7. Average number of failed IdPs in Shibboleth

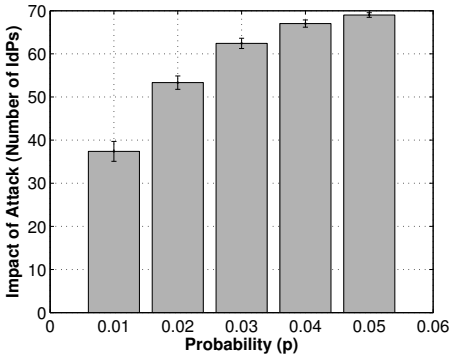


Fig. 8. Average number of failed IdPs in CAFe

number of failed IdPs is denoted as: $\eta = \frac{1}{k}\sum_{i=1}^{k} NFIdP_i$, where k is the number of samples for one value of p and $NFIdP_i$ is the number of failed IdPs to sample i.

Experiments were performed on a Dell PowerEdge T410 computer equipped with a Debian GNU/Linux 6 operating system, kernel version 2.6.32-5-686 and an Intel Xeon E5620 (CPU 2.40GHz), RAM 3GB. Figs. 7 and 8 show the average number of failed IdPs to IdM system from University at Buffalo and CAFe respectively. In both Figures, the average number of failed IdPs was obtained using 95% of confidence interval. Results from the two systems show a increasing number of failed IdPs, generating a interesting situation to analyze the system reorganizations.

Reorganizations of the system are evaluate through an implementation developed with Java Optimization Modeler (JOM) [25]. JOM comprises in an open-source library to model and to solve optimization problems using Java programming. This library solves the 3DM problem using a Mixed Integer Linear Programming (MILP) with support of GNU Linear Programming Kit (GLPK) [26].

In this analysis, it is assumed the elements from ID', IP' and SP' are organized in set of subsets with at least two. Also it is assumed the sets with the same size, i.e, $|ID'| = |IP'| = |SP'|$, and random values to the benefits. Furthermore, it is considered the set of subsets IP' presents always the same size of operational IdPs under the attack. From each sample of M, we pass the operational IdPs to the implementation find an optimal solution to 3DM. Results show SAMOS reorganizing speedily an IdM system with few IdPs. A dispersion graph is used to present the relation between the number of IdPs organized in function of the time, distinguishing each probability variation of binomial distribution with different points. Fig. 9 illustrates these results.

Fig 9 shows the dispersion of the number of IdPs reorganized in function of the time in a shibboleth IdM system from University at Buffalo. The highest execution time lies in less than 0.65 seconds considering the highest number of IdPs reorganized. This result shows that the solution executes in acceptable time to few IdPs, presenting potential to mitigate DoS attacks as the bottom-up approach.

Fig. 10 illustrates the same analysis for the CAFe federation. Results show reorganizations in an IdM system with 20 around IdPs take less than one and half minute. With approximately 30 IdPs, the time obtained increases to 15 hours, and to reorganize 47 IdPs, it is close to 16 days. The long time to compute big inputs in this step can be minimized in two different ways. Optimizations can be applied in fragmented ways in federations, solving the problem firstly in domains and reaching the federation through a bottom-up approach. Beside this, heuristics to find solutions to 3DM can be aggregated in the scheme to decrease the execution time.

VI. Conclusion

This paper presented SAMOS, the first scheme to mitigate the effects of DoS attacks in IdM systems through the reorganizations of the system components. SAMOS is based on the unavailabilities effects such as the exhaustion of CPU and memory resources, eliminating the dependence of intrusion detection systems. Furthermore, it employs operational IdPs from the IdM system to supply the demand of the IdM system, differently from traditional approaches based on replication. Three steps support the reorganizations, *Pre-configuration*, *Optimization*, and *Utilization*. Two case studies involving real datasets are conducted to obtain initial results considering *Pre-configuration*, and *Optimization* steps. Results indicate the applicability the scheme as a bottom-up approach, mitigating the effects of the attack firstly in domains, and lastly in federations. As future works, we intend to *(i)* implement a prototype of the solution through the Security Assertions Markup Language (SAML) standard, and *(ii)* evaluate the performance in other IdM systems.

References

[1] J. Torres, M. Nogueira, and G. Pujolle, "A survey on identity management for the future network," *IEEE Communications and Surveys Tutorials*, vol. 15, no. 2, pp. 787–802, Second 2013.

[2] L. Barreto, F. Siqueira, J. Fraga, and E. Feitosa, "An intrusion tolerant identity management infrastructure for cloud computing services," in *IEEE International Conference on Web Services*, June 2013, pp. 155–162.

[3] P. Arias Cabarcos, F. Almenárez, F. Gómez Mármol, and A. Marín, "To federate or not to federate: A reputation-based mechanism to dynamize cooperation in identity management," *Wirel. Pers. Commun.*, vol. 75, no. 3, pp. 1769–1786, Apr. 2014.

[4] K. Wan and V. Alagar, "Context-aware security solutions for cyber-physical systems," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 212–226, 2014.

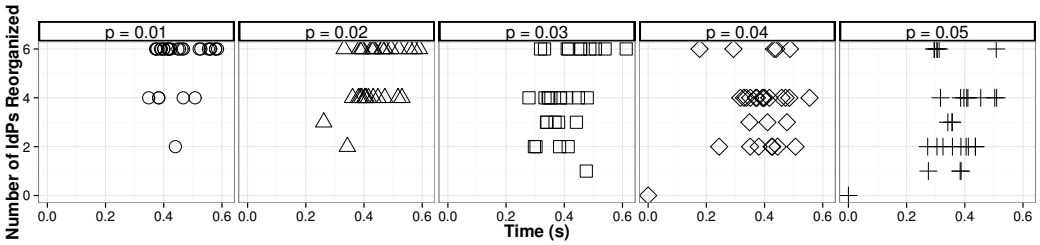


Fig. 9. Dispersion of Speed of Reorganization on *Shibboleth* IdM System

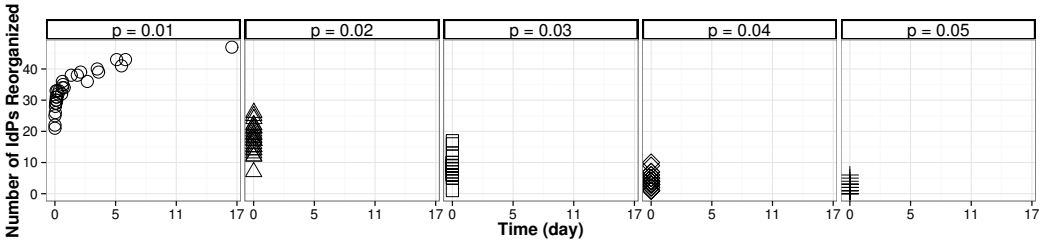


Fig. 10. Dispersion of Speed of Reorganization on *CAFe* Federation

[5] J. Torres, M. Nogueira, and G. Pujolle, "Secure and revocable node authentication in vehicular ad-hoc networks," in *IEEE Symposium on Computers and Communications*, July 2013.

[6] A. Lonea, H. Tianfield, and D. Popescu, "Identity management for cloud computing," in *New Concepts and Applications in Soft Computing*, ser. Studies in Computational Intelligence, V. E. Balas, J. Fodor, and A. R. Várkonyi-Kóczy, Eds. Springer Berlin Heidelberg, 2013, vol. 417, pp. 175–199.

[7] F. R. Carlson, "Security analysis of cloud computing," *CoRR*, vol. abs/1404.6849, 2014. [Online]. Available: http://arxiv.org/abs/1404.6849

[8] H. Shah, S. S. Anandane, and Shrikanth, "Security issues on cloud computing," *CoRR*, vol. abs/1308.5996, 2013.

[9] Z. Fu, M. Papatriantafilou, and P. Tsigas, "Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 3, pp. 401–413, May 2012.

[10] Q. Jia, H. Wang, D. Fleck, F. Li, A. Stavrou, and W. Powell, "Catch me if you can: A cloud-enabled ddos defense," in *IEEE/IFIP International Conference on Dependable Systems and Networks*, June 2014, pp. 264–275.

[11] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Computer Networks*, vol. 62, no. 0, pp. 122 – 136, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613004003

[12] A. Alsumayt and J. Haggerty, "A survey of the mitigation methods against dos attacks on manets," in *Science and Information Conference*, Aug 2014, pp. 538–544.

[13] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding ddos attacks in named data networking," in *IEEE Conference on Local Computer Networks*, Oct 2013, pp. 630–638.

[14] D. Kreutz, O. Malichevskyy, E. Feitosa, K. R. Barbosa, and H. Cunha, "System design artifacts for resilient identification and authentication infrastructures," *International Conference on Networking and Services*, vol. 11, p. 12, 2014.

[15] "Guide: Local monitoring of a shibboleth identity provider," https://shib.kuleuven.be/docs/idp/2.x/install-idp-2.1-rhel-monitoring.html, last access: April. 2015.

[16] "M/monit: Proactive monitoring of unix systems, network and cloud services," http://mmonit.com/monit/#about, last access: April. 2015.

[17] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, Sep. 1994.

[18] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable Secure Computing*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[19] Y. Tan, S. Sengupta, and K. Subbalakshmi, "Analysis of coordinated denial-of-service attacks in ieee 802.22 networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 890–902, April 2011.

[20] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[21] "UB Identity Management and Authentication Metrics," https://ubidm.buffalo.edu/stats/, last access: Oct. 2013.

[22] "*CAFe* Statistical Information," https://portal.rnp.br/web/servicos/estatisticas, last access: Jan. 2015.

[23] Organization for the Advancement of Structured Information Standards, *Security Assertion Markup Language (SAML) v2.0*, Std., 2005.

[24] J. Watt, R. Sinnott, G. Inman, and D. Chadwick, "Federated authentication and authorisation in the social science domain," in *International Conference on Availability, Reliability and Security*, aug. 2011, pp. 541 –548.

[25] J.-L. Izquierdo-Zaragoza and P. Pavon-Marino, "Educational and research tools for network optimization," in *International Conference on Transparent Optical Networks*, June 2013, pp. 1–4.

[26] "GLPK (GNU linear programming kit)," http://www.gnu.org/software/glpk, last access: Sept. 2014.

Privacy-Aware Personal Information Discovery Model based on the cloud

Thiago Moreira da Costa
and Hervé Martin
Laboratoire d'Informatique de Grenoble
Université de Grenoble
Saint Martin d'Heres
{thiago.moreira, herve}@imag.fr

Nazim Agoulmine
IBISC
Université d'Evry
Evry
nazim.agoulmine@ibisc.fr

***Abstract*—Data collection, storage and manipulation have become more critical due to the growth of magnitude of their misuse or mismanagement impact in business and political scenarios nowadays. While research has pushed technology to deliver more powerful information discovery algorithms, and responsive on-demanding storage and processing capacity through data analysis and distributed cloud infrastructure, concerns about privacy have globally raised several discussions involving different sectors of the society. In particular, individual rights are highly impacted by privacy issues due to nowadays geographic distribution of sensitive information and its discovery.**

In this work, we present a model for privacy awareness during the data analytics process in a context of scalable computing using the cloud. Our approach addresses privacy issues both in data analytics process and in the infrastructure resource allocation according to privacy regulation in Service Level Agreements (SLA). The proposed model for Privacy-Aware Information Discovery (PAID-M) provides privacy awareness by executing data analytics algorithms encapsulated with privacy preserving techniques. The model also presents how it intends to address the privacy issue in the cloud deployment process by considering differences in privacy regulations and jurisdictions.

I. INTRODUCTION

The pace on which information has modeled the world where we live has amazingly increased. Indeed, technology has changed the way organizations and people produce and treat data, making possible to produce and share information unprecedentedly faster. Furthermore, Data Science has allowed us to consume information quicker and more efficiently by aggregating and selecting relevant information from large amount of data using data mining techniques. This myriad of aspects of data, and its production, analysis and dissemination techniques have also led to the change of how individual's privacy has been compromised.

A recent report of the Execute Office of the President (USA)[1] about big data and privacy highlighted several problems about personal privacy, the society is facing now or in a near future: intrusion upon seclusion, public disclosure of private facts, disclosure of inferred private facts (sometime false positives), defamation using inferred private facts, invasion of unencrypted private communication, invasion in personal virtual space, stalking and violation of locational privacy, foreclosure of self-determination, lost of autonomy, and others.

While some of these issues are related to information leak, and consequently involve broader governance and systematic to minimize the collection and analyze of data by third-parties, some of privacy threats ramifications are related to new big data technologies that are able to extract valuable information from large volumes[2]. In fact, the continuous ratio growth of data volume, velocity, variety and veracity in the big data scenario is its main challenged and distributed computing are one of the strategy to overcome these technical constraints [3].

The modern technology used to mining semantic enriched data and the computational power provided by nowadays distributed and cloud computing to process huge data sets are permitting fast investigation/correlation of relevant information (decreasing time to discovery private personal facts) and high significance level of inferred information (minimizing the ratio of false positives). Big data storage and analytics technology made it possible to store data produced from web and social interaction, machine logs, sensing, transactions and Internet of Things[4], not limiting in some selective relevant information only. The services provided by modern applications in portable devices that are context-sensitive, social and location-based depend on this data to deliver smart feature and customize users' experience, behaving accordingly to the individual's context and situation. However, the indiscriminate storage of personal digital data has led to a practice of *life logging* [5] which has the potential to be intentionally exploited to extract private personal information that was not initially intended by individuals who use these services.

Furthermore, big data technology is increasing the complexity in privacy policy implementation, since it differs from the traditional privacy management where private information was meaningful for the individual who owns it, and it could be classified as sensitive or not, such as race, health status records and salary. Big data technology can infer information from raw data that are not clear for those who produce it, for instance, work place location or risk of loan eligibility.

Besides that, algorithms to prepare data, reduce sensor noises, remove outliers, compress data, integrate heteroge-

978-1-4673-9408-6/15/$31.00

neous data are continuously proposed, leveraging the quality of data stored. In the field of semantic trajectory analysis, for instance, algorithms for outlier removal, kernel smoothing, and compression prepare spatiotemporal data (GPS points) for posterior processing[6]. Other techniques for normalization, integration and filtering may be applied to prepare data for analytics algorithms as well. Furthermore, the best practices for publishing and connecting structured data on the Web, called Linked Data[7], are changing the way data is stored, retrieved, and integrated by using semantic web technology and adding contextual information to them. This allows data analytics algorithm to take into account several aspects of context on which these data were produced, leveraging the level of significance of information extracted from personal data of individuals[8]. Recent cases in Facebook perceiving individuals' moods have disclosure the level of detail currently achieved by big data technology in social networks using history of annotated digital traces [9][10]. This threat to personal privacy become more complex when added to the fact that distributed infrastructure, such as the cloud, is used to achieve this analytics results.

Cloud Computing has a particular role in this scenario specially because its distributed infrastructure bring complex privacy issues [11], [12]. Hashem et al. [4] describe several security and privacy concerns in the cloud that needs to be accounted according to the recent privacy regulations, such as encryption, privacy-safe query, data protection architecture, social network sensitive information publishing, and statistical privacy attack. However, it is in privacy jurisdictions that most regulation decision have focused, pushing personal privacy relevance to a critical level, similar to what was formerly found in business multi-tenant requirements for the cloud [13].

The same characteristics that are required by multi-tenant environments, such as privacy governance and accountability, compliance to regulations, jurisdictional clearance, and service-level agreements also apply to software and services that storage and process personal data of individuals. Therefore, jurisdiction remains a problem to be solved in cloud computing. Restrictions for cross-border data flow for storage or processing may applied in order to guarantee that personal privacy conditions are respected despite of the service provider's location, as currently states in the Russian Statute on *Roskomnadzor* (Russian Federal Service for Supervision of Communications, Information Technology, and Mass Media) and in the European Data Protection Regulation, for instance.

Nonetheless, cloud's technology competitiveness relies heavily on its delivery and deployment models [11] that allow cost reduction and efficient resource allocation due to the cloud elasticity across multiple IaaS providers. How cloud's server vendors can provide such elasticity and still be compliant to multi-jurisdiction privacy regulations?

In this context, three aspects of privacy must be addressed to propose a privacy-safe environment for personal information discovery: i) policies that take into account the direct or indirect relation between raw data and information semantically relevant; ii) privacy-aware programming platform that provides *composable* and extensible data mining modules; iii) privacy sensitive distributed platform that provide elastic storage and processing capacity.

For this propose, we introduce in this work a Privacy-Aware Information Discovery Model (PAID-M) to deploy a reliable, scalable and privacy sensitive data mining system. The main contributions of our work are:

- A privacy policy management that is able to preserve data that can be used to infer sensitive information from being published. By applying privacy policy based on taxonomies, this approach uses semantic reasoning to find direct and indirect relations between privacy policy data.
- *Privacy encapsulation* of traditional data mining algorithms by modularizing them while verifying their inputs and outputs. We define solution for reuse traditional data mining algorithms by supplying a preparation phase and a post-processing phase to verify privacy. Furthermore, a concept of data mining module allows combining an array of data mining algorithms, providing a solution to build information discovery requests.
- Scalable processing and storage using the cloud and interacting with MOSt, a Multisite Orchestration System[14], in order to provide privacy sensitive cloud elasticity.

The rest of this paper is organized as follows: Section 2 describe the privacy-aware information discovery model, describing step by step the rational used to address the privacy problem in the two levels of the solution (programming and infrastructure abstractions). In Section 3, the global architecture of the solution is explained, each components and the technology that is expected to be implemented. Finally, conclusions and next steps of the work are presented.

II. PAID-M - Privacy-Aware Information Discovery Model

[15] classifies technology required for Data Analytic in two categories: *Programming Abstraction* and *Infrastructure Abstraction*. The *Programming Abstraction* aggregates those algorithms that make sense of data, extract non obvious patterns, and predict future trends and behaviors. The *Infrastructure Abstraction* is the technology responsible to provide a scalable, fault-tolerant, and safe environment for data analytics algorithms. In this context, we propose a Privacy-Aware Information Discovery Model as a platform-independent model to address the problem of privacy of personal data in a scenario of big data information discovery focusing in the programming and infrastructure abstractions. Our model address privacy in the level of *Programming Abstraction* by supporting the data analyst through the process of building an analytic process in order to be able to intermediate the execution of analytic processes and verify privacy according to the individual's policy. For this matter, it is necessary to cover what are the types of data analytics algorithms and understand how these algorithms can be concatenated. Furthermore, for the *Infrastructure Abstraction* our model propose an infrastructure that provide parallelized, scalable, fault-tolerance and safe using the cloud. In this level of infrastructure, privacy is

implemented by intervening deployment plans and resource allocation in a way that SLAs for privacy and regulation are accounted. Different components in our model can be deployed according to the individual's privacy policy and SLA. In the next subsections, different aspects of our models are discussed.

A. Programming Abstraction

The *Programming Abstraction* is part of the technology intended to provide an accessible, small and composable interface to reuse techniques of KDD, data mining, text mining, statistical and quantitative analysis, explanatory and predictive models, and advanced and interactive visualization to drive decisions and actions, as described in Big Data Analytics[16]. Several works have studies the common characteristics of analytics algorithms and the Data Analytic Workflow (DAW)[17], [18], [19], [20], [21], [1], [22], [23], [16], [24]. The definition of a high-level class of data analytics algorithms on top of big data allows algorithms reuse, modularization and possibility to express different kinds of relationships among classes. As an example, the work described in [24] proposes mega-modules as super classes of analytics algorithms capable of compose any DAW using these modules. Figure 1 depicts the DAW overview for a single step of data analysis and what is expected in each step. This strategy for encapsulating the analytic steps in modules containing preparation, analytics and result phases has been successfully implemented in several projects, such as GeoDeepDive[22], M-Atlas [25], and Apache Flink (former Stratosphere[18]).

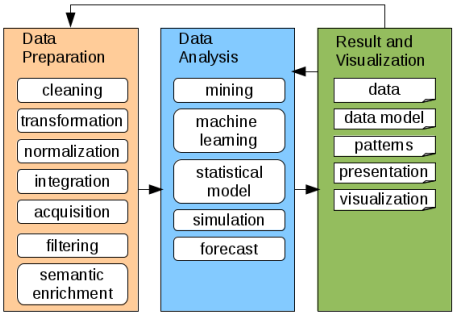


Fig. 1. Data Analytic Workflow Overview

Although some of these projects has partially covered privacy concerns, *privacy enforcement* is not formerly considered into their DAW. As an example, M-Atlas, which works with spatiotemporal data, provides a high-level query language called DMQL [26] that can be used to express DAW's. Nonetheless, DMQL lacks expressiveness to incorporate other privacy preserving techniques than *k*-anonymization[27]. Besides that, the privacy preserving techniques is only used when expressed by the data analyst. Another issue concerning the privacy in the process of data analysis is interpreting its result. Sometime the results are not intelligible or not conclusive. In a traditional Data Analytic scenario, the data analyst concludes if the result is meaningful or if it needs to be re-executed, discarded, validated or adjusted. The output of data analytics algorithms are mainly data, data models and patterns[20]. Data and patterns are mostly result of data mining techniques, while data models can be output from machine learning, statistical model algorithms, simulation and forecast. For the matter of understandability of the results, visualization and presentation can be necessary as well.

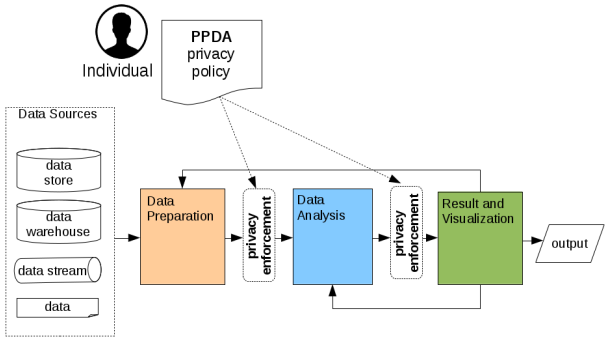


Fig. 2. Privacy Enforcement in Data Analytic Workflow

In PAID-M, we propose enforce privacy in the DAW by incorporating privacy preserving techniques according to the personal privacy policy, as seen in Figure 2.

Among the myriad of existing techniques to be applied to the data during the preparation and post-analysis steps, data characteristics and data types are properly accounted by the data scientist during the Data Analytic Process (DAP[1]) definition. It is no different when considering the privacy preserving techniques. According to [28], privacy-preserving data mining and algorithms can be classified in randomization methods, group-based anonymization, distributed privacy-preserving data mining, privacy-preservation of application results. Some examples of such techniques are as follows:

- Randomization method by adding noise in order to mask attribute values;
- *k*-anonymity and *l*-diversity methods reduce the possibility of indirect identification of individual who produced by generalizing (reducing granularity of the data representation) and suppression;
- Horizontal and vertical partitioning of data sets cross multiple sites in the data mining process, sharing partitions in a way that minimize the possibility of an individual entity;
- Downgrading of application effectiveness, such as rule mining hiding, classifier downgrade, and query auditing to minimize privacy violation by reducing algorithm effectiveness.

Both data analytics algorithms and privacy preserving techniques have a semantic in the DAP that must be observed. That means these techniques are context-sensitive and its application varies according to the data (and its characteristics)

[1]Data Analytic Process is an instance of data analysis following the Data Analytic Workflow

to be processed and to the result analysis. For the PAID-M, this semantic and meta-data (of techniques and data), must be mapped in order to provide adequate techniques during the process of composition of the DAP. Furthermore, this semantic annotation and meta-data can trigger the automatic privacy enforcement by using semantic web technology for reasoning which privacy preserving techniques could be used according to each step of the DAP.

Concerning the privacy policy, the solution proposed in our model is composed by two parts which are implemented in different abstractions of the process. In the *Programming Abstraction*, the Privacy Policy for Data Analytic (PPDA), as depicted in Figure 2, describes the individual's intention for data use, indicating in terms of what aspects of her life should be considered private. In the execution of a DAP, for each step, semantic annotation of data analytics algorithm will guide the system to apply privacy preserving techniques according to PPDA. By doing so, the strategy of PAID-M for privacy preservation goes beyond the all-or-nothing paradigm of policy data access control, taking advantage of privacy preserving techniques to prepare data and to verify the result of data analytics algorithms.

B. Infrastructure Abstration

The *Infrastructure Abstraction* is the technology that provides the solution for the big data challenges using cloud computing. Cloud computing provide many features such as parallelization and distributed computing that are extremely important to overcome big data barriers. Although, while some of the concepts behind cloud computing are not new, from the point of view of scalability, virtualization technology aggregate the most convenient on-demand capacity for data analytic processes. The elasticity property of the cloud allows a very fast increment or decrease of resources, being transparent for the Virtual Machines (VMs). This is done by the virtualization technology (hypervisor) that intermediates physical machines and operating systems (OS), as shown in Figure 3, in order to manage and allocate resources.

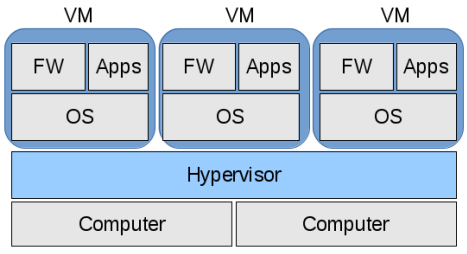


Fig. 3. Example of cloud virtualization

In this context, infrastructure are offered as a service (IaaS[2]), where cloud providers can provide a physical infrastructure (mainframe, servers, network, storage). In order to make this physical infrastructure available, cloud providers must use a IaaS cloud platform to manage the physical resources, such as OpenStack[3] or Amazon Web Service[4].

In a market that starts to become more heterogeneous, with the advantage of private clouds and the increase of cloud vendors, choosing which IaaS to contract is not an easy task. SLA4Cloud[14] is a project that aims to dynamically provide optimal and composition placement of virtual machines delivering better network capabilities and performance trades according to the specified SLA. The Multisite Orchestration System (MOSt) intermediate the process of deployment serving as a hub for several IaaS cloud providers. However, some SLA requirements are still difficult to achieve, due in part to the lack of mechanisms and tools to ascertain if the cloud vendor maintains compliance with the contracted SLA, in part because there are still different SLA requirements that is not easily implemented and monitored due its multidisciplinary characteristics. Privacy can be considered as such a challenge [11], [12] that involves policy, regulation and technology[1]. Its implementation is not trivial, in view of the nowadays constant changing in regulation in the world. This question issues the cloud computing directly, because the cloud elasticity is directly affected and restricted by regulation of personal privacy in some countries that restricts cross-border data flow, for instance. In our model, we propose to cover privacy regulations in the *Infrastructure Abstraction* by extending MOSt[14] with the capability to identify which IaaS cloud vendors are not eligible due privacy regulation restrictions. The PPDP will contain SLA that will specify privacy requirements, such as jurisdiction restrictions and other constraints related to Privacy Regulation. Figure 4 present the interaction between the system and the cloud.

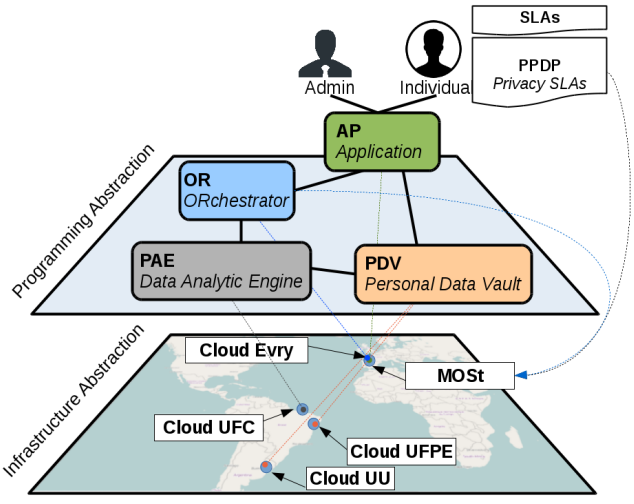


Fig. 4. Example of privacy enforcement in the cloud

[2]Infrastructure as a Service

[3]http://www.openstack.org/

[4]http://aws.amazon.com/

III. Global Architecture

The system to be implemented using the PAID-M will be based in open-source technology and the infrastructure of cloud federation currently maintained by universities in Brazil, France and Uruguay. The *Infrastructure Abstraction* implementation of this work intend to use the infrastructure set up for the SLA4Cloud project, using the IaaS provided by the universities cited above and noted in Figure 4.

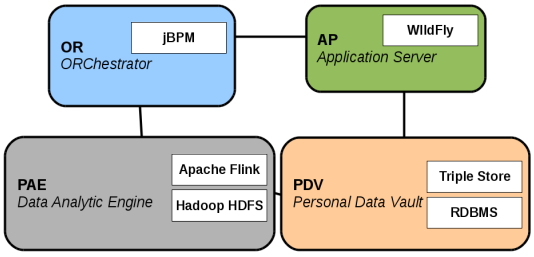


Fig. 5. Architecture Overview

For the *Programming Abstraction* implementation, the four main components, presented in Figure 5, are the followings:

- The ORchestrator (OR) component is the responsible for starting and stopping the other three components. It is the first to be deployed and also controls the queue for data analysis. The Orchestrator is intended to be implemented using Wildfly[5] and jBPM[6] technology;
- The Application Server will be responsible for interacting with administrator, data producers and data analysts. Interfaces to upload data, submitting DAP and visualizing results is expected to be implemented in AP using WildFly;
- The Personal Data Vault (PDV) will be in charge of storage of personal data. For each data producer, one account will be created in order to protect and isolate private data. Two type of storage will be available to support different type of data, a Triple Store and a Relational Database Management System (RDBMS);
- The Data Analytic Engine (DAE) is responsible for processing the DAPs. In order to do that, the Distributed FileSystem HDFS[7] should be implemented to provide cache and support for the Apache Flink.

There are several open-source projects for large data sets processing, such as Apache Spark, Apache Storm, Apache Flink, Apache Tez, and the traditional Apache Hadoop-MapReduce [8] and all variation based on market's implementation and requirements. Apache Hadoop 2 and its YARN (Yet Another Resource Negotiator - Resource Manager) predominates in the large data set processing engines currently available. For a matter of succinctness and objectiveness, the parameters used to compare the available engines are not going to be presented. Although, it is important to highlight what makes Apache Flink[9] the best candidate for the DAE.

Formerly known as Stratosphere, Apache Flink is a project sponsored by public and private sectors in Europe Union to empower data scientists to conduct complex data analysis while providing parallelization, query optimization, broad infrastructure and source connectivity, flexible composition of analytics process, automatic optimization in different stages of the process, high performance run-time, rich programming languages to interact along the stages of the analytics process [29]. Besides that, Apache Flink is a data analytic platform that enables the extraction, analysis, and integration of heterogeneous data sets, being capable of performing information extraction and integration, traditional data warehousing analysis, model training (and machine learning), and graph analysis using a programming model based on second order functions[18]. The platform can run standalone, natively in computer clusters, or Hadoop clusters via YARN(see Figure 6).

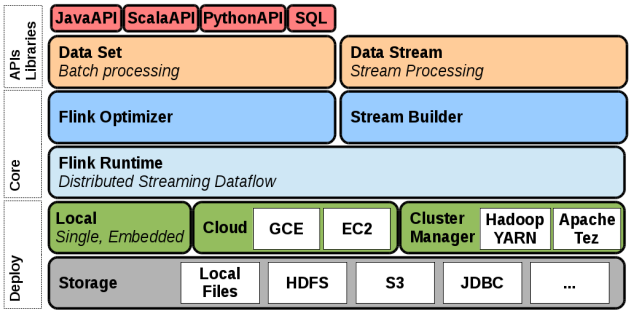


Fig. 6. The Flink Stack

IV. Conclusions

The model presented in this work aims to provide a systematic that will support data consumer to control privacy in the era of big data. Two challenges related to privacy are faced by the Data Analytic in large data sets. The first related to the extraction of private information from data using data analytics algorithms (not clear to the individual who produced). The second related to the cloud computing elasticity, that poses a problem of privacy jurisdiction and regulations. In the next phase, we aspire to implement the PAID-M and run case studies using spatiotemporal data sets, evaluating if individual' s privacy policy were satisfied.

Acknowledgment

This research is partially funded by the Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education (CAPES/MEC) and the EU EASI-CLOUDS project (ITEA 2 #10014) and STIC-AmSud SLA4CLOUD project(14 STIC #11).

[5]http://wildfly.org/
[6]http://www.jbpm.org/
[7]http://hadoop.apache.org/hdfs/
[8]https://hadoop.apache.org/
[9]http://flink.apache.org/

Thanks to all the partners of the project who have helped with their discussions to improve the research work presented in this paper.

REFERENCES

[1] P. C. o. A. o. S. PCAST and Technology, "Big data and privacy: a technological perspective," Tech. Rep. May, 2014.

[2] A. Cavoukian and J. Jonas, *Privacy by design in the age of big data*. Information and Privacy Commissioner of Ontario, Canada, 2012.

[3] A. Jacobs, "The Pathologies of Big Data," *ACM Queue*, vol. 7, no. 6, p. 10, 2009.

[4] I. A. T. Hashem, I. Yaqoob, N. Badrul Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of Big Data on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437914001288

[5] K. O'Hara, M. Tuffield, and N. Shadbolt, "Lifelogging: Privacy and Empowerment with Memories for Life," *Identity in the Information Society*, no. 2008, pp. 155–172, 2009. [Online]. Available: http://eprints.soton.ac.uk/267123/

[6] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "Semantic Trajectories: Mobility Data Computation and Annotation," vol. 9, no. 4, 2012.

[7] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009. [Online]. Available: http://eprints.soton.ac.uk/271285/

[8] C. Bizer, P. Boncz, M. L. Brodie, and O. Erling, "The meaningful use of big data: Four perspectives - Four challenges," *SIGMOD Record*, vol. 40, no. 4, pp. 56–60, 2011. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84857148573{\&}partnerID=40{\&}md5=4662920b41e2753d23919fb73ee5c6dc

[9] E. Bakshy, S. Messing, and L. Adamic, "Exposure to ideologically diverse news and opinion on Facebook," *Science Press*, 2015.

[10] E. C. Tandoc, P. Ferrucci, and M. Duffy, "Facebook use, envy, and depression among college students: Is facebooking depressing?" *Computers in Human Behavior*, vol. 43, pp. 139–146, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0747563214005767

[11] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and Privacy Challenges in Cloud Computing Environments Cloud," *Cloud Computing*, no. December, 2010.

[12] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371–386, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2013.04.028

[13] European Network and Information Security Agency (ENISA), "Cloud Computing: Benefits, Risks and Recommendations for Information Security," Tech. Rep., 2009. [Online]. Available: www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment/at{_}download/fullReport

[14] E. H. Cherkaoui, E. Rachkidy, M. Santos, P. A. L. Rego, J. Baliosian, and J. N. De, "SLA4CLOUD : Measurement and SLA Management of Heterogeneous Cloud Infrastructures Testbeds," *3rd International Workshop on Advances in ICT*, pp. 1–6, 2014.

[15] A. Kumar, F. Niu, and C. Ré, "Hazy: Making it easier to build and maintain big-data analytics," *Communications of the ACM*, vol. 56, no. 3, pp. 40–49, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2428570

[16] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. a.S. Netto, and R. Buyya, "Big Data computing and clouds: Trends and future directions," *Journal of Parallel and Distributed Computing*, 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0743731514001452

[17] R. L. Grossman, "What is analytic infrastructure and why should you care?" *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, p. 5, 2009.

[18] A. Alexander, B. Rico, E. Stephan, F. Johann-Christoph, H. Fabian, H. Arvid, K. Odej, L. Marcus, L. Ulf, M. Volker, N. Felix, P. Mathias, R. Astrid, J. S. Matthias, S. Sebastian, H. Mareike, T. Kostas, and W. Daniel, "The Stratosphere platform for big data analytics," *VLDB*, 2014.

[19] R. Bordawekar, B. Blainey, and C. Apte, "Analyzing analytics," *ACM SIGMOD Record*, vol. 42, no. 4, pp. 17–28, feb 2014. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2590989.2590993

[20] S. Ceri, E. D. Valle, D. Pedreschi, R. Trasarti, and L. B. Pontecorvo, "Mega-modeling for Big Data Analytics," pp. 1–15, 2012.

[21] A. O'Driscoll, J. Daugelaite, and R. D. Sleator, "'Big data', Hadoop and cloud computing in genomics," *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 774–781, 2013.

[22] C. Zhang and C. Ré, "GeoDeepDive : Statistical Inference using Familiar Data-Processing Languages," pp. 993–996, 2013.

[23] P. Atzeni, D. Cheung, and S. Ram, "Conceptual Modeling," in *ER*, 2012. [Online]. Available: http://scholar.google.com/scholar?hl=en{\&}btnG=Search{\&}q=intitle:No+Title{\#}0

[24] S. Ceri, T. Palpanas, and E. Valle, "Towards mega-modeling: a walk through data analysis experiences," *SIGMOD*, vol. 42, no. 3, pp. 19–27, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2536673

[25] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti, "Unveiling the complexity of human mobility by querying and mining massive trajectory data," *VLDB*, vol. 20, no. 5, pp. 695–719, jul 2011. [Online]. Available: http://link.springer.com/10.1007/s00778-011-0244-8

[26] R. Trasarti, S. Rinzivillo, M. Nanni, and F. Giannotti, "Types and Operators in M-Atlas system," no. i, pp. 1–12, 2011.

[27] R. J. Bayardo and R. Agrawal, "Data Privacy Through Optimal k-Anonymization," *ICDE*, no. Icde, 2005.

[28] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," 2008.

[29] V. Markl, "Breaking the Chains: On Declarative Data Analysis and Data Independence in the Big Data Era," *Proceedings of the VLDB Endowment*, pp. 1730–1733, 2014. [Online]. Available: http://www.vldb.org/pvldb/vol7/p1730-markl.pdf

An Adaptive Random Heuristic in Virtual Networks: Dependability Analysis

Marcelo Santos, Ricardo Martins, Matheus Santana, Stênio Fernandes
Informatics Center (CIn)
Federal University of Pernambuco (UFPE)
Recife, Pernambuco, Brazil
{mabs, rmas, embs, sflf}@cin.ufpe.br

***Abstract*— Network Virtualization has been a central matter for the future of the Internet. Nevertheless, virtualized structures are as failure-prone as their underlying physical infrastructure. Hence, dependability attributes constitute relevant metrics for virtual network allocation decisions. This work proposes and evaluates a greedy solution to mapping virtual networks that is a well-known NP-Hard problem. We take into account a dynamic networking scenario where the time to map requests and availability are important factors. Results show that the proposed heuristic achieves an availability above 99% in the different scenarios simulated.**

Keywords—Networking Virtualization; Virtual Networks Mapping; Availability

I. INTRODUCTION

With more than three billion users[1], we can say that the Internet has been working relatively well. However, currently, the Internet is a complex system that comprises a great number of devices, Internet Services Providers (ISPs) and Cloud Systems. The complexity implies a higher difficulty in managing resources efficiently and making changes in the network. The phenomenon of ossification of the Internet [1] is as an example of this.

Network Operators need to overcome barriers in relation to network complexity and its limitations to offer quality services to end users. On the other hand, Network Operators try to use available resources in the best way aiming to: (1) Minimize bottlenecks; (2) Support more customers; (3) Respect SLAs agreements; (4) Decrease the amount of failures; and (5) Get more profit. Several technologies have been proposed for this to be done easily and with a high management capacity of the network, for instance, trends such as Software Defined Networking (SDN) [2], Networking Functions Virtualization (NFV)[3] and Network Virtualization (NV)[4] arise as possible solutions. A key aspect of these technologies is the ability to manage the network through virtualization over nodes, services, links and network functions. It is important to note that these approaches can manage network resources in a totally different way from traditional networks. The network operator has the power to use a set of resources to meet specific objectives (e.g. load balancing, energy efficiency or minimize faults). Therefore, from this perspective, it is essential to have an algorithm that coordinates the use of available resources to meet requirements of the client and of the infrastructure provider. Due to the characteristics of the problem and attributes involved, we have an NP-hard problem [5]. which consequently can't be solved in polynomial time in a known way, because of this, heuristics and greedy algorithms have been used to find an approximate optimal solution.

When considering the use of network equipment (e.g. routers, servers and links) through network virtualization technologies, a risk of failure is involved. Obviously risks are inherent to physical infrastructure because its hardware is failure-prone as well as the respective software infrastructure. For instance, the crash of Amazon's EC2 cloud[2] can be used as an example that failures due to either human intervention or technical glitches are a reality and should not be neglected. Thus, it is necessary to take into account attributes of dependability in Algorithms that perform the mapping of virtual resources on physical resources. Ignoring the characteristic of failure may cause a high rate of problems in virtual requests, causing a low quality of service and even causing costs to infrastructure provider due to SLAs violations.

In this paper, we consider a scenario that aims to reduce the probability of failure, at the same time respecting the restrictions imposed by clients and infrastructure providers. We take into consideration the time to make a decision as an important factor in our scenario. Therefore, we propose a greedy heuristic for mapping virtual network requests on physical infrastructure considering attributes of dependability modeled by Reliability Block Diagram (RBD). We analyze the results obtained by our proposed algorithm with other's well-known in the literature. The results show that the proposed heuristic achieves an availability above 99% in the virtual networks mapped.

The remainder of this paper is organized as follows: Section II describes the related work while and gives some background on dependability. Section III shows a formal model about virtual network mapping problem. Section IV presents the proposed heuristic. Section V describes the experimental methodology adopted in this work. Section VI shows the main results from our experiments. Section VII discusses the obtained results and presents the lessons learned. Finally, Section VIII concludes the paper and provides directions for future work.

[1] http://www.internetlivestats.com/internet-users/#trend

[2] http://www.businessinsider.com/amazon-lost-data-2011-4

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

II. Related Work and Technical Background

A. Related Work

D. Sun *et al*. [6] propose a cloud dependability model, called Cloud Dependability, by using System-level Virtualization (CDSV). Their work focuses on assessing security issues and components dependability attributes when virtualized on operating system level (e.g. virtual machines, hypervisors and others). In [7], the authors follow a similar approach presenting a quantitative assessment and a risk assessment methodology in order to assess security risks regarding cloud computing environments. Despite showing how to address dependability assessment on virtualized stages, the scope of the work is simpler than network virtualization scenario.

In [8], S. Fernandes *et al*. propose a dependability analysis method on virtualized networks environments based on Reliability Blocks Diagrams (RBD) and Petri Stochastic Networks (SPN). This is the first work analyzing dependability impact over dynamical network virtualization environments. However, this work doesn't propose any algorithm for mapping virtual requests. The paper is based on the R-ViNE algorithm [9] and then performs a dependability analysis. Therefore, it is important because it is among the first studies of what dependability assessment on network virtualization stages should be.

The most similar work in the literature was performed by Lira *et al*. [10]. This paper proposes a heuristic based on the GRASP meta-heuristic [11] for allocation of virtual networks. Although this work considers dependability attributes, the proposed algorithm does not have the objective of maximizing dependability. The dependability is only one restriction. The objective of the allocation is to minimize the cost in relation to the use of physical resources that does not take into account attributes of dependability in its objective function. Furthermore, the paper presents preliminary results and does not provide data in relation to load of physical resources used and the requests acceptance ratio. Thus, we can say that embryonic works such as Fernandes *et al*. [8] and Lira *et al*. [10] were useful as motivation for a more extensive and in-depth analysis.

S. Shanbhag et al. [12] present's a solution to allocate virtual nodes and virtual links called VHub. This solution is based on the p-hub facility location problem and execute in a single step. The authors solved the problem as a mixed integer program considering bandwidth, processing and node location. The optimization objective is to maximize revenue and perform load balancing. However, this approach does not take into consideration any dependability attribute.

We describe two well-known algorithms that were used to compare with our proposed heuristic. It is worth emphasizing that these studies do not take into account failure characteristics when mapping virtual network requests. However, the results comparison is important because it shows how much dependability varies when its attributes are neglected during the mapping process.

In [13], Zhu and Ammar propose a greed algorithm which focuses on balancing link and node loads. Their approach, however, does not take into account capacity aspects: the balance is done based on only the quantity of virtual nodes and links mapped over the network infrastructure. The main idea of this strategy is mapping new virtual nodes on physical nodes considering how many virtual nodes are already mapped to these physical nodes and loads of links and neighbor nodes.

In [9], M. Chowdhury *et al*. model virtual networks mapping problem as a function of server revenue costs considering nodes, links, and geolocation constraints. The resulting problem is NP-hard and it is reduced to a mixed integer optimization problem. They proposed two approximation algorithms: D-ViNE and R-ViNE. D-ViNE has a deterministic approach to nodes mapping based on a linear relaxed solution. R-ViNE is similar but uses a random nodes mapping. Both algorithms have versions in which virtual links are mapped using the shortest path or slicing the virtual flow to a number of physical paths. It is worth noting that this work does not take into account links delays as a constraint parameter for mapping. In practice, links delays are relevant constraints for most application requirements. Furthermore, it is assumed that a single physical machine cannot host more than one virtual machine for one particular request, which makes it difficult searching for the mapping solution. Another drawback is the fact that some D-ViNE and R-ViNE algorithms variations perform virtual paths slicing. In practice, this is a difficult task to accomplish. Based on this qualitative evaluation, we decided to compare our approach to the D-ViNE-SP variation only, which maps links using the shortest path approach.

For a complete review on virtual network mapping problems algorithms, we refer the reader to [14].

B. Technical Background: Dependability Analysis

The dependability of a system, put simply, is the system's capacity to supply a set of services which can be justifiably trusted. It is widely seen as an umbrella concept covering many attributes such as fault tolerance, availability and reliability [15][16]. Dependability metrics can be calculated by combinatorial models like Reliability Block Diagrams (RBD) and fault trees or by stochastics state-based models like, for instance, Markov Chains and Stochastics Petri Nets (SPN). We refer the reader to [17][18] for a better understanding of reliability models.

Availability attribute has fundamental importance on system dependability assessment. System dependability can be measured (e.g.: using the already cited combinatorial models) by the assessment of its components availability. Component availability (A) is measured by its time to failure (TTF) and time to repair (TTR) metrics. Average values can be used once the exact values for these metrics may be not available. Stationary-state availability (A) may be represented by the values for mean time to failure (MTTF) and mean time to repair (MTTR) for the given component:

$$A = \frac{MTTF}{MTTF+MTTR} \quad (1)$$

Mean time to failure metric may be assessed by taking into account system reliability (R), as follows (2):

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty tf(t)dt \quad (2)$$

The current work adopts the availability modeled by RBD as dependability parameter. Reliability block diagrams supply the necessary method for availability or reliability assessment by

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

mapping systems (and its underlying subsystems / components) to blocks arranged in series or parallels. Assessment is done by using assessment rules for each one of the possible arranges types after system modeling. It is recommended refer to [19] for a better understanding of modeling through reliability block diagrams.

III. VNMP MODEL

This section aims to define the problem treated in this paper by a mathematical model. In summary, we have a substrate network (SN) that receives a set of virtual networks (VNs). Each substrate network has nodes and links with limited capacities. Virtual Networks are composed by a set of virtual nodes and virtual links with requirements defined previously.

Substrate network is modelled as an undirected graph $G = (V, E)$ with nodes set V and arc set E represents the physical nodes and physical links. Each physical node i ∈ V has a generic capacity of $c_i \in \mathbb{N}^+$, we assume this capacity as CPU capacity. A virtual node makes use of the CPU resource available on the physical node i that hosts it. Physical links e ∈ E has a bandwidth capacity $b_e \in \mathbb{N}^+$ and a delay $d_e \in \mathbb{N}^+$. The virtual network is modelled by another undirected graph $G' = (V', E')$. Each virtual node k ∈ V demands a CPU capacity $c_k \in \mathbb{N}^+$. Each virtual link $f \in E'$ demands a bandwidth capacity $b_f \in \mathbb{N}^+$ and have a maximum allowed delay $d_f \in \mathbb{N}^+$. If a virtual link f is mapped in M physical links, the sum of delay of these physical links should not exceed d_f. Finally, the sum of the demand of each virtual node mapped into a physical node i cannot exceed c_i.

Quite simply, a virtual request is defined by a set of virtual nodes and virtual links. The availability of a request r is given by the function $A(r)$. Based on an RBD model described in the methodology section, each mapped virtual request has the availability calculated as the product of the availability of each block that composes the RBD model (series RBD model). The objective function is to maximize the availability of each virtual network request.

IV. VIRTUAL NETWORK MAPPING HEURISTIC

Some approaches allow optimal solutions to be obtained, such as, for instance, linear programming, but for the problem addressed in this paper they are not possible due to consideration of dependability attributes. In other words, the problem becomes nonlinear requiring the use of non-optimal solutions to solve the problem.

Finding a solution to this type of problem is not an easy task due to the fact the problem be NP-hard. Certain heuristics may be used as Simulating Annealing [20], GRASP and Tabu Search [20] in order to find a solution in a short time. Due to GRASP meta-heuristic adaptive characteristic we develop a heuristic based on it. In this way, the following section describes the main contribution of this paper: a greedy solution focusing on maximizing the availability for mapping virtual networks into a physical infrastructure. This algorithm is called ARH (Adaptive Random Heuristic).

ARH consists of a greedy algorithm that seeks an allocation that maximizes the availability of a virtual request. The requests are allocated one at a time based on a random choice that respects the constraints established. A request is to define (1) the number of virtual nodes; (2) connections between the virtual nodes; (3) capacity constraints for the virtual links and virtual nodes; and (4) delay of the virtual links. Obviously, the capacity of each component of the physical infrastructure can't be less than the sum of virtual components allocated into itself.

The allocation process can be simplified in three main steps. The first step to allocate a virtual request (set of virtual nodes and links) is the random choice of the first virtual node to be allocated. One list with limited size is built with the physical nodes that can host the chosen virtual node. Physical nodes with better availability constitute this list, then a physical node is randomly chosen from this list and the first virtual node is allocated. Randomness is an important factor to avoid solutions with local solutions. This technique is used in a range of meta-heuristics -- GRASP, for example.

The second step consists in performing a random Breadth-First Search (BFS) from the firstly allocated virtual node. The graph that represents the substrate network is then pruned to satisfy the constraints of the virtual link and the new virtual node that will be allocated. A new list composed by physical nodes with the highest availabilities is built, then a random search is performed on this new list to find the physical node and the set of links that maximize availability for the allocation of current request. Note that a physical node that has the highest availability does not necessarily indicate better total availability when considering the physical links used to connect two virtual nodes. The third step consists in performing the second step until all virtual nodes are allocated.

A. ARH: Pseudo code

Note that the presented pseudo code is a simplification of the real algorithm due to space restrictions. Heuristic's general functionality is shown in *Figure 1*. A key point to notice is the *allocate_request* sub procedure invocation (line 6) that that is displayed in Figure 2.

```
Input: SN = substrate network,
       VNS = virtual network requests collection,
       MAX_TRIES = integer for maximum possible number of iterations
1.  start
2.    for i ← 1, .., |VNS| do
3.      allocated ← false
4.      tries ← 0
5.      while (tries < MAX_TRIES) and (allocated == false) do
6.        allocated ← allocate_request(VNS[i], SN, MAX_TRIES)
7.        tries ← tries + 1
8.      end-while
9.    end-do
10. end
```

Figure 1. Heuristic's main procedure

The *allocate_request* sub procedure accomplishes allocation for specific virtual network request. This is initially done by random choice of a virtual node that will be the first to be allocated. Once first virtual node allocation is complete, the algorithm performs a breadth search from it and tries allocate the first node neighbors (line 11 to 20). It's worth noting that this ultimate allocation is accomplished in peers of nodes by *allocate_nodes* (line 13).

```
Input: VNR = virtual network request, SN, MAX_TRIES
1  start
2.    allocated ← true
3.    virtual_node ← choose_random_node(VNR)
4.    allocated ← allocate_first_node(virtual_node)
5.    if (allocated == false) do
6.       return false
7.    end-if
8.    virtual_base_node ← virtual_node
9.    remaining_nodes ← []
10.   do
11.      for i ← 1, .., |neighbors(virtual_base_node)| do
12.         virtual_destiny_node ← neighbors(virtual_base_node)[i]
13.         allocated ← allocate_nodes(virtual_base_node,
virtual_destiny_node, SN, MAX_TRIES)
14.         if (allocated == false) do
15.            return false
16.         end-if
17.         remaining_nodes ← add_element(remaining_nodes,
                                          virtual_destiny_node)
18.         remaining_nodes ← remove_element(remaining_nodes,
                                          virtual_base_node)
19.      virtual_base_node ← choose_random_node(remaining_nodes)
20.   end-do
20.   while (|remaining_nodes| > 0)
21.   return true
22. end
```

Figure 2. *allocate_request* sub procedure

Figure 3 depicts *allocate_nodes* pseudo code. It uses a list of physical nodes capable of hosting the input virtual node. This list supports an adaptive random search which goal is to maximize the dependability of virtual network trunk to be allocated (line 18 to 30). A key point here is that dependability maximization for two connected virtual nodes and links between them leads to the dependability maximization for the virtual network in its entirety. Such behavior is explained by modeling the system through series RBD (in which total availability is given by the product of components availability).

```
Input: virtual_base_node, virtual_destiny_node, SN, MAX_TRIES
1.  start
2.    physical_source_node ←physical_underlying_node(
virtual_base_node)
3.    solution ← []
4.    if (is_allocated(virtual_destiny_node) == false) do
5.       for i ← 1, .., |physical_nodes(SN)| do
6.          physical_node ← physical_nodes(SN)[i]
7.          if (can_allocate(physical_node, virtual_destiny_node)) do
8.             ok ← check_shortest_path(physical_source_node,
                                        physical_node, virtual_destiny_node)
9.             if (ok) do
10.               solution ← add_element(solution, physical_node)
11.            end-if
12.         end-if
13.      end-for
14.      // Filters nodes for biggest availabilities
15.      solution ← best_physical_nodes(solution)
16.      best_solution ← 0
17.      tries ← 0
18.      while (tries < MAX _TRIES) do
19.         physical_node ← choose_random_node(solution)
20.         availability ← compute_availability(physical_source_node,
                                        physical_node, virtual_destiny_node)
21.         if (availability > best_solution) do
22.            best_solution ← availability
23.            solution_node ← physical_node
24.         end-if
25.         tries ← tries + 1
26.         solution ← remove_element(solution, physical_node)
27.      end-while
28.      if (is_defined(solution_node)) do
29.         allocated ← allocate_node_and_link(physical_source_node,
                                        solution_node, virtual_destiny_node)
30.      end-if
31.      return allocated
32.   end-if
33.   return true
34. end
```

Figure 3. *allocate_nodes* sub procedure

Some procedures can be inferred from their name. Nonetheless, it may be worth noting some of them: (1) *can_allocate*: verifies if physical resources constraints are respected according to the demand of virtual resources allocation; (2) *check_shortest_path*: takes a source and a destiny physical nodes to verify if the physical path between them is the shortest one -- which is equal to the maximum availability; (3) *compute_availability*: computes the availability taking into account two physical nodes (source and destiny) and the physical path between them. It may be used inside the definition of check_shortest_path indeed;

V. METHODOLOGY

We developed an own simulator to test our proposed heuristic. ViNE-Yard simulator [21] was used to execute the others strategies.

A. Mapping Strategy

D-ViNE-SP [9] and G-SP [13] strategies were chosen and compared to our proposed heuristic. Motivation for this choice is that these strategies take into account the shortest path in order to select the set of possible physical host links in which a virtual link may be hosted in contrast to other D-ViNE variations that slice a single virtual link into multiple physical paths. We have simulated a slightly modified version of our heuristic in order to not take into account the delay attribute because D-ViNE-SP and G-SP do not consider this attribute. However, it is possible to add a delay constraint in our simulator, if it is necessary.

The ViNE-Yard simulator has a constraint which requires that each virtual node be allocated in a different physical node. In other words, if we have a virtual request composed by two virtual nodes A and B, these nodes cannot be allocated together in the same physical node. In order to have a fair comparison, we modified our heuristic adding the same constraint and then analyzing the impact on availability. Thus, we have two variations of our heuristic: Adaptive Random Heuristic (ARH) and ARH-PC (Adaptive Random Heuristic – Physical Constraint).

B. RBD Model and Dependability

The first step is to calculate the physical nodes availability through an RBD model. We assume that if any component of a node fails, the node itself fails. Thus, the RBD model can be drawn as a series of blocks connected. Note that the availability of each physical component remains unchanged during the whole of the simulation.

We consider a physical node as an assembly of four parts: CPU, Hard Disk, Hypervisor and RAM memory. It is assumed an RBD system connected by a series configuration (Figure 4). The virtual network availability is calculated through the product of the availability of each component. Note that a virtual link is mapped over one or more physical links and routers.

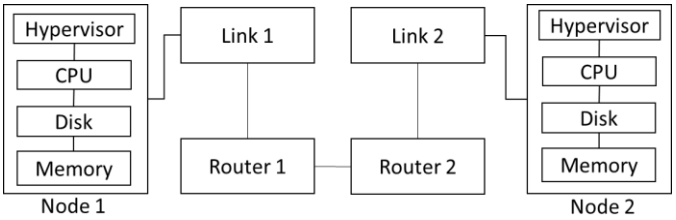


Figure 4. RBD model

MTTFs and MTTRs (in hours) for these components are presented by the Table I and are based on [19]. Random numbers in the interval [35, 100] are generated for each physical network component in order to determine its MTTF based on [8]. This number represents the considered percentage for the MTTF. MTTRs remain the same for all listed components.

TABLE I. MMTF AND MTTR FOR PHYSICAL NODES AND LINKS COMPONENTS

Component	MTTF (h)	MTTR (h)
CPU	2500000	1
Hard Disk	200000	1
RAM Memory	480000	1
Link	19996	12
Hypervisor	2800	2
Router	320000	1

C. *Factors, Parameters and Levels*

The substrate network topologies in our simulations are composed of 20, 30, 50, and 100 physical nodes connected with probability 50%. The substrates networks were randomly generated using the Barabasi-Albert method for creating scale-free networks [22]. We assume the physical node capacity as a unique attribute generated by a uniform distribution between 50 and 100 units. The virtual networks (VNs) arrive based on Poisson distribution with an average rate of 4 VNs/100 time units. The lifetime of each VN is based on an exponential distribution with an average of 1000 time units. The VN size is defined by a uniform distribution depicts in Table II. In order to perform a fair comparison, the methodology is based on [9]. Table II shows a complete view about the factors, levels and parameters used in our simulation.

We performed 30 simulations for each variation of a scenario. The confidence interval with 95% confidence level is displayed in the results section.

TABLE II. FACTORS, LEVELS AND PARAMETERS

Factors	Levels and Parameters
Topology Size	20, 30, 50 and 100 nodes
Physical Link bandwidth	~U{50:100}
Node Capacity	~U{50:100}
VN requests arrival	~Poisson(1/25)
VN life time	~exp(1000)
VN Size	~U{2:10}; ~U{4:12}; ~U{6:14} and ~U{8:16}
Virtual Node Requirements	~U{0:20}
Virtual Link Requirements	~U{0:50}
Heuristic	D-ViNE-SP, G-SP, ARH and ARH-PC
Considering VN life time	Yes and No

The collected metrics are described in the Table III. The simulations were performed on a machine with the following settings: CPU Intel(R) Core(TM)2 Duo CPU E7300 @ 2.66GH; V; 4GB RAM Memory DDR2; and Operational System LinuxMint Release 14 (nadia).

TABLE III. DESCRIPTION OF THE COLLECTED METRICS.

Metric	Description
Availability	Average availability of all virtual requests
Acceptance rate	Average rate of virtual request acceptance
Nodes utilization	Average physycal nodes utilization (stress)
Links utilization	Average physycal links utilization (stress)
Execution time	Average excution time for each problem

VI. RESULTS

In order to have a fair comparison, we created a variation of our heuristic adding a restriction on the allocation of two or more virtual nodes (belonging to the same request) on the same physical node. It variation is referred to as ARH-PC (Adaptive Random Heuristic – Physical Constraint).

A. *Placement constraint (scenario 1)*

In the first scenario, we analyze a general behavior of each heuristics. We assume a virtual network size generated by a uniform distribution between 2 and 10 virtual nodes. The substrate network (physical topology) size varies between 20, 30, 50 and 100 physical nodes. G-SP and DViNE-SP are not designed to allocate two virtual nodes from a request in a same physical node. Due to this limitation, we simulated a variation of ARH in order to quantify the impact over availability metric. However, other metrics need to be analyzed. For instance, it is important to any placement strategies, allocate virtual requests as many as possible. Figure 5 depicts that the ARH and ARH-PC strategy has a high acceptance rate. ARH strategy shows acceptance rate higher than 80% even with variation of the topology size.

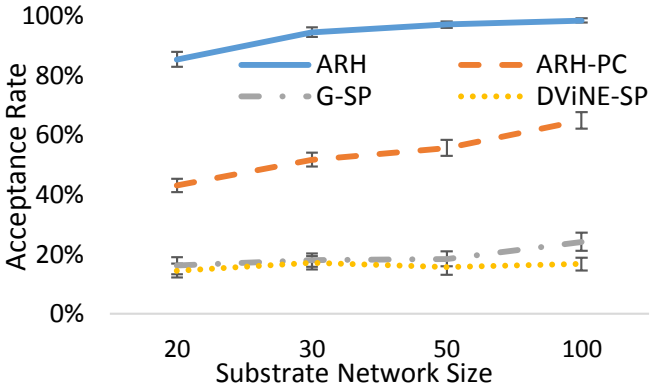


Figure 5. Virtual Network request acceptance rate

ARH have a high acceptance rate due to the simple strategy to allocate in the same physical node one or more virtual nodes of the same virtual request, unlike other strategies that restrict

the allocation of virtual nodes of a requisition in distinct physical nodes. There is an expected growth in the acceptance rate when substrate network size increases due to a greater number of available physical nodes to allocate a virtual node.

Regarding the availability attribute, Figure 6 shows that it does not make sense to limit the allocation of each virtual node (from the same request) to a different physical node. ARH has availability greater than 99%. When the strategies ARH-PC, G-SP and DViNE-SP accept more VN requests the availability decrease because it becomes harder maximize the availability. On the other hand, ARH is the best approach in a scenario with more physical nodes available. Note that ARH can allocate two virtual nodes in the same physical node. Consequently, saving bandwidth in physical links.

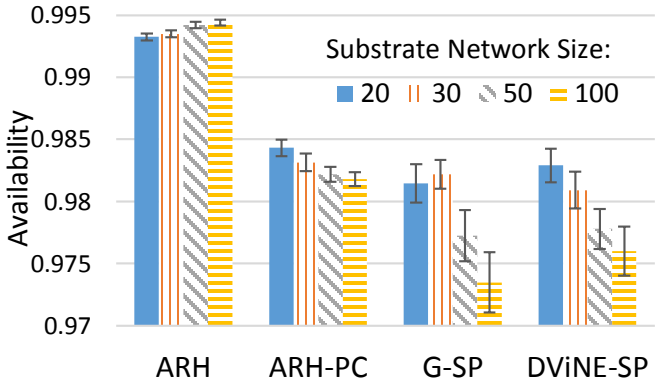


Figure 6. Virtual Network request availability

To analyze the stress on physical nodes and physical links, we calculated the load/capacity for each node and link. Based on each value we estimated the average between all links and nodes. According to Table IV, ARH-PC and ARH heuristic obtained a high utilization of the nodes when compared to the other allocation strategies. This result is justifiable due to the high rate of acceptance of requests. Specifically in ARH heuristic, occurs a concentration of virtual nodes (from the same request) in some physical nodes causing a higher stress on physical nodes and less stress on physical links. In other words, when two or more virtual nodes are allocated on the same physical node there is a reduction of a virtual link that would occupy some physical links. In a topology composed by 100 physical nodes, the ARH approach has the higher acceptance rate (Figure 5), the higher availability (Figure 6) and the lowest average link stress (Table IV).

TABLE IV. AVERAGE NODE AND LINK UTILIZATION

Physical Topology Size	Load/Capacity (Stress)							
	ARH		ARH-PC		G-SP		DViNE-SP	
	Node	Link	Node	Link	Node	Link	Node	Link
20	88%	48%	33%	62%	10%	37%	11%	32%
30	68%	35%	29%	53%	7%	31%	8%	30%
50	42%	19%	19%	39%	4%	25%	5%	20%
100	21%	11%	12%	27%	3%	21%	2%	13%

B. Virtual Network Size Variation (Scenario 2)

We have a fixed topology size of 50 nodes in the scenario 2. We varied the size of virtual requests according Table II. In all strategies when the virtual network size increases the VN placement problem becomes harder to find a solution that maximize the VN availability. Figure 7 shows the acceptance rate for all strategies simulated. We can see in Figure 7 a high impact on the acceptance rate when the VN size is increased. In the best case for ARH we have 90% of acceptance rate against 61% in the worst case. In all simulations performed, the ARH had the best performance.

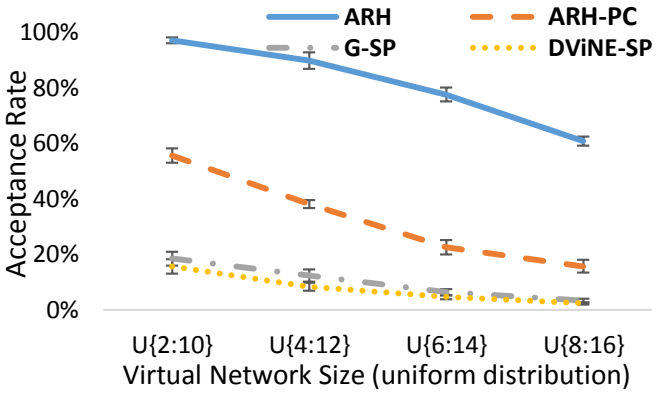


Figure 7. Virtual Network request acceptance rate

Table V depicts the availability for scenario 2. We can realize the impact in the availability according to VN size growth. ARH shows less impact in the availability because it is allowed to allocate more virtual nodes in the same physical node. It causes a reduction in the RBD model and consequently a higher availability compared to other strategies.

TABLE V. COMPARING AVAILABILITY BY VIRTUAL NETWORK SIZE

Heuristic	Availability			
	Virtual Network Size (Uniform distribution)			
	U{2:10}	U{4:12}	U{6:14}	U{8:14}
ARH	0.994	0.993	0.991	0.990
ARH-PC	0.982	0.976	0.969	0.961
G-SP	0.977	0.967	0.956	0.820
DViNE-SP	0.978	0.969	0.925	0.824

Table VI shows the link and node load. It is important to note that ARH accept much more virtual requests and due to it has higher rates of stress in physical links and nodes compared to other approaches.

TABLE VI. AVERAGE NODE AND LINK UTILIZATION

Virtual Network Size	Load/Capacity (Stress)							
	ARH		ARH-PC		G-SP		DViNE-SP	
	Node	Link	Node	Link	Node	Link	Node	Link
U{2:10}	42%	19%	19%	39%	4%	25%	5%	20%
U{4:12}	51%	29%	19%	40%	4%	24%	4%	19%
U{6:14}	50%	30%	14%	32%	3%	20%	3%	17%
U{8:16}	47%	30%	13%	31%	2%	17%	2%	12%

C. Stressful Cenario (Scenario 3)

To analyze the heuristics behavior in an extreme stressful scenario, we simulated the scenario 1 without VN lifetime. In other words, a virtual network is never removed during the simulation.

Figure 8 shows a decline in the number of VN requests accepted. However, this fact can be realized only in small

substrate networks. Considering a substrate network with 100 nodes, scenario 1 and scenario 3 had the same acceptance rate (98.5%).

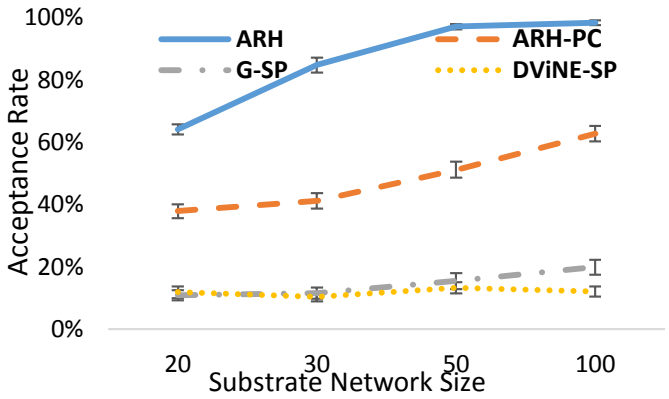


Figure 8. Virtual Network request acceptance rate

Although Figure 8 shows a large decrease in a substrate network with 20 physical nodes, the availability had the same value for all strategies (Figure 9). It again demonstrates the great performance of the ARH approach even in scenarios with high stress.

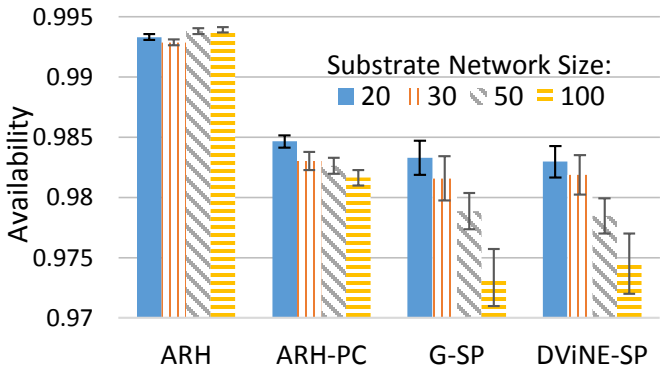


Figure 9. Virtual Network request availability

D. Execution Time

According to the scenario 1 described previously, we collected the execution times' and calculated the average for each simulation.

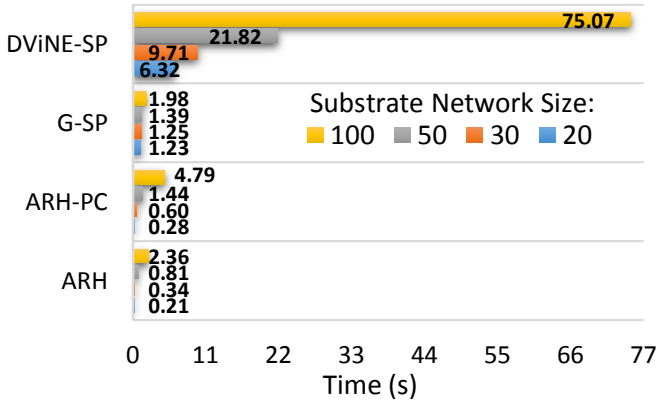


Figure 10. Average time for one simulation execution

Figure 10 shows that D-ViNE-SP spend much more time than ARH-PC and G-SP. This result is due to the fact ARH-PC, ARH, G-SP are a greedy heuristic, and D-ViNE-SP is based on linear programming. D-ViNE-SP was approximately 30 times slower than ARH. For topology size, those of 20, 30 and 50 nodes ARH have the best performance with execution time less than 1 second.

VII. DISCUSSION

It is important to note that this problem is NP-hard, and because of its complexity we propose a non-optical solution called ARH. The ARH heuristic achieved significant results with good rates of availability when compared to the other allocations strategies. With demand for the use of reliable services and rigid Service Level Agreements (SLA) that must be respected by infrastructure providers and service providers, neglecting attributes of dependability is not a viable alternative, given the results presented in this paper.

An important point to note is that the ARH has a variance in relation to other compared strategies: heuristic ARH allows allocation of one or more virtual nodes of the same virtual request on the same physical node. Although simple, this difference was expressive in metrics such as acceptance rate, given that the purpose of the ARH heuristic is to maximize the availability of each allocated virtual network. However, the heuristic has had better results in the acceptance rate of VN requests, ARH did not produce high overhead in physical links when compared to other strategies. Moreover, our heuristic has a high load on the physical nodes. Such behavior is expected as to save on a number of links used and maximize availability.

Each increase in uptime is important in a network scenario. 98% of availability results in a total downtime of 14.4 hours during a month of continuous operations. Hence, 1% of the time in a month means 7.2 hours of either downtime or uptime. As a comparison example, Amazon EC2 and Amazon EBS offer at least 99.95% on their virtual machines. Therefore, any cloud provider can use the proposed ARH approach to maximize availability during the process of mapping virtual networks onto physical nodes.

VIII. CONCLUSION

Allocating virtual network requests demands effective mapping strategies. In this paper, we present an adaptive random heuristic to allocate virtual networks considering dependability attributes, delay constraints, bandwidth and CPU requirements. The aim of heuristic is to maximize the availability for each mapped virtual request. The proposed heuristic showed better results as compared to other approaches in metrics, such as acceptance rate and availability.

The ARH heuristic achieved in all experiments availability greater than 99%. Considering the growth of virtualization use in computer networks, our results show that neglecting dependability attributes make a virtual network more prone to failure.

In the future, we intend to take into account specific restrictions of dependability in client and server side. We also intend to analyze and estimate software aging effects and rejuvenation techniques.

ACKNOWLEDGMENT

This work is supported by the FACEPE (Brazil) under Grant no. IBPG-1321-1.03/11 for Marcelo Santos; CNPq 304422/2013-4 for Stenio Fernandes.

REFERENCES

[1] R. P. Esteves, L. Z. Granville, and R. Boutaba, "On the management of virtual networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 80–88, Jul. 2013.

[2] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[3] "GS NFV-INF 001 - V1.1.1 - Network Functions Virtualisation (NFV); Infrastructure Overview," 2013.

[4] M. Napierala, "Problem Statement: Overlays for Network Virtualization," 2014.

[5] A. Haider, R. Potter, and A. Nakao, "Challenges in Resource Allocation in Network Virtualization," 2009.

[6] D. Sun, G. Chang, Q. Guo, C. Wang, and X. Wang, "A Dependability Model to Enhance Security of Cloud Environment Using System-Level Virtualization Techniques," in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*, 2010, pp. 305–310.

[7] P. Saripalli and B. Walters, "QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security," in *2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 280–288.

[8] S. Fernandes, E. Tavares, M. Santos, V. Lira, and P. Maciel, "Dependability assessment of virtualized networks," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 2711–2716.

[9] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual Network Embedding Algorithms With Coordinated Node and Link Mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[10] V. Lira, E. Tavares, S. Fernandes, and P. Maciel, "Dependable virtual network mapping," *Computing*, vol. 97, no. 5, pp. 459–481, Oct. 2014.

[11] J. Inführ and G. R. Raidl, "GRASP and Variable Neighborhood Search for the Virtual Network Mapping Problem," 2013, pp. 159–173.

[12] S. Shanbhag, A. Reddy Kandoor, C. Wang, R. Mettu, and T. Wolf, "VHub: Single-stage virtual network mapping through hub location," *Comput. Networks*, vol. 77, pp. 169–180, Feb. 2015.

[13] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, 2006, pp. 1–12.

[14] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[15] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein, "A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability," *IEEE Commun. Surv. Tutorials*, vol. 11, no. 2, pp. 106–124, 2009.

[16] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.

[17] P. R. M. Maciel, K. S. Trivedi, R. Matias, and D. S. Kim, *Performance and Dependability in Service Computing*. IGI Global, 2012.

[18] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-based evaluation: from dependability to security," *IEEE Trans. Dependable Secur. Comput.*, vol. 1, no. 1, pp. 48–65, Jan. 2004.

[19] D. S. Kim, F. Machida, and K. S. Trivedi, "Availability Modeling and Analysis of a Virtualized System," in *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, 2009, pp. 365–371.

[20] "Integrated Methods for Optimization | John N. Hooker | Springer." [Online]. Available: http://www.springer.com/us/book/9781461418993. [Accessed: 01-Aug-2015].

[21] "ViNE-Yard." [Online]. Available: http://www.mosharaf.com/ViNE-Yard.tar.gz.

[22] A. Barabási, "Emergence of Scaling in Random Networks," *Science (80-.).*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

Solution for Spectrum Monitoring of the Industrial, Scientific and Medical (ISM) Radio Bands

Vinicius C. Ferreira
and Ricardo C. Carrano
Departamento de Engenharia de Telecomunicações
Universidade Federal Fluminense
Niterói - RJ, Brazil
Email: contato@midiacom.uff.br

Bruno Peres
Departamento de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
Rio da Janeiro - RJ, Brazil
Email: brunoperes89@gmail.com

***Abstract*—The growing need for Internet connectivity allied with technological advances in the production of mobile devices have resulted in the proliferation of local networks with wireless access techniques. These networks allow greater mobility and comfort for its users, and provide multiple access and satisfactory transfer rates. These networks also became popular due to the low cost of its operational infrastructure, partly by the use of unlicensed bands, of free and public use. The widespread use of these frequency bands cause the interference from signals transmitted by multiple devices, resulting in the need of spectral analysis to obtain a local frequency spectrum diagnosis, thus providing a more effective use of it.**

The goal of this work is creating an infrastructure for spectral analysis of an unlicensed band, using low cost elements, allowing the monitoring of the environment and the access of that information in a simple and practical way.

I. Introduction

Wireless communication is one of the most successful histories of engineering in the last twenty five years, not just from a scientific point of view, but also, from market and social impact.

In the last two decades, communication through cellular systems and their new services had great relevance for general population. From 2003, a series of new advances increased the interest in wireless communications area, including the huge success of Wireless Local Area Networks (WLANs).

Devices following the standard created by the Institute of Electrical and Electronics Engineers (IEEE), IEEE 802.11, enabled the use of computers in a versatile manner with mobility similar to mobile phones. This standardization had its beginning in the mid 90's and became popular after several versions and intense competition between manufacturers. Today, we have several access points scattered around houses, offices, airports, restaurants, among others, which allow free access to the Internet, following the philosophy: "anytime, anywhere" [1].

Since the demand for higher data rate connection is increasing, as well as the number of devices using technologies based in the ISM band, studies and analysis of this frequency range ocupation will become mandatory. Seeking the best solution to avoid problems caused by radio frequency congestion and also to increase spectral efficiency is necessary to achieve these goals.

978-1-4673-9408-6/15/$31.00

In this presented scenario, this work has as its main purpose the creation of a tool for frequency spectrum analysis in the 2.4 GHz band (ISM band), making use of low cost elements and the existing LAN infrastructure.

The rest of the paper is organized as follows. We introduce the concept of spectrum sensing and related works in Section II. In Section III, we characterize wireless short-distance networks and describe its main industrial specifications. In Section IV we detail each item used in this solution and the steps taken to create the tool. We show the results and a quick analysis of the kit usage in Section V. Finally, we conclude the paper with a brief discussion of the points achieved and next steps to complement this work in Section VI.

II. Related Work

Spectrum Sensing is being studied for decades and is used to support a number of different purposes as: improving quality of service in wireless networks, detection of unauthorized transmitters for regional spectrum enforcement, and Signals Intelligence (SIGINT).

Spectrum analyzers have been improved from the past decades to match high-resolution and wideband requirements for studies in both acoustic and RF. In [2] a software reconfigurable spectrum analyzer for real-time analysis in high data rates is described.

Also, the interest in spectrum sensing as major research topic returned since the cognitive radios paradigm became popular. Cognitive radios' goal is to increase spectrum usage efficiency by monitoring and making an opportunistic use of it. It takes advantage of unused frequencies, often designated as spectrum holes or white spaces, both in time and in space [3].

Several sensing techniques were developed for this so called "spectrum awareness" purpose. The most common are: Energy Detector Based Sensing, Waveform Based Sensing, Cyclostationarity-Based Sensing, Radio Identification Based Sensing and Matched-Filtering [4].

In [5], an approach for infrastructure aided cooperative spectrum sensing scheme for vehicular ad hoc networks is

presented. It uses cars as sensors for the Road Side Units prospective decisions of spectrum usage.

There are different technologies used for spectrum sensing, among them we have three different approaches [6]:

- Traditional Spectrum Analyzers: often used to survey the signal environment to ascertain what frequencies are occupied. There are two basic types of systems in use today: FFT analyzers and radiometers. FFT analyzers digitize the input signals over some frequency range and then do digital FFT processing to determine the power spectral density across that range. Radiometers use an RF receiver to tune to a each frequency across a band and then measure the received signal power at each frequency. Many spectrum analyzers utilize a combination of both of these techniques to define spectral occupancy.
- Spectrum Monitoring Systems: unlike traditional spectrum analyzers, whose primary purpose is for RF test and measurement, spectrum sensing systems are specifically designed to measure spectrum occupancy over time. These systems typically incorporate complex logging features and uses sophisticated algorithms to maximize the probability of signal detection and minimize false alarms. These systems also typically include Direction Finding capabilities to help in finding unlicensed or unintended emitters.
- Signal Identification and Modulation Recognition Systems: include an ability to classify detected signals, identifying modulation structure and signal type. These types of systems generally fall into one of two categories: blind detection where the signal is identified through analysis of signal parameters (amplitude, phase and/or frequency vs. time) or a directed search, where the detected signal is compared against one or more known signal types using a matched filter algorithm or other similar technique.

In this paper we propose a solution that makes use of a low cost traditional spectrum analyzer and network elements of an existing WLAN infrastructure to make the distributed spectrum sensing.

III. Short Distance Wireless Communication Systems

As any network, wireless networks transmit data through a medium, a physical layer. This medium, in wireless networks is a form of electromagnetic radiation, which is made to cover a certain area. Two of the most used mediums for local area applications are infrared radiation (IR) and radio waves. The infrared radiation, electromagnetic radiation in the range of approximately 300 GHz to 430 THz, has a very strict and limited coverage. IR is easily blocked by any obstructions, such as walls, partitions, etc. Therefore, currently IEEE 802.11 devices use radio waves or radio frequency (RF) as physical medium [7].

To simplify the use of RF in specific applications and not to overload the requests for licenses in regulatory institutions, unlicensed bands were created. The devices which usually makes use of this bands are: wireless microphone, cordless phone systems, remote controls of vehicle alarms, environmental sound systems and equipment for wireless local area networks, that use low transmission power and tend not to cause interference with other RF system.

In Brazil, the applications of unlicensed use are governed by the Resolution no. 506 of 1st of July of 2008 of Brazilian Telecommunications National Agency, ANATEL. The frequency bands used for these applications are the ISM band (Instrumentation, Scientific and Medical), comprising three segments of the spectrum: 902 MHz to 928 MHz, 2400 MHz to 2483.5 MHz and 5725 MHz to 5850 MHz; and the U-NII band (Unlicensed National Information Infrastructure), which contains the 5250 MHz frequency bands to 5350MHz and 5470MHz to 5825 MHz [8].

In addition to the 802.11 standard and still in the context of short range wireless networks, we have the personal wireless LANs, wireless personal area networks (WPANs), which generally use the family of standards IEEE 802.15.X. As the 802 comprises local and metropolitan area networks and the suffix 15 refers to the personal networks, and the "X" corresponds to the protocol specific application. This standard governs the wireless connections at the range of short centimeters to a few meters, for personal purpose, connecting devices on a desk, cordless phones, headsets, etc.

A. Structure of a WLAN

The wireless LANs have some basic elements, such as:

- Wireless hosts. Similar to the wired network, they are the endpoints of a system, running applications. E.g.: Laptops, Smartphones.
- Wireless link. The connection between the hosts and the base station is via wireless communication links. The main characteristics of these links are their range and transmission rate, as will be exemplified later for some technologies.
- Base station. They are the central part of a wireless network infrastructure, responsible for sending and receiving data to a wireless host associated with it. When a host is said *associated* with a base station it means the host is within the station's coverage range and uses it to transmit the data.
- Network infrastructure. A larger network with the host wants to communicate. In general it is directly connected to the base station and is traditionally a wired network.

Wireless networks that contains all of these basic elements are said to operate in infrastructure mode because all the traditional network services (addressing and routing) are offered by the network infrastructure connected to the base station. However, there are networks that operate without using a traditional network infrastructure, and the hosts are responsible for network services. These networks are called *ad hoc* networks [9].

Specifically for the architecture of wireless LANs IEEE 802.11, the fundamental block of network elements is called *basic service set* (BSS), which contains one or more wireless

stations, hosts, and a base station, known as *access point* (AP) in the terminology used by the IEEE 802.11 standard.

B. Radio and Physical Layer Characteristics

Both hosts and base stations are equipped with radios to establish wireless links. These links have some specific characteristics to avoid overhead of the used frequency band and to ensure robustness in radio connections used in wireless networks.

The IEEE 802.11 and IEEE 802.15 standards use some techniques and modulations that optimize the use of the electromagnetic spectrum and enable various applications to use the same spectrum. Often these techniques are of mandatory use, required by regulatory institutes and have technical specifications so that there is interoperability between different equipments and an efficient use of the electromagnetic spectrum.

1) Spread spectrum: Spread spectrum techniques are used to take advantage of the electromagnetic spectrum to establish communication resistance to multipath and narrow-band interference. In these techniques the narrow-band signal to be transmitted has its spectrum spread over a larger band by a mathematical function. At the reception the inverse process is performed and the signal is recovered, returning to its original shape while the noise is dissipated and has its power dispersed over a larger frequency band. In this way we get a better performance in noisy channels, as the signal to noise ratio is increased and therefore the resistance to noise. The types of spread spectrum in certain IEEE 802.11 and IEEE 802.15 are basically three types:

- *Frequency Hopping Spread Spectrum* (FHSS). The transmitter jumps from frequency to frequency hundreds of times per second. The transmission becomes so difficult to detect and almost impossible to be blocked, it also offers high resistance to fading by multipath and narrow-band interference. This technique is used in some IEEE 802.15 standard applications, as Bluetooth. This technique is extremely used for military purposes, to offer greater information security.
- *Direct Sequence Spread Spectrum* (DSSS). It uses a sequence of codes to spread the data at a higher frequency band. It is a technique widely used commercially as a way to share the same frequency band by multiple users. These codes used can be different for each user in the same system, a method called CDMA (Code Division Multiple Access), used in 3G cellular networks and GPS (Global Positioning System). It was the main technique initially adopted in the IEEE 802.11 standard.
- *Ultra-Wide-band* (UWB). The third method makes use of a higher bandwidth, with at least 500MHz or 20% of the center frequency band used. The UWB sends a series of fast pulses, varying the position of the pulses. By using a wide frequency band UWB can achieve high data rates and because of its larger spread over this great band it is not considered a source of harmful interference to narrow-band systems. Its use is still limited and has few commercial applications such as Wireless USB and WiHD, alternatives created to replace USB and HDMI cables and to stream videos in high definition on a local network.

2) Multiplexing Methods: Multiplexing is a technique used to share a communication channel by multiple users. This technique can be designed in different ways, and is usually performed to be divided by time, frequency or code.

- *Time Division Multiplexing* (TDM). This multiplexing method is based on the alternation of the communication channel usage — one user at a time makes your transmission during a gap of designated time.
- *Code Division Multiplexing* (CDM) is based on DSSS, using a different code for each user, as mentioned earlier when it was addressed in the previous section III-B1.
- *Frequency Division Multiplexing* (FDM). Finally, within the electromagnetic spectrum context usage we have the FDM. This multiplexing method is a modulation scheme that divides the allocated spectrum band into smaller sub-bands, each user will transmit exclusively within a sub-band. This sub-band should therefore have a wide enough range of frequencies to transmit a user's signal. Furthermore it is allocated a spare band called guard band to ensure no interference happens between adjacent channels.

In digital systems when it comes to data networks, it is possible to use a spectrum division without loss to make the frequency guard band. This is possible with OFDM (Orthogonal Frequency Division Multiplexing). The channel is subdivided and the various sub-carriers send data independently, similar to FDM, but these sub-carriers are closely approximated, overlapping each other. However, adjacent carriers amplitudes must be zero at the center frequency of each other. This way we can sample the signal of a sub-carrier on your central point without interference from adjacent signals.

The OFDM is used as an alternative to spread spectrum techniques. It was initially specified in the IEEE 802.11 standard to be used in the range of 5 GHz U-NII frequencies. In a channel using a 20MHz bandwidth, as specified in the standard, this technique can achieve a rate of up to 54Mbps.

3) Currently Used Techniques: The continuous demand for increasing transmission speed, particularly in the 2.4GHz ISM band, resulted in the evolution of existing techniques. The FHSS was abandoned in the IEEE specifications for WLANs by its low transmission capacity, little flexibility, as well as for being a technique which pollutes the whole spectrum range that it uses, but its use was continued in WPANs.

Thus, efforts were concentrated to evolve using the techniques DSSS and OFDM, emerging two new extensions of the old Specifications:

- *Extended Rate PHY* (ERP). The ERP was essentially the use of OFDM in the range of 2.4GHz frequency, but with some adjustments . An effort was made to the current standard be incorporated in this new specification, reducing damage to existing networks. A big technical

barrier was faced, since a transmission of a device is preceded by an channel inactivity check, and it should be possible to operate in both, new and old device specifications. Like OFDM, ERP achieves rates of up to 54Mbps if used in the new devices, but as it is compatible with the techniques like DSSS/CCK (HR/DS) it is necessary a 22MHz bandwidth instead of only the 20MHz used by OFDM.

- *High Throughput* (HT). The HT is a specification that covers both the frequencies of 2.4GHz and the 5GHz. It uses the OFDM, but with some changes. It uses MIMO (Multiple-Input Multiple-Output) technology, which consists of using multiple antennas to transmit and receive data simultaneously. This way of conveying is more efficient than communicating via a single antenna. In addition, the band allocated by channel was doubled to 40MHz. With this increase in bandwidth and the use of multiple antennas there was a big gain in transmission rates, which reached up to 600Mbps. A version of High Throughput was made specially to 5GHz frequencies, allowing more bandwidth allocation, from 80MHz up to 160MHz and more MIMO spacial streams. A theoretical throughput up to 6.9 Gbps is advertised with its maximum setup.

4) Spectral Masks: The spectral masks, also called transmission masks are descriptions of radio power levels along the transmission band. These levels are well defined and must be honored. These settings are designed in order to reduce interference between adjacent channels, limiting exceeded radiation at frequencies beyond the bandwidth required. Unwanted emissions are mitigated by the use of bandpass filters designed appropriately to allow passage of the necessary frequencies.

IV. Sensing Kit

In this section we will specify the material used and the adjustments made on it to allow its use as an electromagnetic spectrum sensor kit. Later, we describe the interface designed to make the spectral analysis.

A. Wireless Router

The use of the wireless router, which was already an integral part of the network, as a platform to operate the spectrum analyzer and transmit the information obtained through the network to a remote server was the cornerstone of this work.

A D-Link DIR-320 was used as a wireless router. The basic features of this model are IEEE 802.11 b and g standards support, 4 LAN ports that operate as a switch, a USB port. The model also features a port labeled as Internet, to connect to an external network and an input for power supply of 5V. For more details on the model, consult the reference [10].

B. OpenWrt

The router's factory configuration and firmware could not meet the requirements of flexibility and reconfiguration, essentials for creating the kit. Therefore it was necessary to install a custom firmware.

The OpenWrt firmware [11], a GNU / Linux distribution for embedded devices, was chosen. OpenWRT has a package management system and provides a framework for building applications without the need to create and compile a complete firmware and distribution. The user then has the freedom to customize his device in a convenient and practical way. This choice was based on prior knowledge and familiarity with the platform.

The main OpenWrt project guidelines are the creation of an open and free code using the GNU General Public License (GNU GPL). Its main benefits are the easy and free access, be always open to new contributors without creating barriers and be addressed to the Community interest. Unlike proprietary software suppliers, which develop according to its own interests and not always consumers, OpenWrt is collaborative, therefore it carries the points of interest of common users and developers.

C. Spectrum Analyzer

Another key player in the sensing kit is the spectrum analyzer used in conjunction with the wireless router to perform the power measurements over the band.

The AirView2 USB [12] spectrum analyzer, from Ubiquiti, was selected. This model features a built-in antenna and works in the 2.4 GHz band, more precisely in the range extending from 2.399 GHz to 2.485 GHz, its resolution have a spacing of 500 kHz and performs 173 samples over the entire band. This spectrum analyzer uses the 2.4GHz transceiver Texas Instruments CC2511F32 [13] with maximum sensitivity of -103 dBm at 2.4 kBaud.

The spectrum scanning takes approximately 270ms, which can generate about four readings of the whole 2.4 ISM band per second. This configuration is sufficient for our purposes in this work, since it allows us to analyze the occupation of the electromagnetic spectrum of interest, the 2.4GHz ISM.

D. Sensing Kit Assembly

Some extra features to enable the USB port, use a serial device and a package for data transmission across the network were needed. For this purpose a few extra packages were installed after installing OpenWRT.

The packages installation was made through opkg, the OpenWRT package manager, which can be used for both download and install packages from local repositories or from Internet repositories.

The modules installed were as follows:

- kmod-usb-core, version 2.4.37.9-1. Kernel support for USB.
- kmod-usb-serial, version 2.4.37.9-1. Kernel support for USB-Serial conversion.
- kmod-usb2, version 2.4.37.9-1. Kernel support for USB2 (EHCI) drivers.
- kmod-usb-acm, version 2.4.37.9-1. Module to support USB Abstract Control Model (ACM) devices.
- netcat, version 0.7.1-2. Service to read and write data across network connections.

Thus it was possible to use the USB spectrum analyzer and CDC ACM driver, which enable handling of the spectrum analyzer by reading and writing commands as a serial device.

After those installations, the spectrum analyzer readings were redirected to a text file in a temporary folder of OpenWrt. Also, a routine was created to insert a timestamps of the scans.

This complete information, timestamps and scans, are transferred over the network using the netcat tool [14]. The netcat allows reading and writing data via a network connection using the TCP / IP protocols. Netcat works by choosing a port for which the information will be targeted and when there is a connection on that port information is transferred to the host over the network.

Data collection is done through a host connection on the specified port using the telnet protocol. The scanning, timestamps insertion, internal storage and redirection of the information for the specified communication port have been automated through scripts.

E. User Interface

As an interface to check the measurements and make the analysis a website was created. In the process of creation it was necessary to: store the information obtained through the router in a database, develop an application to generate graphs based on these stored data and create a page for the graphics visualization.

As stated in the previous section, data is received by establishing a connection to the router on the specified port, using the telnet protocol.

In the whole process of receiving data, inserting them in a database and creating the website we used the Python programming language [15].

The database used in this work was SQLite [16]. The choice was based on its easy of use, maintenance and administration. If a more complex database as PostgreSQL or Oracle is needed for future works, the SQLite is totally portable. Moreover, the existence of sqlite3, an API Python programming language for SQLite databases was of vital importance in its choice. For more information about the sqlite3 module check the references [17].

The website was developed using Django framework. The Django is a framework for high-level Web programming done in Python, which enables a quick and simple application development. With Django it was possible to create an application with database interaction and the matlibplot library was used to plot graphics. Information about the library can be checked in reference [18] and all Django documentation is in the reference [19].

The development of the website has been simplified, using only HTML to generate user search forms and display their graphics to the required content.

At the website one can check instant measurements, the average power transmitted inside a time window or all measures accumulated over the monitored time in a single graphic. Furthermore it is also possible to observe only a desired channel or the entire spectrum monitored.

V. ACHIEVED RESULTS

The sensing kit creation proved to be satisfactory on its purpose. With one kit in hand it is possible to conduct a spectral analysis in an environment, Figure 1 shows how the sensing kit works. First, the spectrum analyzer installed at the router USB port starts reading the whole spectrum of the ISM band, from the 2.399 MHz to 2.485 MHz, with steps of 500 kHz from each reading. The signal strength is read in dBm, and the output that can be read from the console is like the following: “scan|0,-101 -102 -102 -102 -101 -104 -101 -101 -101 -101 -103 -100 -101 [...]”. As said, the value ”-101” corresponds to -101 dBm for the 2.399 MHz frequency and the ”-102” corresponds to -102 dBm for the 2.399,5 MHz frequency, and so on. When the reading reaches the final frequency there is a line break and the reading starts again over the ISM band. Each reading of the whole spectrum takes around 270 milliseconds, so we can have a maximum of 4 readings of the whole spectrum in a single second.

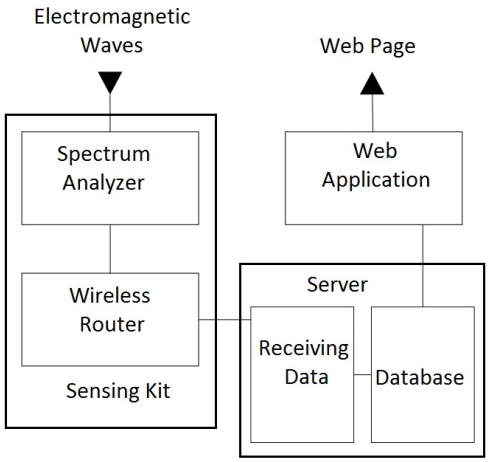


Fig. 1. Workflow diagram of the sensing kit.

A routine at the wireless router was created to save these readings in the router memory and to put a time-stamp information for each reading. So we had full spectrum readings with the information of when it was taken.

With readings in hand, we were able to send them to the server that was running the database and capturing the information sent from the wireless router. That information was stored in a SQL database with the information of time, frequency and power level of each frequency. Those readings were shown at the web interface created.

The kit’s performance on measurement, data transfer, data storage and data visualization through the web application created was tested, as shown at [V-A].

However, the biggest advantage is the possibility to create a distributed sensing infrastructure for spectral analysis.

A distributed sensing infrastructure makes possible a better description of the spectrum. Cross-correlating the data of

multiple sensing points can be used for root cause analysis, device identification, triangulation, and other applications.

It was possible to check interference levels on each channel and the received power average over time. With this information at hand and especially with the cumulative view a range of the spectrum more suitable for use was identified.

A. Testing the sensing kit

One kit was installed in a residential apartment whose environment had multiple devices using the monitored band. In the residence there was an access point, notebooks and mobile phones all associated with it, also microwave ovens and cordless phones, as well as multiple external networks of neighboring residences were present.

Performing an analysis over the data collected during one week in this scenario, we could observe an intense spectrum occupancy in all its breadth and especially on channels 1, 6 and 11 demonstrated in the Figure 2. These channels are particularly important since they are the unique orthogonal among themselves, i.e., the spectral mask of any such channels does not interfere with the spectral mask of the other.

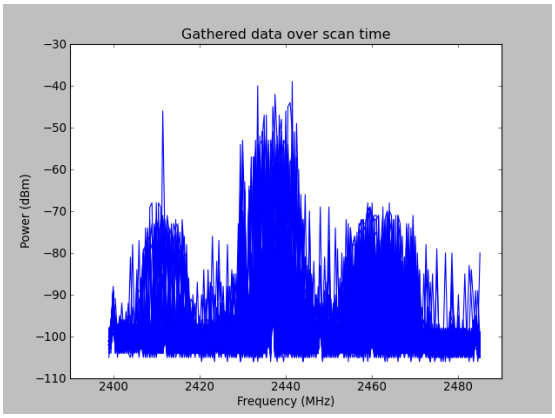


Fig. 2. Cumulative vision of the whole monitored spectrum

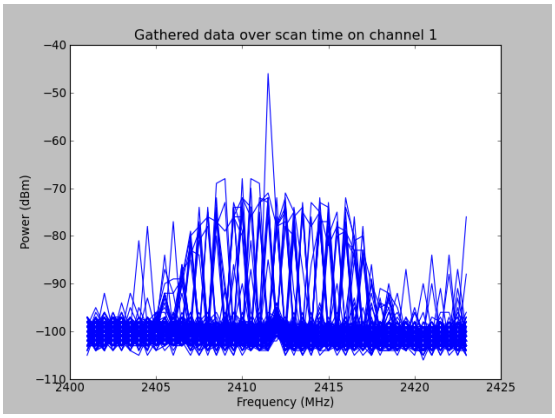


Fig. 3. Cumulative vision of the channel 1

Observing each of three channels separately in Figures 3, 4 and 5, they have similar spectral masks like the 802.11

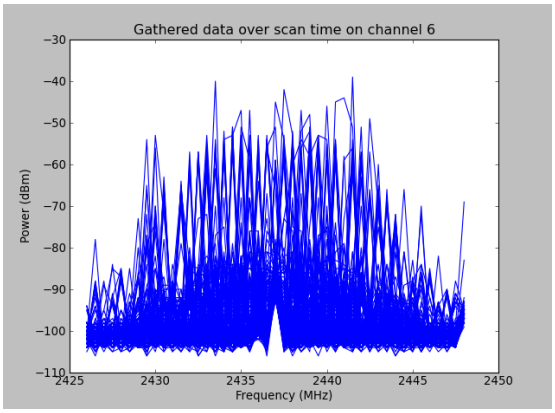


Fig. 4. Cumulative vision of the channel 6

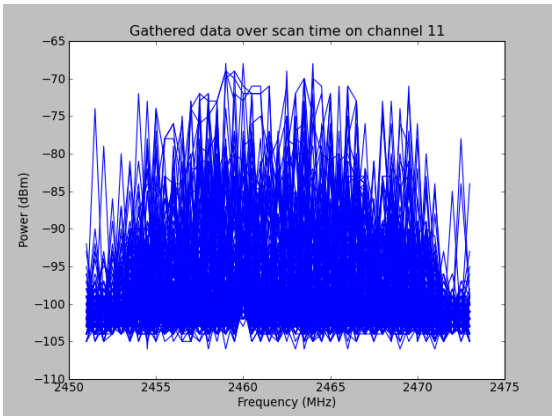


Fig. 5. Cumulative vision of the channel 11

family devices. Furthermore, channel 6 has a power intensity superior to other channels, reaching approximately 45 dBm in its central frequency, while the others are around 70 dBm. So it is possible to affirm the existence of devices either more powerful or close occupying the channel 6. This matches the channel used in the network access point in the residence.

Finally, channel 1 and 11 reaches a gain of SNR up to 25 dB compared with the channel 6. And the gain of channel 1 over 11 happens at the interference points, both controlled as the uncontrolled, which can lead to problems like delays, frame loss, loss of synchronization and cross-talk. Channel 11 presented a greater density of lines and his mask appears to be wider observing the cumulative view of the entire spectrum. That characterizes a heavier use of it, and the existence of more networks in the adjacent channels or the same channel.

So channel 1 is the most suitable for use by a new access point to be installed in the analyzed situation.

VI. Conclusion

This paper presented the assembly of a spectrum analysis kit, discussing each device used and the configuration needed to perform the required task.

A distributed sensing infrastructure can be created using multiple devices. A more detailed description of the spectrum

is possible using this infrastructure and some techniques, as cross-correlating the measures of multiple devices.

A test using the kit was performed. A scan over the ISM band (2.4 GHz) and the conclusion of the channel more propitious to install a new access point was made, that being channel 1.

It is worth mentioning the importance of analyzing this spectrum and to maintain a database of these measures, so that it is possible to detect anomalies in the use of spectrum, identify which type of service is used (taking into account their spectral mask) as well as find possible interference points.

Another point worth to be mentioned is the maintenance of the spectrum usage history. It allows an analysis of the usage progress over the time and assists on forecasting its use and planning.

The sensing kit that has been assembled in this work can be of great help to studies that propose to analyze the frequency range of the ISM band, since it is easy to install and assembly, and has a low cost. Therefore, it is an easy and practical way of creating a distributed sensing infrastructure.

This distributed sensing infrastructure improve the spectral analysis by allowing to create a correlation between the data collected by different environmental related points.

We think that this frequency range will become more and more the research target of research entities and providers companies of telecommunications equipment, so that their equipment stand out from the others in the market, since the market prospects are a true "*boom*" in the use of this frequency, and that the infrastructure assembled in this work can be used to analyze and also be used at case studies in this frequency range, obtaining satisfactory results.

As a future task, more unities of this kit will be installed at the same environment, thus, providing a more detailed analysis.

ACKNOWLEDGMENT

The authors would like to thank MidicaCom Laboratory for the assistance with the equipment and infrastructure used in this project.

REFERENCES

[1] A. F. Molisch, *Wireless Communications*, 2nd ed. Wiley - IEEE, 2011.

[2] M. P. Quirk, M. F. Garyantes, H. C. Wilck, and M. J. Grimm, "A wide-band high-resolution spectrum analyzer," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 12, pp. 1854–1861, 1988.

[3] J. Marinho and E. Monteiro, "Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions," *Wireless Networks*, vol. 18, no. 2, pp. 147–164, 2012.

[4] T. Yücek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 1, pp. 116–130, 2009.

[5] K. Baraka, L. Safatly, H. Artail, A. Ghandour, and A. El-Hajj, "An infrastructure-aided cooperative spectrum sensing scheme for vehicular ad hoc networks," *Ad Hoc Networks*, vol. 25, pp. 197–212, 2015.

[6] L. Pucker, "Review of contemporary spectrum sensing technologies," *IEEE SCC41IEEE*, 1900.

[7] M. Gast, *802.11 Wireless Networks: The Definitive Guide*, 2nd ed. O'Reilly, 2005.

[8] ANATEL, "Regulation on Restricted Radiation Radio Communications Equipment," http://www.anatel.gov.br/legislacao/resolucoes/2008/104-resolucao-506, Jul. 2008, accessed: 05/12/2013.

[9] J. F. K. e Keith W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. Pearson, 2010.

[10] D-Link Corporation / D-Link Systems, Inc., "DIR-320 Wireless G Router," http://www.dlink.com.br/produto/dir-320l, Nov. 2011, accessed: 05/12/2013.

[11] OpenWrt, "OpenWrt Wireless Freedom," http://openwrt.org, Nov. 2013, accessed: 05/12/2013.

[12] Ubiquiti Networks, Inc., "AirView Spectrum Analyzer by Ubiquiti Networks," http://www.ubnt.com/airview, Dec. 2013, accessed: 05/12/2013.

[13] Texas Instruments Incorporated., "Wireless Conectivity - Proprietary 2.4 GHz - CC2511F32," http://www.ti.com/product/cc2511f32, Dec. 2013, accessed: 05/12/2013.

[14] Giovanni Giacobbi, "The GNU Netcat – Official homepage," http://netcat.sourceforge.net/, Nov. 2006, accessed: 05/12/2013.

[15] Python Software Foundation, "Python Programming Language - Official Website," http://www.python.org/, Dec. 2013, accessed: 05/12/2013.

[16] "SQLite Home Page," http://www.sqlite.org/, Jan. 2014, accessed: 05/12/2013.

[17] P. S. Foundation, "sqlite3 - DB-API 2.0 interface for SQLite database - Python v2.7.6 documentations," http://docs.python.org/2/library/sqlite3.html, Jan. 2014, accessed: 05/12/2013.

[18] J. Hunter, D. Darren, E. Firing, M. Droettboom, and the matplotlib development team, "matplotlib: python plotting - Matplotlib 1.3.1 documentation," http://matplotlib.org/, Oct. 2013, accessed: 05/12/2013.

[19] D. S. Foundation, "The Web framework for perfectionists with deadlines — Django," https://www.djangoproject.com/, Jan. 2014, accessed: 05/12/2013.

A Holistic Approach to Enable Perceptive, Instrumental and Ubiquitous Smart eHealth

F. Ramalho[1], A. Neto[1], K. Santos[1], J. B. Filho[2], N. Agoulmine[3]
[1]Federal University of Rio Grande do Norte (UFRN), Natal-RN, Brazil
[2]State University of Piauí (UESPI), Teresina-PI, Brazil
[3]University of Evry Val d'Essonne, France
augusto@dimap.ufrn.br, {flavio, kelyson}@ppgsc.ufrn.br, bringel@uespi.br, nazim.agoulmine@ufrst.univ-evry.fr

***Abstract*— eHealth technology can bring about many benefits for patients, health professionals, and institutions by providing faster, safer, and better healthcare services through the integration of *Information and Communication Technologies* (ICTs) with eHealth. We believe the success of the next generation of healthcare services requires advances in modern computing to support truly intelligent, instrumental and interconnected capacities for smart eHealth, with proactivity, quality, scalability, security, automaticity, and reliability. This paper proposes a Multi eHealth Cloud Service Framework MeCa) adopting a holistic approach that embodies a hub of emerging services and applications in its architecture that comprise a technology for smart eHealth. A performance evaluation was carried out on a real testbed, and the results demonstrated that MeCa is able to optimize the constraints of mobile eHealth applications by saving energy consumption, as well as both CPU and memory usage in a Home-Assisted Living use cases.**

Keywords—eHealth; cloud computing; patient level biosensors; context-awareness.

I. INTRODUCTION

Information and Communication Technologies (ICTs) are increasingly providing tools, services, and knowledge for the healthcare domain (including patients, health professionals and institutions in both the private and public sectors) to allow changes leading to revolutionary eHealth [1]. eHealth envisages empowering the healthcare system with both affordable and personalized medical services through a cost-effective and secure use of ICTs, including remote patient/medical devices surveillance/sensing, real-time diagnosis, sharing of medical information/exams, and remote appointments/surgery [2]. The Mobile eHealth (mHealth) system leverages the increasingly sophisticated techniques of mobile computing to generate, handle, and disseminate patient health data/information. It provides a great opportunity to the network of eHealth stakeholders perform better through user-friendly applications on mobile devices (smartphones, tablets, laptops, and the like).

Recent advances made in the area of sensitive biological devices (biosensors) are creating new opportunities for value-added human-centered applications, such as Home-Assisted Living (HAL). HAL field is viewed as a revolutionary mHealth scenario where patients can be cared for in their own environments (their workplace, residence, entertainment center etc.). In this scenario, data analytics is carried out for patient-level multi-biosensing (e.g. measurements of the heartbeat, temperature, position and location, at the same time and in their specific technologies), and this enables specialists/technicians to monitor/act on them remotely [3]. For example, HAL mHealth application cloud configure both the ECG and glucose biosensors to send data in real time to a certain remote application upon detecting that a patient has exceeded his blood pressure health level, and is thus able to make a suitable and timely diagnosis.

As the number of patients increases the computing costs of mHealth applications (transmission, storage and data analytics) also increase and the processing time can become a critical factor as a result of the growing number of multi-biosensing operations [4]. Moreover, there is a sharp rise in the energy consumption of mobile devices with increasing requirements for both processing and wireless communication [5]. In addition to their worldwide penetration and facility, the features offered by mobile devices make them ideally suitable for the immediate physiological monitoring of patients, as with greater access to mobile phones to all segments of a country, including rural areas, the potential of lowering information and transaction costs in order to deliver healthcare improves. However, for more complex eHealth decisions and operational activities (e.g. a real-time diagnosis based on image processing), other models must be considered.

We believe the next generation of healthcare patients must be regarded in the light of the advances of modern computing so that they can support truly intelligent, instrumental and interconnected capacities for smart eHealth. Moreover, we expect that the service and applications of smart eHealth will allow shifting paradigms to provide facilities that ensure proactivity, quality, scalability, security, automaticity, and reliability, while at the same time, enhancing the quality of care, reducing the number of decision errors, and encouraging collaboration and healthy behavior [6]. Our previous work introduced the Context-Aware Mobile Approach (CAMA) [7][8] which is designed to turn eHealth raw data into context information on smart phones by semantically describing complex healthcare event patterns (i.e. situational, behavioral and conditioned) with a high degree of accuracy, and making them available in a central infrastructure. An example of a CAMA-assisted smart context is "*the patient has had a heart attack and requires immediate assistance at particular GPS location*", into a friendly enough structure to permit end-applications by employing low-complexity and resource-efficient methods to process the information. Moreover, this information is easily accessible to the user (through SMS

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

messages, instant messaging, web posting, etc.), and self-contained enough to optimize both the workload and the activities of health professionals. CAMA achieves this by employing intelligence-gathering techniques (integration of ontologies, fuzzy logic and neural networks) to deal with the real-world complexity and uncertainty for which traditional approaches (i.e. first principles modeling, explicit statistical modeling, and conditional statement) are ineffective or not feasible. However, CAMA is not suitable for the mHealth scenario, since its cumbersome and complex approach jeopardizes its survivability by aggressively overloading both the performance and sustainability of the mobile devices.

The time sensitivity of critical processing, along with the increasing amount of data that is expected to be generated in the future eHealth system requires a powerful, stable, and safer infrastructure that is well defined and can provide a seamless context-awareness for CAMA (with an mHealth scenario that is cost-effective). This means that the flexibility and high-performance capabilities, (in terms of low-cost computing, storage and software services), makes cloud computing a promising technology for this scenario. In addition, it provides the following improvements [9]: *(i)* a scalable and robust system, with powerful processing/storage capabilities and ubiquitous access; *(ii)* unified access to the processing and storage infrastructure; *(iii)* it makes it easy to share the processed data with common interfaces; *(iv)* flexibility in adopting new types of services and applications without the intervention of end-entities, among other factors.

In this study, we propose MeCa (Multi eHealth Cloud Service Framework), a holistic approach that embodies a hub of emerging services and applications in its architecture. This provides a seamless infrastructure for the integrated use of: *(i)* Context as a cloud service, to provision smart eHealth context information in compliance with application-specific demands; *(ii)* Sensor as a Cloud Service, to enable a multi-biosensing approach to gather multiple-patient biometrics, at same time, from different sensing technologies (i.e. platform, communication and access approach) and store it on the cloud; *(iii)* eHealth Application as a Cloud Service, enable smart eHealth applications to run on the cloud by leveraging high resource capacities when applying computational intelligence to provide powerful eHealth insights, in this way, minimizing battery, memory and CPU consumption on the mobile side. This also seeks to allow targeted applications to make complex decisions at different levels (i.e. involving a diagnosis and reactions that go beyond simply monitoring).

From the perspective of MeCa, the aim is to allow mHealth applications to adopt a notification-driven approach (rather than a coupled-access one), acting in a standby mode for the reception of on-demand notifications. Thus, MeCa's hub-service infrastructure makes the mHealth applications lightweight, simple and cost-efficient, since the underlying operational costs (i.e. memory, CPU and energy usage) are optimized by creating an infrastructure for all the required technologies. The purpose of this is: to gather raw data directly from the multiple sensors (and in their technologies); produce application-required information; and provide assistance for final decision-making and action. The performance evaluation of MeCa was carried out through prototyping in a real testbed, to accurately assess the expected benefits. The experimental results confirm that MeCa's hub-service infrastructure allows mobile devices to achieve optimized operational costs in terms of energy consumption, as well as CPU and memory usage rates when compared with the regular coupled-access scheme.

This paper is structured as follows. Section II examines important related work. Section III provides an overview of the proposed architecture, including its systems and subsystems. The performance analysis and examination of the results of the new architecture are carried out in section IV. Finally, Section V provides the conclusion and makes suggestions for future work.

II. Related Work

Our literature study provides evidence of the interest shown by the research community in the application of modern computing technologies to improve eHealth systems. Our research explores projects that employ two access paradigms, (couple-based and cloud-based), which are discussed in the following section.

On the one hand, most eHealth applications usually adopt the widely used couple-based access approach for application-to-sensor intercommunications, such as [10][11][12][13][14]. The general purpose of these works is to operate a wireless multi-parameter monitoring system that gathers data (ECG, blood pressure, body temperature and respiration) from biosensors, and sends this data to a central system to be analyzed by different specialists (medical personnel, applications). In the regular couple-based access approach, eHealth applications must implement all the required technologies to gather biosensing data directly and in their specific technologies (i.e. communication, platform, language, and the like). This approach is very demanding and expensive owing to the increasing sensor density, and this can impair its performance in terms of processing, time response, scalability and other factors, as well as increasing the complexity of the system [3]. On the other hand, emerging works, such as [15][16][17][18][19], are exploiting cloud infrastructures as an alternative to the couple-based access scheme, as a means of leveraging robustness, high-performance and stability cloud stack capacities which can provide an improved service and applications. Finally, the works [20][21][22] propose context-aware approaches for eHealth systems. In all works, the context represents a very simple structure, even when the biosensor raw data is taken into account. Thus, end-applications are avoided when the data analytics are deployed.

None of the works referred to above, deploy a holistic service infrastructure, with common interfaces and value-added hub of an integrated service, nor do provide holistic context-awareness by considering smart context information. The limitations pointed out in this section led us to adopt the MeCa approach, which is described in the next section.

III. Overview of the MeCa Proposal

The MeCa proposal exploits modern computing techniques as a means of enabling ICTs to allow the current healthcare

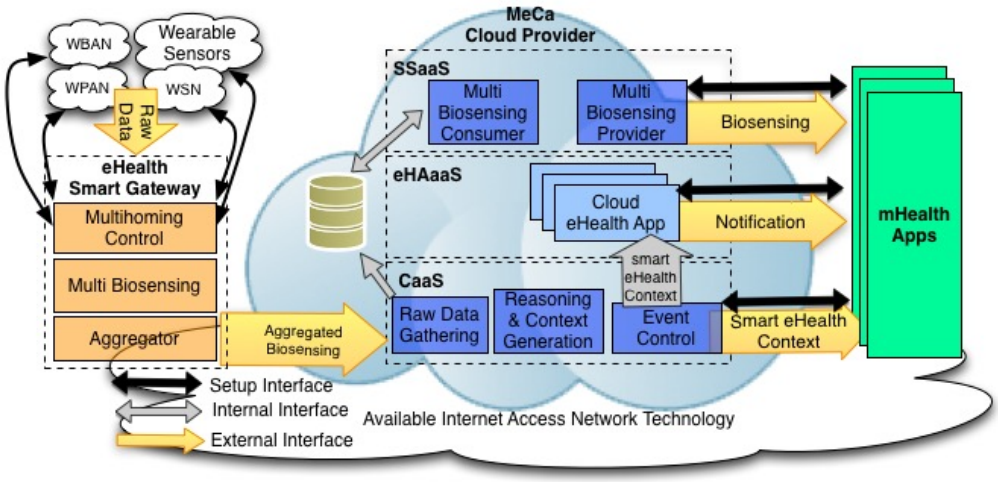


Figure 1. Architecture of the MeCa Approach

system to be turned into a proactive, faster, safer, reliable and much better service. Moreover, it assists in reducing the workload and activities of the health domain (professionals, institutions, stakeholders, patients, etc.), while improving the quality of life of patients. To achieve this, MeCa aim to deploy a holistic cloud infrastructure for a hub of modern services and applications that seek to seamlessly provide a set of integrated facilities with ubiquitous access, in terms of smart eHealth, multi-biosensing and eHealth smart Applications. The MeCa cloud provider allows eHealth applications running on resource-constrained fixed/mobile devices to benefit from the notification-driven approach. This entails act as a front-end to high-resource demands (processing and storage) eHealth application(s) running on the MeCa cloud provider to achieve their specific goals in a cost-effective manner (i.e. optimized CPU and memory usage, as well as energy consumption).

For instance, an eHealth expert system running on the MeCa cloud provider is able to implement heavy smart surveillance algorithms for real-time patient video monitoring. This way, the eHealth expert system can produce on-demand application-specific notifications (through preconfigured thresholds) and convey unexpected critical patient-level smart context information. In this scheme, end applications can be standby for short/long periods. As a result, they are able to exploit the infrastructured support facilities and only react when they receive the MeCa's cloud context notification. For example, this can be carried out by calling on a nearby ambulance service and/or authority in response to critical circumstances or even notifying remote health professionals (medical doctors, nurses, even family members) and/or systems to take action accordingly. We expect that the MeCa hub-service infrastructure will provide huge opportunities for implementing new and value-added eHealth applications intended to improve the planning and managing of healthcare resources. It should be stressed that the specification of eHealth applications, data analytics techniques, and other related fields are not within the scope of this paper. Figure 1 illustrates the architecture of the MeCa-enabled system.

In the MeCa approach, the complex context-awareness support is created in a fixed and high-computational cloud-extended (Infrastructure as a Service – IaaS -level) information structure. This makes mHealth applications lightweight and means they have lower computational costs. The MeCa architecture comprises three systems, each with several embedded sub-systems that provide internal and external interfaces for inter-system communication and displays of MeCa operations, respectively.

A. eHealth Smart Gateway (eHSG):

The eHSG is a MeCa component which capabilities includes multi-biosensing data, data aggregation, and network transmission. The eHSG is composed of the following sub-systems:

- The Multihoming Control (MhC) sub-system that is responsible for enabling the underlying network technology (e.g. Bluetooth, Wi-Fi, ZigBee) according to the required biosensing data;
- Multipart Biosensing (MB): The MB is responsible for receiving all the collected data sensing from the MhC and storing it in a local database;
- Aggregator (AG): The AG processes the received data provided by the MB, and sends aggregated information to the Cloud provider

B. MeCa Cloud Provider (MCP):

The MCP well-defined hub-service cloud infrastructure incorporates the following components:

i. Context as a Service (CaaS):

eHealth is context-sensitive in nature, and thus behavioral and situational patterns guide decisions at different levels of complexity. Without its context-awareness capability, it would be very difficult to make intelligent decisions based solely on raw data sensing. It is a hard task to produce contextual healthcare information, especially in mobile scenarios due to their unpredictable situational patterns. The efficiency of context-awareness depends on accurate reasoning (sensor data

processing), which is the driving-force for cooperative and intelligent decisions [3]. The autonomous properties of a context-aware environment are promising because they are able to guide dynamic, automated and real-time activities [3].

For a well-structured approach, a set of formalisms is required to control the generated context and resulting high level of information, which is handled by MeCa. Hence, as our focus is not on algorithms for the context generation, the CaaS implements our previous work on CAMA [7] to generate eHealth smart context information with a high degree of accuracy. The CaaS stores the context information into a cloud storage service, which is available for internal and external consumers. The CaaS architecture is as follows:

- The Raw Data Gatherer (RDG) accesses biosensing raw data in the database inside the MCP, which is designed to follow the connection-decoupled Tuple Space concept so that it can cope with the dynamicity of the mobile scenarios, as well as allow an improved indexing scheme to retrieve a set of information that involves low-cost processing;
- The Reasoning & Context Generator (RCG) is responsible for generating eHealth smart context information by matching the requirements of particular eHealth (preconfigured) applications. Hence, the applications use a well-defined interface to configure the context process, which involve knowledge of the required context, data sources, the way the contextualization will be carried out and other relevant information. The CG is a well-defined interface that allows an eHealth application to explicitly request an existing body of sensing data and/or the contextual information of a patient. In carrying this out, the eHealth application must show the identity of the data, as well as other relevant information, so that the CG can index the stored information and return the intended context information to the interested eHealth applications.

ii. eHealth Application as a Service (eHAaaS):

The eHaaS provides an environment for hosting eHealth applications, which is based on the information provided by the CaaS component. These applications benefit more from the cloud computing infrastructure than the traditional IT service environment, because they have more scalability, flexibility, reduced capital and higher resource utilization capabilities [23]. The goal of these applications is to notify external application(s) of information that is contextualized owing to certain pre-defined conditions. Thus, the applications are connected to a standby system instead of consuming resources for monitoring entities all time.

iii. Sensor as a Service (SSaaS):

The SSaaS is responsible for providing raw biosensing for applications interested in this kind of content. The SSaaS is able to do this because it consists of the Multi Biosensing Consumer (MBC) subsystem. This allows it to collect aggregated raw data from the local database, and provide it to specific applications through the Multi-Biosensing Provider (MBP).

IV. EVALUATION

The aim of carrying out an evaluation of MeCa is to assess the benefits of the proposed MeCa Architecture, and compare it with that of the regular couple-based access.

A. Description of the Testbed and Experiments

The evaluation is carried out for a testbed configured with real services, resources, devices, and technologies. The Testbed architecture embodies a Body Area Network (BAN) system integrated with a single-board computer, a cloud infrastructure, and an mHealth application. The BAN system is built with the aid of an Arduino board MEGA 2560 [24], which connects an e-Health Sensor Platform 2.0 [25] and includes five biosensors (i.e. an electrocardiogram, galvanic skin response, airflow, temperature and position). The single-board computer is implemented on the Raspberry Pi [26] platform, and hosts a HAL application. A desktop hosts an OpenStack [27] compliant cloud infrastructure by deploying the MeCa approach. The mHealth application is Android-compliant and runs on a Samsung GT-I9192 mobile device. It is responsible for receiving the contextualized information and providing the medical report to the doctor/patient. In the MeCa scenario, this report is a simple notification that is launched when an emergency event is detected in the monitored patient, which means that while the patient is healthy, no notification will be sent. The set of experiments used in the MeCa assessments are described in the following sections.

i. Regular set of Experiments

The architecture adopted in the Regular set of Experiments is shown in Figure 2, which comprises the BAN system, the single-board computer and the mHealth application. The BAN system periodically (at each second) stores all the biosensor data (described above) in a MySQL database implemented by the single-board computer. The HAL application (on the single-board computer) waits for the mHealth application requests so that it can provide the biosensor data of its interest. The mHealth application must gather biosensor data from the HAL application via API sockets. Moreover, the mHealth application is responsible for processing the biosensor data to generate the eHealth context information, as well as generate on -demand eHealth notifications for the end-user.

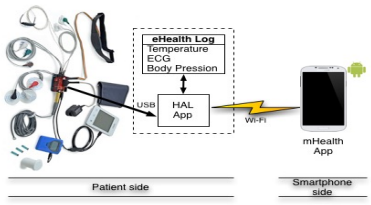


Figure 2. Regular testbed architecture

ii. MeCa-based set of Experiments

The architecture deployed for the MeCa-based set of experiments is depicted in Figure 3, and adopts the BAN system, the single-board computer, the cloud infrastructure and the mHealth application. In this case, the BAN system writes all the biosensors in the MySQL database that is implemented in the cloud infrastructure, via a web service

scheme. Three application instances are implemented in the cloud: *(i)* Biosensors as a Service, to provision patient level biometrics; *(ii)* Context as a Service, to generate eHealth context information; *(iii)* and HAL Application as a Service, to provide eHealth context-driven on-demand notifications to the mHealth application at the smartphone. In this set of experiments, the mHealth application at the smartphone is much simpler and more lightweight than the regular set of experiments, with the prospect of staying connected to standby. In addition they spend most of the time waiting for eHealth notifications that are generated on-demand by the HAL cloud application, thus denoting a cost-effective and sustainable approach.

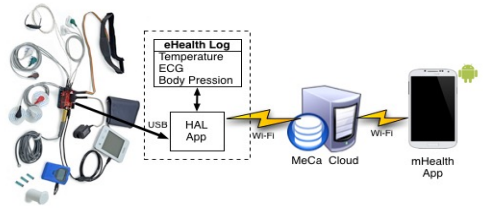


Figure 3. MeCa-Based testbed architecture

B. *Methodology employed for the Assessments*

When conducting the experiment, we used the PhysioBank collection [28], which is readily available for the research community. The PhysioBank allowed us to collect a total of 3,600 eHealth notifications sent out during the period of one hour of experimental time. We changed the biometrics records so that we could generate four datasets, each with a variation in the number of eHealth notifications (i.e., 10%, 25%, 50% and 80%). The first dataset has 10% of notification events (i.e. 360 notification), and so on. The benchmarking considers measures of battery consumption, CPU and RAM usage on the smartphone obtained via a specific android application developed by our team.

C. *Results of the Analysis*

To study the impact of MeCa system on the services lifetime, we measured the battery consumption rate in both sets of experiments. The results obtained are shown in Figure 4.

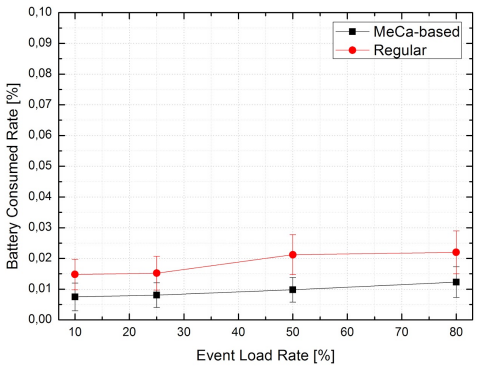


Figure 4. Remaining battery capacity compared with the varying event load

As expected, when the event load increases, the percentage of energy consumed in both scenarios remains high. This is because while there are more notifications, the mHealth application has to process the events and send the medical reports more frequently. The MeCa proposal allows an average improvement in the battery consumption rate of 47% as the mobile device only receive the notification, it does not process any data. The results of the CPU load obtained in both experiments are shown in Figure 5. The purpose of the experiment was to examine the impact of MeCa hub-service facilities on the mHealth resource allocation in the mobile device for performance assessments.

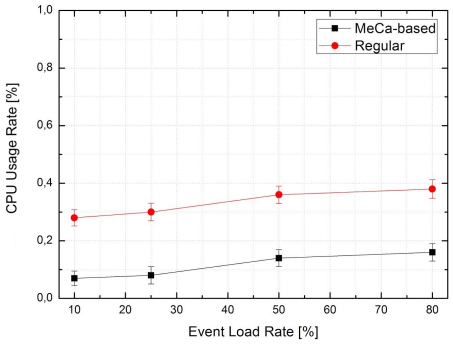


Figure 5. CPU usage rate compared with the varying event load

As shown in Figure 5, the CPU load certainly increases with the event load and this can be explained by the need to report a user-event whenever there is a cloud notification. Thus, the more notifications arrive, the more user reports are generated. Even so, MeCa allows the CPU load to make an average improvement of 65% compared with the Regular set of experiments, which is described in section 4.1. The main reason for the MeCa's improvement is that all the context generation processing is carried out on the cloud, whilst the mobile node deploys that on the Regular approach. This behavior can be confirmed in the analysis of memory usage, which traces the amount of bytes allocated to the RAM of the mobile node, as shown in Figure 6.

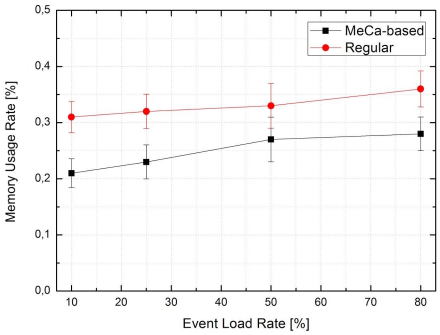


Figure 6. Memory usage rate compared with the varying event load

As shown in Figure 6, the RAM usage increases as the event load increases. Similarly to the battery and CPU measurements, the memory consumption for the MeCa-based set of experiments is also always smaller, an average optimization ratio of 25%. This behavior is also achieved because the MeCa approach transfers most of the complex processing operations to the cloud, whilst in the Regular approach are transferred to the mobile node.

The significant improvements in the battery, CPU and memory consumption rates obtained from the MeCa-based set of experiments compared with the Regular one, are essentially explained by the influence of the notification-based scheme, which allows a sharp decline in the number of communication demands on the mobile device. While the mHealth application generally remains connected in standby mode in the MeCa-based set of experiments, the coupled-based access approach deployed by the mHealth application in the Regular set of experiments, requires frequent data exchanges to gather biosensing raw data from the remote system of the biosensors.

V. Conclusion and Future Work

This work proposes MeCa as a means of allowing the current healthcare systems to evolve towards the future smart eHealth. The MeCa proposal deploys a holistic cloud infrastructure that embodies a hub of modern services and applications seeking to provide a seamless, integrated Context, Sensor and eHealth Application as a cloud service with ubiquitous access. The main benefit of MeCa is that it can alleviate the task of designing eHealth applications by deploying a notification-based approach to ensure an optimized performance and the survivability of mobile devices. Moreover, MeCa supports context-awareness to provide enhanced insights while keeping mHealth applications lightweight.

The MeCa proposal is still an ongoing project, and we know we will have to face many challenges to be able to accomplish it within a couple of years. As future work, we plan to apply MeCa in other eHealth scenarios and address the task of adding further refinements.

Acknowledgment

This work has been developed under the ASgARD project (CNPq Edital Universal 14/2014, grant agreement n. 457051/2014-0). Authors also thank to CNPq and CAPES.

References

[1] G Resolution 58/28 of the World Health Assembly, Geneva, 2005.

[2] International Telecommunication Union. "*Implementing e-Health in Developing Countries: Guidance and Principles*". Draft September 2008. Retrieved in 2nd July 2013. Available in http://www.itu.int/ITU-D/cyb/app/docs/e-Health_prefinal_15092008.PDF.

[3] J. Antoniou, C. Christophorou, A. Neto, S. Sargento, F. Pinto, N. Carapeto, J. Simoes and A. Pitsillides, "*Context-Aware Self-Optimization in Multiparty Converged Mobile Environments*". In AUTONOMICS'09, Limassol, Cyprus, Sep 2009.

[4] Rolim, Carlos Oberdan, et al. "A cloud computing solution for patient's data collection in health care institutions." *eHealth, Telemedicine, and Social Medicine, 2010. ETELEMED'10. Second International Conference on*. IEEE, 2010.

[5] G. Kreps, L. Neuhauser, "*New directions in eHealth communication: Opportunities and challenges*", In: Patient Education and Counseling, Vol. 78, Issue 3, pp. 329-336, March 2010, ISSN 0738-3991. doi 10.1016/j.pec.2010.01.013.

[6] G. Fortino, D. Parisi, V. Pirrone, G.Fatta, *"BodyCloud: A SaaS approach for community Body Sensor Networks"*, In: Future Generation Computer Systems, Vol. 35, pp. 62-79, June 2014. doi 10.1016/j.future.2013.12.015.

[7] A. Neto, J. Junior, J. Neuman, E. Cerqueira, "*Context-aware eHealth information approach for the Brazilian primary healthcare system*", in: Proceedings of 15th IEEE International Conference on e-Health Networking, Applications & Services (Healthcom), pp. 274 -276, Lisbon, Portugal, 2013. Doi: 10.1109/HealthCom.2013.6720682

[8] F. Silva, J. Castillo, A. Neto, F. Silva, P. Rosa. "*Software Defined eHealth Networking Towards a Truly Mobile and Reliable System*". In: IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom 2014), 2014, Natal-RN, Brazil. Doi: 10.1109/HealthCom.2014.7001903.

[9] G. Fortino, M. Pathan, G. Di Fatta, "*BodyCloud: integration of cloud computing and body sensor networks"*, in: Proceedings of 4th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2012, pp. 851–856, 2012.

[10] S. Dai, Y. Zhang, "*Wireless Physiological Multi-parameter Monitoring System Based on Mobile Communication Networks"*, in: 19th IEEE Symposium on Computer-Based Medical Systems Based on Mobile Communication Networks, Washington, DC, USA: IEEE Computer Soceity, pp. 473-478, 2006.

[11] S.M. Shahrokhi, Y. He, "*Energy-saving MAC scheme with dynamic transmission thresholds for body sensor networks* ",Int. J. Sensors Sensor Netw., 1 (6), 2013. Doi: http://dx.doi.org/10.11648/j.ijssn.20130106.11

[12] B. Lo, S. Thiemjarus, R. King, G.Z. Yang, "*Body sensor network—a wireless sensor platform for pervasive healthcare monitoring",* in: Proc. of the 3rd International Conference on Pervasive Computing, PERVASIVE 2005, Vol. 13, pp. 77–80, 2005.

[13] M. Patel, J. Wang, "*Applications, challenges, and prospective in emerging body area networking technologies",* in: IEEE Wirel. Commun., 17 (1), pp. 80–88, 2010

[14] Koumaditis, Konstantinos, et al. "Cloud Services for Healthcare: Insights from."*Delivery and Adoption of Cloud Computing Services in Contemporary Organizations* (2015): 292.

[15] O. Diallo, J. Rodrigues, M. Sene, J. Niu, "*Real-time query processing optimization for cloud-based wireless body area networks"*, in: Information Sciences, Volume 284, Pages 84-94, ISSN 0020-0255, 10 November 2014. Doi: http://dx.doi.org/10.1016/j.ins.2014.03.081.

[16] A. Bourouis, M. Feham, A. Bouchachia "*A new architecture of a ubiquitous health monitoring system: a prototype of cloud mobile health monitoring system",* in: Int. J. Comput. Sci. Issues (IJCSI), 9 (2), pp. 434–439, 2012

[17] H. Lin, J. Shao, C. Zhang, Y. Fang, "*CAM: cloud-assisted privacy preserving mobile health monitoring",* in: IEEE Trans. Forensics Secur., 8 (6), pp. 985–997, 2013.

[18] C.O. Rolim, F.L. Koch, C.B. Westphall, J. Werner, A. Fracalossi, G.S. Salvador, "*A cloud computing solution for patient's data collection in health care institutions"*, in: Proc. of ETELEMED, IEEE, pp. 95–99, 2010.

[19] Koumaditis, Konstantinos, et al. "Introducing a Patient-Centered e-Health Record Over the Cloud." *GLOBAL HEALTH 2014, The Third International Conference on Global Health Challenges*. 2014.

[20] R. Alharthi, R. Albalawi, M. Abdo, A. El Saddik, "A context-aware e-health framework for students with moderate intellectual and learning disabilities,". In: *2011 IEEE International Conference on Multimedia and Expo (ICME),* July 2011. doi: 10.1109/ICME.2011.6012218

[21] J.O. Oladosu, J.O. Emuoyibofhare, S.O. Ojo, M.O Adigun, "Framework for a Context-Aware Mobile Ehealth Service Discovery Infrastructure For Rural/Suburban Healthcare", Journal of Theoretical and Applied Information Technology, vol 6, no 1, pp 81-91, 2009.

[22] F. Bergenti et al. "Context-aware Service Coordination for Mobile e-Health Applications", In: First European Conference on EHealth, 2006.

[23] S. Nandgaonkar, A. Raut, "*A Comprehensive Study on Cloud Computing*". International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, pp. 733-738, April 2014.

[24] http://arduino.cc/en/Main/arduinoBoardMega2560

[25] http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical

[26] http://www.raspberrypi.org/

[27] http://www.openstack.org/

[28] http://www.physionet.org/cgi-bin/atm/ATM

An iRODS-based Distributed and Federated Data Repository for a Multi-CMF Network for Experimentation

Thiago S. Hohlenweger, Marcelo Pinheiro, Adriano L. Spínola, Igor L. Macedo, Joberto S. B. Martins
Computer Network Research Group (NUPERC)
Salvador University (UNIFACS)
Salvador, Brazil
{thiagosh81, marcelo.mpinheiro, adriano.spinola, igorleoem, joberto.martins}@gmail.com

Raphael A. Dourado, José A. Suruagy Monteiro
Centro de Informática (CIN)
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil
{rasd2, suruagy}@cin.ufpe.br

***Abstract*—A Network for Experimentation (NfExp) is, typically, a large (regional, national or international) and specialized testbed (GENI, PlanetLab, FIBRE, other) developed in order to support the experimentation of new protocols, services and applications mostly in the context of Future Internet (FI). In effect, a Network for Experimentation allows an experimenter to configure instantiate, run and collect measurements of distributed experiments on top of a testbed managed by a Control and Monitoring Framework (CMF). Recent Network for Experimentation deployments are Multi-CMF testbeds in which various existing CMFs are integrated in order to achieve a larger set of supported applications and services. This paper describes a distributed and federated storage solution for monitoring data in a Multi-CMF scenario (FIBRE testbed). The solution adopts a distributed iRODS-based monitoring data storage scheme, supporting data distribution and federation across islands (institutions) implementing the testbed. A distributed and transparent data storage facility among multiple native database systems was implemented supporting real-time and persistent monitoring data storage. It allows federated seamless monitoring data storage/retrieval for FIBRE data integration.**

Keywords—Network for Experimentation (NfExp); Control and Monitoring Framework (CMF); FIBRE testbed; iRODS; Data Storage; Monitoring; Federation.

I. INTRODUCTION AND MOTIVATION

"Networks for Experimentation" (NfExp) like PlanetLab, GENI and FIBRE have been widely deployed around the world, mostly by academic research projects. These networks, also known as "testbeds", are intended to support "experiment running" and innovative research for new protocol, services and applications as required, for instance, in the context of Future Internet (FI) [1].

A Network for Experimentation is controlled and managed by a Control and Monitoring Framework (CMF). The CMF is, in brief, responsible for resource identification, allocation and overall orchestration. Among the multiple features and capabilities supported by a CMF, we have access control, overall resource allocation and orchestration, instrumentation and monitoring support and federation, just to mention a few [2].

The context of the distributed and federated data repository presented in this paper is the testbed named FIBRE network [3]. The FIBRE testbed is a Multi-CMF network that supports wired and wireless technologies on a transcontinental networking structure between Brazil and Europe. The FIBRE CMF testbed has a control mechanism that orchestrates resources and the monitoring part of the FIBRE CMF is defined by the Instrumentation and Monitoring Architecture (I&M) [4].

The Multi-CMF deployment approach adopted by FIBRE testbed (control and monitoring) consists in, firstly, preserving native CMFs features and capabilities. Following that, the FIBRE CMF implementation challenge consists in developing a new "meta-CMF" placed on top of the existing ones in order to integrate and federate available features and capabilities and incorporate new ones eventually necessary in order to achieve the FIBRE overall objectives and requirements.

The Instrumentation and Monitoring Architecture (I&M) adopted by FIBRE testbed also maintains the basic deployment principle of preserving native CMFs monitoring facilities and introduces the need of a new FIBRE-specific data repository. The motivation is firstly due to the FIBRE requirement that users should be unaware of any native deployment details and perceive the FIBRE testbed as a single entity. The second deployment aspect to consider is due to the overlapped characteristics of the FIBRE network. In effect, monitoring data in FIBRE is collected by both native CMFs monitoring tools and new tools implemented specifically for FIBRE network. As such, a distributed and federated data repository is necessary in order to store/ retrieve monitoring data in accordance with FIBRE Instrumentation and Monitoring Architecture.

The integrated and federated data repository solution for FIBRE is presented in the following sections. Section II and III describe the basic infrastructure of the FIBRE testbed, its characteristics and requirements. Section IV introduces the FIBRE Instrumentation and Monitoring Architecture (I&M).

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

Sections V and VI describe the iRODS-based distributed repository and section VII presents a prototype virtual setup created in order to demonstrate the iRODS-based repository.

II. FIBRE TESTBED – ISLANDS AND "NATIVE" CONTROL AND MONITORING FRAMEWORKS

The FIBRE testbed is an intercontinental network infrastructure composed by 10 islands in Brazil and 03 islands in Europe physically interconnected by RNP (Brazilian Research Network) and GEANT (Figure 1) [3].

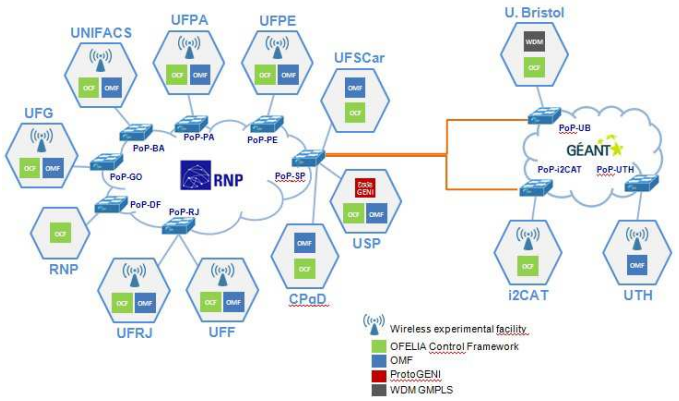


Fig. 1. The FIBRE testbed islands

FIBRE´s island basic infrastructure supports three Control and Monitoring Frameworks (CMFs): OFELIA Control Framework, cOntrol and Management Framework (OMF), and ProtoGENI. These CMFs are integrated in a single entity, the FIBRE testbed [5] [6] [7] [8].

III. FIBRE TESTBED REQUIREMENTS

The FIBRE testbed requirements reflects basically the Multi-CMF structure of the testbed which is composed by the OFELIA, OMF and ProtoGeni frameworks operating concomitantly.

In OFELIA, OMF and ProtoGeni CMFs, the collected experiment data is processed in order to allow two basic monitoring functionalities:

- Experiment (monitoring) data visualization through, typically, a web portal (both "infrastructure" and "experiment" monitored data); and
- Persistent experiment monitoring data storage.

In terms of on-the-fly monitored data access, OFELIA, OMF and ProtoGeni process the captured experiment and infrastructure data according distinct data representation and format that are inherent to their CMF. In the same way, display of real-time monitored data to the experimenter (user) is processed distinctively.

In terms of the persistent storage functionality available on OFELIA, OMF and ProtoGeni ("native" CMFs), the monitored parameters are stored on a specific database system (SQL, Postgres, RRD, other database) which is in fact a CMF implementation choice. Beyond that, the format and semantics associated to the monitored data is not necessarily compatible among the native FIBRE´s testbed CMFs.

As such, the FIBRE testbed data repository has to consider specific requirements such as:

- FIBRE monitoring data format, representation and semantics;
- FIBRE seamless data storage; and
- FIBRE monitoring data federation.

The FIBRE´s testbed monitored data format, representation and semantics define a uniform way to represent and publish captured data (FIBRE standard). As such, data manipulation and visualization tools implemented by FIBRE testbed can retrieve process and display data to FIBRE users using a common standard. This standardization is necessary since OFELIA, OMF and ProtoGENI store their measurement data with distinct formats, usually not compatible among them.

FIBRE seamless data storage is fundamentally a user requirement defined for the FIBRE testbed deployment. In effect, OFELIA, OMF and ProtoGENI store their monitored data using their own storage methodology using distinct locations and strategy. As such, retrieving the monitored data from native CMFs concomitantly implies in having FIBRE testbed supporting a set of methods for CMFs used and, beyond that, the FIBRE user would be supposed to have some knowledge about monitored data location. Aiming to simplify FIBRE users interface in accessing and manipulation the monitored data, a seamless data repository was defined as required by the FIBRE testbed.

Federation capability of monitored data is another important FIBRE testbed requirement. Beyond the Multi-CMF characteristics of FIBRE, its actual deployment is realized across multiple administrative domains. Each island of the FIBRE testbed is implemented in most cases at different Institutions (university, research facility, other) and possibly in different countries. As such, federate the monitoring data repository is necessary in order to deal with the technical and managerial aspects involved in having monitored data spread among distinct administrative domains.

The federation requirement also leads to the issue of adopting distributed file storage for the monitored data repository. The FIBRE testbed defines that the monitored data repository should be distributed in order to support more flexible management and control of the Multi-CMF testbed. It is assumed that each administrative domain may have its own CMF data repository, which in turn, should be seamless interconnected and shared with other administrative domains repositories. All experiment data repository facilities should be transparently accessed.

IV. FIBRE TESTBED – INSTRUMENTATION AND MONITORING (I&M) ARCHITECTURE

The FIBRE Instrumentation and Measurement (I&M) Architecture (FIBRE I&M) (Figure 2) corresponds to the "monitoring" part of the FIBRE testbed and, by definition, has the capability to configure, monitor, collect, store and display both infrastructure and experiment specific monitored data for federated and/or individual CMFs aggregates (OMF – OMF, OFELIA – OMF, OFELIA – OFELIA, ProtoGENI – OFELIA, other) in FIBRE testbed [4].

One important component of the FIBRE I&M Architecture is the Measurement Data Integration Point (MDIP). The MDIP

is a CMF-dependent component (OFELIA, ProtoGENI and OMF specific deployments) that collects monitored data on any native CMF and brings it to a FIBRE-specific monitored data standard. MDIP does guarantee that currently available monitoring tools for OFELIA, OMF and ProtoGENI can remain being used by FIBRE testbed.

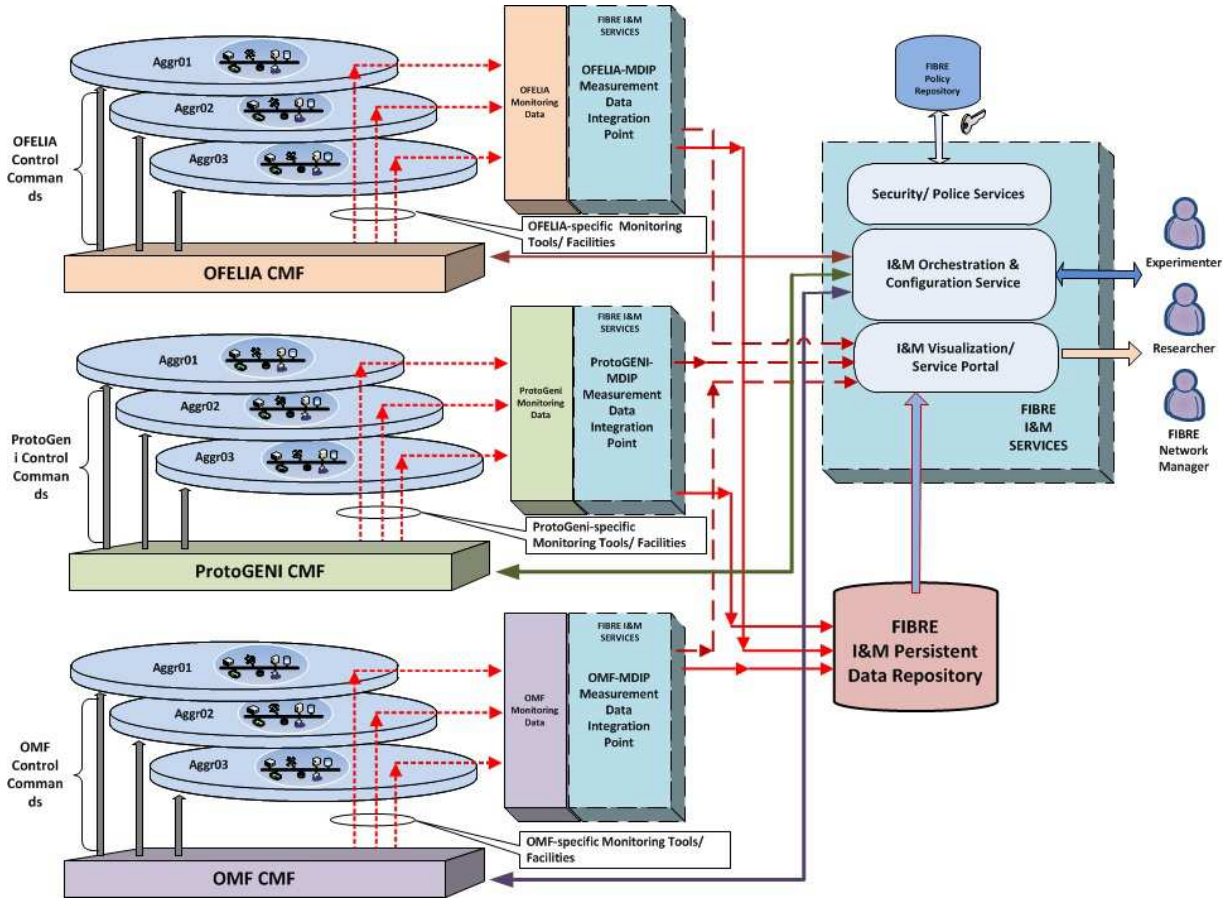


Fig. 2. FIBRE Instrumentation and Monitoring Architecture

I&M Architecture also specify the approach adopted for user access, monitoring tools instantiation, monitoring data capture, visualization and storage. In effect, there are I&M services for user authentication, monitoring tools instantiation, and a service portal allowing both tools configuration and monitored data visualization.

In relation to the object of this paper, I&M specify the need of a distributed persistent monitored data repository that can be seamlessly and transparently be accessed by FIBRE users (experimenters, network managers and other institutional authorities, other users).

In the next section, a technical alternative for a distributed and federated data repository is introduced and a suitable deployment alternative for FIBRE testbed is presented.

V. FIBRE TESTBED – MONITORING DATA AND DATA REPOSITORY

Considering the FIBRE´s data repository requirements, the testbed requires a solution adopting a standard monitoring data format and representation associated with a distributed data repository infrastructure supporting seamlessly operation.

- FIBRE Testbed Monitoring Data Format

In terms of the monitoring data format and basic manipulation capabilities, the FIBRE I&M Architecture uses NM-WG (Network Measurements Working Group), an Open Grid Forum standard for measurement information exchange. NM-WG uses (but is not restricted to) XML as markup language, and besides natively supporting a set of metrics, is also extensible to incorporate new ones [9].

- FIBRE Testbed Distributed and Federated Data Repository

The FIBRE testbed data repository solution is based on iRODS that has distributed storage capabilities and support for diverse experimental network storages [10] [11]. In sequence, iRODS is briefly introduced in terms of its architecture, main operating characteristics, components and technical aspects that motivate its adoption for FIBRE.

The Integrated Rule-Oriented Data System (iRODS) is a data grid solution supporting a centralized or distributed data storage allowing data management, sharing and protection [10]. iRODS has three main components: the iCAT Metadata Catalog, the iRODS Server and iRODS clients.

The iRODS Metadata Catalog (iCAT) is a server storing database metadata for iRODS Server stored data from distinct DBMS like PostgreSQL, MySQL or Oracle. The metadata catalog is an important asset of iRODS in relation to FIBRE

requirements. In effect, metadata can be used in FIBRE in order to provide a semantically uniform monitoring data representation and storage.

The iRODS Server is a storage server using access protocols supporting distributed and seamless data storage and retrieval in distinct administrative domains. The relevant technical aspect of iRODS in relation to FIBRE requirements is its seamless data storage and retrieve facility. Beyond that, iRODS does provide a federation capability which is essential to FIBRE in order to cope with its inherent multi-domain deployment characteristics (across multi administrative domains).

iRODS clients are Web applications supporting a set of functionalities such as data search, access and metadata management.

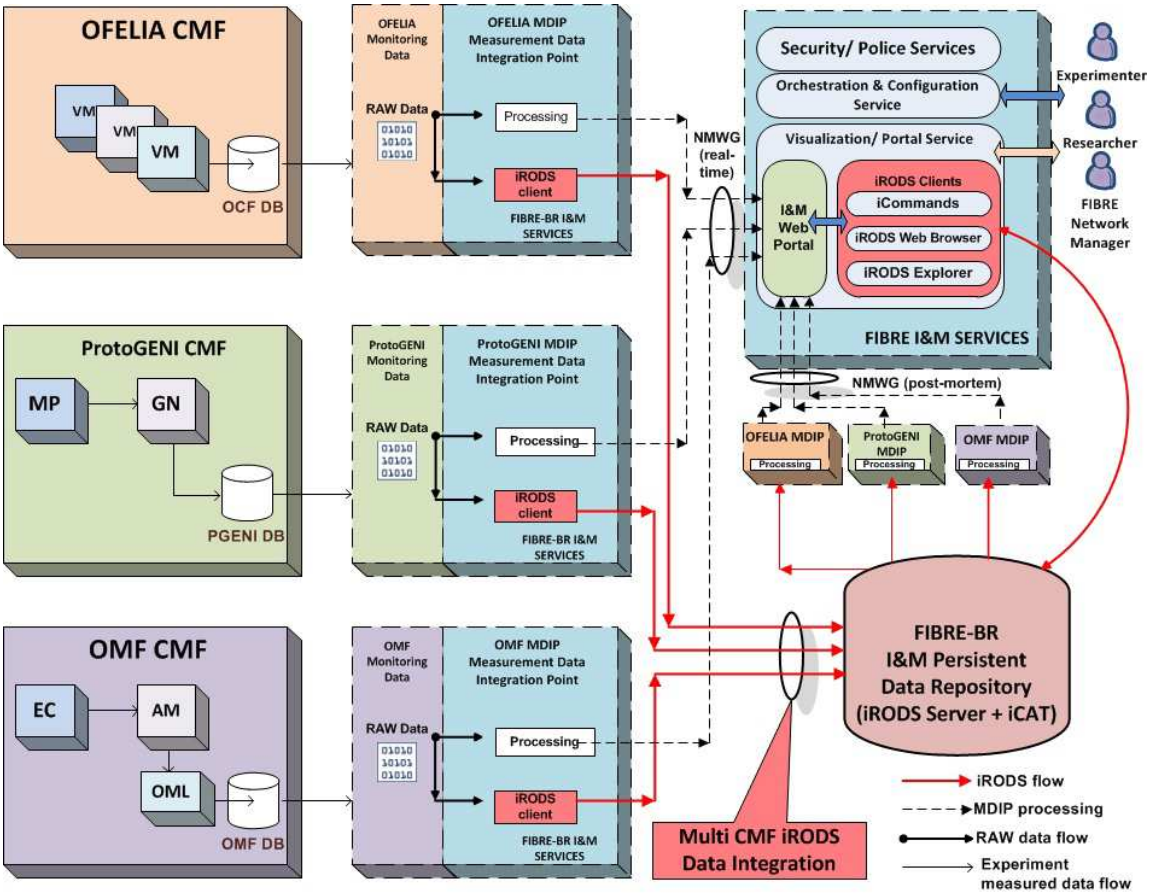


Fig. 3. iRODS Monitoring Data Storage in FIBRE I&M Architecture

Next section presents an effective iRODS deployment supporting the FIBRE testbed requirements, characteristics and operation. The discussion includes the monitoring data processing, testbed scalability, seamless data storage/ retrieval operation and federation across multiple administrative domains.

VI. THE IRODS-BASED FIBRE DATA REPOSITORY

The iRODS-based FIBRE data repository is a distributed solution with two sets of interactive and complementary operating components:

- The "iRODS island" facility; and
- The "iRODS testbed" facility.

The "iRODS island" facility basically capture monitored data in slices per island and per CMF and, subsequently, interact with the iRODS Server/iCAT and iRODS FIBRE testbed facility components in order to provide a distributed seamless repository for FIBRE.

The iRODS island components are illustrated in Figure 3:

- The iRODS "Slice Client" runs on the experimenter slice at the MDIP (Measurement Data Integration Point) for each specific native FIBRE´s CMF (OFELIA, OMF or ProtoGENI); and
- The iRODS "Portal Client" infrastructure is composed by iCommands, iRODS Web Browser and iRODS Explorer facilities and runs at the FIBRE testbed Portal.

The iRODS "Slice Client" is located at the experimenter´s slice (MDIP) and does user authentication, metadata definition and/or data transfers on behalf of the experimenter. In order to facilitate and have a more transparent user authentication, configuration and operation, the script "StorageData" runs at MDIP and uses iRODS iCommands ("iput", "imeta", …) to configure iRODS environment variables according with user authentication parameters (Figure 4). Options available with the script include user authentication, iRODS environment variables configuration, experiment data and metadata storage and retrieval.

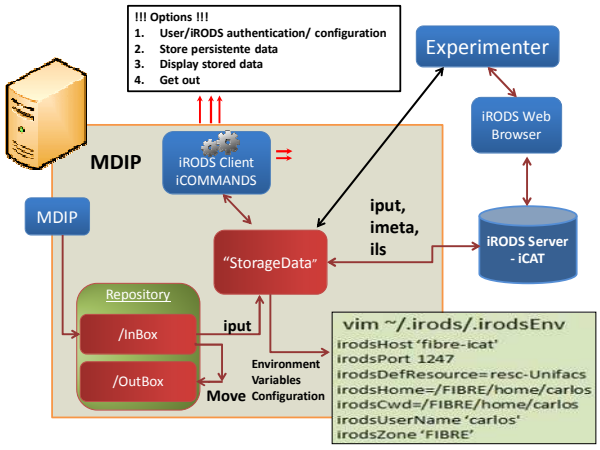


Fig. 4 – iRODS Slice Client

The iRODS "Portal Client" infrastructure allows basically monitoring data access.

The "iRODs testbed" facility effectively stores and retrieves monitoring data for FIBRE islands on a per-CMF basis. It is composed by the iRODS Server and iCAT Catalog Server components. The iRODS Server and metadata catalog (iCAT) components storage data and allows metadata management.

Considering the iRODS testbed components described, at least three iRODS-based architectural alternatives are possible using iRODS Server and iCAT Catalog Server for FIBRE:

- A centralized data repository;
- An iRODS-federated data repository; and
- A distributed data repository with seamless data access.

The centralized data repository approach (Figure 5) does not adequately comply with FIBRE testbed requirements. In this alternative it is more difficult to scale, it requires the utilization of the FIBRE testbed backbone for intensive data monitoring storage and retrieval on a centralized server and, finally, it has a single point of failure. In this structure all islands belong to a single iRODS domain or administrative zone allowing as such seamless access to monitored data.

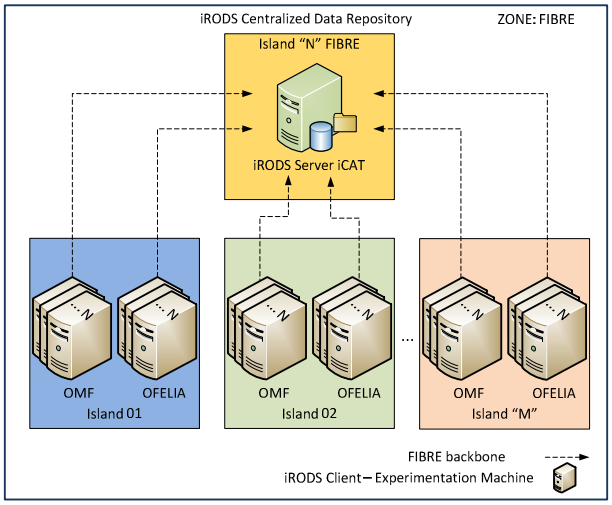


Fig. 5. iRODS centralized data repository

A second alternative for the iRODS-based FIBRE testbed facility is to adopt an iRODS-federated structure in which each island is fully independent and typically associated with its host institution (Figure 6). In this approach, each FIBRE island has an iRODS Server and an iCAT Catalog Server, corresponding to an independent domain or administrative zone per island. This alternative was not adopted by the FIBRE testbed because it is not adequately transparent for data storage/ retrieval by users/ experimenters.

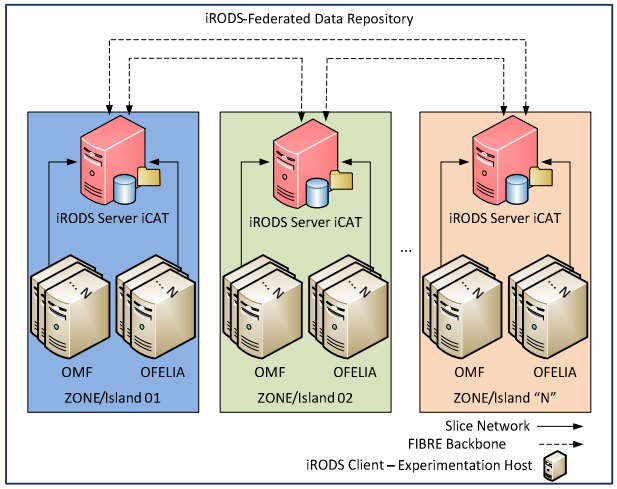


Fig. 6. iRODS-Federated data repository

The third approach is the one adopted by FIBRE testbed. It is a distributed data repository having an iRODS Server per island with a common iCAT Server supporting the FIBRE testbed (Figure 7). The iCAT Catalog Server supports access control and data/ metadata processing and management of persistent monitoring data storage for all FIBRE islands. This alternative also supports "iRODS federation" which is based on iRODS "zones". A "zone" is defined as an iRODS system formed by iRODS Servers, iCAT Servers and iRODS clients. Distinct iRODS zones can be used in FIBRE in distinct islands and may interoperate in terms of data storage exchange. For that, iCAT Server uses a trust relation among islands in order to provide communication and remote access.

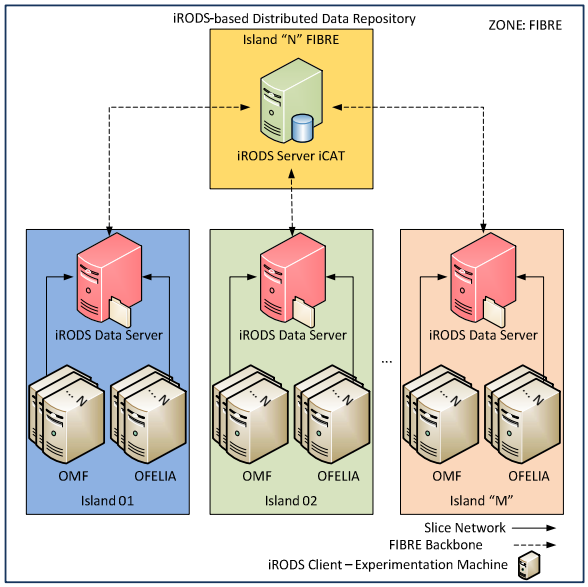


Fig. 7. Distributed data repositorywith seamless access

In terms of the FIBRE testbed requirements, this approach is a fully distributed solution, supports seamless access to monitoring data and scales adequately with new islands (new administrative domains) being incorporated to the FIBRE testbed.

VII. FIBRE iRODS-BASED DATA REPOSITORY – PROTOTYPYNG THE CONCEPT

A prototype implementation based on virtual machines was realized in order to validate FIBRE basic requirements compliance by the iRODS-based solution and demonstrate its operation. Aspects demonstrated were the seamless storage/ retrieval operation, metadata creation and federation support.

The prototype implementation is illustrated in Figure 8 and is composed by:

- 03 islands with 07 hosts (experimenters using OCF and OMF and servers);
- OFELIA (OCF) and OMF operating on each island; and
- iRODS Server, iCAT Server (metadata and access control) and MDIP iRODS Clients (iCommands).

The iCAT Server manages user´s data, metadata and iRODS service access control. It identifies users and their associated resources pointing to experiment data iRODS storage server. The iRODS Server is responsible by experiment data storage for each island. The MDIP iRODS Client (iCommands) is responsible by the experimenter identification and monitoring data dispatch to iRODS Servers.

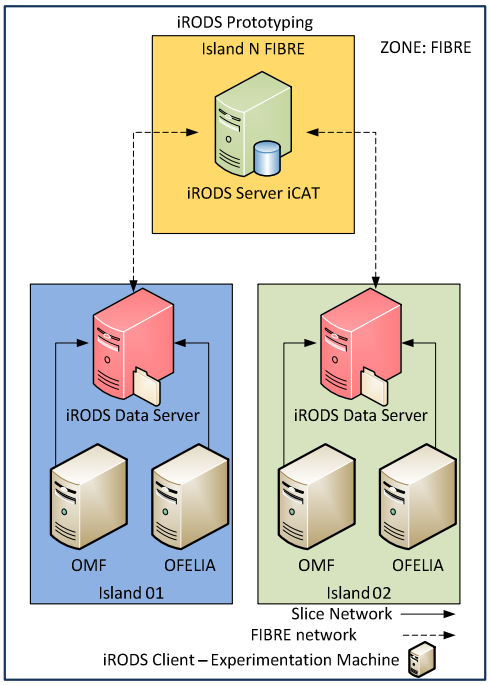


Fig. 8. FIBRE iRODS-based repository prototyping

Table 1 presents the prototype configuration scenario created in order to demonstrate iRODS repository operation. Five users are simulated and they collect monitored data with distinct sizes from OFELIA and OMF MDIPs in distinct islands.

ISLAND	CMF	USERS				
		Thiago	Marcelo	Adriano	Igorluiz	Igorleonardo
UNIFACS (unifacs-Resc)	OFELIA	host_ifacebw.rrd	Flowtp.rrd		host_ifacebw.rrd	
	Metadata		attribute=metrics value=throughput		attribute=metrics value=bandwidth consumption	
	OMF		OML001.sql	OML002.sql		OML004.sql
	Metadata		attribute=metrics value=delay			attribute=metrics value=throughput
UFPE (ufpe-Resc)	OFELIA		host_ifacebw.rrd	host_ifacebw.rrd	Flowtp.rrd	
	Metadata				attribute=metrics value=throughput	
	OMF	OML001.sql	OML002.sql			OML003.sql
	Metadata					attribute=metrics value=delay

Table 1. iRODS components and configuration

- Each user uses "StorageData" to collect monitoring data, insert metadata (island, CMF identification, other) and store it;
- Experiment monitored data collected is simulated with distinct file sizes

The monitored data is stored in iRODS servers at distinct islands and can be seamless retrieved through iRODS Web Browser (Figure 9) as required by FIBRE testbed. Any logical name used to access data will map to physical locations on different islands used by the experiment.

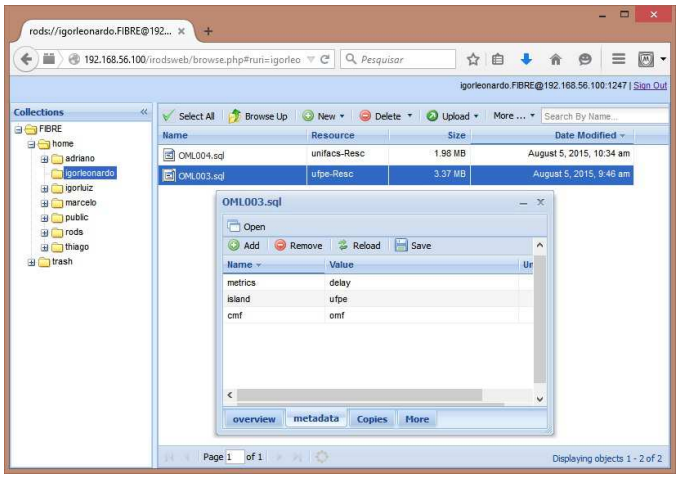


Fig. 9. Experiment seamless data access in different islands

A federated island structure was also prototyped in order to demonstrate the FIBRE iRODS-based repository ability to include new islands belonging to different administrative domains. A new island in a different zone (IFBA) was created with a basic experimental setup: iRODS iCAT Server, iRODS Server and 02 machines running OFELIA and OMF CMFs (Figure 10).

Federation is achieved with iRODS by declaring a new zone (FIBRE zone command: admin mkzone IFBA remote ifba-icat:1247) from each federated zone (IFBA remote to FIBRE and FIBRE remote to IFBA). Federation allows local users to become remote users with authentication. Authentication in FIBRE is always realized by the local zone, assuming as such a thrust relationship among zones that is previously negotiated by zone administrators.

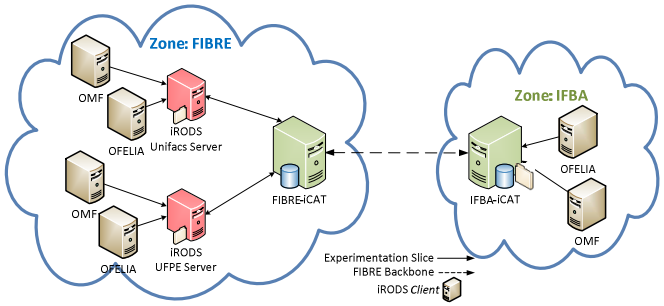


Fig. 10. FIBRE zone federated with IFBA zone

VIII. Final Considerations

An iRODS-based distributed and federated monitoring data repository for FIBRE multi-CMF network for experimentation was developed and described in relation to various architectural and deployment aspects. The overall strategy allows FIBRE testbed monitoring data integration leading to a seamless data storage while maintaining the native´s CMFs supported databases and repositories for storage (OFELIA, OML and ProtoGENI).

Monitoring data storage and retrieval also make use of iRODS federation facility allowing federation on multiple FIBRE islands through a simple naming scheme and management defined by administrative zones based on iRODS iCAT operation.

The FIBRE iRODS-based architecture adopted has an independent storage by islands and an overall domain controlled by iCAT Server. This approach supports adequately the scalability of the testbed which is necessary for an experimental testbed supporting constant and multiple islands incorporation and multiple configured experiments by different users (experimenters) on a single and/or distinct administrative domains.

The prototype implemented has demonstrated seamless and federation basic features and capabilities of the FIBRE testbed using a virtualized test infrastructure. Actual and future work include the incorporation of iRODS servers on the FIBRE backbone and evaluation of the iRODS-based architecture performance.

References

[1] Jianli Pan, Subharthi Paul and Raj Jain. “A Survey of the Research on Future Internet Architectures,” IEEE Communications Magazine, V. 49, n 07, pp 26-35, 2011.

[2] Marcondes, Cesar; Martins, Joberto Sérgio Barbosa ; Monteiro, Jose Augusto Suruagy ; Cardoso, Kleber ; Antônio Abelém ; Vagner Nascimento ; Machado, I. ; Tereza Cristina ; Charles Miers ; Marcos Salvador ; Christian Rothenberg . Estado da Arte de Sistemas de Controle e Monitoramento de Infraestruturas para Experimentação de Redes de Comunicação. In: Antônio Jorge Gomes Abelém; Dorgival Olavo Guedes Neto;Jussara Marques de Almeida. (Org.). Minicursos Livro Texto do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 1ed.Porto Alegre: Sociedade Brasileira de Computação - SBC, v. 1, p. 99-159, 2012.

[3] FIBRE. *Future Internet Testbeds Experimentation Between Brasil and Europe* [Online]. Available: http://www.fibre-ict.eu. 2013

[4] M. M. Pinheiro, I. L. E. Macêdo, I. L. O. Souza, T .S. Hohlenweger, P. R. R. Leite, A. L. Spínola, H. Monteiro, R. A. Dourado, L. N. Sampaio, J. A. S, Monteiro, and J. S. B. Martins, “An Instrumentation and Measurement Architecture Supporting Multiple Control Monitoring Frameworks,” in III Workshop on Experimental Research on the Future Internet (WPEIF), Ouro Preto, Brazil, 2012.

[5] OFELIA Control and Monitoring Framework (CMF) [Online], Available: http://www.fibre-ict.eu/index.php/cmf/ofelia.

[6] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar. “OMF: A Control and Management Framework for Networking Testbeds". In: ACM SIGOPS Operating Systems Review 43.4. Ed. by M. E. Fiuczynski and J. Matthews, pp. 54-59, 2010.

[7] O. Mehani, G. Jourjon, T. Rakotoarivelo, and M. Ott. “An Instrumentation Framework for the Critical Task of Measurement Collection in the Future Internet". In: Computer Networks. Ed. by J. P. G. Sterbenzet al. 2014.

[8] ProtoGENI [Online]. Available: http://protogeni.net/.

[9] Zurawski, Jason, Martin Swany, and Dan Gunter. "A scalable framework for representation and exchange of network measurements." Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on. IEEE, 2006.

[10] iRODS. *iRods: Data Grids, Digital Libraries, Persistent Archives and Real Time Data Systems* [Online]. Available: https://www.irods.org/. 2013

[11] Arcot Rajasekar, Michael Wan, Reagan Moore, Wayne Schroeder, Sheau-Yen Chen, Lucas Gilbert, Chien-Yi Hou, Christopher A. Lee, Richard Marciano, Paul Tooby, Antoine de Torcy, and Bing Zhu. *iRODS Primer: Integrated Rule-Oriented Data System*. Morgan & Claypool Publishers. 2010.

Monitoring-based Validation of Functional and Performance Aspects of a Greedy Ant Colony Optimization Protocol

Raul FUENTES*, Ana CAVALLI*, Wissam MALLOULI†, and Javier BALIOSIAN‡
*TELECOM Sudparis, Evry, France. E-mail: {fuentess, ana.cavalli}@telecom-sudparis.eu
†Montimage, Paris, France. E-mail: wissam.mallouli@montimage.com
‡ University of the Republic, Montevideo, Uruguay. E-mail: baliosian@fing.edu.uy

***Abstract*—Delay Tolerant Networks (DTN) are well adapted for situations where the network nodes suffer from intermittent communications due to the high mobility of the nodes and the constantly changing environment. Several research works tried to address this problem and lately, an ants-based protocol named GrAnt, has been proposed as one of the best solutions. In this paper we firstly assess GrAnt performance in much more challenging conditions than those presented by its authors, and secondly, we present a generic methodology based on MMT, an online security monitoring tool that enables real-time analysis of network traffic, to correlate the performance of a DTN protocol (such as GrAnt) with the information stored in its messages in order to validate the reliability of the protocol.**

I. Introduction

Delay Tolerant Network (DTN) is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. DTN was proposed by the NASA in 2007 to model satellite communication [1]. Their initial studies concluded that 1) DTN decreases the labor costs of the operations control center OPEX; and 2) DTN decreases the infrastructure costs CAPEX [2].

In this paper, we present the validation of the GrAnt protocol concerning the functional and performance aspects. The validation has been performed using two tools, the simulator Opportunistic Network Environment (ONE) and a monitoring tool, MMT [10], developed by the Montimage society. Different experiments have been performed on a case study and the results show the advantages and limitations of the GrAnt protocol. In order to illustrate the application of the GrAnt protocol, we consider an underground mining scenario which shows all the previous restrictions. It is assumed that all the workers in a mine carry devices supporting DTN for the transmission of data between deep places in the tunnels and a main server in a central point at the surface. We use the MMT Tool for detecting the social interaction between nodes in the mine and to evaluate the performance of GrAnt based on the messages conditions when reaching the main server.

The paper is organised as follows. Section II presents a short description of the GrAnt protocol. Section III introduces the ONE simulator and the results on performance analysis and Section IV presents the application of monitoring techniques to analyse the behavior of the protocol. Finally, section V gives the conclusions of this work.

978-1-4673-9408-6/15/$31.00 © 2015 IEEE

II. The GrAnt protocol

The original Ant Colony Optimization (ACO) algorithm, known as the Ant System [6], was first proposed in the early nineties [7], [8] inspired in the manner in which ants mark and choose their paths. In ACO, a number of agents simulating ants build solutions to a given optimization problem and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants [8].

The ACO algorithm has been often considered as a good choice for routing Mobile Ad-hoc Networks (MANET), however, the ACO's random nature makes it slow in front of sudden network changes [6].

Figure 1 presents an execution of GrAnt in a small network. Node *s* sends an FA *k* with destination *d* together with a data message *m* (Figure 1(a)). The path to *d* is computed based on some knowledge acquired by this FA, which decides where to be forwarded at each node and tries to infer the capability of good next forwarders towards *d*. While being forwarded, each FA *k* collects the quality of every node x (Q_x) along the path to *d* (Figure 1(b)). After reaching its destination, a Backward Ant (BA) is sent in reverse through the path recorded in the FA. The BA stores the followed path's total quality (Q^k_{path}) and deposits a pheromone which is proportional to (Q^k_{path}) at each link of the reverse path from *d* to *s* (Figure 1(c)). Shall new messages be forwarded to *d*, the already deposited pheromone is reinforced, guiding the forwarding of future FAs to *d* (Figure 1(d)). To direct DTN traffic to the most promising contacts, GrAnt uses information about opportunistic social connectivity between nodes. All the details on how this is performed can be read in [6].

III. GrAnt Analysis using ONE Simulator

In the following GrAnt behaviour analysis we try to asses if GrAnt's implementation satisfies its functional requirements and if it performs correctly with a given application design and configuration.

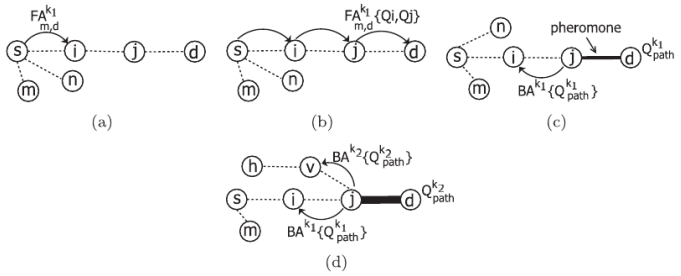


Fig. 1: Overview of the GrAnt protocol execution (Source [6]).

For the experiments we use the same simulation platform used by GrAnt creators, the Opportunistic Network Environment (ONE) simulator [9]. This simulator is adapted to study DTNs and has the ability of using different realistic mobility models such as Point of interest movement mode (POI) and Working day Movement Model (WD) [9]. We take as a starting point the best settings for GrAnt that are proposed by its authors.

All tests were performed with the same movement patterns, execution time (800.000 seconds), number of messages (11 458), the same number of nodes with the same technology and all moving in a model of the centre of Helsinki. These tests were divided into 3 categories: 1) Stress Scenario; 2) Impact of TTL; and, 3) saturation buffer.

Experiments on Buffer Saturation with Fixed Sizes

While dropping messages due to TTL expiration is a natural element of the algorithm, messages removed due to buffer overflow is not. If this happens too often there is a risk of losing key communication messages in an application. In the worst case, a message could be an FA removed that has not yet expired, has not been duplicated and, thus, its contents will be completely lost.

In this first series of tests, we vary message size (as a percentage of buffer size) keeping buffer size and TTL constant:

- Exp. 12 - TTL: 400 min, BS: 4MB, MS: 5% (200KB)
- Exp. 17 - TTL: 400 min, BS: 4MB, MS: 12% (480KB)
- Exp. 2 - TTL: 400 min, BS: 4MB, MS: 37.5% (1.5 MB)
- Exp. 4 - TTL: 400 min, BS: 4MB, MS: 57.5% (2.3MB)
- Exp. 10 - TTL: 400 min, BS: 4MB, MS: 75% (3 MB)
- Exp. 11 - TTL: 400 min, BS: 4MB, MS: 95% (3.8 MB)

In these tests, we set the message size as a percentage of the buffer. Table I shows the results obtained from the simulations.

If the message size is too small, the number of duplicated FA increases with a small impact on the number of dropped or aborted messages. Remember that duplicate FA ants carry the same messages but follow different paths. The reason that the number of aborted messages has so small impact on the throughput is that when the message is extremely small, any chance of passing it will be sufficient. The contact time may last only few seconds and in these experiments we can effectively see that when the message size increases, the proportion of aborted messages also increases.

TABLE I: Fixed-size messages.

Parameters	EXP 1	EXP 2	EXP 4	EXP 10	EXP 11	EXP 12
Started	57108	31928	22801	18621	16516	78324
Relayed	56975	31293	18037	8232	4672	78275
Aborted	133	635	4764	10388	11844	48
Dropped	29427	34639	27094	18632	15577	4212
Removed	19005	3822	1091	484	252	43265
Delivered	7105	2426	852	426	222	9515
Hopcounts FA	12450	3302	1073	459	235	20949
Hopcounts BA	6422	2055	617	259	108	8461
Drop ratio rate	0.43	0.8103	0.9186	0.9463	0.9657	0.0469
Delivered ratio	0.62	0.21	0.07	0.04	0.02	0.83

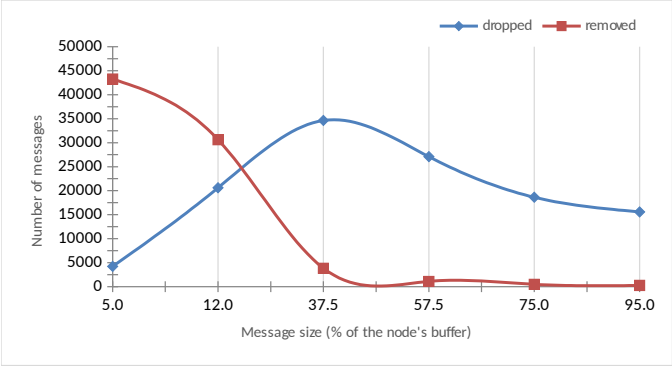


Fig. 2: Comparison between dropped and removed messages (EXP 12).

Regarding removed and dropped messages something interesting occurs. Experiment 12 has the highest value of messages removed and a small number of messages dropped. Regarding messages removed, we can observe that the number decreases when the size of the message is increased. This is not the case for dropped messages which seem to follow this trend but suddenly increases and then decreases. We can better observe the behavior of both fields in Figure 2.

The behavior regarding the number of removed messages as the message size grows is easy to understand. The bigger the messages to be transmitted, the number of messages that fits in the node decreases; therefore, although each new message could force the removal of old messages, the total number of messages generated is also much lower. This is shown by the almost 78% decrease of messages initiated in Experiment 11 (3.8 MB) compared to Experiment 12 (200 KB).

So why did the Experiment 12 with the smaller message

sizes have to remove so many ants from its buffers? Note that the total number of generated ants is highest for the messages and virtually all were transmitted. The high number of small messages saturated the nodes' buffers before the TTL of the ants began to expire. That is why in experiment 12 (5%), compared to Experiment 1 (12%) and finally to Experiment 2 (37.5%) the behavior allows the following observation: the larger the messages, the lower number of messages, reducing the number of messages aborted and increasing the number dropped.

However, the peak between experiments 2 (37.5%) and 4 (57.5%) is due precisely to the completely different behavior in the first three experiments. The nodes are not capable of storing more than one message at a time, so that the number of messages removed drastically decreases. But in turn, messages dropped increase at a completely different rate than the first three scenarios.

We could say that applications that need to handle only small messages with respect to buffer sizes give a good result as the percentage of successfully delivered messages increases. However, this is not always possible. Nodes that have huge buffers would be fine but this implies a very high cost. If nodes have a buffer of moderate size (as in the simulations) we would then need to limit the size of messages, which may limit the ability of applications. As we can see, DTN environments require making consensus between the size of the application messages and the size of the buffers of the nodes.

Our conclusion is that GrAnt has to work with a message size of less than a half of the buffer size in order to transmit data at acceptable rates.

IV. Network Traffic Analysis using MMT

A. GrAnt Trace Analysis

Telecom SudParis has designed and implemented a plugin for communication analysis of applications using the GrAnt algorithm for routing messages. This plugin allows extracting and analyzing the following components in the messages:

- **Type** - Identifies type of ant, possible values: FA or BA.
- **ID** - Message ID.
- **I-N** - Identifies intermediary I-Node. When a message is created by first time I-N will be the same as S-N.
- **J-N** - Identifies intermediary J-Node. When a message is created by first time I-N will be the same as D-N.
- **Size** - Size of the application's message being passed (in Bytes).
- **HOPS** - Total hops the ant has performed.
- **TTL** - (remaining) TTL value.
- **Timestamp** - Time stamp with the message's time of creation.
- **S-N** - Source node.
- **D-N** - Destination Node.
- **Path** - A list with the intermediary nodes the FA or BA has already visited.
- **Priority** - Priority of the message, possible values are: Expedite (BA) and Normal (FA)
- **(q|Q)uality** .- Sum of path's quality (FA) or Total Quality Inverse Path (BA)

MMT allows updating the plugin to suit different transport technologies, in other words, on a real GrAnt implementation, it would be easy to update the plugin to recognize new communication mediums, e.g., Bluetooth or 802.11n protocols. These changes do not affect the elements designed and MMT will continue offering the same fields.

Regarding the fields we use in GrAnt, we can observe quite simply the paradigms differences between IP and DTN. In protocols such as GrAnt, elements for making routing decisions are appended to the message, whereas in IP, the message format is agnostic of the routing protocol. To avoid confusion, packets passed from one node to another will be called ants, where a Forward Ant (FA) carry the message generated by an application, and a Backward Ant (BA) serves as a message reception acknowledgment.

The ONE traces are processed by the MMT plugin and then analysed following two approaches: *security rules* to be validated and *attack behaviors* to be detected. We put special emphasis on the former security rules, which are useful for checking the ants' fields and thus, it is possible to detect anomalies in the topology.

Security rules define expected behavior of functionality and/or security concerns of the element that is being monitored, while the attack describes malicious behavior that can be an attack model, a vulnerability or a description of node misbehavior.

MMT rules are defined in an XML format using properties composed of a context and a trigger. The context describes the state of the system at a given time, and the trigger is the event or set of events that constitute the attack or misbehavior that occurs when the context holds. Depending on whether the property is a security rule or an attack we interpret it as described bellow:

- If the context is true, and the trigger is detected, then the security rule has been satisfied and everything is ok. Here we state that the security rule has been respected.
- If the context is true, but the trigger is not detected for a certain time period then the rule has been violated. Here we have a misbehavior, a vulnerability or an attack since the rule was not followed as required.

As mentioned in the Section I, an underground mine is an ideal environment for a DTN. Given the strong attenuation that mine walls impose to radio signals, the workers movements cause intermittent communications between their devices that follows closely the workers' scheduling. Almost all the workers, or even mobiles objects such as carts, will part from the central point to different parts of the mine, and some of them, will be moving from tunnel to tunnel or returning periodically to the central point. Using MMT security rules it is possible to characterize the behaviour of GrAnt on a given topology and pattern of movements, detecting weaknesses in the communications potential of the workers' scheduling by inspecting the ants' fields.

B. Rules to Trace

In general, the information flow ends at the exits of mines, such as information captured by air quality sensors, conditions of machines and parts, etc. This information is transported by carriers that could be mine workers or other moving elements in the mine and even could be the sensors themselves, that are able to deliver messages to a central point for decision-making. For any real world application using DTN protocols, the nodes flow will be the key for how messages are forwarded. Usually, we have nodes collecting information to be sent to headquarters or central points. If there is no nodes moving between those sensors and central points the messages will never be delivered on time. If nodes do not have a proper movement schedule messages can be lost.

Who makes decisions, what they will be and where they will be sent depends on the application. Nevertheless, what is needed is general purpose analysis means for DTNs, in particular, tools to confirm that the nodes are moving in with the expected pattern and, if not, identify the anomaly.

We define three MMT rules with the objective of detecting the anomalies, the first rule aims to detect when the distance between the central point and the nodes is too far, the second is used for analyzing the social metrics of the nodes and the last is focused in detecting abnormal movements.

1) Critical distance: It is necessary to assure that the movement of all nodes successfully allowed the message to be transmitted from the tunnels to the central point. Otherwise, there is a risk that the ants will expire before reaching their destination.

To do so, we designed a security rule, named *critical distance*, that detects when a BA reaches destination with a TTL value below a certain threshold. If this occurs, we can assume that other messages are being lost.

2) Confirm scheduling planning: For any application to have success transmitting their messages from source to destiny on DTN environments, one or more nodes need to have a very good social parameters. For real world environment where those nodes are people with a clearly scheduling this should be enough to guarantee a high social degree. However, this is difficult to confirm.

MMT is used to follow the meetings between nodes to confirm that their scheduling is truly helping the communication flow.

Defining properties composed of contexts and triggers are not sufficient to obtain this information; however, MMT allows more than this, since when these rules are detected, we can perform additional actions that could be, for instance, generating a report for each node indicating the number of encounters with other nodes.

For this property, MMT generates one additional report on the social metrics (i.e., Centrality and Betweenness) for each node.

3) Tracking unexpected movements: While the Schedule planning rule is intended for determining the centrality of a particular node which is supported by its position within the mine, here what we want is to detect unusual patterns.

Such patterns could be due to nodes that normally should not possess a centrality greater than a certain threshold, but do. In this case, the rule is executed for all nodes and their centrality is verified, at the same time as the previous rule, but only the nodes not in the group defined by the rule schedule planning are listed.

V. Conclusions

In this paper we have provided performance-based and functional analysis of the GrAnt routing protocol for DTNs. The analysis has been performed using two tools, the ONE network simulator and the monitoring tool MMT. In order to illustrate the application of the GrAnt protocol, we consider an underground mines scenario that presents all the characteristics of DTNs. Using MMT we have performed several experiments to analyse the behaviour of GrAnt protocol on a given topology and pattern of movements, detecting some weaknesses in the communications potential of the mine workers' scheduling. The experiments results show limitations that should be considered during the development of real DTN applications. We have also developed a general methodology based on MMT, to correlate the performance of a DTN protocol (such as GrAnt) with the information stored in its messages. For future work we are planning the development of attacks rules based on MMT, in order to test the robustness of the GrAnt protocol.

References

[1] NASA, "NASA Successfully Tests First Deep Space Internet," NASA, 18 November 2008. [On line . Available: http://www.nasa.gov/home/hqnews/2008/nov/HQ_08-298_Deep_space_internet.html.

[2] NASA, "Disruption Tolerant Networking for Space Operations (DTN)," 03 March 2013. [On line]. Available: http://www.nasa.gov/mission_pages/station/research/experiments/730.html.

[3] Google/Jet Propulsion Laboratory; NASA/Jet Propulsion Laboratory; The MITRE Corporation; Intel Corporation; SPARTA, Inc., "RFC 4838: Delay-Tolerant Networking Architecture," IETF Trust, 2007.

[4] M. Dorigo y G. d. Caro, Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization, McGraw-Hill, 1999.

[5] M. Dorigo, V. Maniezzo y A. Colorni, "The ant system: Optimization by," *IEEE Transactions on Systems, Man,,* nÂº 26, pp. 29-41, 1996.

[6] K. V. Ana Cristina, M. Anelise, R. D. Myriam y C. V. Aline, "GrAnt: Inferring best forwarders from complex networks' dynamics," nÂº 57, pp. 997-1015, 2012.

[7] M. Dorigo, M. Birattari y T. Stützle, "Ant Colony Optimization, ARtificial Ants as a Computational Intelligence Technique," *IEEE Computational Intelligence Magazine,* November 2006.

[8] M. Dorigo, V. Maniezzo y A. Colomi, "Positive feedback as a search strategy," *Dipartimento di Elettronica, ,* pp. 91-016, 1991.

[9] A. Keränen, J. Ott y T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," ICST, Rome, 2009.

[10] W. Mallouli, B. Wehbi, E. Montes de Oca y M. BourdellÃ¨s, "Online Network Traffic Security Inspection," de *9th workshop on system testing and validation (STV)*, Paris, France, 2012.

Approach to Power Prediction in WSN Using Propagation Models: Practical Analysis Applied in Water Reservoirs

Teles de Sales Bezerra,
José Anderson Rodrigues de Souza,
Saulo Aislan da Silva Eleutério

Federal Institute of Education, Science and Technology
of Paraíba - IFPB, Campina Grande - Brazil
Email: teles, andersonrodrigues@ieee.org
saulo_eleuterio@ieee.org

Jerônimo Silva Rocha

Federal Institute of Education, Science and Technology
of Paraíba - IFPB, Campina Grande - Brazil
Institute of Advanced Studies on Communications - IECOM
Email: jeronimo@iecom.org.br

***Abstract*—Wireless Sensor Networks (WSN) are already trend in many applications, environmental modeling in this network is of fundamental importance to maintain communication between modules always in acceptable power transmission and reception levels. Faced with this challenge the aim of this work is to conduct a study on the propagation of radio frequency signals in WSN applied in an environment used as water reservoir, this analysis will be done by comparing the real values Received Signal Strength Indicator (RSSI) with the estimates proposed by some of the most used power prediction models for WSN, thus achieving the best modeling of the proposed environment and thus determine which power prediction model more suitable for the application.**

I. Introduction

History shows that progress and technological development are inevitable, actually becoming a need in the world of nowadays. Telecommunications are currently essential in world economy operations of any modern society, being, wireless network systems are one of the greatest developments in the area of communications. An example of such a network, is a WSN (Wireless Sensor Network), where spatially distributed autonomous devices, equipped with sensors, are used in environment monitoring, traffic control, healthcare, home automation, and others applications [1].

Wireless Sensor Networks (WSN) gain increasing attention by researchers as well as industry and governments, they provide the ability to monitor large areas for events efficiently and with small effort. WSN is an infrastructure comprised of sensing (measuring), computing, and communication elements that gives an administrator the ability to instrument, observe, and react to events and phenomena in a specified environment. WSN has been developing rapidly in recent years and been applied to many fields including military, business, and agriculture [2].

The present article is aimed at analyzing the problems encountered in many Brazilian regions where the lack of water is constant and the need for constant monitoring of water reservoirs is important. However to implement this network in any environment modeling the same becomes important to predict the possible challenges that will be encountered, modeling environments will be made from the metric Received Signal Strength Indicator (RSSI), and from values collected in application environment can trace the best model that fits the situation proposed. This paper is organized as follows: in Section II shows the concepts of metric modeling to be used and the importance of prior modeling in wireless communications for the correct prediction of the power, in Section III shows the propagation models used in this work. In Section IV shows related work that served as the theoretical background, in Section V shows the methodology in the measurements, in Section VI shows the results on the accuracy of propagation models compared to real values collected and finally in Section VII shows conclusions and future works.

978-1-4673-9408-6/15/$31.00

II. Received Signal Strength Indicator

RSSI (Received Signal Strength Indicator) is a measure from the received radio signal's power. It is a metric used to estimate the transmission quality between two nodes as the distance between them is varied, implemented under the IEEE 802.11 standard [3]. It works by using the distance between transmitter and receiver to designate the quality from the received signal, even considering variations on signal strength by comparing the received signal level with probability distributions and localization measures based on statistic analysis [4].

There are various electromagnetic wave is fading processes, for example fading caused by signal reflections on objects, and they all quite affect the transmission between nodes. The waves travel through different ways, that not necessarily have the same length, and interactions between them and the objects and barriers during their travel is responsible for great part of the fading phenomenon on transmission and reception processes. Fading during electromagnetic wave is travel is also caused by reflection, diffraction and scattering [5].

To ensure an acceptable level of quality of service for users in wireless data network, network designers rely on signal propagation path loss models. Radio wave propagation models are a series of mathematical calculation developed to predict path characteristics and losses in a given environment [6]. For example, propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, plus the variability in the signal intensity near a particular location area. Thus, propagation models are mathematical tools used by engineers and scientists to plan and optimize wireless network systems [7].

Coverage problems are one of the most active research topics related to WSN. It is generally necessary to deploy multiple sensors to cover an entire WSN area so as to provide services within the area. Each sensor used in WSN has a limited sensing radius range [2].

III. RF Propagation and Propagation Models

When propagation is considered in an outdoor environment, one is primarily interested in three types of areas: urban, suburban and rural areas. The terrain profile of a particular area also needs to be taken into account. The terrain profile may vary from a simple curved Earth to a highly mountainous region. The presence of trees, buildings, moving cars, and other obstacles must also be considered. The direct path, reflections from the ground and buildings, and diffraction from the comers and roofs of buildings are the main contributions to the total field generated at a receiver, due to radio-wave propagation.

Propagation models are fundamental tools for designing and deploying any wireless communication system including WSN in outdoor environments. The models are closely related to the system working environment and characteristics. In general, propagation models are methods and algorithms used to predict the signal strength level along with description of signal level variability. Their main purpose is to predict the distortion and attenuation of the RF signal that will reach the receiver [8].

Currently there are various mathematical models with the objective of predict the average strength of the wireless signal transmission between two network devices. These models are useful in estimating the radio area of coverage of a transmitter and are called propagation models, featuring the signal strength when there is separation between transmitter and receiver.

A. Free Space Model

This model determines the power at the receiver only in function of transmit power, the gain of the antennas and the distance between sender and receiver. The attenuation (path loss) for the Free Space model is defined as follows:

$$PL_{dB} = -10 * log\left(\frac{G_t * G_r * \lambda^2}{(4 * \Pi)^2 * d^2}\right)$$

B. Log-Distance Model

Defined in [6] [9], Log-Distance model considers that the average received power decreasing logarithmically with distance from the transmitter. The coefficient n has a value ranging from 2 to 6 and this model is characterized by the following equation:

$$PL_{dB} = PL(d_0) + 10 * n * log\left(\frac{d}{d_0}\right)$$

C. Shadowing Adapted Model

Adapted models are implemented from the classic models by the adjusting (adaptation) of their coefficients relation to field measurement by the minimum mean square error technique. The main advantage of this approach is that fact of the same "encapsulate" some model input parameters, thus avoiding problems related to bad dimensioning the same, which can lead to considerable errors of prediction. Defined in [10]:

$$PL_{dB} = -10 * \beta * log(d) + X[dB],\ X[dB] = 9$$

D. Tewari, Swarup e Roy Model

Model defined by [11], this model was developed based on measurements performed in the forest of India, which resembles in some factors with the Amazon rainforest [11].

$$PL_{dB} = 88 + 20 * log(f_{MHz}) + 40 * log(R_{Km})$$

$$-20 * log[H_t(m) * H_y(m)] + L_f(dB)$$

E. Weissberger Model

For empirical models, it was found that the model developed by Weissberger esteem the excess of attenuation produced by vegetation, which is a model of interest in provide for the existence of foliage, and to make prediction for small stretches [12]. The loss model is:

Case $d \leq 14m$:

$$PL_{dB} = 0.45 * f^{0.284} * d$$

Case $14m \leq d \leq 400m$:

$$PL_{dB} = 0.45 * f^{0.284} * d^{0.588}$$

F. ITU-R Model

$$PL_{dB} = 0.2 * f^{0.3} * d^{0.6}$$

G. COST 235 Model

$$PL_{dB} = 15.6 * f^{-0.009} * d^{0.26} - With\ leaf$$

$$PL_{dB} = 15.6 * f^{-0.2} * d^{0.5} - Whitout\ leaf$$

H. RIM Model

RIM (Radio Irregularity Model) is a model developed purposefully for wireless sensor networks. This model is defined by [13] and as the literature already mentions that the radio coverage is not a perfect circle in real environments [9], neither as little resembles a circle. The RIM model is based on this irregularity. To symbolize the irregularity of the radio coverage model, the parameter DOI (Degree of Irregularity), was introduced in the RIM model. The description of the DOI calculation irregularity is given by:

$$PR = PE - (PL * K_i) + F$$

where:

- F = Component of *fading*.
- K_i = Coefficient representing the difference in losses *path loss* in different directions, $K_i = 1, case(i = 0)$;
- i = Coefficient of i-nth degree.

i.e., the angle being the angle 0 is analyzed, in reference to line of sight.

IV. Related Works

Various papers have been published with the purpose of investigate the effects on the propagation of radio signals in the devices ZigBee. The authors in [14] have investigated the effects caused by external factors in the RF signals, specifically analyse the RF activity outdoors for 24 hours in order to investigate the influence of time on the RSSI measurements and therefore to estimate the difference between day and night measurements, due the effects of the communication are aleatory and moving human that possibly were present in the area. On the other hand, some works analysed the effects of internal factors about on RSSI measurements, as the effect of polarization antenna between the transmitter and the receiver [15], or the effect of the conception of hardware devices [16]. And other authors in [17] perform a RF propagation analysis using collected RSSI values indoors and the authors in [5] did measurements in outdoor environment and showed the performance of WSN ZigBee applied in grass environment.

During the studies were also consulted references that not only verified the performance of wireless systems, but it also performed the systems modelling procedures. The authors in [18] have carried out studies with the COST 235 models and ITU-R to model forest environments in Turkey, the authors in [9] performed measurements of performance in wireless systems in an open courtyard at a university in Lisbon - Portugal, the authors in [11] propounded a new propagation model for the characterization of forests in India, which have some features present in the Brazilian forests finally the authors in [19] performed measurements of RSSI values compared to propagation models in a pine forest in Portugal.

V. Measurements Methodology

The methodology for measurements and analysis data were taken 2 steps, namely:

- Step 1: The experiment was performed in an area of water reservoir located in Campina Grande city in Paraíba, Brazil. With fixation of the transmitter module at one point and performing the displacement of the receiver module along of water, the Fig. 1 shows the methodology of measurements.

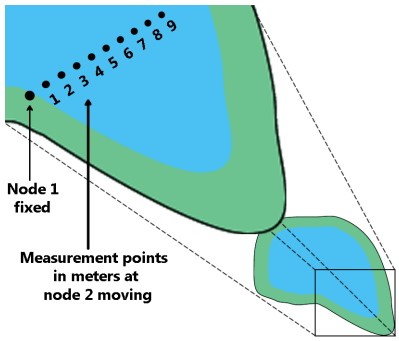


Fig. 1. Methodology of measurements.

- Step 2: In collecting the data for each point 100 samples were collected RSSI values. In order to evaluate the samples collected was initiated the step of processing the samples with 10 measuring points collected 100 samples and these samples the average RSSI, in total of 10000 samples of RSSI values.

VI. Accuracy of Propagation Models

The results obtained have undergone a process of statistical analysis for validating data, such reliability is based on these methods the following.

A. Mean square error (MSE)

In practical terms, the Mean Square Error (MSE) equals the sum of the variance and tendentiousness square estimator. An estimator is used to deduce the value of an unknown parameter in statistical model. Such estimate of MSE is expressed by (1).

$$MSE = \frac{\sum_{t=1}^{n}(A_t - P_t)^2}{n} \quad (1)$$

B. Mean absolute percentage error (MAPE)

The average absolute percentage error calculation estimates an exact value of error, and such error is expressed by percentage to estimate just how was necessary the actual value with the estimated such a relationship is expressed by (2).

$$MAPE = \frac{\sum_{t=1}^{n}\left|\frac{(A_t - P_t)}{A_t} * 100\right|}{n} \quad (2)$$

Where A_t is real value on t period and P_t is prevision for t period.

The presented methods were used for statistical analysis to measure the accuracy of the samples, and based on the results of these analyzes was the best way to power prediction model for the proposed environment. Table I shows the results and the Figure 2 shows the values of propagation models used in this work and the real RSSI measurements.

TABLE I. ACCURACY OF PROPAGATION MODELS: MSE AND MAPE

Model	MSE	MAPE
Log-Distance n =2	360.369216	24.893448
Log-Distance n=3	365.795296	14.920217
Log-Distance n=4	1198.015650	34.827049
Log-Distance Shadowing	225.470965	17.406552
Free Space (Friis)	**122.383602**	**8.826252**
RIM-DOI	372.203934	24.965006
Weissberger d<14	165.707058	15.187434
Weissberger d>14	547.903012	30.472475
Tewari, Swarup e Roy	1649.123357	44.975889
ITU-R	826.048589	37.439258
COST 235 With leaf	167.001281	15.101572
COST 235 Without leaf	722.448214	34.896682

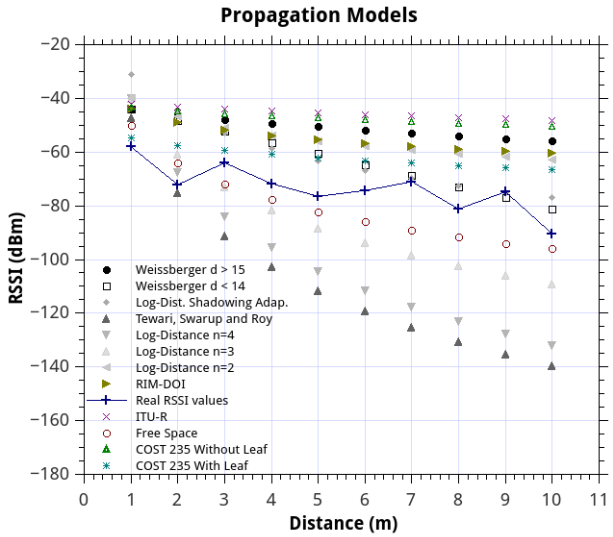


Fig. 2. Accuracy of Propagation Models and Real RSSI values.

VII. CONCLUSION AND FUTURE WORKS

Deployment of a WSN in water reservoirs requires the study of signal propagation in order to find the best way of positioning sensor nodes and the power prediction of RSSI values is very important to building this network. This work contributes to the deployment of WSNs in the region of the Campina Grande - Paraíba in order to optimize the use of resources such as water, where this region doesn't have good precipitation of water and controlling the use of resources is of paramount importance.

From the accuracy of propagation models to power prediction, we conclude which the Free Space Model is the best way to modelling the power prediction in this area, but other models also has good results, Free Space Model had a precision about 91% in comparison with the real RSSI values. For this work, we choose to study only the transmission range of the sensors in water reservoirs in Brazilian North-east as a future work we will study the impact of climate on these measurements, varying schedules of data collection and on different days with different climates and temperatures.

ACKNOWLEDGEMENT

The authors thanks IFPB Campina Grande and National Council for Scientific and Technological Development (CNPq).

REFERENCES

[1] G. Gonçalo and S. Helena, "A novel approach to indoor location systems using propagation models in wsns," *International Journal On Advances in Networks and Services*, vol. 2, no. 4, pp. 251–260, 2010.

[2] W. Kong, M. Li, L. Han, and A. Fukuda, "An smt-based accurate algorithm for the k-coverage problem in sensor network," in *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2014, pp. 240–245.

[3] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*. IEEE, 2008, pp. 303–306.

[4] C. Park, D. Park, J. Park, Y. Lee, and Y. An, "Localization algorithm design and implementation to utilization rssi and aoa of zigbee," in *Future Information Technology - FutureTech, 2010 5th International Conference on*. IEEE, 2010, pp. 1–4.

[5] T. Bezerra, S. Silva, E. Silva, M. Sousa, and M. Cavalcante, "Performance evaluation of zigbee transmissions on the grass environment," in *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2014, pp. 287–290.

[6] T. S. Rappaport, *Comunicações sem fio: Princípios e Práticas*. Pearson Prentice Hall, 2009.

[7] R. Timoteo, D. Cunha, and G. Cavalcanti, "A proposal for path loss prediction in urban environments using support vector regression," in *AICT 2014, The Tenth Advanced International Conference on Telecommunications*, 2014, pp. 119–124.

[8] T. Stoyanova, F. Kerasiotis, A. Prayati, and G. Papadopoulos, "A practical rf propagation model for wireless network sensors," in *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*. IEEE, 2009, pp. 194–199.

[9] R. M. P. Jacinto, "Modelação da propagação numa rede de sensores sem fios," 2012.

[10] A. Fanimokun and J. Frolik, "Effects of natural propagation environments on wireless sensor network coverage area," in *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on*. IEEE, 2003, pp. 16–20.

[11] R. Tewari, S. Swarup, and M. Roy, "Radio wave propagation through rain forests of india," *Antennas and Propagation, IEEE Transactions on*, vol. 38, no. 4, pp. 433–449, 1990.

[12] T. C. Braga, "Monitorização ambiental em espaços florestais com rede de sensores sem fios," 2010.

[13] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 221–262, 2006.

[14] E. Jafer, B. O'Flynn, C. O'Mathuna, and R. Spinar, "A study of the rf characteristics for wireless sensor deployment in building environment," in *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*. IEEE, 2009, pp. 206–211.

[15] M. Barralet, X. Huang, and D. Sharma, "Effects of antenna polarization on rssi based location identification," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 1. IEEE, 2009, pp. 260–265.

[16] J. Hightower, C. Vakili, G. Borriello, and R. Want, "Design and calibration of the spoton ad-hoc location sensing system," *unpublished, August*, 2001.

[17] R. M. Pellegrini, S. Persia, D. Volponi, and G. Marcone, "Rf propagation analysis for zigbee sensor network using rssi measurements," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*. IEEE, 2011, pp. 1–5.

[18] O. Kurnaz, M. Bitigan, and S. Helhel, "Procedure of near ground propagation model development for pine tree forest environment."

[19] J. A. R. Azevedo and F. E. Santos, "Propagação em ambientes florestais," *Floresta*, vol. 2, no. 35, pp. 4–37, 2008.

Analysis of the Integration of WiMAX and Cellular Networks

Matheus Queiroz, Felipe Atourassap, Suellen Reis, Rafael Sander, Werley Santos, Anna Tostes
and Fatima Duarte-Figueiredo
Department of Informatics
Pontifical Catholic University of Minas Gerais
Belo Horizonte, Minas Gerais
Email: [matheusmirandaqueiroz, faspm, suellenoliveira, sandernogueira,
werleypsantos, annatostes, fatimaduartefigueiredo]@gmail.com

***Abstract*—Next-generation IP-based wireless network environments (NGN) will allow users to freely switch between different network technologies while preserving connectivity. Therefore, integrating heterogeneous networks is fundamental to the development of NGN models. In this article we present an analysis of the integration of WiMAX and cellular networks, specifically UMTS. We opted to use a multi-layer integration solution based on MIH, MIP and SIP. MIH facilitates media-independent handover preparation procedures, while MIP and SIP are responsible, respectively, for network and application layer mobility. The goal of the analysis is to evaluate the continuity of service during and after network transitions, and also to assure that minimum quality levels for the 3GPP QoS classes (streaming, conversational, interactive and background) are respected. Simulations were conducted to assess service continuity and evaluate QoS parameters. Data on average network delay, jitter and throughput were collected, and average handover time was measured. Results showed that network transitions occurred in a seamless manner and minimum quality levels were attained for all four QoS classes.**

***Keywords*—Vertical Handover, Heterogeneous Networks, Multi-layer Integration, MIP, SIP, MIH**

I. Introduction

Contemporary mobile devices are capable of much more than simply making phone calls or accessing simple web-based applications. A typical mobile device, such as a tablet, smartphone or laptop computer, has a varied offering of web-based service at its disposal, all of which can be enjoyed on the go. The advent of social networks and multimedia streaming services entices mobile users to be always connected and reachable no matter where they are. Providing reachability and uninterrupted connectivity, however, is not an easy task. Networks have limited coverage, medium access technologies are diverse and user mobility is unpredictable, which means that mobile users may come across several distinct networks during their course, and switching to a different network due to loss of signal for example can be necessary.

Similarly, the device could identify a nearby network that is able to provide better service quality, making the vertical handover (switching to a different access technology) interesting. In order to assure service continuity and reachability, these heterogeneous networks need to integrate seamlessly, and user mobility needs to be managed efficiently.

978-1-4673-9408-6/15/$31.00

The integration problem can be viewed as two partially intersecting problems: (i) mobility management, and (ii) softly handing the connection over to a different network, without affecting ongoing application flows.

Mobility management concerns how users move within a network environment. Users can move in different speeds and in different patterns, affecting how they are served by different network technologies. For users moving at faster speeds, like car passengers, it may not be prudent to switch to short-coverage networks like WiFi, because they will move out of range very fast and handovers will be necessary.

The soft handover problem consists of transparently transferring the connection to a different type of network, without interrupting ongoing application flows. A user could be connected to a domestic WiFi network and, as they move out of its range, their connection could be seamlessly handed over to a cellular network, for example. Many solutions to solve both problems can be found in the literature.

Regarding user mobility, a common and established solution is Mobile IP (MIP) [1], which is a protocol developed by the Internet Engineering Task Force (IETF) with the goal of enabling transparent user mobility across networks. It does so by introducing network elements called the home agent (HA) and foreign agent (FA). Whenever a mobile host (MH) enters a new network, it receives a temporary care-of address (CoA). The CoA is relayed to the MNs HA by the foreign networks FA, to enable packet forwarding via tunneling. That way, an MH can be located outside of its original network and receive packets appropriately.

With respect to heterogeneous networks integration, a possible solution lies in the IEEE 802.21 Media Independent Handover (MIH) standard [2]. It provides a logical entity called media independent handover function (MIHF) capable of offering media independent services to facilitate the initiation and preparation stages of handovers between IEEE and 3GPP/3GPP2 networks.

Concerning service continuity, a possible solution is to use SIP (Session Initiation Protocol) [3]. It is an application layer protocol capable of initiating, maintaining and finalizing sessions between two or more hosts. It was designed to be readily compatible with widely-used Internet protocols. Application flows could be provided through SIP sessions.

When one of the hosts moves, a re-invite command can be issued to reestablish the previous session.

Assuring continuous connectivity throughout different networks alone is not enough, however. Whenever users switch networks, it is not only expected that ongoing applications will not come to a stop, but quality levels will be up to par with whatever network resources an application demands. To define what qualifies as adequate quality levels for different types of services, 3GPP (Third Generation Partnership Project) divided services into four quality of service (QoS) classes: streaming, conversational, background and interactive.

Each of the four QoS classes [4] has metrics that impact its performance the most, and 3GPP stipulates thresholds that relevant metrics must or must not reach in order for service quality to be considered adequate. Applications belonging to the conversational class, such as VoIP sessions, are very delay-sensitive, therefore delay must be as low as possible. For the interactive class, comprising services like web browsing and database querying, throughput is the most important metric, and it must be as high as possible. Streaming applications, like YouTube, need jitter levels to be kept as low as possible, because packets arriving with different delay intervals hinder fluid streaming service. Throughput is also relevant for high-definition video streaming, for example. Packet losses are what matter the most for background class applications, such as e-mail and file transfers. Having a trustworthy link and reliable transport protocols is crucial for this class.

Previously proposed solutions approach the problem in different ways. Single-protocol integration models range from models that perform link layer integration, using MIH and focusing on quality of experience (QoE) [5] or context-awareness [6]; transport layer integration using mSCTP [7]; network layer integration using MIP [8]; and application layer integration using SIP and focusing on VoIP applications (a class of applications that relies heavily on SIP). Solutions that integrate multiple protocols to achieve seamless vertical handover also exist, such as MIP-SIP integration [9], SIP-MIH integration [10] and MIP-MIH integration [11], but to the best of our knowledge no other solutions exist which integrate MIP, SIP and MIH to achieve seamless handovers. Other integration mechanisms do not involve multiprotocol integration, but decision schemes such as fuzzy logic systems [12], [13].

In this paper, we conduct an analysis of the integration of WiMAX and cellular networks (specifically UMTS) using a multi-layer, multi-protocol integration approach. Our analysis aims to verify if mobile users can roam seamlessly in an environment in which Universal Mobile Telecommunications System (UMTS) [14] and Worldwide Interoperability for Microwave Access (WiMAX) [15] networks are present. MIH is used to manage and report network events, to aid in the handover decision, while MIP and SIP are employed, respectively, in mobility management and service continuity.

The remainder of this article is organized as follows. In section II we expand on how MIP, SIP and MIH work. In section III, we present our integrated scenario. Section IV details the steps involved in the handover procedure. In section V we present our simulation results, in which we evaluated handover duration and QoS parameters such as throughput, latency and jitter to evaluate the seamlessness and quality of the integration solution. In section VI, we conclude the paper and suggest future research paths.

II. Protocols used in the integration solution

A. Mobile IP

MIP [1] is a mobility management protocol developed with transparent, network-independent mobility in mind. In networks that implement MIP, even if a mobile node (MN) moves away from its home network and enters a foreign network (FN), it can still be located and have packets addressed to it correctly delivered. Each network has an infrastructure element that acts as a location management and forwarding agent. It is called the home agent (HA) to MN that are originally from the network and a foreign agent (FA) to visiting nodes. Every MN has a home address (HoA), which is whichever address is assigned to it for a long time, and a corresponding home network (HN), which has a matching IP prefix to that of the HoA. Any other network is called a foreign network. MIP basically works in three stages: discovery, registration and tunneling. Agents constantly advertise their presence by means of special advertisement messages. MN then receive these messages and discover whether they are in an FN or not. If they have indeed entered an FN, they begin the registration stage, in which they receive a temporary address in the FN called care-of address (CoA). The HA is notified of the new CoA so it can start tunneling packets from correspondent nodes (CN) addressed to the MN on its HN to the FN where it is located using the CoA. Alternatively, if a CN knows the CoA, it can correspond directly with the MN without the HAs intervention, incurring in smaller latencies.

B. IEEE 802.21 Media Independent Handover

MIH [2] was developed with the goal of providing link layer information to upper layers in order to facilitate transparent, seamless handover across networks employing different link layer technologies, wired or wireless. It supports IEEE 802, 3GPP and 3GPP2 networks and takes part in the initiation and preparation stages of a handover, but does not directly participate in network selection and handover execution. MIH is composed of three basic services: (i) Media Independent Event Service (MIES), (ii) Media Independent Command Service (MICS), and (iii) Media Independent Information Service (MIIS). These services are provided by an entity denominated Media Independent Handover Function (MIHF), which resides within network protocol stacks as a middleware between link and network layers. The MIHF entity provides service access points (SAP) for higher layers to access the collected information and use available services. MIES handles link layer events, and is able to detect changes in link parameters or properties. The detected changes can be communicated to MIH users (layer 3 or higher of any entity that is compatible with MIH) that are subscribed to MIES events. MIES can, for example, detect that a certain link is losing signal strength, therefore will probably soon be unavailable, and report this to subscribed MIH users by means of an MIH event. MIES messages flow from lower to upper layers. MICS provides MIH users with a series of configuration commands and has an opposite flux: it originates from MIH users and flows from higher to lower layers. Using MICS, users can, for

example, subscribe or unsubscribe to MIES events, inquire about network parameters and configure network thresholds for link adaptation handovers. These thresholds are used in case a user comes upon a more capable network that meets their performance criteria for certain applications. MIIS is used by MIHF to obtain static information about networks present in the environment. Information obtained is used, for instance, to feed network selection algorithms, which need precise data about surrounding networks to make correct handover decisions.

C. Session Initiation Protocol

SIP [3] is an application layer protocol proposed by the IETF that is capable of establishing, maintaining and terminating multimedia sessions between two or more participants. It was not meant to be a standalone solution, but rather to integrate easily with other protocols such as Session Description Protocol (SDP), Real-time Transport Protocol (RTP) and Real-Time Streaming Protocol (RTSP). From the users point of view, the main purposes of SIP are to aid in locating prospect session participants, querying them about their interest in partaking in a session, negotiating session initiation parameters, modifying session parameters (if applicable), and terminating a session. Actual data exchanged in a session is not the responsibility of SIP, since other protocols like RTSP and RTP are used for that. SIP user agents (any SIP-enabled entity) usually have a unique identifier used to track them down across different networks or end-points. SIP uses a mix of proxy servers, redirection servers and registrar servers to enable location services, and user agents can query these servers for information on how to contact other user agents to establish sessions. Proxy servers are capable of receiving, forwarding and responding to requests. It does not issue requests and does not process its contents, relying on SIP headers to perform its functions. Usually a proxy server has access to a location database so it can determine the next hop for a message. Redirect servers, unlike proxy servers, do not forward requests, simply responding to them instead. It is used to inform agents of another agents location, making use of the same location database as the proxy servers. Registrar servers are servers that deal exclusively with SIP registration messages. If a registrar server receives any other kind of request, an error message is sent in response. When a registrar server receives a registration request, it updates the location database so other servers can correctly locate user agents.

III. UMTS/WiMAX Integration

In this section we describe our WiMAX/UMTS integrated scenario. 1 shows an overview of the scenario, with the main network components and how they interconnect. UMTS and WiMAX have distinct yet complementary characteristics that make their integration attractive. While UMTS offers a wider coverage area, it has lower bandwidth, which can compromise service quality for applications that rely heavily on it, such as multimedia streaming and large file transfers. WiMAX, on the other hand, offers higher bandwidth rates but has a smaller coverage radius. In a fully integrated UMTS/WiMAX environment, mobile users would be free to choose the network that best suits their application needs.

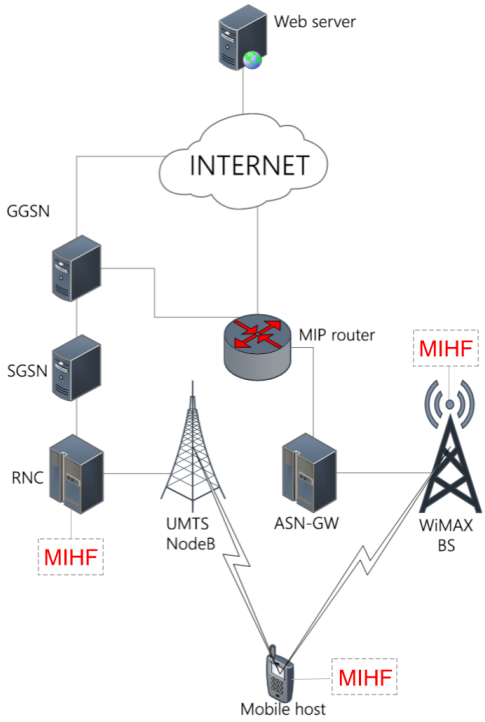


Fig. 1. Loosely Coupled Integration Solution

Our solution is loosely coupled, that is, the UMTS network and the WiMAX network have independent elements, protocols and methods for user authentication and billing, for example, and are integrated by means of an external element that acts as a mobility manager and gateway, the MIP router. A loosely integrated solution means that one network does not have to implement another networks protocols and does not need to be directly connected to another networks infrastructural elements. Mobility management is centralized and handled by MIP routers.

Some changes need to be made to existing network elements in order for the integration to work. The UMTS and WiMAX Internet gateway servers, Gateway GRPS Support Node (GGSN) and Access Service Network gateway (ASN-GW) respectively, need to support SIP proxy functionality in order to be able to handle SIP requests used by mobile hosts and corresponding nodes to establish application sessions. They are also responsible for authentication, authorization, access control and request forwarding. These gateway servers are also MIP foreign agents, having the additional task of assigning care-of addresses to visiting mobile hosts. Other network elements such as base stations remain unaltered. Additionally, a new element is introduced: the MIP router. It is added to act as a centralized mobility management unit, handling the registration of mobility terminals (SIP registrar functionality) and be their home agent. This means that it has to handle incoming requests from UMTS and WiMAX gateway servers informing them when a mobile host gets assigned a CoA.

MIH function entities reside between layers 2 and 3, link and network layers respectively. MIHF is installed in mobile

hosts, in UMTS RNC units and WiMAX base stations. MIHF entities have to be implemented in these elements because MIHF needs to have access to lower level network resources in order to properly trigger link events and gather precise information about available network resources, and those elements are responsible for managing their respective networks physical layer capabilities. Other features are included in mobile hosts. Figure 2 displays mobile host functional elements in a layer-like fashion. Starting from the bottom layers we have the two network interfaces, required to connect to both networks. Immediately above the link layer interfaces, the MIHF entity resides, with its three functional elements. In the network layer, we have the MIP implementation, to allow for network layer mobility across networks. The handover management element is responsible for handover initiation, which means that in our integration model handovers are always initiated by the mobile hosts. This proposition is due to the mobile hosts complete knowledge of currently running applications and their resource requirements. Therefore the mobile host was deemed to be the most capable element to make handover decisions based on its needs. The handover policy element implements algorithms to select which network a mobile host will switch to, when appropriate. The topmost layer is the application layer, in which the services that require connection continuity reside, and where SIP sessions are established.

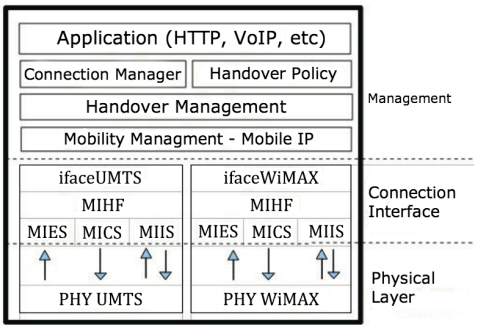


Fig. 2. Mobile Host Functionality Layers

IV. Handover Process

The WiMAX/UMTS and UMTS/WiMAX handover processes involve many steps that are mostly the same for both handovers, but due to the different network architectures and composing elements, some steps are slightly different. Figures 3 illustrates the UMTS/WiMAX (top) and WiMAX/UMTS (bottom) handover steps respectively. The handover process is initiated by the network discovery process, in which a mobile terminal scans the environment for available networks. Once the available networks are discovered, they can be analyzed by the mobile host to determine which one to switch to, taking available network resources such as bandwidth in consideration. The mobile host then signals the appropriate home network element that a handover process is being initiated. The signaled element is the RNC for UMTS networks and the base station for WiMAX networks. MIH handover preparation messages are then exchanged between home and foreign network controllers, followed by an MIP notification sent to the MIP router, to trigger packet buffering. The home network controller then sends an MIH configure message to the mobile host, informing it what WiMAX base station or UMTS NodeB is the best option to connect to. Next, the mobile host sends its home network an MIH link down message, terminating the current connection. After the connection to the home network is terminated, the mobile host can then connect to the foreign network. Following the MIH link down message, the mobile host initiates the connection process by sending an AAA (authentication, authorization and accounting) message to the foreign network (if it is a UMTS network) or a registration message (if it is a WiMAX network). The following step, PDP context activation (activation of a user session), is only present in WiMAX to UMTS handovers. After the connection to the foreign network is established, the mobile host receives its care-of address, through which it will receive packets in the new network. The mobile host proceeds to send an MIH link up message to the foreign network controller, followed by a SIP registration message to the MIP router, so that its new location and care-of address can be known by the mobility management element.

After registering the mobile hosts new IP address, the MIP router can start forwarding buffered packets to the mobile host. The last step is to reestablish the previous application session with the correspondent node, by means of a SIP REINVITE message, containing new session parameters like the new IP address and new network resources. The reasons behind the incorporation of MIH into the model are twofold. Since received signal strength (RSS) is used as a handover triggering criterion, devices need to have at their disposal an easy way to manage incoming radio signals and related events, which MIH offers (MIES). Additionally, MICS and MIIS are very useful when it comes to scanning neighboring networks for information and starting the handover process. MIP was added to the model as the network-layer mobility manager. With the use of foreign networks and care-of addresses, mobile hosts can be reached when they are away from their home networks. Agent advertising can also be used to identify foreign networks. SIP was included to eliminate the need for MIP tunneling, which was found to be inefficient. The use of application-layer sessions and REINVITE messages allow home agents, foreign agents, correspondent nodes and mobile hosts to bypass MIP tunneling, which speeds up the handover process. SIP registrar server functionality can also be used to track mobile hosts across distinct networks, as an alternative form of mobility management, using the application layer instead of the network layer.

V. Simulations and Results

In this section we discuss the results obtained after simulating our integration model. The objective of the simulations was to verify if handover times were acceptable and service quality levels were adequate. For the simulations, we used the network simulator NS-2 v2.31 and its trace file processing features were used to collect the results presented in this section. Some of the network elements were configured in a way that would not be optimal for an actual deployment setting. The MIP router, for example, acts as the home network for mobile hosts at all times, independently of the actual network of origin. It also accumulates the functions of MIP mobility manager, SIP proxy and SIP registrar. SIP proxy functionality was also added to GGSN and ASW-GW in the UMTS and WiMAX networks respectively. These simplifications were introduced in order to make implementations less complicated and keep

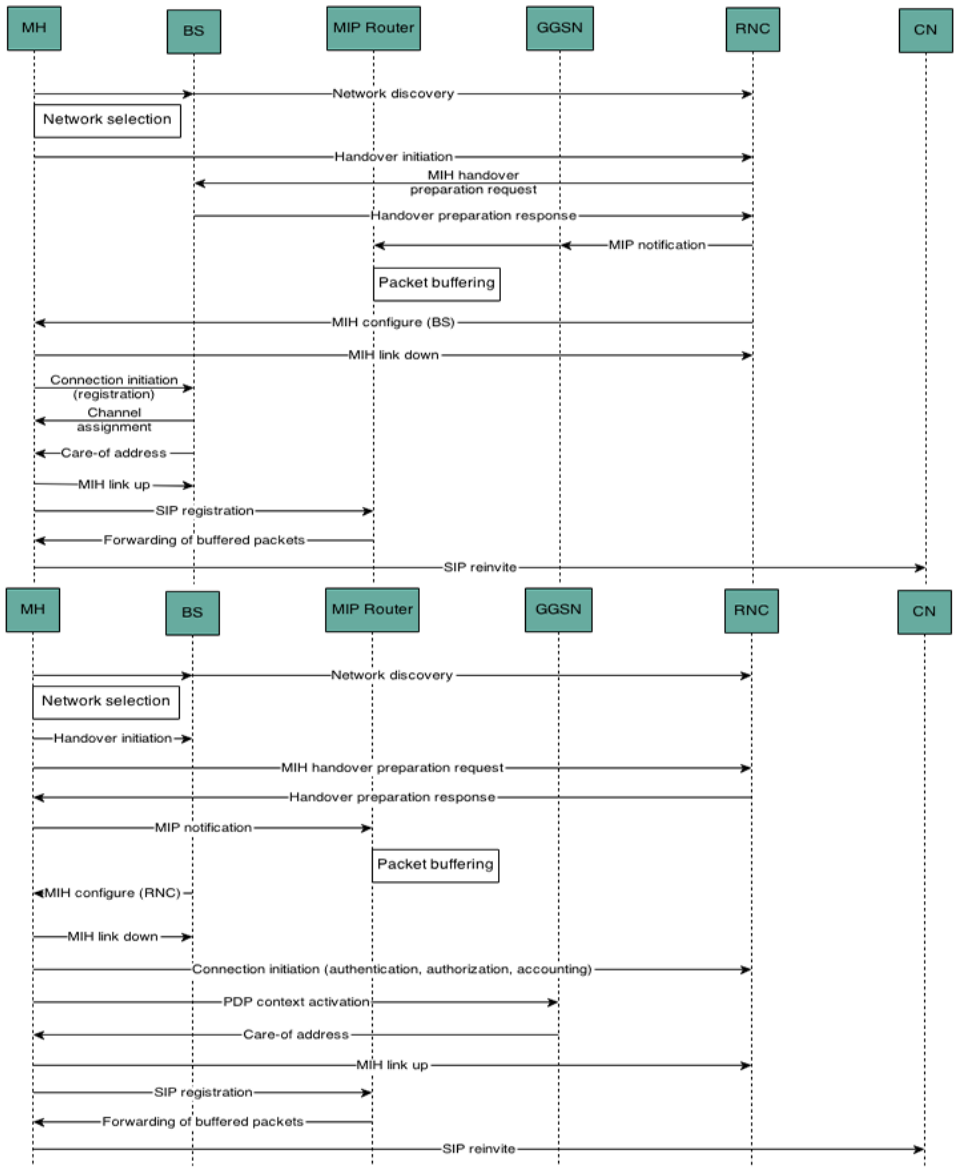


Fig. 3. Handover Steps

the integration architecture simpler, without adding too many infrastructural elements. While the version of NS-2 used in this work has native MIP functionality, external modules for SIP and MIH had to be incorporated to our code to add the desired functionality. Both modules were developed by the National Institute of Standards and Technology (NIST).

The simulation scenario consisted of a heterogeneous UMTS-WiMAX scenario containing 1000 users. 10% of the users were active at any given time during the simulation. Applications were distributed among users in the following way: 15% of users ran a Background class application; 15% of users ran Conversational class applications; 30% of users ran Streaming applications; and 40% of users ran Interactive applications. A total of 33 executions of the simulated scenario were performed and, for each of them, performance results were collected. The confidence interval for the simulations is 95%.

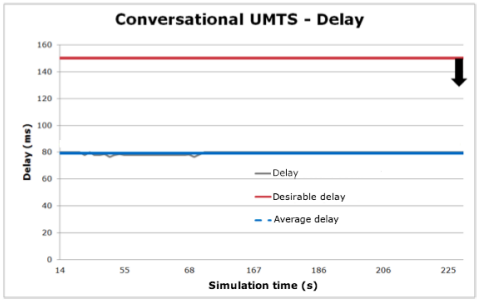


Fig. 4. Delay results for the Conversational class in the UMTS network

The first metric evaluated was total handover time. Time measurement started at the network discovery stage, which marks the beginning of the handover process, and

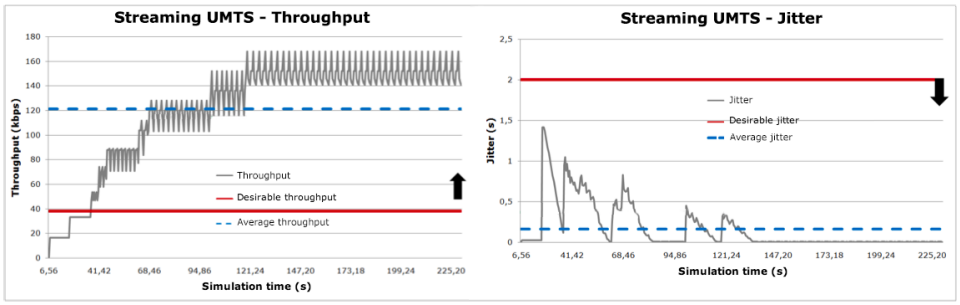


Fig. 6. Throughput and jitter results for the Streaming class in the UMTS network

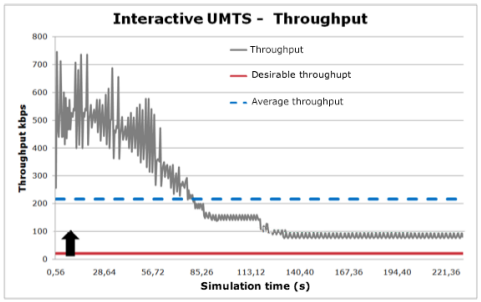


Fig. 5. Throughput results for the Interactive class in the UMTS network

ended after the SIP REINVITE message was sent. Average WiMAX/UMTS handover time was 77 ms, and average UMTS/WiMAX handover time was 56 ms. The difference can be attributed to AAA and PDP context activation, steps that are absent in UMTS/WiMAX handover. Figure 4 shows performance values for delay in the conversational class. For this class, the stipulated 3GPP limits for delay are 150 ms, because conversational applications are very delay-sensitive. In our simulations, the average measured delay for conversational applications was around 80 ms, which places our results within the expected 3GPP limits. The good delay results can be explained in part by the use of SIP to avoid MIP tunneling. Instead of routing the packets through a third network element, the use of SIP sessions, along with requests to the proxy to find the new address of nodes that underwent handovers, allows packets to be sent directly between corresponding nodes, thus allowing them to get to their destination in less time. If MIP tunneling was used instead of SIP sessions, packets would have to take a longer route to their destinations, and delay rates would end up being higher.

Figure 5 shows throughput performance for the interactive class. Albeit throughput is the most important parameter for the interactive class, typical interactive applications such as website browsing do not typically require very high network performance, therefore 3GPP stipulates a minimum of 20 kbps of throughput in order for network performance to be regarded as acceptable. In our simulations, we observed that throughput values for the interactive class suffered many variations, but always stayed well above minimum recommended values. We measured an average throughput of 175 kbps, with peaks of more than 700 kbps.

Figure 6 shows jitter and throughput performance for the streaming class. For this class, 3GPP determines that minimum throughput values must be between 32 and 384 kbps, and jitter values must not be above 2 seconds. Our results show that throughput for the streaming class starts off below the minimum acceptable value during the first seconds of the simulation, but, in spite of some observed variation, stays above the 32 kbps mark, yielding an average of 121 kbps. Jitter took some time to stabilize, but stayed below the 2 second ceiling for the entire duration of the simulations.

As was previously mentioned, the most important metric for background applications is packet loss. Since application packets are buffered in the MIP router during the handover process, there are not packet losses. Therefore we considered our model to deliver acceptable performance for background applications.

VI. Conclusions and Future Work

In this article we presented a heterogeneous network integration analysis which aimed to evaluate service continuity and QoS guarantee in UMTS/WiMAX environments. A multilayer integration mechanism was implemented, with the goal of executing fast, seamless vertical handovers. Simulations were conducted using NS-2 to verify that handover processes were successfully completed and service quality parameters were adequate. Results showed that both UMTS/WiMAX and WiMAX/UMTS handovers were successfully executed, and average performance values for the four QoS classes were enough to fulfill 3GPP requirements. Therefore, our network integration, service continuity and quality of service objectives were reached.

As future work, we suggest adding other network technologies to the analysis, such as LTE and Wi-Fi, and modeling more complex scenarios, involving different user mobility patterns and many different applications being executed at the same time. Adding new networks should not be a problem, because the integration mechanism does not rely on any specific infrastructure component and is therefore reusable.

References

[1] C. E. Perkins, “Mobile ip,” *Communications Magazine, IEEE*, vol. 35, no. 5, pp. 84–99, 1997.

[2] A. De La Oliva, A. Banchs, I. Soto, T. Melia, and A. Vidal, “An overview of ieee 802.21: media-independent handover services,” *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 96–103, 2008.

[3] A. B. Johnston, *SIP: understanding the session initiation protocol*. Artech House, 2009.

[4] T. ETSI, “125.401,” *UTRAN Overall Description*.

[5] J. Jailton, T. Carvalho, W. Valente, C. Natalino, R. Frances, and K. Dias, "A quality of experience handover architecture for heterogeneous mobile wireless multimedia networks," *Communications Magazine, IEEE*, vol. 51, no. 6, 2013.

[6] M. Xiong and J. Cao, "A clustering-based context-aware mechanism for ieee 802.21 media independent handover," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 1569–1574.

[7] L. Ma, F. Yu, V. C. Leung, and T. Randhawa, "A new method to support umts/wlan vertical handover using sctp," *Wireless Communications, IEEE*, vol. 11, no. 4, pp. 44–51, 2004.

[8] M. M. A. Khan, M. F. Ismail, and K. Dimyati, "Seamless handover between wimax and umts," in *Communications (MICC), 2009 IEEE 9th Malaysia International Conference on*. IEEE, 2009, pp. 826–830.

[9] N. M. Alamri and N. Adra, "Integrated mip-sip for ims-based wimax-umts vertical handover," in *Telecommunications (ICT), 2012 19th International Conference on*. IEEE, 2012, pp. 1–6.

[10] G. P. Silvana and H. Schulzrinne, "Sip and 802.21 for service mobility and pro-active authentication," in *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*. IEEE, 2008, pp. 176–182.

[11] K. Taniuchi, Y. Ohba, V. Fajardo, S. Das, M. Tauil, Y.-H. Cheng, A. Dutta, D. Baker, M. Yajnik, and D. Famolari, "Ieee 802.21: Media independent handover: Features, applicability, and realization," *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 112–120, 2009.

[12] S. B. Johnson, S. Nath, and T. Velmurugan, "An optimized algorithm for vertical handoff in heterogeneous wireless networks," in *Information & Communication Technologies (ICT), 2013 IEEE Conference on*. IEEE, 2013, pp. 1206–1210.

[13] A. Singhrova and N. Prakash, "Adaptive vertical handoff decision algorithm for wireless heterogeneous networks," in *High Performance Computing and Communications, 2009. HPCC'09. 11th IEEE International Conference on*. IEEE, 2009, pp. 476–481.

[14] F. Hillebrand, *GSM and UMTS: the creation of global mobile communication*. John Wiley & Sons, Inc., 2002.

[15] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "Ieee standard 802.16: a technical overview of the wirelessmantm air interface for broadband wireless access," *IEEE communications magazine*, vol. 40, no. 6, pp. 98–107, 2002.

Evaluating Performance Degradation in NoSQL Databases Generated by Virtualization

Gustavo Martins, Petrônio Bezerra, Reinaldo Gomes, Fellype Albuquerque
Science Computer and Systems Department
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
{gustavomartins, petroniocg}@copin.ufcg.edu.br,
reinaldo@dsc.ufcg.edu.br,
fellype.albuquerque@ccc.ufcg.edu.br

Anderson Costa
Computer and Technology Department
Federal Institute of Education, Science and Technology of Paraiba, Campina Grande, Paraiba, Brazil
anderson@ifpb.edu.br

Abstract— **There are many services in Cloud Systems currently on the market for hosting servers with different purposes. These environments are IT infrastructures frequently deployed in large data centers using Virtual Machines. Moreover, an increasing number of users are using massive online application resources with hosted databases on virtual machines. In light of this, there is a critical need to evaluate databases performance in virtual environments. This work aims to evaluate overhead generated by virtualization in two common NoSQL databases, Cassandra and MongoDB, by different virtualization techniques - full virtualization and paravirtualization. For this, we used the YCSB benchmark (Yahoo! Cloud Serving Benchmark) to drive performance tests, with the results being evaluated through statistical analysis. As findings, the performed experiments in both virtualization techniques demonstrated that the experimental design comprises most of the factors in MongoDB, different from Cassandra which comprises of just a few. In paravirtualization scenarios the environment factor was more sensitive to experimental variations overcoming factors such as threads and numbers of transaction. Both databases in full virtualization scenarios reached significant variation, however the threads and transactions factors were more significant.**

Keywords—Virtual Machine; Performance; Experimentation; Cassandra; MongoDB

I. INTRODUCTION

Machine virtualization and cloud software systems were a segment that had shown rapid growth in the software market in recent years. This market is fast as ever, with the rise of several companies bringing their respective solutions in this segment. However, with so many virtualization companies competing for users' attention, choosing a solution that best addresses a set of requirements for a specific purpose may prove to be a hard work and deserve attention in the development of architecture host solution [1].

Over the years, VMware Inc. has established itself as one of the main players on virtualization market growth. However, the highlight product of the VMware, vSphere Hypervisor, which provides a virtual operating platform for guest operating systems, faces increasing competition from companies such as Microsoft, Citrix and Oracle. Currently, VMware holds approximately 56% of the virtualization market, but its market share has declined steadily in recent years. Although VMware continues to provide a high-level virtualization platform, other market players have started to gain space in terms of performance and features [2].

However, this market growth does not happen in all segments. TechNavio analysts predict that the global market for server virtualization will grow an average rate of 31.07% in the period 2012-2016, based on the CARG (Compound Annual Growth Rate). One of the main factors contributing to this market growth is the need for better productivity of enterprise servers. On the contrary of market servers, virtualization used by end customers has been falling rapidly [3].

Furthermore, data centers have adopted virtualization technologies for many benefits achieved through its use: power and space consolidation, hot migration, high availability, and fault tolerance. Moreover, an increasing number of users are using online application resources with databases hosted on virtual machines. Therefore, currently there is a clear critical need to evaluate database performance in virtual environments [4]. The large number of solutions provided by the main players of the virtualization market encourages many research initiatives to evaluate and compare the performance platforms, to try to highlight the best solutions and its features. This work seeks to evaluate the overhead generated by virtualization in NoSQL databases' performance, which were deployed in two different virtualization platforms based on *full virtualization* and *paravirtualization* techniques. NoSQL databases adopted in this work, Cassandra [5] and MongoDB [6], are commonly used in online applications that deal with large volumes of data and meet several requirements such as availability, scalability, fault tolerance and high performance [7]. In this experiment, we focused on databases performance in physical and virtual environments and compare them. To achieve this, we developed an experimental analysis defining some factors that may affect database performance. From this, we analyzed the influence of each experimental variation about databases performance. The remainder of the paper is structured as follows: In section II we present the related works. Section III we discuss about defining the problem, followed by experimental hypothesis descriptions in Section IV. The experimental detailing is presented in Section V. Data Analysis and Discussion is then discussed in section VI. Some

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

Threats to Validity are also given in Section VII. Finally, we conclude the work in section VIII.

II. Related Work

As the virtualization solutions have grown, new techniques have been proposed by branch companies that aim to provide better performance and resources through improvements. Two examples of these techniques are paravirtualization and hardware-assisted virtualization, each one implementing different paradigm of virtualization, and it may drive users (architect or design solution) to ask themselves about which of these solutions can best address the desired requirements. Moreover, the following question may arise: May there be a significant impact on performance in a given application due virtualization technique adopted? In attempt to answer this question, some studies were performed in order to highlight on the impact of virtualization in some contexts.

As a contribution given by [8], a study was carried out to analyze the impact that virtualization overhead can generate in NoSQL databases and Relational Database - SQL, implemented on virtual machines. Cassandra, MongoDB and PostgreSQL databases were selected to be used in the experiment. To conduct the performance analysis of the databases in physical and virtual machines, an environment was set up to perform reading and writing transactions. For this, a Java application was developed. Results showed negative and positive points in database performance due to virtualization compared to a physical machine. An example of this was obtained when the PostgreSQL database score overcame Cassandra, both running in virtual machine, about single reading for several threads (hosts) scenario. Remembering that databases like Cassandra are marked by high scalability feature. In this feature, PostgreSQL reached better performance than Cassandra. However, the results from this experiment were not statistically treated in order to analyze the influence of the factors discussed in the experiment, and our paper differs from their approaches because we used YCSB benchmark, which is standard for evaluate NoSQL databases currently. Another difference is that they used only Xen as Virtual Machine Monitor (VMM).

In [9], another initiative also aimed to analyze the performance of different VMMs and compare them concerning overhead. The experiment comprises VMMs from different manufacturers and also considers other technical approaches such as full virtualization and paravirtualization. For this, the Phoronix Test Suit benchmark was used as workload to analyze processing performance, reading and writing in main and secondary memory and network traffic. The results obtained shows that there was a considerable performance difference among VMMs evaluated: VMware Player, Virtual Box and Xen. *Xen* reached very good performance in workload execution of CPU bound, questionably, better than physical machine results. In other tests, VMware achieved the best results in main and secondary memory workload scenarios. It was expected by the authors that paravirtualization technique yielded best results due to system kernel modifications made by *Xen*, but findings did not show that. VMware stood out in most scenarios. This work was important to give us background in performances comparisons with virtualization techniques.

In [10], the authors made a comparative study of six virtualization technologies distributed under GPL license. They evaluated the imposed overhead by the virtualization layer through benchmark software, both in the real machine and virtual hosts Linux as operating system. In addition, the scalability of hypervisors was assessed by concurrent execution of the benchmark suite through multiple virtual machines. This is similar to our work since the authors also checked performance using a benchmark in database servers, however they used SQL databases. The results indicated that virtualization technologies such as paravirtualization (used in Xen) and OS-level virtualization (as in Linux-VServers and OpenVZ) got better use of available physical resources. Our research aims evaluate NoSQL databases due to adoption growth of these databases in large web applications.

The authors of [11] collaborated with an experimental evaluation of the performance overhead caused by VMware Player, QEMU, Virtual PC and Virtual Box. The performance of VMMs with Linux OS was evaluated running several benchmarks, which tested CPU, disk I/O and network traffic. The authors concluded that results from virtual machines performance could, simultaneously, depend on two factors: the virtualization technique used and the type of application evaluated. Furthermore, they also found that applications limited by CPU bound have less impact than those associated with I/O. These finding were so important to highlights our research since important factors were appointed as some performance issues, as example: technique and type application. However, there is no evidence how significant are the factors effects under performance application.

Another similar research initiative in order to compare performances between NoSQL databases using the YCSB benchmark, can be found at [14]. The authors state that IT professionals work hard to decide which databases comprise more performance requirements in order to set an optimized database for their application's user cases. For this, it's necessary to compare performances between them. In this paper, the YCSB was adopted to measure the performance of four NoSQL databases: Redis, MongoDB, ElasticSearch and OrientDB. They performed this using a single PC, but they did not handle these databases in VMMs.

Another contribution that is similar to our work can be seen in [15]. In their work, they had presented a method and the results of a study that selected among three NoSQL databases for a large, distributed healthcare organization. They showed the performance evaluation method and results to the following databases: MongoDB, Cassandra and Riak. The authors endorse the YCSB benchmark as the *de facto* standard for evaluate NoSQL databases. About the performance, they found that Cassandra database provided the best throughput performance, but with the highest latency.

Many research initiatives were taken aim to evaluate NoSQL databases as well as SQL databases, however none of

these works took into consideration VMMs as deployment platform to analysis the impact of virtualization overhead in databases' performance, mainly by the large use of Cloud Computing by the companies.

As we can see from related works, the issue is very relevant and has been exploited by the scientific community. In light of this, we feel the need to identify, based on statistical analysis, the main factors responsible for performance overhead in two NoSQL databases and confirm whether virtualization factor is relevant in the experiment.

III. PROBLEM DEFINITION

Given the wide applicability of virtualization in data centers and the wide use of NoSQL databases that support large demands of data in a scalable and flexible manner, we aim evaluate through performance test using YCSB [12] the Cassandra and MongoDB databases in real and virtual machine. The main purpose is to analyze virtualization overhead impact in databases performance. For this, consolidated solutions in market virtualization based on different techniques were adopted: VMware and Xen Server. These VMMs implement the different virtualization techniques *full virtualization* and *paravirtualization*, which address a different approach. This work seeks to answer the following issues that generate the research questions:

- **Business Problem:** Can the virtualization approach add business value to host service due performance improvements related to technique?
- **Technical Problem:** Can a given solution hosted in virtualized servers suffer slow-down due to virtualization technique adopted?

IV. HYPOTHESIS DEFINITION

The research goal is to evaluate the performance of NoSQL databases in virtualization context, Cassandra and MongoDB, as well as across physical and virtualized environments in order to check virtualization overhead. Based on a large amount of applications deployed in VMM, research questions (RQ) below come elicit the following hypotheses based on previously described scenario:

RQ 1: Can the virtualization technique (full virtualization and paravirtualization) be a reason to difference in NoSQL databases performance?

- H1-0: Both techniques, full virtualization and paravirtualization, offer a deployment environment with performance equivalent - equivalent performance (Null Hypothesis).
- H1-1: Full Virtualization provides a deployment environment with better resource management for NoSQL databases, hence resulting in better performance (Alternative Hypothesis).
- H1-2: Paravirtualization provides a deployment environment with better resource management for NoSQL databases, hence resulting in better performance (Alternative Hypothesis).

RQ 2: Which one database with respective VMM presented better performance compared to physical hosts?

- H2-0: Comparing physical and virtual environments, there is no significant performance overhead (Null Hypothesis).
- H2-1: Cassandra database performs better in terms of overhead when compared to MongoDB due to virtualization - Full Virtualization (Alternative Hypothesis).
- H2-2: MongoDB database performs better in terms of overhead when compared to Cassandra due to virtualization - Full Virtualization (Alternative Hypothesis).
- H2-3: Cassandra database performs better in terms of overhead when compared to MongoDB due to virtualization - Paravirtualization (Alternative Hypothesis).
- H2-4: MongoDB database performs better in terms of overhead when compared to Cassandra due to virtualization - Paravirtualization (Alternative Hypothesis).

V. EXPERIMENTAL DETAILMENT

For the experiment, we adopted Xen-Server 4.1 as the paravirtualization technique and VMware Workstation 11.0 as full virtualization on both real and virtual machines running on Ubuntu 12.04 O.S.. In virtual instances as well as in real machine, Cassandra 2.0 and MongoDB 2.6.6 were installed. Regarding experiment workload, the YCSB benchmark was used to drive performance tests on databases installed on physical and virtual environments. Based on the factors set forth in the experiment, we defined a mathematical model based on detailed experimental design, in the next session.

A. Experimental Design

The design adopted for the experiment was $2^k r$ factorial. Mathematical model (1) below address the factors and levels presented in Table 1:

$$y_{ij} = \mu_0 + x_{ai}q_a + x_{bi}q_b + x_{ci}q_c + q_{ab}x_{ai}x_{bi} + q_{ac}x_{ai}x_{ci} + q_{bc}x_{bi}x_{ci} + q_{abc}x_{ai}x_{bi}x_{ci} + \epsilon_{ij} \quad (1)$$

TABLE 1 - FACTORS AND LEVELS

Factors	Clients	Transactions	Environment
Levels	1	100000	Physical
	10	1000000	Virtual
Output Variable	Transactions per Seconds		

Describing the model, we have μ which represents the arithmetical mean of results, followed by each factor showed in Table 1 and represented by x_a, x_b, x_c terms and then followed by its interaction between them in *i*th experiment treatment. Each term can assume two values, -1 and 1,

representing the higher and lower levels. The remaining terms q_a, q_b and q_c with respective interaction among them, represent the experimental effects. The difference between the estimate and the measured value y_{ij} in the *j*th replication of the *i*th treatment represents the experimental errors through ϵ_{ij}. As we can see in Table 1, the levels of each factor are addressed in each level term with higher and lower values.

B. Data Collection

To estimate the experimental error, the treatments were replicated three times. Given the number of factors, levels and replications, we obtained a total of 24 runs from 8 experimental treatments for each of VMMs and physical instance. Tables 2 to 4 show the performance data collected through YCSB Benchmark in two commons workloads: Load operation and Select/Update operations. Load operations means record insertion in database and Selection/Update operations means 50% of record selection and 50% record update. Threads and transactions are parameters related quantity of clients, simulated by thread, and quantity of operations, respectively. The throughput unit is given in transactions/seconds.

TABLE 2 - PHYSICAL ENVIRONMENT - YCSB WORKLOAD A - LOAD OPERATION ON THE LEFT AND 50%/50% SELECT AND UPDATE OPERATION ON THE RIGHT

	Cassandra Database				**MongoDB Database**			
	Load Operations		**Select/Update Operation**		**Load Operations**		**Select/Update Operation**	
Transactions	**Physical**		**Physical**		**Physical**		**Physical**	
	1 Thread	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**
100000	3250	6720	1977	3461	5548	8631	8714	9607
	5988	8126	4023	6108	5599	8722	9082	3112
	4196	6505	1625	3093	5742	8569	8262	11576
1000000	3484	6946	1280	2436	5064	7402	10093	17903
	6486	7146	4558	5375	5073	7806	10048	19223
	4514	5821	3382	3850	5009	7879	10136	19740

TABLE 3 - VIRTUAL ENVIRONMENT XEN - YCSB WORKLOAD A - LOAD OPERATION ON THE LEFT AND 50%/50% SELECT AND UPDATE OPERATION ON THE RIGHT

	Cassandra Database				**MongoDB Database**			
	Load Operations		**Select/Update Operation**		**Load Operations**		**Select/Update Operation**	
Transaction	**Virtual**		**Virtual**		**Virtual**		**Virtual**	
	1 Thread	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**
100000	3591	4765	2893	3656	3474	4405	5079	7241
	3687	3479	3470	3568	3373	4367	5453	6544
	3430	4549	2297	3433	3508	5132	4928	8140
1000000	3614	3220	2509	2119	3171	4386	6204	11161
	2996	2916	3361	3896	3165	4661	6163	9455
	3532	3185	2905	3058	3156	4161	6155	11228

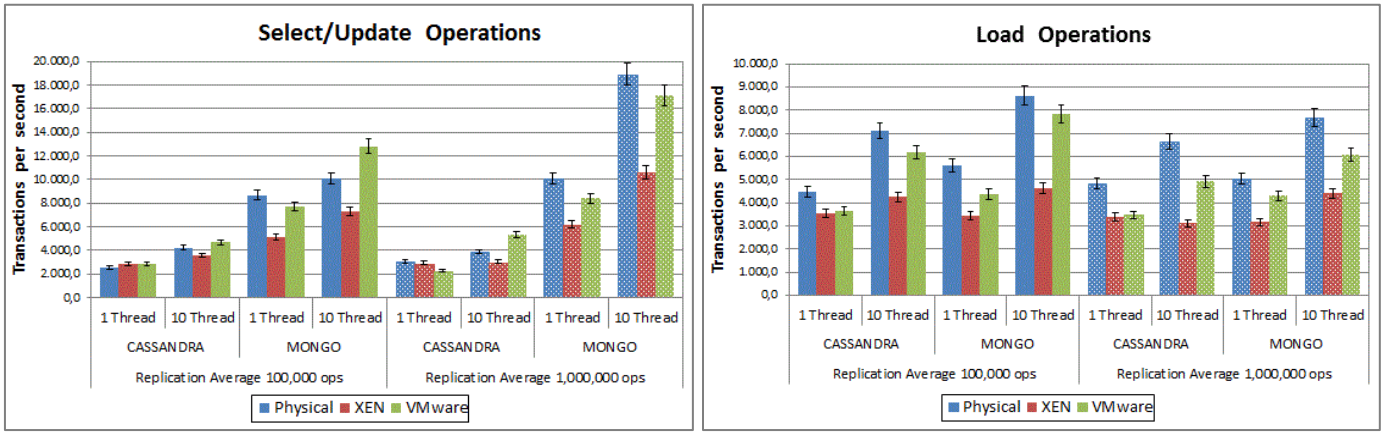
TABLE 4 - VIRTUAL ENVIRONMENT VMWARE - YCSB WORKLOAD A - LOAD OPERATION ON THE LEFT AND 50%/50% SELECT AND UPDATE OPERATION ON THE RIGHT

	Cassandra Database				**MongoDB Database**			
	Load Operations		**Select/Update Operation**		**Load Operations**		**Select/Update Operation**	
Transaction	**Virtual**		**Virtual**		**Virtual**		**Virtual**	
	1 Thread	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**	**1 Thread**	**10 Thread**
100000	3499	5249	3155	3861	4164	8178	7271	11503
	4180	7861	2268	6031	4137	8233	8044	15555
	3288	5447	3046	4166	4833	7175	7908	11414
1000000	3542	4858	2071	5476	4166	5878	8339	17067
	3425	4989	2823	5208	4488	6016	8453	17181
	3444	4940	1924	5410	4241	6373	8481	17216

VI. DATA ANALYSIS AND DISCUSSION

Checking scores from the execution of YCSB, we can analyze that Cassandra was beaten by MongoDB in all cases, both running on a physical machine and on virtual machines (Xen Server and VMware). Aiming to analyze database performance, Fig. 1 *a* and *b* shows the average achieved by databases. The confidence levels are indicated above in each bar graph, with 5% of confidence interval. As we can see, physical host reached the best performance

(transaction/second) in load operations and almost all select/update operations, followed by VMware and Xen, due to overhead imposed by the virtualization layer. In the select/update scenario, we can question some results that virtualization overcame the physical machine as we can see in Fig. *b*. Similar results also were found in [9], when Xen overcame the physical machine in given scenario through its benchmark. Furthermore, MongoDB had been less susceptible database in this context. Seeking to analyze the influence of the factors discussed in the results, we carried out an experimental analysis, as established in the experimental design section, using the statistical software Minitab 17 [13], trial version for academic purpose. ANOVA (Analysis of Variance) was conducted in order to verify the statistical significance of each factor in the output variation (transaction/second). For this, statistical significance level was set up in 5% (α=0.05).



a) *b)*

Figure 1 - Benchmark Operations – Databases' Average Score

Below, Fig. 2 and 3 show the significant factors relating to treatments through Xen virtualization and load and select/update scenarios for both databases. The points on the graph represent the experimental factors, in which red represent statistically expressive factors (it means p-value less than 5%) and blue not expressive. Also, the distance of red point from normal line represents factors' size effect regarding influence on the output variable. Between the graphs, the scale should be considered.

Analyzing the environment variation factor, from ANOVA results, it was responsible about 49% of the variation on the Cassandra database in Load operation, with 21% of experimental error. In Cassandra's select/update scenario, no factors were significant.

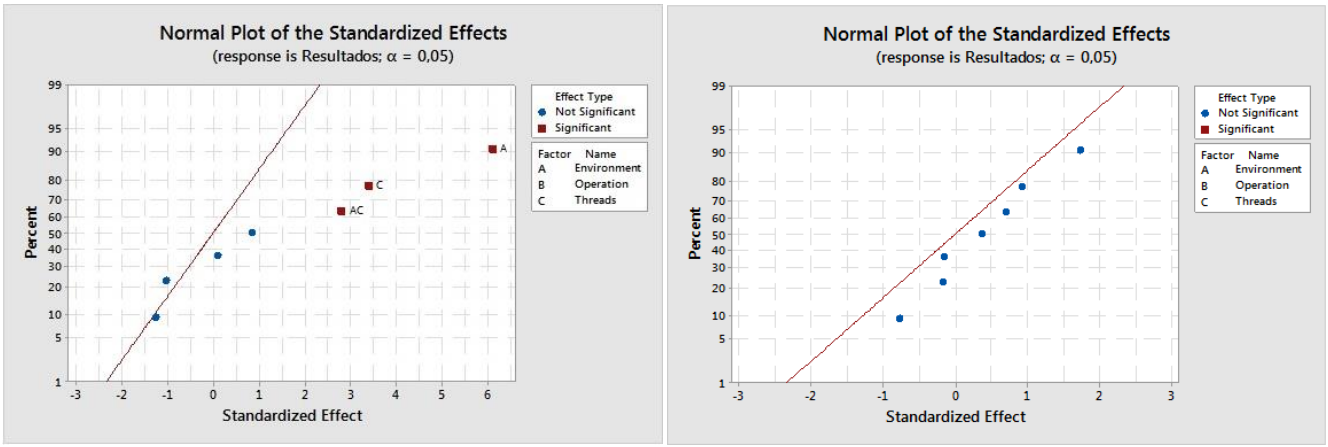


Figure 2 - On the left Cassandra on Xen Server -YCSB Workload a Load Op. and on the right Select/Update Op. (50%/50%)

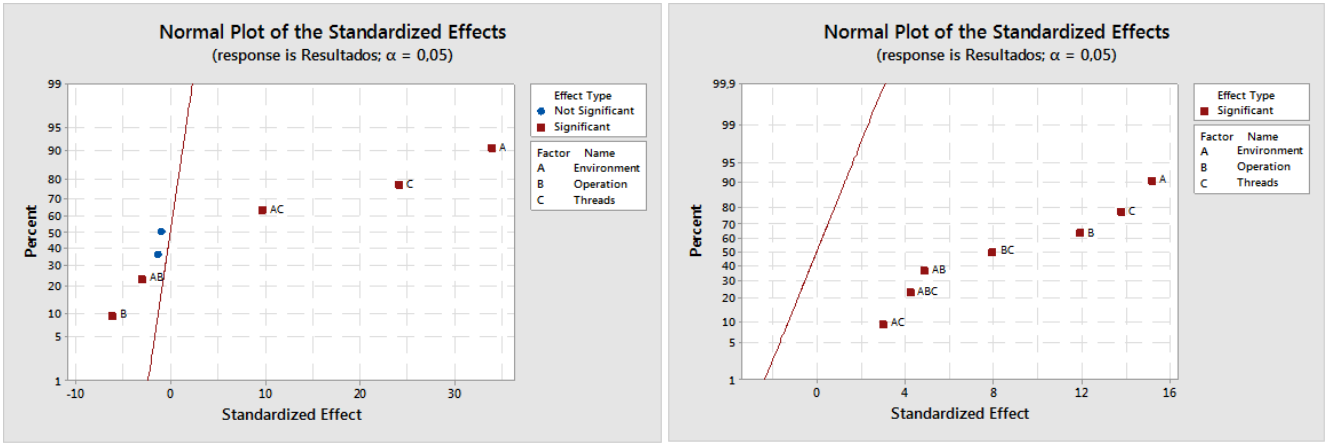


Figure 3 - On the left MongoDB on Xen Server -YCSB Workload a Load Op. and on the right Select/Update Op. (50%/50%)

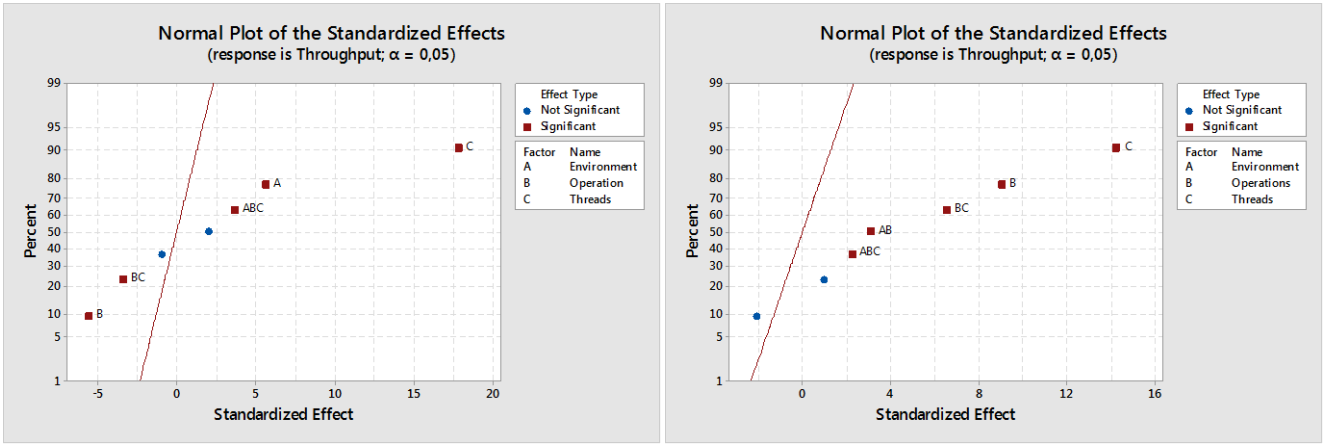


Figure 4 - On the left MongoDB on VMWare -YCSB Workload a Load Op. and on the right Select/Update Op. (50%/50%)

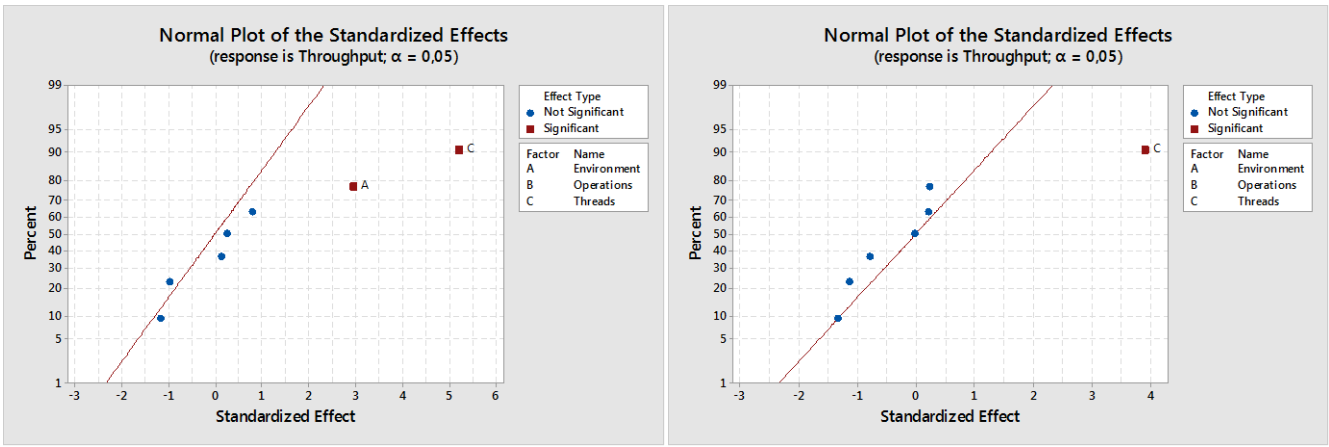


Figure 5 - On the left Cassandra on VMWare -YCSB Workload a Load Op. and on the right Select/Update Op. (50%/50%)

Also Cassandra presents a lower score in terms of performance, as we can check in Table 3, it did not achieve significant results in the selection/updating scenario. As we can analyze in Fig. 3, MongoDB in the select/update and load scenario had an impact on all factors of its performance, besides the environmental factor (physical or virtual) showed the greatest variation in the experiment. In load case the variation of the environment factor was about 60% followed by threads with approximately 30%. The experimental error in this case was 0.8%, thus fitting the model very well. Regarding the select/update scenario, the environment reached 33% of variation followed by threads with 27%. The experimental error in this case was 2%. It means that virtualization has a significant influence on output variable in these scenarios.

The results obtained by running the benchmark on VMware presented very different conclusions from the Xen. As we can see in Fig. 4 and 5, the environment factor did not appear as so expressive as previous scenarios, which had presented itself as the most significant factor regarding databases' performance impact. For instance, Fig. 5, in load case, Cassandra has Threads as the greater significant factor with 49.5%, followed by Environment with 16%. The error in this case was 29%. Thread appears alone in the select/update scenario with almost 44% and error about 47%. The experimental model did not fit well with Cassandra as ANOVA shows. It may imply uncovered factors in Cassandra's scenarios. One more time, MongoDB fitted the model better than Cassandra. In the last scenarios, in both load and select/update operations Threads factor represent the greater variation with 75% in load case and 56% in select/update. The error for both operations, in these last cases, was less than 5%. Furthermore, the most significant factors in the latter scenario are those well known as influential in distributed databases: number of threads (host) and transaction. To validate this linear model of experimental design, we have to look the residuals and verify if it meet three basic assumptions: the residuals distribution must be normal, residuals must be independent and variance has to be constant [16]. From descriptive statistical analysis perspective, all graphs generated by Minitab tool met these assumptions. Figure 6 shows an example of the residuals distribution behavior, variance and independence, from MongoDB in Select/Update workload. For each database and workload, the same descriptive analyses was performed for both databases.

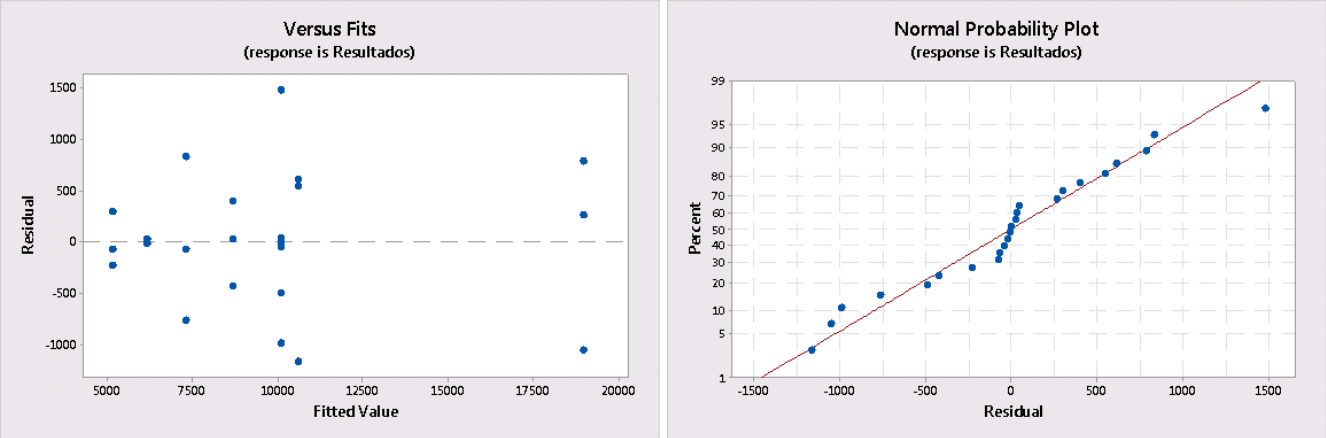


Figure 6 - On the left residuals' variance and independence graph and on the right residuals' distribution behavior (MongoDB).

VII. Threats to Validity

In this experiment, we identified some threats to validity. Below are the possible threats to validity:

- **Treatment and Selection Threat:** The results reported in this paper are only relate to the virtual machines covered in the experiment, with NoSQL databases, Cassandra and MongoDB, using the benchmark YCSB. Therefore, from the findings we cannot generalize exceeding this threshold. For further conclusions, it is necessary to extend the study using other virtual machines based on full virtualization and paravirtualization techniques.

- **Treatment and Environment Treat:** The experiment was conducted using a small local server and not a real data center. However, the same settings of CPU resources, memory and storage were given in all treatments.

- **Random Irrelevance:** There is not much guarantee that CPU resources and memory were equally available in virtual machines while the benchmark was performed. Somehow, this variation should be minimal because identical OS versions had been installed in both real and virtual hosts.

VIII. Conclusion

As we had seen, virtualization techniques can cause overhead performance in databases platforms deployed on VMMs, in our case, NoSQL databases were explored as an example. From the findings, we could reject the H1-0 (null hypothesis) and H1-2 (alternative hypothesis) of RQ1. Thus, we cannot refute the hypothesis H1-1 with 5% significance level, it means full virtualization provides a deployment environment with better resource utilization for NoSQL databases. Regarding RQ2, we should note which virtualization techniques were less responsible for fluctuation in output variable, hence the database that had equivalent performance between virtual and physical instances. Therefore, we can say that the most impartial technique on the output variable was full virtualization. In

other words, hypothesis H2-0, H2-3 and H2-4 were refuted. Only the MongoDB scenarios were shown to be less susceptible to virtualization overhead using the full virtualization technique. Of course, this study is not conclusive and we must observe aspects related to the overhead of VMMs in various applications context commonly used in deployment environments that benefit from virtualization. Moreover, such research requires the use of VMMs with different techniques, but it is also necessary to expand the range in NoSQL solutions to replicate this experiment, adding more scientific relevance.

ACKNOWLEDGMENT

We would like to thank Teleinformatics Research Group, CNPq, IFPB, UFCG and CAPES for supporting this research.

REFERENCES

[1] N. Martin. Top 10 virtualization companies emerging in 2014. Search Server Virtualization, Jan. 2014, [Online; accessed 15-Oct-2014] Available: http://searchservervirtualization.techtarget.com/photostory/2240211913/Top-10-virtualization-companies-emerging-in-2014/1/Emerging-virtualization-companies-poised-to-disrupt-the-market.

[2] Trefis. Growing Competition For VMware In Virtualization Market. Nasdaq.com, Jan, 2014, [Online; accessed 09-Jan-2014] Available: http://www.nasdaq.com/article/growing-competition-for-vmware-in-virtualization-market-cm316783.

[3] David Thomason. Server Virtualization Market 2012-2016. Before It's News, Apr, 2013, [Online; accessed 04-Nov-2014] Available: http://beforeitsnews.com/science-and-technology/2013/04/server-virtualization-market-2012-2015-2573912.html,.

[4] Sharada Bose, Priti Mishra, Priya Sethuraman, and H. Reza Taheri. Benchmarking database performance in a virtual environment. In *TPCTC*. LCNS, v. 5895, pages 167–182, 2009.

[5] Apache Cassandra Database. "http://cassandra.apache.org/", [Online; accessed 02-Oct-2014]

[6] MongoDB Database, "http://www.mongodb.org/", [Online; accessed 10-Oct-2014]

[7] http://planetcassandra.org/nosql-performance-benchmarks/, acessado em 14/09/2014.

[8] J. van der Veen, B. van der Waaij, and R. Meijer, Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual. In *Proc. of the 5th International Conference on Cloud Computing (CLOUD)*, pp. 431–438, IEEE, 2012.

[9] Bezerra, P. C., Gomes, R. C. M., Comparing Performance Overhead of Virtual Machine Monitors. In IADIS 11th International Conference on Applied Computing, 2014, Porto - Portugal. In *Proceedings of the IADIS International Conference on Applied Computing*, 2014. pp. 206-213.

[10] Camargos, F. L. et al, 2008. Virtualization of Linux servers: a comparative study. In *2008 Ottawa Linux Symsposium*, pp. 63-76.

[11] Domingues, P. et al, 2009. Evaluating the performance and intrusiveness of virtual machines for desktop grid computing. In *Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing (IPDPS '09)*. IEEE Computer Society, Washington, DC, USA, pp. 1-8.

[12] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. 2010. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing (SoCC '10)*. ACM, New York, NY, USA, 143-154.

[13] Minitab Statistical Software, "http://www.minitab.com/", [Online; accessed 08-Dec-2014]

[14] Y. Abubakar, T. S. Adeyi, and I. G. Auta, Performance evaluation of NoSQL systems using YCSB in a resource austere environment. *Performance Evaluation*, vol. 7, no. 8, 2014.

[15] John Klein, Ian Gorton, Neil Ernst, Patrick Donohoe, Kim Pham, and Chrisjan Matser. 2015. Performance Evaluation of NoSQL Databases: A Case Study. In *Proceedings of the 1st Workshop on Performance Analysis of Big Data Systems (PABS '15)*. ACM, New York, NY, USA, 5-10.

[16] Jain, R. 1991. The Art of Computer Systems Performance Analysis; Techniques for Experimental Design, Measumerement, Simulation and Modeling. Wiley professional computing. Wiley.

An adaptive approach for real-time communication of multi-robots based on HLA

Rivaldo Simo, Leandro Henrique S. Santos, Alisson V. Brito
Programa de Ps-Graduao em Informtica - PPGI
Universidade Federal da Paraiba - UFPB
João Pessoa, Brazil
Email: rsym95@gmail.com, leandrohenriquecomp@gmail.com,
alisson@ci.ufpb.br

***Abstract*—Robot soccer is one example of a real-time multi-robot system. The cooperation among them is one of the challenges like to be overcome. Distributed approaches for local state exchanging like Real-time Databases (RTDB) are CPU demanding and complex to write and maintain. Thus, this work presents a new communication approach based on a centralized middleware named High-Level Architecture (HLA), where the robots adapt their transmission rate based on the performance of the other robots. At same time, we investigate the benefits and impacts of using HLA for real-time applications. Experiments demonstrated that all real-time requirements of a robot soccer application were reached using this approach, pointing to a new possibility for real-time communications between robots.**

Keywords*—computer network, synchronization, real-time systems

I. Introduction

Amongst all technological changing, an area that deserves attention and has been increasing quickly more and more, is the applications with robots. For example: industrial automation, medicine, search and rescue, urban monitoring, bomb disposal, mine clearance, space exploration, etc.

The advances in robotic are allowing researchers and engineers design systems that can interact with humans and among themselves cooperatively [1]. The interaction among several robots in real time, which is typical of robotic applications, is not an easy task, as it is becoming increasingly complex due to the treatments that should be given in the communication process.

According to [2], such applications require data delivering on time and a simple delay may constitute a catastrophic injury, like in controlling of nuclear power plants. These requirements constitute a major challenge for researchers because the protocol that provides the interaction should also be able to deliver only relevant information to robots; therefore, it must delivery only integrated data and on time [3].

In order to encourage the development of applications for robots, the Robocup launched a challenge which aims to build a team of 11 robots operating autonomously, able to win the world champion soccer team in 2050 [4]. The ultimate goal of this incentive is the development of relevant solutions for society, while working for soccer problems.

978-1-4673-9408-6/15/$31.00

As a result of these demands, some projects dedicate their efforts to build communication infrastructures in real time in order to satisfy the requirements of such systems. In [5] the authors present an architecture for cooperation of mobile systems in real time, where communication and reliability are the main goals.

Another is the RTDB (Real-Time Database) middleware. Created by CAMBADA project [6], it provides the infrastructure used by the robot soccer team from the University of Aveiro. The middleware provides a distributed shared memory model, where each team member, called agent receives the local copy of the states of all other agents at each control loop [7]. Some of the challenges faced by the creators of the project are: ”How to assign different tasks to a variety of robots?”, ”How to control the movement in a formation of robots?”, ”What information must the robots change to enable coordination and cooperation between them?”.

The RTDB uses UDP, therefore, becomes likely to lose many messages, as well as having a limit on the amount of robots, because the scenery would be in a difficult situation when it exceeds 6 machines in a same network [8]. Loss of packets are tolerable until reach a limit, because there is plenty of redundancy between the robots in the communication process. For example, if a robot B leaves the coverage area, a robot A may not get the state robot B directly. However, a robot C may have stored an old state of B and send it to A.

In order to mitigate these limitations, this paper explores the use of another communication approach, based on the High-Level Architecture (HLA), a middleware mostly used for Parallel and Distributed Simulation (PADS) [9]. The HLA has a centralized architecture where no message is lost. It is based on TCP and may organize the synchronization of messages. But due to these features (TCP acknowledgment, flow control, synchronization), it tends to add delay to the messages [10]. Thus, this work presents a new communication approach based on HLA, where a robot adapt its transmission rate based on the performance of the other robots. At same time, we investigate the benefits and the impact of using HLA for real-time applications, like multi-robot communication. Experiments demonstrated that all real-time requirements of a robot soccer application were reached using this approach, pointing to a new possibility for real-time communications between robots.

In this context, this paper presents related works in sections

II, and a brief explanation of HLA in the section III. The configuration of the experiments are presented in section IV. In the section V is presented the performance analysis of the HLA in a multi-robot scenery. Finally, in the section VI the adaptive algorithm and their results are presented, followed by final considerations in the section VII.

II. Related Work

There are several middlewares that present themselves as a communication solutions for multi-robot systems. However, as there is no standardization, and each project builds its own middleware designed to solve the needs of a specific team or project. Thus, the fact that there are many middlewares makes the literature even wider. On the other hand, it is even farther to reach a single solution that covers all the requirements of an ideal middleware for communication of these systems.

Seeking to raise the particularities of the various existing middleware, draw what these projects are already able to provide with good functionality and what they still cannot offer, in [11] [12] [13] are presented various middleware projects designed to robot communication. Among the many outstanding features, usability, extensibility, connectivity and security are highlighted. These are some of the items noticed and evaluated by the authors in relation to the investigated middleware.

In this evaluation list are the Miro [14] and the RT-middleware [15], both are based on CORBA, which consists of a standard architecture created by the Object Management Group [16] and utilized by distributed systems. Although CORBA is fairly complete, possessing various services, its implementation generate a lot of complexity that makes it extensive. In this way, the usability, which is a very important issue to be considered in these middleware, is reduced. [17]. In this same work, other middlewares are presented and analyzed, such as UPnP Robot, Player / Stage, ESRP Kernel, MARIE, RSCA, AWARE, Sensory Data Processing and Middleware Layer for Incorporation. After presenting the characteristics of the briefly studied middleware, the authors identify and point out some issues to be covered such as: security, advanced integration capabilities and high-level abstraction of hardware and software.

Another work worth mentioning is the work of [18]. In this work, characteristics and some evaluation criteria as: Security, Dynamic wiring, Distributed environment, Real time, Behavior coordination, standards, technologies and operating system support are shown. The authors present not only middlewares for robots networks, but also how the robots from [11] and others, like Orocos, Pyro, Orca [19] and others.

Our proposal aims to present an attempt of using HLA presented in section III for communication of teams of robots. Works have demonstrated the capacity of HLA for real-time and distributed systems communication [20] [21] [22], turning it into appropriate for robot soccer. As HLA is based on a centralized approach, it does not result into complexes codes running in each robot, as happens in CORBA [13]. Another reason for using HLA is that is based on international IEEE 1516 standard [23].

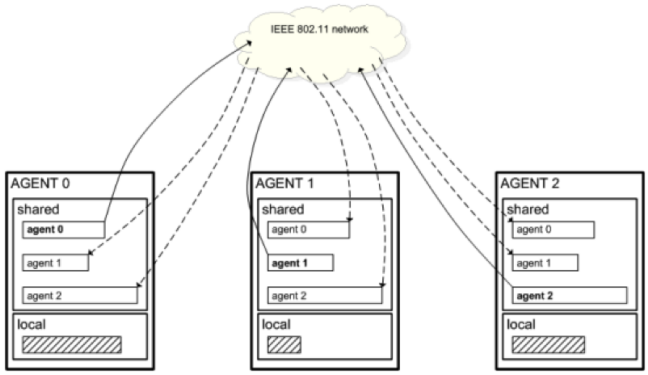


Fig. 1. Structure of a RTDB network.

A. Real Time Database (RTDB)

A real-time system may be characterized as any information system with a basic requirement of responding to a request within a certain period of time. The positive result of the transaction does not only depend on the successful delivery of information, but mostly whether the time requirements were obeyed [24].

In [5] is presented an architecture for cooperation of mobile systems in real time reliably. This approach guarantees real-time requirements for applications using the same wireless channel, which is error-prone and unreliable. The Cooperative Autonomous Mobile Robots with Advanced Distributed Architecture (CAMBADA) has developed a middleware called Real Time Database (RTDB) that captures the states of the data of each agent shared by other members of the soccer team as informs [8]. Regarding the robot soccer, the information within the RTDB are composed by the states of all the players, as well as the position of the ball.

Once this information is shared, each robot locally guards your necessary data for communication along with data from other team members. A robot can use up the information of other robots to complement its own database. An example would be a situation in which a robot somehow loses for a specified period the position of the ball. In this case it may use the position of the ball that another robot shared. In RTDB, communication between members using this middleware is performed by IEEE 802.11 network using an Access Point (AP), as shown in Figure 1, which is a limitation since the communication using wireless networks has lots of packet losses, and the coverage area provided by the AP is limited.

As presented in [25], for robot soccer teams, communication is crucial, given the need to update the states of all robots. Therefore, this update must be done for better coordination of teams, ie. roles exchanging, strategies and major decisions inherent in the ongoing process of robot soccer game.

To analyze the communication process of a robot soccer match, [25] points out that this period between updates, in the case of Aveiro soccer team, using the RTDB, is 100ms. This time is divided by K team members to send their states using TDMA.

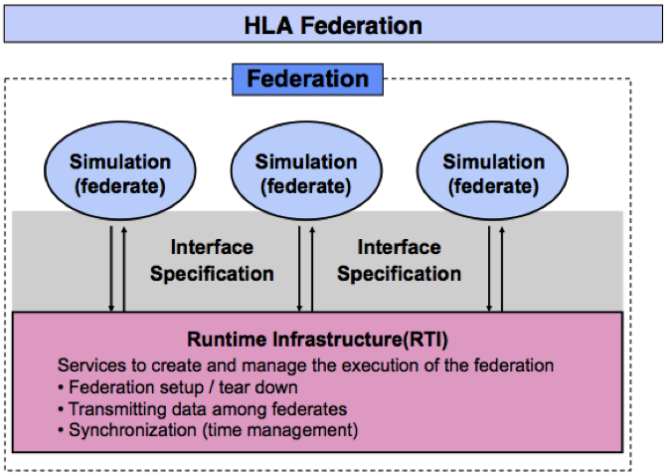


Fig. 2. Architecture of HLA Federation

III. High-Level Architecture (HLA)

High-Level Architecture (HLA) is a general-purpose architecture defined by Defence Modelling and Simulation Office (DMSO) to support reusability and interoperability using a wide number of different types of simulators, which are maintained by U.S Department of Defense (DoD) [23]. The standard defines this architecture having three parts: the first explore the framework in a general way and its main rules, the second deals with the interface specifications between simulators (HLA), and the third deals with the specification model of data (OMT) which are transferred between simulators. The HLA has a time managing service to synchronize data between heterogeneous models. The mains goals of HLA are to make possible the interoperation of distinct models and reuse them when necessary to provide a distributed simulation environment for systems that need large scale computing.

The principal idea of HLA is to separate the specifics functionality of each simulator using a general proposal infrastructure (see Figure 2). Each simulator needs to use the Runtime Infrastructure (RTI) to communicate to HLA and others simulators. The RTI is responsible for the specific structures of each simulator to interface with the global structure of HLA. Each simulator that is connected to one RTI is called a Federate. The set of all federates managed by one RTI is called a Federation. In cases of geographically distributed simulations, it is also possible to have many Federations in a same simulation environment.

IV. Experiments

For the experiments, the open-source implementation of HLA, Certi (http://savannah.nongnu.org/projects/certi) was used. Computers were used to emulate, having different configurations, such as notebooks, netbooks and a Raspberry Pi. The goal is to study the behavior of HLA running different configurations and the impact to performance.

A. Equipments

In Table I is possible to see all computers used during the experiments.

TABLE I. Equipments used in the experiments

Equipment	Configuration
Notebook HP	Intel i5-3220M with 8GB DDR-SDRAM
Notebook Dell	Intel i7-4510U with 4GB DDR-SDRAM
Notebook IBM	Intel i5-2210M with 4GB DDR-SDRAM
Netbook Acer	AMD C-50 with 2GB DDR-SDRAM
Netbook Positivo	Atom N2600 with 2GB DDR-SDRAM
Raspberry Pi	AMR BCM2708 with 512MB DDR-SDRAM
Access Point	Dlink DSL 2730R
Switch	3Com 3C17561-91 with 26 ports

TABLE II. Scenarios with objectives

Group 1		
Scenario	Computers	Objectives
1	3 notebooks	Testing HLA with high performance computers
2	2 notebooks + 1 Raspberry	Testing the performance when a low performance computer is joint to high performance ones
3	2 netbooks + 1 Raspberry	Testing the performance when a low performance computer is joint to high performance ones
Group 2		
4	2 netbooks + 3 notebooks	Testing HLA with high performance computers mixed with middle performance ones
5	3 notebooks + 1 Raspberry + 1 netbook	Testing the performance when using a heterogeneous configuration

B. Scenarios

Tests were performed on five different scenarios in order to analyze the operation of HLA in different situations. These scenarios are organized into two groups to facilitate analysis. Thus, the first group is formed by the set of three robots with different characteristics. The second group is formed by the set of five robots. The scenarios and their objectives can best be seen in Table II.

C. Configuration

At first, all robots were interconnected in a network via UTP cable using a switch. Thus, computers were configured so that the highest performance computer also plays the role of RTI Gateway (RTIG), centralizing and synchronizing all communication. The other robots can only send their positions after they join the Federation, which is held after a request made to RTIG. Only after that the communication starts.

The operation of Certi was adapted to our experiment so that the sequence of steps after connection of all the robots is repeated in a cycle of 10,000 iterations, as shown in Figure 3.

To perform the experiments, a short code was run in every robot (see Listing 1. At each iteration, the robot update its position (simulating a movement) and send it in a message to RTIG, which will eventually dispatch it to the other robots enrolled in the federation. After that the robot asks do advance its logical time (synchronization approach). This method is blocking and makes the robot wait until the RTIG reply authorizing the logical time to advance. This will guarantee a synchronous communication among the robots.

In our tests, synchronous and asynchronous communications were performed and evaluated. Furthermore, in order

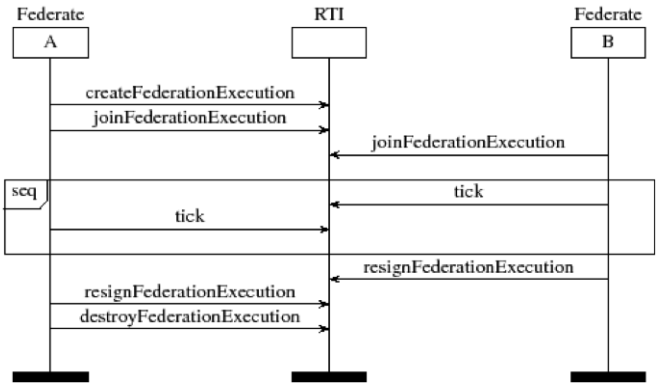


Fig. 3. Sequence of operations

Listing 1. Logical loop of robots.

```
1
2  // Call me main loop and set the iteration number
3  loop(10000, objectHandle);
4
5  // Main loop
6  void loop (int iteractions, RTI::ObjectHandle objectHandle) {
7          int i;
8          for (i=1; i < iteractions; i++ ) {
9          setPosX(getPosX() + 1);
10         setPosY(getPosY() + 1);
11
12         // Send positions to RTIG
13         updateAttributeValues(objectHandle);
14
15         // Ask RTIG to send the next message
16         advanceTime( 1.0 );
17         }
18 }
19
20 // Waiting RTIG to release the next message
21 void advanceTime (double timestep) {
22         while ( fedamb->isAdvancing ) {
23         sleep(100);
24         rtiamb->tick();
25         }
26 }
```

to establish a more realistic measure of the experiment, each scenario was repeated 5 times and the results were represented by the total time necessary to send 10,000 messages.

Upon completion of the 10,000 iterations, the robots are able to disconnect from federation and finish their executions. Once finished running, messages are exchanged only between the robots that are still active in the federation until all finish and the federation is destroyed.

V. Analysis of performance with HLA

In this section are presented the results of two different configuration, synchronous and assynchronous.

A. Results using synchronous communication

In synchronous mode, or conservative [9], HLA works in order to maintain all the robots at the same logical time, causing the robot to act always slower to guarantee that the number of messages sent by all robots be the same.

TABLE III. Results with synchronous communication

Scenario	Total time (sec.)	Period between messages (ms)	Sent message in 100ms
1	15,81	1,58	63
2	126,4	12,64	7
3	137,8	13,78	7
4	63,8	6,38	15
5	188,9	18,89	5

For this, the advance time routine becomes blocking and the grant to advance time (and send message) is sent by RTIG only when all robots reach the same barrier.

This operating method is more appropriate when there is a requisite to keep all robots synchronized, independent on the performance of each machine. In the first scenario, all robots are composed of computers with high processing power, so no one becomes a bottleneck. Nevertheless, scenario 4 is also made up of robots with high processing power, which makes the whole system works at that pace, yet there is no significant bottleneck.

However, this restriction inherent in synchronous algorithms scenarios becomes prohibitive for soccer robots, since the processing time required for forming the message may vary greatly depending on various factors such as occlusion amount of objects to be processed , outside interference among others.

For example, the results of scenarios 2, 3 and 5 from Table III show that performance were strongly affected by the slowest computer (ie. Raspberry Pi). This becomes more evident when scenarios 1 and 2 are compared. The simulation became 8 times slower just because the Raspberry was inserted to the configuration.

Nevertheless the synchronous algorithm version used here is much simpler than that used by RTDB, it can be observed in the reduced interval between two messages found in all the experiments performed here. As can be seen in Table III, the worst case scenario with 5 robots, including the Raspberry PI, obtained a period of 18.89 ms between messages, which is far below the particular restriction for robot soccer, which is 100 ms. This demonstrates that the HLA has a great potential for critical real-time applications, like soccer robot.

B. Results with asynchronous HLA

In asynchronous or optimistic synchronization [9], HLA does not avoid any Federate to advance its local time. Thus, all robots are authorized to send all your messages when requested. This method has the major advantage of not being vulnerable to low performance of one of the robots in the network. Then, all the other robots will be able to work with their maximum performance throughout the execution of the application.

In order to measure that effect in a multi-robot system, scenario 3 was repeat, because it has the lowest performance used computer, the Raspberry Pi. Thus, as can be seen in Table IV, the total time of simulation was reduced from 137 seconds to only 61 seconds, showing the asynchronous

TABLE IV. RESULTS WITH ASYNCHRONOUS COMMUNICATION

Total time (sec.)	Netbook 1	Netbook 2	Raspberry Pi
61,2	Messages: 10,000 Period: 6.12 ms	Messages: 8,780 Period: 6.97 ms	Messages: 4 Period: 15,300 ms

algorithm increased the overall performance of the system by more than 2 times.

On the other hand, it is important to observe that the asynchronous limited the number of messages that the Raspberry Pi computer was able to send. Just 4 in comparison with 10,000 of Netbook 1 and 8,780 of Netbook 2. In practice, it means reduction of accuracy. The robots controlled by Netbook 1 and 2 would be able to send 10,000 and 8,780 new positions, respectively, while the robot controlled by the Raspberry Pi would send only 4 positions. At the same way, a period of two messages achieved by the Raspberry Pi was 15,300 ms, too far away from the requirement of maximum 100 ms of soccer robots. This lack of accuracy was reduced by a novel adaptive algorithm presented in the next section.

VI. THE ADAPTIVE ALGORITHM

After the study HLA behavior presented in this paper, an adaptive algorithm was developed in order to better manage the flow of messages transmitted between the robots, allowing everyone send messages within the established limit for the soccer robots and avoiding the entire system to be limited to robots who are acting much slower.

To verify the efficiency of this algorithm, scenario 3 was again used, however, all robots started to be connected through a wireless network, thus generating a more realistic environment and hence with more delays involved. Then, a performance comparison with the regular asynchronous model of HLA was executed.

A. Basic concept

In this algorithm, the messages are grouped into cycles, which is a fixed number of messages exchanged between the robots. When all robots finish one cycle, each one must calculate your speed relative to the slower robot during that cycle. This value is called *delay* and is calculated as follows:

$$delay = \frac{myIterationPerCycle}{lowestIterationPerCycle} \quad (1)$$

The variables *myIterationPerCycle* and *lowestIterationPerCycle* are, respectively, how many iterations the robot itself run during that cycle, and same number for the slowest robot. After this calculation is performed, iteration values (*myIterationPerCycle* and *lowestIterationPerCycle*) are reset and the cycle restarts, which will occur throughout the process of execution of the application dynamically and adaptively. Thus, after the delays are calculated, it is possible to determine how quickly each robot is compared to the slowest one, and adapt its internal cycle speed for the next cycle according to a predefined damping parameter (*damping_value*) as can be seen following:

Listing 2. Logical loop of robots.

```
void calculateNextDelay() {
    // Calculate delay
    float lipc = (float)lowestIterationPerCycle;
    float delay = myIterationPerCycle/lipc;

        // Calculate delay next cycle
        float delayNextCycle = delay/DAMPING_VALUE;

        // Check minimum delay
        if (delayNextCycle < MINIMUM_DELAY) {
        delayNextCycle = MINIMUM_DELAY;
        }
}
```

$$delayNextCycle = \frac{delay}{damping\_value} \quad (2)$$

The variable *delayNextCycle* means how many iterations the robot will run in the next cycle without sending messages. For example, if *delayNextCycle* is 3, the robot will run 3 iterations without updating its status to the other robots, sending it just in the fourth iteration. The *damping_value* represents the speed reduction factor that is applied to a robot when it realize it is running faster than the other.

Through the tests, it was found that a satisfactory value for the number of iterations required to restart the cycle is 500 iterations. Thus, during the 10,000 sent messages in the experiments, there were 20 cycles in total. Furthermore, it was found that a suitable value for *damping_value* was 2. In Figure 5, it can be seen how penalized was the fastest robot to avoid flooding the robots that are acting slower.

To prevent robots which operate slowly over a long time to reduce the performance of the other robots, a minimum value in the calculation of the delay was added. This limiting value is predetermined and, to our tests, the minimum threshold of 20% was used. So, no matter how many cycles a robot remains much slower than the others, the delay ratio will not exceed the ratio 8:10. In Listing 2 it can be seen part of the calculation of the delay used in the next cycle.

B. Experiments and results

For this experiment, some improvements were made and the experiments with synchronous and asynchronous communications were repeated. Here, a delay of 5.5 milliseconds was add to the control loop of each robot. This extra time helps to avoid a too fast robot to over flood the over robots with messages.

By repeating the Scenario 3, now using a wi-fi network, it was found that the slowest robot (Raspberry Pi) could only send 59 messages during the first cycle, as can be seen in Figure 4 (see first cycle). This value is almost 10 times less than the total messages sent by the fastest robot. With this message flooding, the slowest robot get into starvation.

This is, so far, the major impediment in the use of HLA in their standard form for real-time robot applications. In case

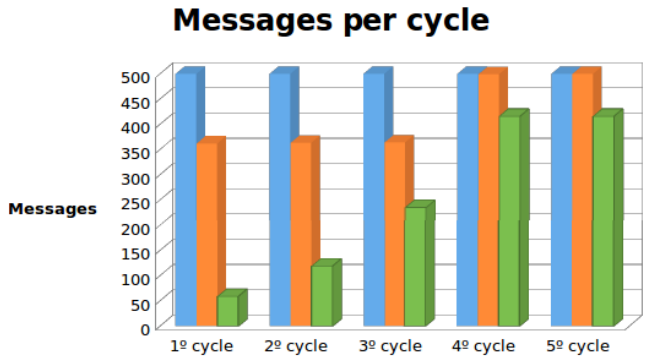


Fig. 4. Messages per cycle with adaptive algorithm

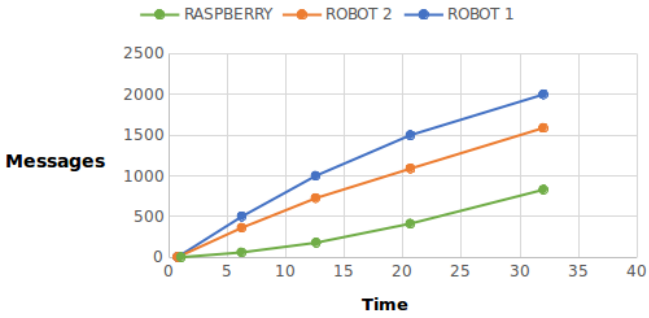


Fig. 5. Messages per cycle with adaptive algorithm

of working asynchronously, the behavior seen in Figure 4 for the first cycle is propagated to the rest of the cycles. On the other hand, in a synchronous manner, all robots will have to reduce their sending rate by almost 10 times in order to equalize with slowest robot, resulting in a general degradation of performance.

Thus, the algorithm presented here reduces these limitations making robots gradually and automatically adapt their updating rate focusing a better configuration for the whole system. As can be seen Figure 4 in the second cycle the slowest robot now has the opportunity to send two time more messages than sent in previous cycle. The same is repeated for the third cycle until achieving a minimum of the a pre-defined 80% of messages of the fastest robot.

The adaptation of message updating frequency can be better seen in Figure 5. Here is possible to see the decreasing of message exchanging rate of Robots 1 and 2. Moreover, with the reduced number of messages traveling in the network, the slowest robot is not overloaded and had time process all received messages and thereby increased its own sent message rate.

In order to verify the real gain of the developed adaptive algorithm in relation to the others, the same experiment was carried out with synchronous and asynchronous versions. The main relevant requirements for a robot soccer application were measured and presented in Table V.

Through these results, you can see that the HLA Asynchronous has the fastest total execution time (113ms). However, the fastest robots now have a high message period while the slowest robots are overloaded with messages and can only send at one message at each 95.3ms, which can easily exceed the limit for soccer robots (100ms) when adding a fourth or fifth robot.

TABLE V. COMPARING ADAPTIVE APPROACH WITH SYNCHRONOUS AND ASYNCHRONOUS

	Synchronous	Asynchronous	Adaptive
Total time (sec.)	311.77	113.09	219.75
Exchanged messages	30,000	18,334	27,703
Message period of slowest robot (ms)	31.77	95.35	29.26

On the other hand, HLA synchronous has as main advantage all robot are able to send all 30,000 messages during the simulation. In addition, it presents a significant improvement in message period of slowest robot (one message at each 31.77ms). However, the main disadvantage of this method is the total application execution time, which happens to be almost 3 times slower than that found in the first model, featuring a large degradation in overall system operation. In practice, it means all robots will move slowly, because they are not able to update their internal states to the other robots faster.

Finally, the adaptive algorithm presented achieved the best of two worlds. It has a better trade-off between the amount of messages sent and the total execution time. The execution time (219.75 seconds) has been reduced by 30% compared to model synchronous HLA and the total number of messages still remained at a very high value (27,703 messages). In addition, there was a gain in the period of messages sent by the slowest robot. With adaptive approach, the slowest robot sent one message at each 29.26ms, which represents 8% and 69% more than the synchronous and asynchronous approaches, respectively. This metric is very important because it is the main requirement for real-time applications such as robot soccer. Also, now the communication does not limit the fastest robots.

VII. FINAL CONSIDERATIONS

After analysis of HLA usage and experiments focusing on a robot soccer application, it is possible to say that the developed approach presented in this work got a satisfactory result. The slowest robots do not degraded the whole robots performance (like in synchronous approach), there was not a high difference between the number of messages sent by each robot, at same time, the requisite of sending at least one message at each 100ms was reached and exceeded in 3 times.

The fact of being adaptive also brought other benefits. Now, the robots will dynamically adapt their message frequencies based on the slowest machines. It is important because in real applications the performance is variable, new robots can join a team, while some others are removed. Our approach promises to face this fact very well.

For future studies it is expected to improve the developed approach to adapt to other real-time applications requirements, e.g. response time, bandwidth, etc. Furthermore, the algorithm will be tested in error prone channels to test the vulnerabilities to package losses. Other communication middleware different

from HLA can also be explored, like MPI, MPICH, OpenMP etc.

REFERENCES

[1] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Network robot systems," *Robot. Auton. Syst.*, vol. 56, no. 10, pp. 793–797, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2008.06.007

[2] J.-B. Chaudron, M. Adelantado, E. Noulard, and P. Siron, "Hla high performance and real-time simulation studies with certi," in *25th European Simulation and Modelling Conference- ESM'2011*, Guimaraes, Portugal, 2011. [Online]. Available: http://oatao.univ-toulouse.fr/4972/

[3] M. F. Martins, F. Tonidandel, and R. A. Bianchi, "Um protocolo confiável e flexível de comunicação para futebol de robôs," *IV Latin American IEEE Student Robotics Competition*, 2005.

[4] H.-D. Burkhard, D. Duhaut, M. Fujita, P. Lima, R. Murphy, and R. Rojas, "The road to robocup 2050," *Robotics Automation Magazine, IEEE*, vol. 9, no. 2, pp. 31–38, Jun 2002.

[5] E. Nett and S. Schemmer, "Reliable real-time communication in cooperative mobile applications," *Computers, IEEE Transactions on*, vol. 52, no. 2, pp. 166–180, Feb 2003.

[6] L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, and L. Lopes, "Coordinating distributed autonomous agents with a real-time database: The cambada project," in *Computer and Information Sciences - ISCIS 2004*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3280, pp. 876–886. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30182-0-88

[7] A. Neves, A. Pinho, A. Pereira, B. Cunha, D. Martins, F. Santos, G. Corrente, J. Rodrigues, J. Silva, J. Azevedo *et al.*, *CAMBADA Soccer Team: from Robot Architecture to Multiagent Coordination*. INTECH Open Access Publisher, 2010.

[8] N. Figueiredo, A. Neves, N. Lau, A. Pereira, and G. Corrente, "Control and monitoring of a robotic soccer team: The base station application," in *Progress in Artificial Intelligence*, ser. Lecture Notes in Computer Science, L. Lopes, N. Lau, P. Mariano, and L. Rocha, Eds. Springer Berlin Heidelberg, 2009, vol. 5816, pp. 299–309. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04686-5-25

[9] R. M. Fujimoto, *Parallel and Distribution Simulation Systems*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[10] J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly, "The department of defense high level architecture," in *Proceedings of the 29th Conference on Winter Simulation*, ser. WSC '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 142–149. [Online]. Available: http://dx.doi.org/10.1145/268437.268465

[11] N. Mohamed and J. Al-Jaroodi, "Characteristics of middleware for networked collaborative robots," in *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*, May 2008, pp. 524–531.

[12] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "A review of middleware for networked robots," *International Journal of Computer Science and Network Security*, vol. 9, no. 5, pp. 139–148, 2009.

[13] ——, "Middleware for robotics: A survey," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, Sept 2008, pp. 736–742.

[14] S. Enderle, H. Utz, S. Sablatng, S. Simon, G. Kraetzschmar, and G. Palm, "Miro: Middleware for autonomous mobile robots," in *In Telematics Applications in Automation and Robotics*, 2001.

[15] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "Rt-middleware: distributed component middleware for rt (robot technology)," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug 2005, pp. 3933–3938.

[16] O. M. Group, *The Common Object Request Broker (CORBA): Architecture and Specification*. Object Management Group, 1995.

[17] M. Namoshe, N. Tlale, C. Kumile, and G. Bright, "Open middleware for robotics," in *Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008. 15th International Conference on*, Dec 2008, pp. 189–194.

[18] A. Elkady and T. Sobh, "Robotics middleware: A comprehensive literature survey and attribute-based bibliography," *Journal of Robotics*, vol. 2012, 2012.

[19] A. Makarenko and A. Brooks, "Orca: Components for robotics," in *In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06*, 2006.

[20] A. Boukerche, A. Shadid, and M. Zhang, "Efficient load balancing schemes for large-scale real-time hla/rti based distributed simulations," in *Distributed Simulation and Real-Time Applications, 2007. DS-RT 2007. 11th IEEE International Symposium*, Oct 2007, pp. 103–112.

[21] H. Zhao and N. D. Georganas, "Hla real-time extension," in *Distributed Simulation and Real-Time Applications, Fifth IEEE International Workshop on, DS-RT 2001.*, Aug 2001, pp. 12–21.

[22] C. Gervais, J. Chaudron, P. Siron, R. Leconte, and D. Saussie, "Real-time distributed aircraft simulation through hla," in *Distributed Simulation and Real Time Applications (DS-RT), 2012 IEEE/ACM 16th International Symposium on*, Oct 2012, pp. 251–254.

[23] IEEE, "Ieee standard for modeling and simulation - high level architecture (hla)– framework and rules," *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)*, pp. 1–38, Aug 2010.

[24] S. Malik and F. Huet, "Adaptive fault tolerance in real time cloud computing," in *Services (SERVICES), 2011 IEEE World Congress on*, July 2011, pp. 280–287.

[25] F. Santos, L. Almeida, P. Pedreiras, and L. Lopes, "A real-time distributed software infrastructure for cooperating mobile autonomous robots," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, June 2009, pp. 1–6.

MUV-Bee: Using WSN to Monitoring Urban Vehicles

Iury Araujo
Informatics Center
Federal University of Paraíba
João Pessoa, Brazil
Email:iuryrogerio@gmail.com

Jessica Castro
Informatics Center
Federal University of Paraíba
João Pessoa, Brazil
Email:jesscmaciel@gmail.com

Fernando Matos
Informatics Center
Federal University of Paraíba
João Pessoa, Brazil
Email:fernando@ci.ufpb.br

Eudisley Anjos
Informatics Center
Federal University of Paraíba
João Pessoa, Brazil
Email:eudisley@ci.ufpb.br

***Abstract*—The number of urban vehicles has increased in the past years causing innumerous problems such as traffic jams, accidents and pollution. One of the main solutions for this problem is to improve the quality of public transportation by making it more reliable, secure and efficient. In this paper, it is proposed a system for monitoring urban vehicles based on ZigBee technology, called MUV-Bee. Such system uses wireless communication to integrate the main server to public transportation buses and bus stops. Its several services provided to society allow a better quality of public transportation which make them more attractive.**

Keywords - intelligent transportation system, ZigBee, bus monitoring.

I. Introduction

Public transportation systems can sometimes be unattractive to the passengers [1]. This happens due to many reasons, mostly, safety problems and delays. As a result, many people have chosen to have their own vehicles [2]. The increase in the number of vehicles leads to a serie of problems such as traffic jams, accidents and air pollution [3].

Transportation cooperatives have made many efforts to solve these problems and make public transportation more attractive [4]. One of the best solution, adopted by several systems, is to use an intelligent transportation system (ITS) to offer the passenger new services, as the arrival time and occupancy estimation. In the town of Nis, Serbia, for example, an ITS was implemented using GPS and GPRS in oder to monitor buses and offer services to the passengers [5]. Another example is the implementation of an ITS to the Beijing public transportation for the 2008 olympic games aiming at improving the transportation quality for tourists [6].

Numerous researches have implemented ITS using GPS [5] [6] or RFID [7] [8] [9] [10] [4] [11] as the main technology to gather the buses location, however some obstacles were observed by them. The GPS is an exact and precise method for determining vehicle position using a 24-satellite cluster, nevertheless the presence of obstacles(buildings, trees, tunnels) can cause the loss of signal [12]. The RFID is a radio-frequency technology, frequently used in studies about identification and monitoring due to its low-cost. On the other hand, there is a major problem that RFID can suffer interference from others dispositives [13]. Recent works have been made using ZigBee as an alternative for these technologies as in [14] [15] [16]. However most of them have focused on implementing an ITS system with basic arrival time predictions, wasting the ZigBee power and the possibility of a more scalable architecture, allowing the coupling of existing systems and new features.

This work presents the MUV-Bee, a system for monitoring urban vehicles using wireless sensor networks (WSN). The system is based on ZigBee Protocol, a wireless communication technology whose main characteristics are low-power, low-cost and low-complexity [14]. Besides the advantages of ZigBee devices, the architecture of MUV-Bee is modular and extendible allowing the coupling of other technologies, such as climate monitoring, vehicle automation or even animal behavior monitoring [17]. Currently, the city hall of Joo Pessoa - Brazil is making a partnership with this project to have tests with the first MUV-Bee prototypes to ever be made in the city be initiated.

Subsequently, a short review of related researches is provided (Section II). After that, descriptions of the proposed system, architecture and services are laid out in Section III. Thereafter, Section IV concludes with the main aspects of this proposal and future researches.

II. Related Research

Bus monitoring systems has grown and several technologies have been proposed [18]. Depending on each project requirement, those technology are set together to obtain the best architecture. For example, in developing countries, some considerations must be taken, such as: the public transport is unsafe, the waiting time for the bus is uncertain, since it is not easy to follow the schedule along the day due the traffic status. Those conditions influence the decisions about the project [19].

RFID technology is one of the most used for vehicular monitoring due to its cost-effective. In [7] [8] [9] [10] [4] [11], RFID is used for identification and consists in RFID tags and readers. The location system uses GPS. In the work proposed in [7] [8] [9] [10] [4],the RFID tag is located in the bus stop and a RFID reader in the bus. Thus, every time the bus get close to a bus stop, the RFID reader gets in contact with the RFID tag and the bus sends a signal to a server with the location required from the GPS. The server uses the location

978-1-4673-9408-6/15/$31.00

data to predict the time to the next bus stop. Moreover, [11] proposes a system where the RFID reader is in the bus stop and the RFID tag is in the bus, so it works similar to the previous one, except that data is sent to the server by the bus stop.

An alternative for RFID has been the ZigBee protocol.The main reasons leading to the use of ZigBee are its low-cost of acquisition and low-power consumption [14]. Almost all works have applied ZigBee in the same way: each bus and bus stop has its own ZigBee module, and the modules find each other and establish a connection for communication. [14] [15] [16]. The connection has a verification process to assure the security of the information [14]. The location of the bus can be estimated using GPS, as proposed by [16], but [14] [15] have shown that it is unnecessary, once the system stores all the bus stops that the bus pass by, when it is known in which bus stop the bus is, it is known its location. Besides location and estimated arrival time, the use of ZigBee allows the configuration of several sensors, such as: temperature (inside and outside the bus), passengers estimation, level of pollution and so forth. In [15] [16] they monitor the environment using various types of sensors, for example: in both infrared sensor is used to identify passenger volume.

III. MUV-BEE

This Section presents the general architecture of MUV-Bee (Monitoring Urban Vehicles), detailing the networks aspects and architectural layers, including communication, sensors and services.

A. MUV-Bee Architecture

The objective of MUV-Bee is collect data from buses and use them data to provide useful information to users, such as arrival time and number of passengers on the bus, so the user can choose to wait for the next bus in case the closest one is crowded. These pieces of information are made available to users by the means of services offered through mobile applications or a website.

Inside the bus a microcontroller equipped with severals sensors will be coupled to gather data, which has the task of being in control of the sensors and send the data to the server. Moreover it is possible to expand the system adding new sensors to gather other pieces of information. The sensors works separated to gather information from the environment with exception of the ZigBee module. ZigBee collects data to infer the locations of a bus during a pre-established route. Each bus stop equipped with a ZigBee module has its unique code and can communicate with the ZigBee on the bus. When a bus arrive at a bus stop the ZigBee module in the bus receives data from the ZigBee module on the bus stop. This data contain the identification of that bus stop, and send a message to the server identifying the bus and the bus stop.

Sending the collected data by the sensors to the server is necessary to use a wireless network. Inumerous monitoring researches decided to use mobile networks to transfer the data, although this is a good solution some of these networks are more expensive to send the amount of data required for the real-time monitoring, in addition another problem is the speed to send this data through the mobile network. One of the solutions for this researches is to use the GPRS, which is a new service for GSM that simplifies and improve the wireless access to packet data networks [20]. In this proposal the WiMax is the most suitable technology. WiMax is a telecommunication protocol who provides Internet access through mobile and fixed station [21]. Although the WiMax does not appear often in the monitoring researches as a solution to send data, the use in this paper is based on the network that is under construction by the city hall of Joo Pessoa - Brazil to provide internet acess around the city using the WiMax technology. It will be a lot better to utilize the soon to be created network.

Fig. 1. MUV-Bee Network Architecture

The data sent by the bus will be processed in the server and if necessary, they will be stored in a database. The server also has the task to send information for the info screens using the WiMax network. The info screens are LED display boards which will be installed on important bus stops on the pre-established route. This screens will show the arrival time of the next buses going in direction of the bus stop. In the future, these screens can also show an estimated number of passengers on the bus, air quality, temperature and so forth. Finally, the server will process the user requests of services and will use the data stored in the database to answer this request or make processes to acquire new information and after that answer the user. In the next section the services will be listed.

B. System Services

The system services are all the functionalities which can be used for the users. The possible services will appear to the user in the applications. It is possible to add new services for the user, if necessary, for the exdepandable capacity of this system. Some of this services can be seen in the figure 2.

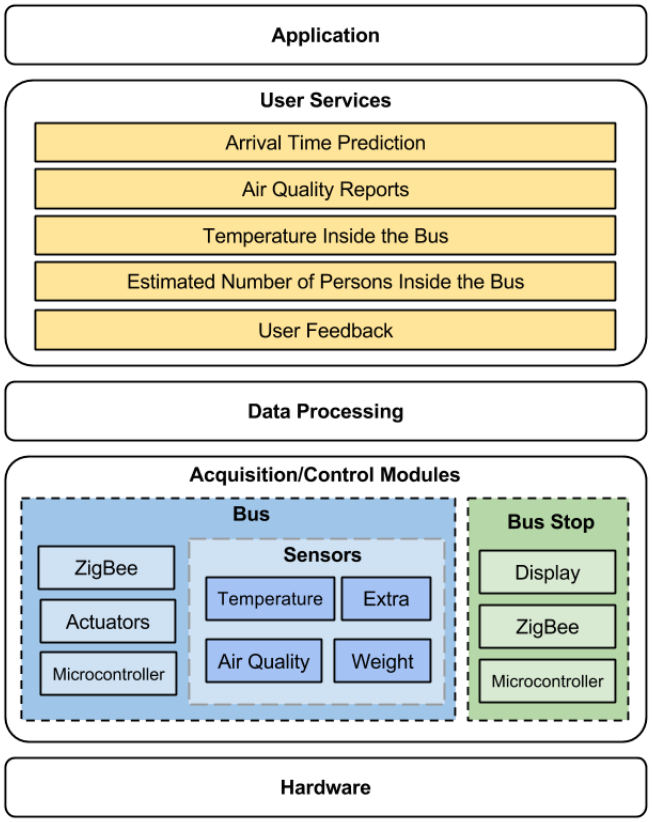


Fig. 2. MUV-Bee System Architecture

The most important service for an user is the arrival time prediction. In the application this service will show a map of the city where it will be possible to select the bus stop and see the arrival time predictions of all the buses for that bus stop. The other services have the intention to present to the user information about the bus condition as temperature, occupancy, air quality. This services will be processed using the data acquired by the bus sensors.

The feedback service is the user channel to help the system to detect errors and improve. For example the user can use the feedback system to alert about an error in the arrival time or about a driver error. If necessary the server will verify the problems recognized by the feedback service and make adjustments in the processing or will report to the administrator the problems related by the users.

IV. Conclusion

This paper presented an intelligent transportation system to improve the public transportation services and make it more attractive to the passengers. The proposed system is based on ZigBee to gather information about the localization of a bus in a pre-determined route. The key feature of this system is its modular structure and capacity to expand when necessary, adding new sensors or services. As future researches it will be decided the algorithms to provide services and for data processing, besides its advantages and disadvantages, as well as the implementation and tests of the system.

References

[1] D. Banister, *Unsustainable transport: city transport in the new century*. Taylor & Francis, 2005.

[2] V. B. C. Da Silva, T. Sciammarella, M. E. M. Campista, and L. H. M. Costa, "A public transportation monitoring system using ieee 802.11 networks," in *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*. IEEE, 2014, pp. 451–459.

[3] M. Shekarrizfard, M.-F. Valois, M. S. Goldberg, D. Crouse, N. Ross, M.-E. Parent, S. Yasmin, and M. Hatzopoulou, "Investigating the role of transportation models in epidemiologic studies of traffic related air pollution and health effects," *Environmental research*, vol. 140, pp. 282–291, 2015.

[4] A. Vasal, D. Mishra, and P. Tandon, "Deployment of gsm and rfid technologies for public vehicle position tracking system," in *Advances in Networks and Communications*. Springer, 2011, pp. 191–201.

[5] B. Predic, D. Stojanovic, S. Djordjevic-Kajan, A. Milosavljevic, and D. Rancic, "Prediction of bus motion and continuous query processing for traveler information services," in *Advances in Databases and Information Systems*. Springer, 2007, pp. 234–249.

[6] H. Niu, W. Guan, and J. Ma, "Design and implementation of bus monitoring system based on gps for beijing olympics," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, vol. 7. IEEE, 2009, pp. 540–544.

[7] M. Hannan, A. Mustapha, A. Hussain, and H. Basri, "Communication technologies for an intelligent bus monitoring system," in *Sustainable Technologies (WCST), 2011 World Congress on*. IEEE, 2011, pp. 36–43.

[8] A. M. Mustapha, M. Hannan, A. Hussain, and H. Basri, "Ukm campus bus monitoring system using rfid and gis," in *Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on*. IEEE, 2010, pp. 1–5.

[9] ——, "Ukm campus bus identification and monitoring using rfid and gis," in *Research and Development (SCOReD), 2009 IEEE Student Conference on*. IEEE, 2009, pp. 101–104.

[10] ——, "Implementing gis in bus identification and monitoring system," in *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE)*, 2011.

[11] L. Cachulo, C. Rabadão, T. Fernandes, F. Perdigoto, and S. Faria, "Real-time information system for small and medium bus operators," *Procedia Technology*, vol. 5, pp. 455–461, 2012.

[12] V. Milanés, J. E. Naranjo, C. González, J. Alonso, and T. de Pedro, "Autonomous vehicle based in cooperative gps and inertial systems," *Robotica*, vol. 26, no. 05, pp. 627–633, 2008.

[13] A. Papapostolou and H. Chaouchi, "Rfid-assisted indoor localization and the impact of interference on its performance," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 902–913, 2011.

[14] H. Feng, L. Lulu, Y. Heng, and H. Xia, "Bus monitoring system based on zigbee and gprs," in *Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on*. IEEE, 2012, pp. 178–181.

[15] C.-q. Cai, Z. Zhang, and S.-D. Ji, "The intelligent bus scheduling based on zigbee," in *Computer Science & Education (ICCSE), 2012 7th International Conference on*. IEEE, 2012, pp. 1002–1005.

[16] Y. Xu, R. Jiang, S. Yan, and D. Xiong, "The research of safety monitoring system applied in school bus based on the internet of things," *Procedia Engineering*, vol. 15, pp. 2464–2468, 2011.

[17] E. Nadimi, H. T. Søgaard, and T. Bak, "Zigbee-based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees," *Biosystems engineering*, vol. 100, no. 2, pp. 167–176, 2008.

[18] A. Antoniou, A. Georgiou, P. Kolios, C. Panayiotou, and G. Ellinas, "An event-based bus monitoring system," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 2882–2887.

[19] R. Mandal, N. Agarwal, P. Das, S. Pathak, H. Rathi, S. Nandi, and S. Saha, "A system for stoppage pattern extraction from public bus gps traces in developing regions," in *Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*. ACM, 2014, pp. 72–75.

[20] C. Bettstetter, H.-J. Vögel, and J. Eberspächer, "Gsm phase 2+ general packet radio service gprs: Architecture, protocols, and air interface," *Communications Surveys, IEEE*, vol. 2, no. 3, pp. 2–14, 1999.
[21] A. Bhambri and N. Kansal, "Survey on wimax technology and its protocol-a review."

Improving group decision-making in IT service management by the use of a consensus-based MCDM method

Igor P. da Silva, Alberto S. Lima, Neuman de Souza
Federal University of Ceará
Fortaleza, Brazil
igorpimentel@gmail.com, {albertosampaio, neuman}@ufc.br

Flávio R. C. Sousa, Lincoln S. Rocha, Thomaz E. V. da Silva
Federal University of Ceará
Fortaleza, Brazil
{flaviosousa, lincolnrocha}@ufc.br, thomazveloso@virtual.ufc.br

Abstract—**Group decision-making in companies that practice IT service management (ITSM) is a difficult and challenging activity. There are particular issues that hinder the performance of IT management committees, such as the lack of productivity, duration of the meetings, physical distance between members, and low quality of some complex decisions, among other restrictive factors. In this paper we present a method based on wisdom of crowds' theory and Analytic Hierarchy Process (AHP), designed to automate the process of group decision-making for IT management committees. We conducted a case study in a Brazilian IT company, and the results indicate that managers considered our method and tool useful, preferable to the current decision-making method, complete and easy to use.**

Keywords- IT service management; group decision-making; IT management committees; winsdow of the crowds; AHP.

I. INTRODUCTION

Companies that practice IT service management (ITSM) using management recommendations of Information Technology Infrastructure Library (ITIL) [1], are getting greater transparency in IT management and delivering high quality IT services, in an IT governance [2] approach.

Business-driven IT management (BDIM) [3] is a recent research area that involves a set of models, practices, techniques and tools, in order to map and quantitatively assess interdependencies between business performance and delivered IT services. The decision-making involved in theory and practice in BDIM was discussed in [4].

IT management committees (ITMC) are very important tools to align both, business and IT goals. These committees need to meet periodically to discuss issues of various natures, issuing opinions or deciding on them. We can cite as examples of committees activities: Consultations, decisions and approval of IT budget, strategic plans, rules and regulations related to IT, among others [5]. Among the main difficulties faced by IT committees' actors, we can mention: Remote member participation in meetings; Guidelines of long time-consuming meetings; Difficulties on the choice in-group decision-making; Low quality decisions.

In this paper, we proposed a new method to support group decision-making in IT service management.

II. LITERATURE REVIEW AND RELATED WORK

IT Management Committee (ITMC) objectives are: align IT actions to the organization strategic guidelines; promote and support the IT projects prioritization to support planning strategies needs; identify and implement opportunities for improvement. ITMC formulate and implement IT strategies and plans, aligned with organizational high-level goals. It directs, monitor and evaluate IT management, observing the IT operations performance, strategies and plans implementation and compliance with IT policies [5]. The ITMC operating cycle comprises three stages (see Figure 1): 1. IT Committee constitution; 2. Communications planning; 3. Meeting schedule execution.

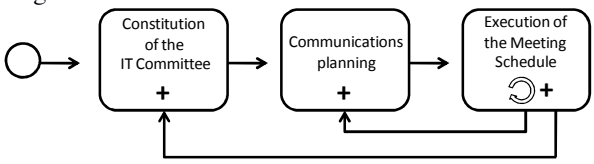


Figure 1. ITMC operating cycle [5].

The ITMC constitution deals with the establishment of a new committee or the reform of existing committee [5]. In communications planning phase, we identify the stakeholders and plan the communication events to meet information needs. Each meeting follows a cycle that goes from agenda preparation to information distribution. Members can revise the committee constitution or its communication planning (See feedback arrows in Figure 1).

Decision-making is a cognitive process by which one chooses an action plan among many others (based on varying scenarios, environments and factors) to a problem situation [6]. People make decisions often based on subjective aspects. When deciding to use group decision-making, one must question whether the efficiency gains will be sufficient to overcome the losses in efficiency. Techniques for group decision-making can help in classification and prioritization of best-presented alternatives in a timely manner. Decisions can be taken by: *Unanimous vote*: Everyone agrees with the decision taken; *Majority:* over 50% of those present agree;*Plurality:* Greater group decides even if there is no

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

majority; *Dictatorship:* Someone decided by the group. In our proposal, we adopted the majority decision type.

The consensus decision-making is a dynamic way to reach an agreement among all group members. When one votes directly each item of the agenda, usually the majority of voters do prevail their opinion. When a group considers consensus, seeks solutions that are actively supported by all. This approach ensures that all opinions, ideas and concerns are considered in decision-making. Our consensus decision-making strategy was inspired by the Delphi technique [9] and "wisdom of crowds" theory [8].

Multi criteria decision-making (MCDM) methods can be divided in those based on *Multi-Attribute Utility Theory* (MAUT) and those based on outranking. The most common MAUT methods are the *Weighted Sum Model* (WSM), the *Weighted Product Model* (WPM) and the *Analytic Hierarchy Process* (AHP); the most common outranking methods are the ELECTRE method and the TOPSIS method [14]. In [17], authors reviewed literature of the multi-criteria decision making approaches. AHP is quite a basic and popular decision-making method in IT contexts [18]. It is designed to cope with both the rational and the intuitive to select the best from a number of alternatives evaluated with respect to several criteria. AHP was used to support IT decision-making process, based on best practices guides [10], structure outsourcing problems [19], construct the objectives of ERP selection [20], aid information retrieval and improve web search results from a controlled vocabulary [21], to obtain better outsourcing provider selection for small and medium enterprises [22], to analyze a IT service management framework and associated processes [25] and to proceed a web site selection for online advertising [26]. In [11], authors cited some inefficiencies of group decision-making in committees, as the excess of caution, the vote and the delay. Fuzzy logic was proposed in literature to improve some MCDM models. We can cite the Fuzzy AHP (FAHP) [23] and Fuzzy ELECTRE III [16].

III. PROPOSED METHOD

The proposed method was based on the phenomenon called "wisdom of crowds"[8], whereby the judgment and collectively constructed perceptions by a group of people, is properly inserted in a context, outweigh the individual perceptions in terms of foresight and quality of the choices made on a set of offered options. The method capture estimates directly from perceptions of the committee members group (crowd) and, then the result converges towards a common, which represents the "average" of captured perceptions. In addition to capturing the perceptions of the "crowd" (several members of the management committee), the method seeks to allow comparison between the evaluated items, so that committee members are able to decide, based on what was told by the other members. Figure 2 shows a view of our group decision-making method. We have examined the MCDM methods and have chosen to base our method on Analytic Hierarchy Process (AHP) [24] support for the following reasons: it is by far the most popular method, it has been used in IT contexts and is easily amenable to extensions (can be improved in a future work). The AHP both allows for inconsistency in the judgments and provides a means to improve consistency [20]. When IT committee needs to assess complex decisions involving multiple factors, our method includes AHP [24, 25, 26] as an option to support group multi-criteria decision-making. In this case, the committee members can set criteria, weights and discuss about the values of AHP parameters. When members are voting using our method, the AHP module results are shown to members to guide group decision-making.

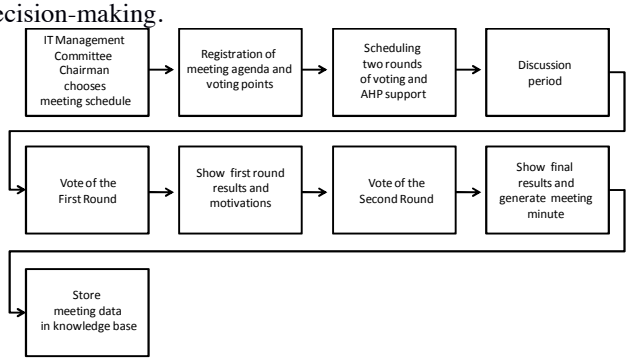


Figure 2. Proposed method.

Our method was automated in a system that supports the ITMC operating cycle, specifically the third stage, meeting schedule execution. As shown in Figure 2, the decision-making process must be performed in two rounds. In the first round, the committee chairman presents the entire agenda of the meeting to the members, who will have access to all information pertaining to committee activity (decision, AHP module or consultation on a subject). In multi-criteria decisions, members choose the AHP parameters values to input our model and get the results. After usual discussions, which are also supported by the system via forum, the committee chairman places each agenda item to a vote in its first round. Members vote and justify their motivation for performed choosing. At first round end, all members can view the results with the selection process justification for each group member, in an anonymous form, to not suffer influence. The second round is the time when the final choice is actually taken. The member can change his/her choice, due to some argument in the first round justifications list that has convinced, or can keep his/her previous vote.

We observed all the steps needed for design, development, testing and validation to develop our *Support Group Decision-Making System* (SGDMS) [7]. Due to time restrictions, the first version of our tool was developed using the Portuguese language.

IV. CASE STUDY AND RESULT ANALYSIS

We planned and carried out a case study in a Brazilian telecommunications company. The company name will not be revealed due to business confidentiality.

The sample used in the study included the members of the IT Committee, with five business managers, the IT manager and five IT project managers, and five technicians (IT experts).

After the presentation of the proposed method, an interview was conducted with 11 IT managers of industry, financial and IT areas, in a face validity exercise [16, 17].

So, our sample involved 27 evaluators. Our proposal was used in 10 IT management committee meetings. The context involved ITMC meetings analysis. We evaluated the hypothesis related to usefulness, completeness, preference and ease of use. We attempted to observe [12] and [13] recommendations, during the research planning phase. We used a controlled experiment to validate our tool.

The case study involved the following steps: Literature review; Method development; Design and implementation of software tool; Meeting with managers; Using the tool in ten meetings; Results presentation and discussion with managers; General results tabulation; Method validation with managers; Publish final results.

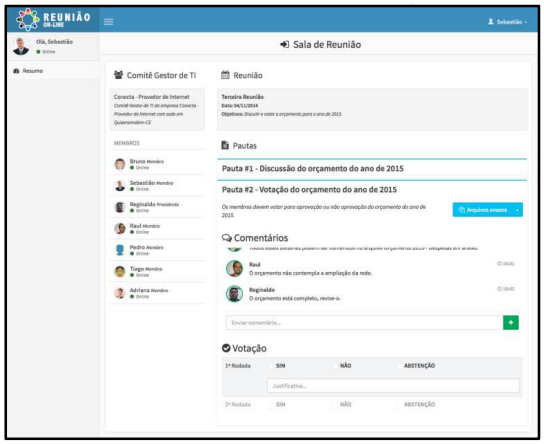


Figure 3. Virtual meeting room.

In our software tool, the profile *Admin* can edit and delete existing entries; register new users; and manage the list of registered committees. Chairman profile includes options to manage committee meetings and agenda, see the meetings list, start/stop meetings, start/stop the vote of the first round, start/stop the vote of the second round and generate meeting minutes. As a member, chairman can access the virtual meeting room. Member profile can view a summary of upcoming meetings on the system's home screen. A member can access the meeting room (Figure 3), and view all the committee's information, such as description, which the other members, meeting aspects such as date and objectives and the agenda list, with its annexes. A member stillcan comment on the guidelines and vote (first and second round). As a member, Chairman can perform all these actions too.

During our case study, the committee decided about migrating applications to software as a service (SaaS) in the cloud, using our tool AHP support module. The SaaS services are those representing the adoption of the cloud model in its most comprehensive form. The user of software on the SaaS model is also user of platform as a service (PaaS), and infrastructure as a server (IaaS), indirectly. The SaaS provider offers the use of software over the Internet and charge for use without the need for investment in hardware, software and specialized IT staff for environmental management. For various types of business, this transformation of capital expenditures in operating expenses is extremely attractive. Furthermore, the total cloud service cost is generally lower than the cost of the service available in the customer environment (on-premises).

In AHP, we decomposed the problem into a hierarchy decision criteria and alternative using paired comparisons to express the relative importance of a criterion with respect to each other. Thus, it becomes possible to construct an array of pairwise comparisons and calculate the eigenvector, to finally calculate a score for each alternative, and the alternative with the highest score by the selected method.

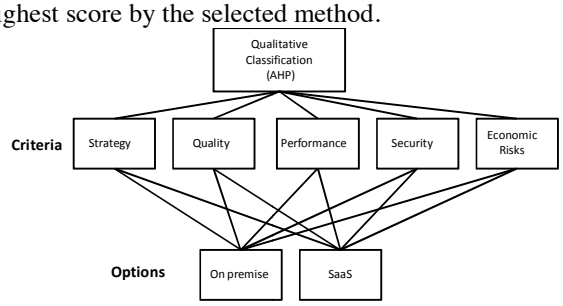


Figure 4. Migration decision modeled in AHP.

To support the Committee multi-criteria decision, members combined cost and key qualitative factors aiming to generate notes to be calculated for the software as a service (SaaS) and also for the on-premises version. That option that obtains the highest score (best cost-benefit) should be the preferred option. To calculate the scores value, the following steps were performed: Estimate the cost of software hosted internally (on-premises): C_h; Estimate the cost of software as a service (SaaS): C_s; To evaluate the benefits and risks of software as a service (SaaS): B_s and also from internally hosted software (on-premises) B_h using AHP (the end result is a value between 0 and 1 for each solution); Normalize the costs obtained in steps 1 and 2 that represent values between 0 and 1; Calculate the cost-benefit of each solution.

Figure 4 shows the AHP model view, which was proposed to evaluate this decision. Committee members used the methodology presented in [15] to develop the AHP model for the decision, using our tool. Due to space restrictions we will show some AHP used data (See Tables 1 and 2).

TABLE I. PAIR COMPARISON BETWEEN CRITERIA

	On premise	***SaaS***
Strategy	5	9
Quality	5	7
Performance	9	5
Security	9	3
Economic risks	9	1

TABLE II. AHP FINAL RESULTS

	On premise	**SaaS**
Year 1	0,8312	1,4952
After 5 years	1,1188	0,8522

When IT management committee members considered the 1st year, the SaaS option would be much more interesting, because it received a higher note. However, over 5 years, the on premise option becomes more interesting, because the initial costs are amortized. It is noted that despite the cost comparison always be in favor of SaaS option, the favorable qualitative assessment to on premise option slightly reduces this cost advantage. The AHP results are very important to a decision-making process, and should be discussed by the committee. We observed that in second round, after seeing

AHP support and decision motivations in first round, committee members followed the AHP results for year 1 and chose the SaaS option.

TABLE III. FACE VALIDITY RESULTS

Hypothesis	% who believes	Is there enough statistical evidence to support the hypothesis?
Preference: evaluators preferred the method presented in relation to the current form of group decision-making group.	100	yes
Utility: Evaluators considered the method useful.	100	yes
Ease of use: Evaluators considered the proposed method easy to use and apply.	93	yes
Completeness: Evaluators considered the presented method complete in relation to objectives.	100	yes

Our face validity exercise [12, 13] obtained 100% of positive evaluations in almost all evaluated hypothesis, except for the relative ease of use of the method/tool (93%). We used a binomial test at 5% significance level, to produce the results shown in Table 3.

V. FINAL CONSIDERATIONS AND FUTURE WORK

In this work, we proposed a consensus-based MCDM method that can improve the productivity in IT management committees. Our proposed software tool can be used in computers, tablets and smart phones, automating meetings, decision-making process and supporting the use of AHP.

We observed that the AHP use to support complex decisions improved the speed on group decision-making. The voting process was conducted in two rounds, which can allow reaching a consensus among decision makers. In addition to this possibility, our proposed tool also allows the adoption of conventional decision-making process, the use of AHP in a multi-criteria decision-making, as well as consultations and meetings with merely informative agendas. Our main contribution was the proposed method and software tool. The initial results indicated that our proposal is useful, preferable, complete and easy to use. As threats to validity, we can cite that the case study was executed in a single company, in 10 meetings of 1 IT management committee, and it is difficult to generalize the results. Although these limitations, our initial results were promising. In a future work, we plan to repeat the study in IT committees of different business areas companies.

REFERENCES

[1] OGC (Office of Government Commerce), ITIL V3 PUBLICATIONS, "Service Strategy", "Continual Service Improvement", "Service Design", "Service Operation", "Service Transition", 2007.

[2] Weill P., Ross J. "IT Governance: How Top Performers Manage IT Decision Rights Results", Harvard Business Press, Cambridge, MA, 2004.

[3] Bartolini C., Stefanelli C. "*Business-driven IT Management*", Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 963 –969, 2011.

[4] Bartolini C., Stefanelli C, Tortonesi M. *"On decision making in business-driven IT management"*, in Proc. 2011 IFIP/IEEE International Symposium on Integrated Network Management, pp. 1082–1088, 2011.

[5] SISP. *Guia de Comitê de TI do SISP: versão 2.0* / Ministério do Planejamento, Orçamento e Gestão, Available: http://www.governoeletronico.gov.br/biblioteca/arquivos/guia-para-a-criacao-e-funcionamento-do-comite-de-ti, 2013.

[6] Shimizu, T. *Decisão nas Organizações*. 2 ed. São Paulo: Atlas, 2006.

[7] Huber, G. P. Issues in the design of group decision support systems. Mis Quarteley. 3: 8, 1984.

[8] Surowiecki J. "The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations", First Anchor Books Edition, 2005.

[9] Rowe G, Wright G. *The Delphi technique as a forecasting tool: issues and analysis*. International Journal of Forecasting, v. 15, Issue 4, pp. 353-375, 1997.

[10] Simonsson M., Johnson P. The IT Organization modeling and Assessment Tool for decision-making support, Caise, LNCS 5074, 2008, pp. 256-261.

[11] Hao L., Wing S. *Decision making in committees*, white paper, University of Toronto, Available: http://homes.chass.utoronto.ca/~haoli/research/Ralph.pdf, 2009.

[12] Runerson P., Host M. Guidelines for conducting and reporting case study research in software engineering, Springer: Empiric Software Eng. ,14:31-164, DOI 10.1007/10664-008-9102-08, 2009.

[13] Wohlin C., Runerson P., Host M., Ohlsson B. R., Wesslen A. Experimentation in Software Engineering - An introduction, Kluwer Academic Publishers Norwell, MA, USA, 2000.

[14] Triantaphyllou E. Multi-Criteria Decision Making Methods: A Comparative Study, ser. Applied Optimization. Kluwer Academic Publishers, 2000. [Online]. Available: http://books.google.com.br/books?id=tuPGe ur-TYC.

[15] Ribas M., Lima A. S., De Souza J. N., Sousa F. R. C., Fenner G. Multicriteria Decion-making in migrating applications to Software as a Service in the cloud. Revista Brasileira de Administração Científica, v. 5, pp. 20-48, 2014.

[16] Shen F., Xu J., Xu Z. An automatic ranking approach for multi-criteria groupdecision making under intuitionistic fuzzy environment, Fuzzy Optim Decis Making, DOI 10.1007/s10700-014-9201-5, 2015.

[17] Ho W., Xu X., Dey P. K. Multi-criteria decision making approaches for supplier evaluation and selection: A literature review, European Journal of Operational Research, v. 202, pp. 16–24, 2010.

[18] Saaty T. L., Vargas L. G. Models, Methods, Concepts & Applications of the Analytic Hierarchy Process, Cap. 1 - How to Make a Decision. Springer US, DOI 10.1007/978-1-4615-1665-1,2012.

[19] Yand C. Yang , Huang J. B. A decision model for IS outsourcing, International Journal of Information Management, v. 20, n. 3, pp. 225 – 239, 2000. [Online]. Available: http:// www.sciencedirect.com/science/article/pii/S0268401200000074.

[20] Wei C. C. Wei, Chien C.F., Wang M. J. J. An AHP-based approach to ERP system selection, International Journal of Production Economics, vol. 96, no. 1, pp. 47–62, 2005. [Online]. Available: http://ideas.repec.org/a/eee/proeco/v96y2005i1p47-62.html.

[21] Phillips-Wren G. E., Forgionne G. A. Aided search strategy enabled by decision support, Inf. Process. Manage., v. 42, n. 2, pp. 503–518, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.ipm.2005.02.004.

[22] Chang S. I., Yen D. C., Ng C. S., Chang W. T. An analysis of IT/IS outsourcing provider selection for small- and medium-sized enterprises in Taiwan," Information & Management, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.im.2012.03.001.

[23] Van Laarhoven P., Pedrycz W. A fuzzy extension of Saaty's priority theory, Fuzzy Sets and Systems, v. 11, n. 13, pp. 199 – 227, 1983. [Online]. Available: http://www.sciencedirect.com/science/ article/pii/S0165011483800827.

[24] Saaty T. The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation. New york: McGraw-Hill, ISBN-13: 978-0070543713, 1980.

[25] Wan J., Zhang H., Wan D. Evaluation on information technology service management process with AHP, Technology and Investment, v. 2, no. 1, pp. 38–46, 2011.

[26] Ngai E. W. T. Selection of web sites for online advertising using the AHP, Inf. Manage., v. 40, n. 4, pp. 233–242, 2003. [Online]. Available: http://dx.doi.org/10.1016/S0378-7206(02)00004-6.

RAU2 testbed: a network prototype for evolved service experimentation

Eduardo Grampín, Martín Giachino, Jorge Rodrigo Amaro, Emiliano Viotti
Instituto de Computación (INCO), Universidad de la República (UdelaR)
Montevideo, Uruguay
Email: {grampin, giachino, jorge.amaro, eviotti}@fing.edu.uy

***Abstract*—The Academic Network of Uruguay (in spanish *Red Académica Uruguaya - RAU*) comprises several universities, research centres and government institutions. RAU is planning a major upgrade, and Software Defined Networking (SDN) is being evaluated as a technology which may promote the deployment of improved network services, while allowing researchers to keep on investigating over the operational network. To this end, we are building an evolved "RAU2" network prototype, in order to test the feasibility of the proposed architecture.**

***Keywords*-Software Defined Networking; Network Experimentation; Traffic Engineering;**

I. Introduction

RAU is planning a major upgrade, seeking for a boost on coverage and capacity, focused on flexibility for the deployment of new, evolved network services. Being a research and education network, the proposed solution should permit the co-existence of operations and research activities over the same physical infrastructure, following an open software (and hardware) approach. Given these requirements, we have chosen to explore the Software Defined Networking (SDN) paradigm, building our own open-source routers, based on COTS x86 hardware augmented with specialized, programmable NetFPGA[1] network adapters.

The project team have more than 10-years experience working on the concept of Control Plane and Data Plane separation, which is one of the basic principles of SDN. The Routing and Management Agent (RMA) architecture has been proposed in [1], coincidently with the foundational work by Feamster et al. on "taking routing out of the routers" [2]. The IETF Path Computation Element (PCE) Architecture [3] was born those same years, in an effort to standardize a simple, yet powerful idea: let networking hardware do its best in forwarding packets al line speed, leaving the complexity to external, fault tolerant computation gear. We built some PCE implementation prototypes, both at the signalling and the computation level, building up from our experience on MPLS-based Traffic Engineering [4].

Our approach is simple: let's learn from experience, trying to develop a research agenda in the line of hybrid networking, that is to say, admitting the coexistence of legacy and newer, generic hardware with full SDN integration.

This work was partially funded by LACNIC FRIDA Program 2013, the uruguayan Agency of Research and Innovation (ANII), and the uruguayan Public Telcommunication Company (ANTEL).

[1] Online: http://netfpga.org/

RAU evolution main goals are: i) to extend service coverage with improved Quality of Service, and ii) to implement transversal services and infrastructures to maximize synergy among member institutions. The evolution from RAU to RAU2 is supported on three pillars:

- a network topology redesign with bandwidth increase,
- an improved network controllability, which shall empower the development of
- a transversal and integrated academic services platform.

Given these requirements, we decided to explore a SDN-based approach, with emphasis on open-source solutions. In next section we review some relevant related work which help us to build our RAU2 testbed.

II. Related work

There are a number of very good SDN surveys, in particular, the interested reader may examine Kreutz et al. [5], which provides a comprehensive review of several aspects of this rising networking paradigm. In this article we will limit ourselves to analyse specific aspects related to the design and construction of our SDN-enabled router and related aspects of testbed deployment.

MPLS is a de-facto solution for both VPN L2, L3 services and Traffic Engineering, which are fundamental requirements for RAU2.

There are have been a number of efforts trying to realize SDN-enabled MPLS implementations. OpenFlow, proposed by McKeown et al. [6] as an standardized interface to add and remove flow entries in a generic Ethernet switch with an internal flow-table, established a foundation for the implementation of the SDN idea: a vendor-independent protocol which defines syntax and semantics for the programmability of the switch by an (external, third party) controller. Kempf et al. [7] proposed an extension to early versions of OpenFlow for MPLS data plane support; these authors built a low-cost LSR consisting of a NetFPGA board running the modified OpenFlow with MPLS support, embedded in a PC with Quagga[2] LDP and MPLS Linux. Open Source Hybrid IP/SDN networking (OSHI), by Salsano et al. [8], propose to generalize the MPLS tunnel concept to the idea of SDN Based Paths (SBPs), which is a virtual circuit or network path created using various fields of different protocols (TCP/UDP, IP, VLANs, Ethernet, MPLS, among others). The OSHI node is composed by legacy Quagga routing suite and Linux kernel IP forwarding linked by virtual

[2] Online: http://www.nongnu.org/quagga/

978-1-4673-9408-6/15/$31.00

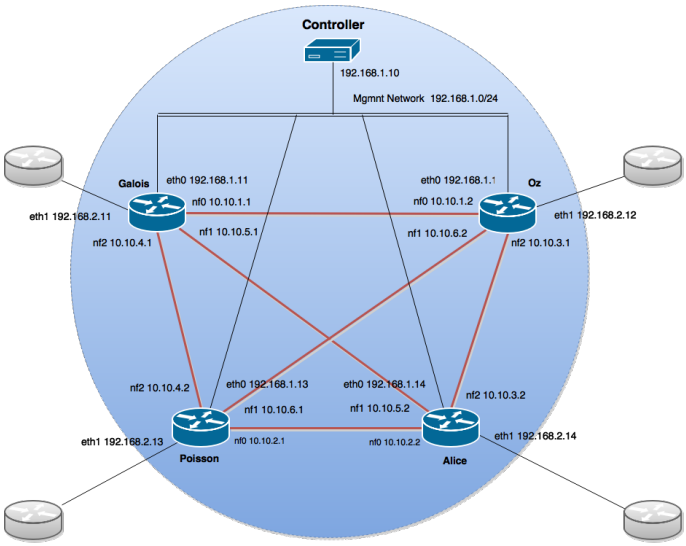


Figure 1. RAU2 testbed basic topology

ports to a software SDN capable Switch (Open vSwitch[3]).

III. Building the Testbed

In light of RAU2 requirements, and considering our previous experience and related work, we decided to build an hybrid prototype. First, we built a four node network using NetFPGA 10G networking cards installed in regular x86 hardware. These nodes have full OpenFlow support, and being regular Linux boxes, they can also run "legacy" routing protocols provided by the Quagga routing suite. There exists an implementation of OpenFlow 1.0 for the NetFPGA 1G platform [9], which has been partially ported to the newer NetFPGA 10G card, and a migration to the newest version 1.3 is needed to support MPLS data plane in OpenFlow. Nevertheless, in the first phase of our testbed deployment we decided to use the well-known Open vSwitch implementation, along with the *Reference NIC* firmware for NetFPGA 10G platform[4]; this decision come with a penalty in performance, but permit to promptly start experimentation, while an OpenFlow 1.3 port is being launched in parallel.

A complete survey of SDN controllers has been conducted, and for the sake of simplicity, we are developing our prototype solution using Ryu[5]. A sample configuration of the testbed using 10Gbps optical links with out-of-band management is shown in Figure 1.

The architecture of the solution, as mentioned above, has to be hybrid in nature, because we want to take advantage of the precious heritage of legacy networking, i.e. distributed network protocols, in order to build fast, resilience and transition-proven solutions. We rely on OSPF for topology learning and routing information dissemination; the controller gathers this information to deploy sample services, such as L2 and L3 VPNs[6].

[3]Online: http://openvswitch.org/

[4]Online: https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-10G-Reference-NIC

[5]Online: http://osrg.github.io/ryu/

[6]Updated code and documentation can be found at: https://github.com/ProyectoRRAP

IV. Conclusion and Future Work

In this paper we analyze the requirements for an academic network major upgrade, and use them to design a network prototype for experimentation of evolved services. We briefly review relevant related research, which lead us to an hybrid SDN-legacy approach. We decided to build our routers using COTS x86 hardware augmented with NetFPGA networking cards. After building and testing the basic functionalities of the routers and the software base, we developed L2 and L3 MPLS-based VPN prototypes over the Ryu controller.

We would like to remark that SDN adoption would definitively be hybrid and progressive, calling for the emergence of new technical and business to facilitate the transition.

References

[1] E. Grampín and J. Serrat, "Cooperation of control and management plane for provisioning in MPLS networks," in *Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium on*, pp. 281–294, May 2005.

[2] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, FDNA '04, (New York, NY, USA), pp. 5–12, ACM, 2004.

[3] A. Farrel, J. P. Vasseur, and J. Ash, "RFC 4655: A Path Computation Element (PCE)-Based Architecture," tech. rep., IETF, Aug 2006.

[4] E. Grampín, J. Baliosian, J. Serrat, G. Tejera, F. Rodríguez, and C. Martínez, "A Trial Experience on Management of MPLS-Based Multiservice Networks," in *Proceedings of the 5th IEEE International Conference on Operations and Management in IP-Based Networks*, IPOM'05, (Berlin, Heidelberg), pp. 191–201, Springer-Verlag, 2005.

[5] D. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, pp. 14–76, Jan 2015.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 69–74, Mar. 2008.

[7] J. Kempf, S. Whyte, J. Ellithorpe, P. Kazemian, M. Haitjema, N. Beheshti, S. Stuart, and H. Green, "OpenFlow MPLS and the open source label switched router," in *Teletraffic Congress (ITC), 2011 23rd International*, pp. 8–14, Sept 2011.

[8] S. Salsano, P. L. Ventre, L. Prete, G. Siracusano, M. Gerola, and E. Salvadori, "OSHI - Open Source Hybrid IP/SDN networking (and its emulation on mininet and on distributed SDN testbeds)," in *European Workshop on Software Defined Networks (EWSDN)*, (Budapest (Hungary)), September 2014.

[9] J. Naous, D. Erickson, G. A. Covington, G. Appenzeller, and N. McKeown, "Implementing an OpenFlow Switch on the NetFPGA Platform," in *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '08, (New York, NY, USA), pp. 1–9, ACM, 2008.

Cloud Enabled Smart Video-Surveillance providing Public Safety Assistance for Vehicles

Hugo Barros Camboim
Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil
hgb@ppgsc.ufrn.br

Augusto José Venâncio Neto
Department of Informtics and Applied Mathematics
Federal University of Rio Grande do Norte
Natal, Brazil
augusto@dimap.ufrn.br

***Abstract*—Smart video surveillance plays a key role in offering technological assistance in public safety scenarios, on account of its potential to allow events and objects to be detected in real-time. In this work, we propose a solution which involves local video streaming processing for the coupled approach deployed between remote Smart Surveillance applications and multimedia sensors embodying vehicles.**

Keywords—Smart Public Safety, Distributed Systems, Ubiquitous Computing, Context Awareness, Object Recognition, Cloud Computing.

I. INTRODUCTION

On the basis of the requirements of vehicular environments, Cloud Computing serves as an excellent tool to support several applications. Cloud Computing can be combined with a new paradigm to provide a computing infrastructure that can reduce the costs of managing hardware and software resources [1]. In addition, this work is carried out by designing a cloud framework able to support several kinds of crime in-vehicle events and support and send generated data which can benefit both vehicle and external side applications. This work seeks to design an intelligent solution based on decoupled modules that can make public transport much more secure by using Cloud Stack Concern techniques once it meet the requirements to enable efficient smart surveillance applications through scalability, security, connectivity and other benefits that will be described below.

II. THEORETICAL REFERENCE

A. Ubiquitous Computing

The constant technical advances in communication and computing have given reality to a scenario which a short time ago was just being glimpsed by Pervasive Computing, and it means that computing is now a common feature of people's activities, and before long already forms part of our daily life.

978-1-4673-9408-6/15/$31.00 ©2015 IEEE

Furthermore, forecasts predict that microprocessors will become smaller and smaller and will soon be cheap enough to be embedded in digital devices, electronic appliances, everyday objects and clothes, as well as in cars.

This scenario is regarded as the new paradigm of the Century [1,2] or the third wave of computing. It allows the physical world to be combined with the world of information and provides a wide range of services and ubiquitous applications that target users, machines, data, applications, and various objects in physical space, so that they can interact with each other in a transparent way [3]. Hence, technology is gradually moving toward the vision of ubiquitous computing, or incrementally having to rely on a set of heterogeneous devices to support a growing range of applications that fit this scenario. According to its creator Mark Weiser, "deeper technologies are those that disappear" [4]. From this perspective, Ubiquitous Computing seeks to include computing in the physical world of the user, by allowing it to concentrate on the tasks that have to be accomplished and not just the tools that are needed for this.

According to [5] ubiquitous systems have at least five principles, Diversity, Connectivity, Decentralization, Invisibility, Context Awareness, Proactivity. On the basis of

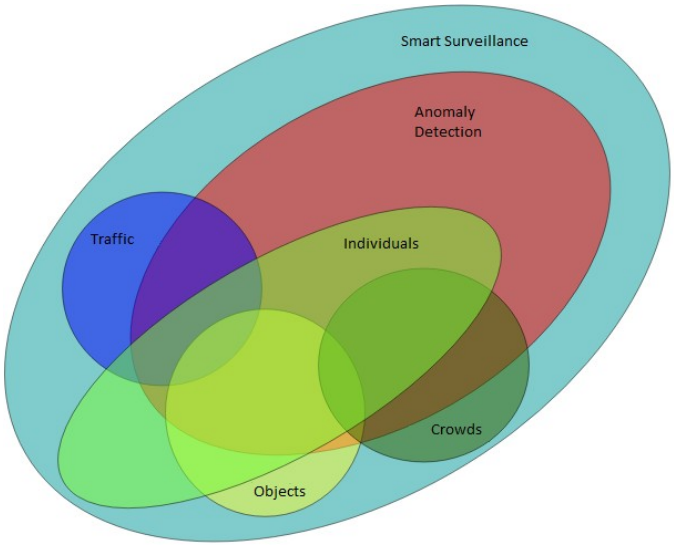


Fig. 1. Venn Diagram – Smart Surveillance Targets.

these principles, the concept of Ubiquitous Surveillance arises [6] as a sub-area of Ubiquitous Computing that is concerned with surveillance.

B. *Smart Surveillance*

With regard to disability surveillance which is only carried out with human endeavor, there are several research papers aimed at automating ubiquitous surveillance Some of them are listed in [7]. These works essentially use all the features combined with extraction techniques for machine learning and these are either Supervised or Non-Supervised. Figure 1 shows a diagram of the intelligent monitoring targets and their interrelationship. Thus the assumptions made about effective automatic surveillance are formed on the basis of a particular target environment. With the aid of this concept, the purpose of this study is to create a mechanism that recognizes objects used in assaults based on a detailed analysis of statistics about criminal acts in city buses in recent years in the capitals of North-East Brazil. Thus the analysis can show the objects used for these criminal acts.

III. Framework Proposal

The proposed framework was designed to work in a citywide scenario, as shown in Figure 3. A part of the framework solution is deployed in the bus cabin of the public transport vehicles (buses).

A. *Components of the Bus Cabin*

Details are given below about the location of the intra-vehicle components. In Figure 3 there is a design of each component and then we describe the functions of each one. The Target Object Detector Module is designed to capture and process video streaming and compare the input images in an attempt to find any registered object template. The minimal resolution required for the expected work is 640x480 pixels.

B. *Cloud Components*

The benefits of Cloud Computing mentioned above, were made use of by designing Cloud Components as part of the proposed framework. In Figure 2, there is a description of the design of the components that will be deployed in a specific cloud provider to support both the Bus Cabin components and Context-Driven Applications.

IV. Conclusion and Remarks

In this poster we presented a framework as tool for improvement of Smart Public Safety in Vehicular Environment, based on local image processing and dangerous objects detection. The main idea of our proposed framework is to make possible the management of dangerous objects templates and it updates “on the fly” for each instance of local solution, reducing human force in treat detection and based on the vehicle location, target more efficiently the police agents.

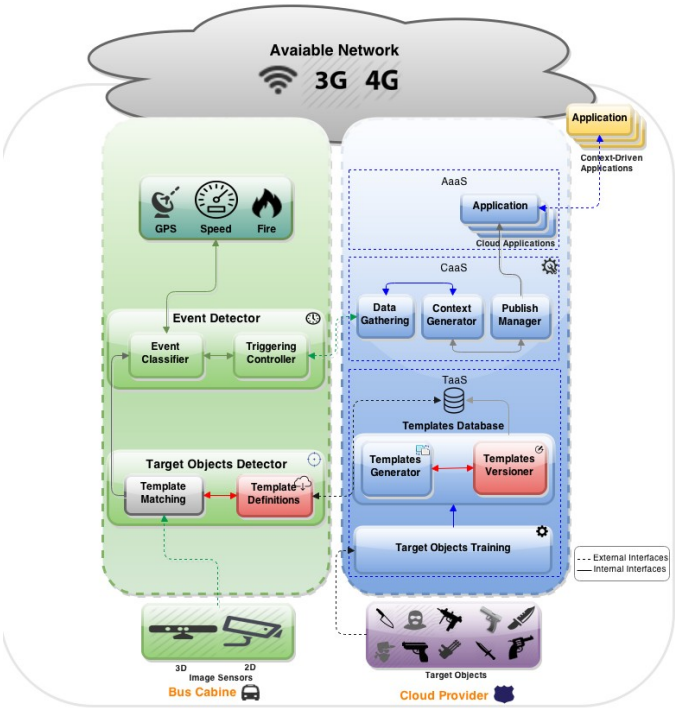


Fig. 2. Proposed Framework Overview

The information extracted from several instances and grouped in the proposed framework can benefit and offers opportunities to governments elaborate and maintain a lot of applications.

References

[1] HAYES, B. Cloud computing. Commun. ACM, ACM, New York, NY, USA, v. 51, n. 7, 9–11, jul. 2008. ISSN 0001-0782. Available at: <http://doi.acm.org/10.1145/1364782.1364786>.

[2] SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21st century. Computer, v. 36, n. 3, 25–31, Mar 2003. ISSN 0018-9162.

[3] SATYANARAYANAN, M. Pervasive computing: vision and challenges. Personal Communications, IEEE, v. 8, n. 4, 10–17, Aug 2001. ISSN 1070-9916.

[4] RANGANATHAN, A. et al. Towards a pervasive computing benchmark. In: Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on. [S.l.: s.n.], 2005. 194–198.

[5] WEISER, M. The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev., ACM, New York, NY, USA, v. 3, n. 3, 3–11, jul. 1999. ISSN 1559-1662.

[6] Hansmann, U., Merk, L., Nicklous, M.S., Stober, T. (2001) "Pervasive Computing Handbook", Ed. Springer. 409 pags

[7] TAYLOR, D. Orwellian Ubiquitous Computing May Build Ultimate Surveillance Society. May 2008. Avalaible at: <http://www.oldthinkernews.com/2008/05/ orwellian-ubiquitous-computing-may-build-ultimate-surveillance-society/>.Acesso em Fevereiro 8, 2014.

An Autonomic Computing-based Architecture for Cloud Computing Elasticity

Emanuel Ferreira Coutinho*¶ and Danielo Gonçalves Gomes§¶ and José Neuman de Souza†¶
*Virtual University Institute (UFC VIRTUAL)
§Teleinformatics Engineering Department (DETI)
†Computer Science Department
¶Federal University of Ceara (UFC) - Fortaleza - Brazil
Email: emanuel@virtual.ufc.br, {danielo,neuman}@ufc.br

***Abstract*—Elasticity is an important feature of cloud computing, and can be understood as how a computational cloud fits to variations in their workload by provisioning and deprovisioning resources. Autonomic Computing brings many concepts quite useful in the construction of elastic cloud computing solutions, such as control loops and thresholds-based rules. This paper proposes an elastic architecture for cloud computing based on concepts of Autonomic Computing. For its validation, we designed two experiments using microbenchmarks, applied in both private and hybrid clouds. Results shown cloud computing and Autonomic Computing may work well together in the elasticity provisioning.**

***Keywords*—Cloud Computing; Elasticity; Autonomic Computing; Performance Analysis.**

I. Introduction

With an increase in cloud services availability it is natural that the number of users and workloads also grow. Consequently, cloud providers should expand their resources without loss of QoS (Quality of Service) or SLA (Service Level Agreement) violation. The computing resources monitoring (e.g. CPU and bandwidth), becomes essential for providers and customers. Often the elasticity is associated to a provider resource. Elasticity can be defined as the degree to which a system is able to adapt to changes in workloads by resources provisioning and unprovisioning in an autonomic manner, so that at each point in time the available resources combine with the demand of workload as close as possible [1]. An autonomous or autonomic system consists of a set of autonomous elements, which is a component responsible for managing its own behavior in accordance with policies, and interact with other autonomous elements that provide or consume computational services [2]. Mechanisms of Autonomic Computing, such as control loops and rules, can be used in monitoring of a computational cloud. Thus, resources can be added and removed from the environment as pre-established thresholds. This type of monitoring strategy is directly associated with one of the main characteristics of cloud computing: elasticity.

This paper aims to propose an architecture for cloud computing using concepts of Autonomic Computing for providing an elastic environment.

II. Autonomic Architecture

As a solution for cloud computing elasticity provisioning, we proposed an architecture based on some concepts of Autonomic Computing described by Kephart and Chess [2], mainly using control loops, collectors, actuators, rules and self configuration [3] [4]. Figure 1 describes the proposed architecture with its components and relationships. Figure 1 does not distinguish the type of cloud, and the architecture can be applied to any of them (private, public, community or hybrid). Figure 2 displays this vision, using a private and public cloud together, constituting a hybrid cloud.

III. Algorithms to Support the Architecture

To support the implementation of the proposed architecture components, we designed some algorithms in high level. We represented the infrastructure by the tuple $I =< M, R, V >$, where M is the set of metrics to be monitored and used for the creation of rules, R is the set of rules, and V is the set of values of the collected metrics. Considering for the rule representation the tuple $R =< m, o, f, a >$, where m is the metric used in the rule, o is the operator applied in the rule $(=, >, <, \leq, \geq)$, f is the reference value used as threshold in the rule, and a is the action to be taken if the rule is true (in case of rule violation). Finally, considering $V =< m, v >$, where m is the metric and v is its collected value. These algorithms are described in Figure 3.

Algorithm 1 describes the pseudocode for the data collection mechanism. The $collectData(m)$ function should be implemented to collect data from metrics that will be used in the rules The $formatData(data)$ function must implement mechanisms to standardize the data. The

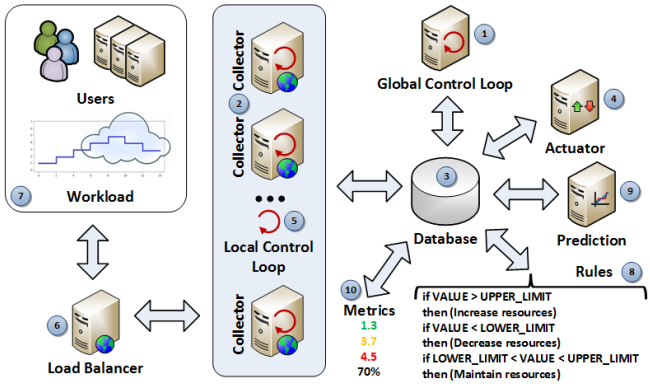


Fig. 1: Autonomic architecture for cloud computing elasticity

978-1-4673-9408-6/15/$31.00

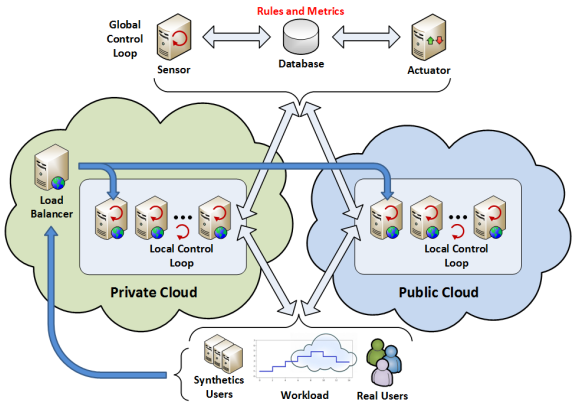


Fig. 2: Autonomic architecture for cloud computing elasticity using private and public clouds (hybrid cloud)

Algorithm 1 Procedure for data log

```
1: procedure LOGDATA(M)
2:     for all m ∈ M do
3:         data ← collectData(m)
4:         formattedData ← formatData(data)
5:         saveData(formattedData, file, type)
6:         V(m) ← formattedData
7:     end for
8: end procedure
```

Algorithm 2 Procedure for rule verification

```
1: procedure CHECKRULE(R, M, V)
2:     for all r ∈ R do
3:         value ← identifyValue(r.m, V)
4:         if checkViolation(r, value) = true then
5:             performAction(r.a)
6:         end if
7:     end for
8: end procedure
```

Algorithm 3 Procedure for global control loop

```
1: procedure ACTIVATEGLOBALMANAGER
2:     loop
3:         if checkPrediction(V) = true then
4:             performAction()
5:         else
6:             CheckRule(R, M, V)
7:         end if
8:     end loop
9: end procedure
```

Fig. 3: Algorithms for cloud computing autonomic architecture

$saveData(formattedData, file, type)$ procedure must be implemented to store the data in the database ($file$) and its type ($type$). Algorithm 2 describes an implementation of the rule verification mechanism. The $identifyValue(r.m, V)$ function should be implemented to obtain the calculated metric value used in the rule. The $checkViolation(r, value)$ function is a mechanism for comparing the collected value of the rule's metric with the reference value, and identify violations of SLA. The $performAction(r.a)$ procedure should be implemented to trigger actions for events of resource adding or removing. Algorithm 3 describes the global manager, represented by the global control loop component. The $checkPrediction(V)$ function must implement a mechanism for prediction of SLA breaks based on collected values and rules, and thus trigger or not the actions of elasticity. If yes, the $performAction()$ procedure acts to anticipate the addition or removal of resources.

IV. Material and Methods

We used two cloud environments for the experiments: a private cloud with OpenNebula and a public cloud with Microsoft Azure. Each experiment used four virtual machines with similar capacities. For elasticity, we built a mechanism based on the proposed architecture. We used an horizontal elasticity strategy, where new virtual machines instances are added to the load balancer when resources are needed, and removed if no longer needed. To trigger elasticity actions, we used the percentage average of virtual machines CPU usage. Figure 4 and Figure 5 show the results, highlighting the experiments thresholds intervals in the red lines.

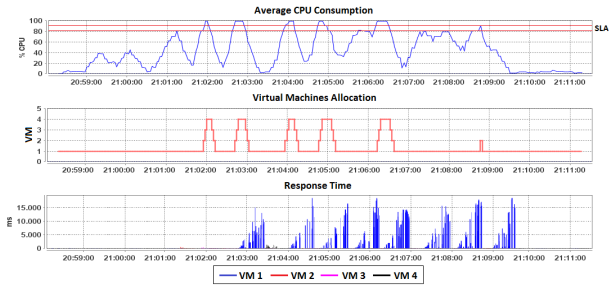


Fig. 4: Average CPU usage (%), allocation, and response time of virtual machines (miliseconds) graphs for Experiment 1

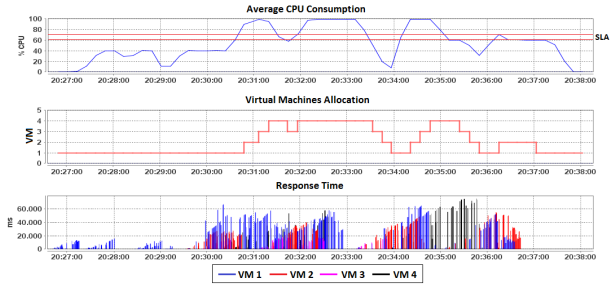


Fig. 5: Average CPU usage (%), allocation, and response time of virtual machines (miliseconds) graphs for Experiment 2

The proposed architecture enabled a mechanism that whenever more resources are needed, as defined capacity, such resources have been added, to allow the maintenance of the defined SLA. When resources were no longer needed, they were deallocated, avoiding idleness and waste, performing elastic actions and maintaining the defined SLA in most cases.

V. Conclusion

The use of Autonomic Computing in cloud environments to support the elasticity provisioning have proven very effective. We propose an autonomic architecture for providing elasticity in computational clouds. Experiments for validating the architecture show it is possible to use Autonomic Computing and cloud computing along with various technologies and different providers. As future work for evolution of the proposed architecture, different criteria should be used for rules design, such as average response time of requests. Finally, large-scale experiments should be performed to validate the architecture.

References

[1] N. R. Herbst, S. Kounev, and R. Reussner, "Elasticity in cloud computing: What it is, and what it is not," in *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013)*. USENIX, 2013.

[2] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[3] E. Coutinho, D. G. Gomes, and J. D. Souza, "Uma proposta de arquitetura autonômica para elasticidade em computação em nuvem," in *IV Workshop de Sistemas Distribuídos Autonômicos - WOSIDA2014*, 2014.

[4] ——, "Elasticity provisioning in hybrid computational clouds based on autonomic computing concepts," in *V Workshop de Sistemas Distribuídos Autonômicos (WOSIDA2015)*, 2015.

Energy Efficient Heterogeneous System for Wireless Sensor Networks (WSN)

José Anderson Rodrigues de Souza,
Teles de Sales Bezerra,
Saulo Aislan da Silva Eleutério

Federal Institute of Education, Science and Technology
of Paraíba - IFPB, Campina Grande - Brazil
Email: andersonrodrigues, teles@ieee.org
saulo_eleuterio@ieee.org

Jerônimo Silva Rocha

Federal Institute of Education, Science and Technology
of Paraíba - IFPB, Campina Grande - Brazil
Institute of Advanced Studies on Communications - IECOM
Email: jeronimo@iecom.org.br

***Abstract*—Mobile devices are increasingly occupying sectors of society and one of its most important features is mobility. However, the use of mobile devices is subject to the lifetime of the batteries. Thus, the use of energy batteries has become an important issue in the study of wireless network technologies. In this context, new solutions that enable aggregate energy efficiency not only through energy saving, and principally they are evaluated from a more realistic model of energy discharge, if easy adaptation to existing protocols. This paper presents a study on the energy needed and the lifetime for Wireless Sensor Networks (WSN) using a heterogeneous network and applying the LEACH protocol.**

I. Introduction

The consumption of electrical energy is becoming an increasingly indispensable with the passing years, through it is possible to perform activities essential to the welfare, safety and economy of a population. In this context, energy consumption grows in proportion to the different ratio to production and storage of energy due to the not meeting the demand.

Diverse applications have been developed using one or more types of sensor nodes. A Wireless Sensor Network (WSN) is comprised of a large number of sensor nodes distributed in one area of interest, interconnected by a wireless communication technology. Each sensor node can acquire relative data on the state of the variables of interest monitored environment, including, for example, temperature, pressure, humidity, noise and luminosity [1].

WSNs can be homogeneous or heterogeneous as regards those types, dimensions and functionality of the sensor nodes. Another characteristic of this application is the large volume of data and the frequency of collection. If the sensor nodes are responsible for processing the images collected can be considered that these nodes will be larger than those of us micro-sensors, i.e., due to the effort demanded by the operations involved with image processing, sensor nodes should present a higher processing power, larger amounts of memory and power consumption.

978-1-4673-9408-6/15/$31.00

One of the biggest advantages in using the WSN is the possibility of having small devices powered by sources such as batteries, making possible the constant feeding of the device for long periods of time considered as months or even years. Utilizing a system sleep state mode, the device spends most of its time in standby mode, which causes the increased autonomy of feeding. Managing and energy conservation are critical functions in sensor networks and there the need to design protocols and algorithms that optimizes energy usage in sensor nodes [2].

The hierarchical structure may be formed by two types of networks, homogeneous or heterogeneous. In homogeneous networks, the nodes present the same characteristics, such as power capacity, processing and radio. In heterogeneous structures, some sensor nodes may present different hardware requirements, such as better energy capacity. These sensor nodes, the network gives the longest period of stability.

II. Related Works

The majority of work is related to the better manageability of energy resources. In [3] other effective solution is proposed, based on Markov mechanism to predict energy consumption of a sensor node in order to build the energy map. Consumption of residual energy concerns the network performance and the ultimate objective is to the extend the life of the network. The authors in [4] have implemented a protocol called IRDT (Intermittent Receiver-driven Data Transmission). This protocol was projected for a real product in developing running battery power. The system that uses IRDT protocol can be operated for a long period of time. With the objective of suppressing the mean of energy consumption, but the differences in energy consumption still exist among us.

The authors in [5] analyzes the basic distributed clustering routing protocol LEACH, which is a homogeneous system then proposed a new routing protocol and data aggregation method in Leach heterogeneous system which the sensor nodes form the cluster and the cluster-head elected based on the residual energy of the individual node calculation with re-clustering scheme is adopted in each cluster of the WSNs.

III. Methodology

During the process creation / adaptation of the network conducted a study of the major points that were essences for the full functioning of the tests, including:

1) Them available sensor nodes in the network are within range wireless communication to communicate with each other or the base station.
2) All available sensor nodes are heterogeneous behavior in the sensing, communication and other capabilities.
3) The random distribution system is applied to create the topology WSN.
4) The position of the base station is located in the center the network sensor and its energy resource is infinite.
5) All sensor nodes available in the network have the same initial energy.
6) The principal factor in the total lifetime of the network is defined as the time from implantation to the instant when the first sensor node dies or when the integer sensor nodes die.

The proposed routing scheme for a network that uses the LEACH protocol heterogeneous is constituted having a specified number of sensor nodes, the formation of groups and well as selection of cluster head that will compare the residual energy of the individual in each round.

IV. Results

In the simulation the network is represented by 100 nodes, where was distributed in an area of 100 X 100 meters. In which the base station is located in the center of the network is the only fixed point in the network, the network nodes have random distances to each simulation. The base station is located at coordinate (50, 50) of network and nodes are randomly distributed. The simulation presents a survey of consumer spending energy in a network that uses the LEACH protocol through of energy consumption is possible to elect a leader node of the cluster by checking the lifetime of each node.

As described in the LEACH protocol, each cluster has a leader, and every round a cluster with different dimensions is created, based on the residual energy of the cluster. Each cluster is created to receive all the nodes of your network not leaders and that the energy consumed by the leader should be the minimum possible to receive and send packets to the base station, Figure 1.

In Figure 2, the network is illustrated to represent the dead nodes in the network by red, in order to check the distance between the base station and the leader node in the round. The measure the rounds increase the number of dead nodes grow due to energy is ending.

V. Conclusions

In this paper the problem of energy consumption in wireless sensor networks using LEACH protocol in heterogeneous model was presented. Based in the cluster, the balance of energy and lifetime of the network, with the primary objective to reduce energy consumed in each group in order to improve

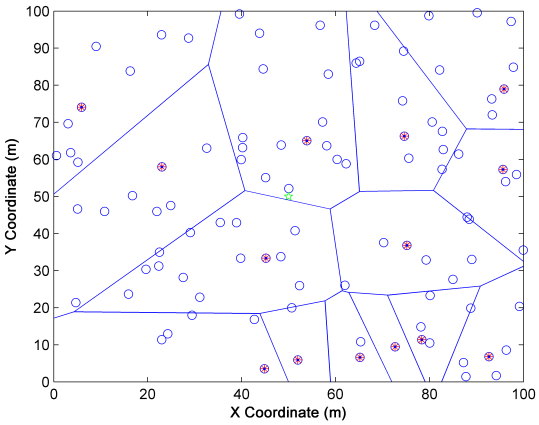


Fig. 1. Formation of groups.

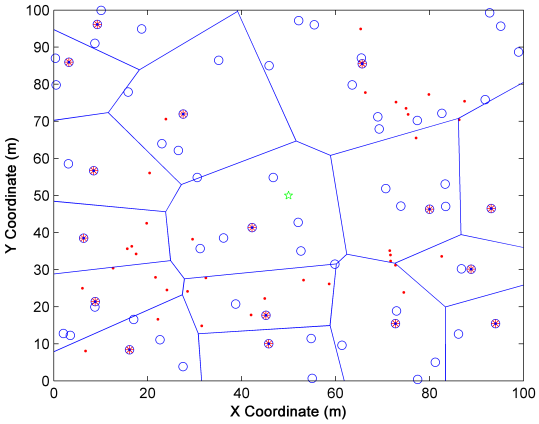


Fig. 2. The network containing nodes dead.

the level and network performance. However, the most efficient differential of the cluster head in the election, by its location and energy level differences, but also in the inclusion new levels of heterogeneity, allowing increase the period of stability of the network.

References

[1] I. Akyildiz, W. Su, and E. Cayirci, "Wireless sensor networks: A survey," in Computer Networks Journal, Vol. 4, New York USA, 2002.

[2] M. P. Sousa and W. A. Lopes, "Desafios em redes de sensores sem fio." revista de Tecnologia da Informação e Comunicação, V.1, p. 41-47, Out. 2011.

[3] R. A. F. e. a. Mini, "Prediction-based energy map for wireless sensor networks." dept. of Electrical and Ad Hoc Net. J., vol. 3, pp. 23553, 2005.

[4] C. Damdinsuren, M. Sugano, and T. Hatauchi, "Lifetime extension based on residual energy for receiver-driven multi-hop wireless network." in Graduate School of Information Science and Technology, Osaka University, 2011.

[5] R. Saravanakumar, S. S. G, and J. Raja, "Energy efficient homogeneous and heterogeneous system for wireless sensor networks," in International Journal of Computer Applications, Volume 17 No.4, March 2011.

Implementing Smart Grid with a CIM-oriented Integration and Data Acquisition Gateway

João Paolo C. M. Oliveira, Vagner Henrique de Souza, Antonio Wendell de O. Rodrigues
Rejane Cavalcante Sá, and Paulo Régis Carneiro de Araújo
Federal Institute of Ceará - Fortaleza, CE, Brazil

***Abstract*—Electric power industries have expanded the automation of their networks in recent years to meet the growing demands for improvements in services. The introduction of the concept of Smart Grid, the increased availability of smart devices and improvements in telecommunications are key factors for that. Besides the benefits in control and management that this concept brings, there are some technical restrictions for the adoption due to legacy or multiple vendor equipments, each one with its own standard, as well as interoperability with supervisory and management softwares. This paper, then, proposes the use of a gateway to enable interoperability between devices using different communication protocols by translating them, centering data and control in a database defined by the Common Information Model(CIM) standard.**

I. Introduction

The electric power industries are undergoing a process of modernization on their structures to meet consumer demands for better quality and service availability. The concept of Smart Grid[1] has emerged as a solution to meet the diverse needs of electric power industries and their consumers. Bringing together concepts of data communication, smart electronic devices, microgeneration, energy storage and information technology, it has enabled the creation of a more secure, efficient, flexible and resilient power grid.

However, the benefits from the Smart Grid also creates challenges to be overcome. The major challenges are related to the acquisition, integration and handling the large volume of data made available by electronic devices that supports communication as well as extraction of useful information from this great mass of data created.

The acquisition problem occurs due to old legacy systems or the vendor strategies for ”customer lock-in” through use of proprietary data models, methods and tools. This problem, on our solution, was solved by providing a extensible translation layer for communication protocols and data models, so that data can flow independent of its origin, allowing integration.

To handle the data, we use the Common Information Model (CIM) applied to a database structure.

The Common Information Model (CIM), a standard developed by the electric power industry that has been officially adopted by the International Electrotechnical Commission (IEC) for power transmission and distribution. The purpose is to allow application software to exchange information about an electrical network[2]. The CIM is object-oriented, can be extended, and consist of classes, attributes and the relationships among them, to describe the behavior of each and every electrical system resource, everything defined in Unified Modeling Language (UML) notation. The main purpose of the CIM is to provide a common language to semantically describe resources and data exchanged among systems[7]. It's usage is supported by the standards IEC 61968, information exchanges between electrical distribution systems[3], and IEC 61970, wich deals with the application program interfaces for energy management systems[4].

978-1-4673-9408-6/15/$31.00

II. Theoretical Revision

The existing devices in the electrical distribution network have intrinsically heterogeneous characteristics. The various manufacturers and solutions result in a mix of proprietary interfaces and software tools that require diverse training and vendor dependency. In addition, the safety factor is decentralized, which causes the need for individual data model and passwords for each device.

All these factors prevent the optimization of the structured management of these equipments. Although the market has proposed solutions that enable the inclusion of certain devices to legacy protocols (Modbus, CAN, etc.), or even create simple translaters, there is a strong demand to have a single data model to store and work with the data. This is the reason of existence of Common Information Model (CIM).

Although the standards-based systems integration of utility software applications using the domain data standard common information model (CIM) exists, many utilities are facing difficulties in managing of data exchanges among software application systems due to lack of standard oriented database management systems. The standard oriented database can be built from the CIM object oriented model using object-relational mapping (ORM)[6] on a database.

With this topology, the integration of devices that are not compatible with the current standards related to Smart Grid structures is made with a lower cost and scalability. Additionally, manageable elements inherent to the devices are listed in a tree of objects allowing read access and/or writing through the operation center. This is of fundamental importance in the current structure of power supply, placing network operation with high levels of responsiveness and performance generating better satisfaction in service.

III. Functional Description of the Gateway

The gateway is designed to be highly transparent and modular to provide flexibility and high scalability. Allowing to adapt more easily to the data models, communication

technologies and devices with non-standard interfaces. Monolithic structures tend to become closed, making the system integration more difficult in heterogeneous structures.

The gateway runs as a serie of services on a POSIX based operating systems (Linux, BSD), providing protocol libraries related to the Smart Grid (IEC61850, DNP3, Modbus) and a DBMS (Database Management System) that will implemente CIM model and be a central point of application. Other then direct SQL query to database, the whole gateway is defined in a client/server architecture of the TCP/IP stack as well as a WebServices to ease the extension of the architecture. An API is provided to easily add protocols and direct communication between new devices.

A. Internals

In the substation control center, all the control and supervision data are available and carried by a SCADA (Supervisory Control And Data Acquisition) software, which performs readings such as voltage, current, circuit status, and switch positions as writings, activating and deactivating elements of network.

The SCADA communicates with the gateway using a standard communication protocol, like DNP3, IEC 61850 or MODBUS, or via Webservice.

The communication between the SCADA and gateway is started by invoking one of the methods GWDNP3(), GWModbus(), GW61850(), GWws() to use, respectivelly, DNP3[5], MODBUS, [8]IEC61850 and Webservice for communication, all belonging to ”core” module on gateway.

When a request comes from SCADA, the gateway selects the appropriate protocol translator and, from this point on, all data and command are converted for a intermediate internal protocol, through the methods ”encode()” and ”decode()” standardized to, respectively, encode and decode a frame from any protocol. Then, if the SCADA request was a reading, a SQL select is issued on database and the result, after retranslated, sent back to SCADA. If, otherwise, an actuation is performed, an SQL insert is made into database. This change triggers a database object that calls a script that send an actuation command to the device.

Every time a new element connects to gateway via a microserver, its protocol is selected on the translating list and the communication starts. These elements can be any hardware, using any protocol that is already implemented, as TCP/IP or RS232 microservers. After that, it invokes a method ”anounce()” from ”core” module. So, all communication structure is created, including instantiation of internal protocol so that readings becomes possible. Then, a field of the database will be set, the device will be seen as ACTIVE from SCADA and will be already able to receive commands and to send readings.

Besides ”anounce()” and the communication methods, four other control methods are implemented on ”core” module of gateway. The first one is the ”sendResponseBack()”, whose purpose is to collect the answer for the query on the repository, regulate it, normalize it and send it back to the database. The second one is ”getMyQueries()”, used by microservers in fixed time cycles to get, from database, requests to the elements directly connected to them. The third method is the ”resposePush()”, also using by microservers, to send the answers to queries to be translated, processed and stored in database. The fourth and final method is the ”queryPush()”, invoked by the microserver to force a special send to database, in an emergency scenario or, in case of a parameter previously set to inform its modification (alarm), is triggered.

These methods from module ”core” are required to create and configure devices and allow a flow of communication and control between the SCADA and any type of device.

IV. Conclusion

This work has presented an integrator gateway that, in adition to create an easy extensible protocol translator and multilink communication solution, proposes a CIM-oriented database as a core of electric system integration. Part of this work has been the solution implemented on an electrical company in the state of Maranhão, Brazil to integrate with some network elements. The research that led to a previous version of this gateway and is part of a bigger solution to integrate and give inteligence to legacy devices on the operation center of the company.

The model is evolving. We are creating different triggers to the database and focusing on data analysis to foresee status of power grid and create a more general model.

Acknowledgment

The authors would like to thanks CEMAR (Maranhão State Electric Company), CELPA (Pará State Electric Company), IFCE (Federal INstitute of Ceará and CNPq (National Counsel of Technological and Scientific Development).

References

[1] NIST, *Smart grid conceptual model - national institute of standards and technology*, 2013. [Online]. Available: http://collaborate.nist.gov/twiki-sggrid/bin/view/SmartGrid/SGConceptualModel

[2] J. SIMMINS. *The impact of PAP 8 on the Common Information Model (CIM)*. In: Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES. IEEE, 2011. p. 1-2.

[3] Working Group 14 of Technical Committee 57, *System Interfaces for Distribution Management Part 11: Distribution Information Exchange Model*, International Standard IEC 61968-11, Geneva, Switzerland, International Electrotechnical Commission, 2002

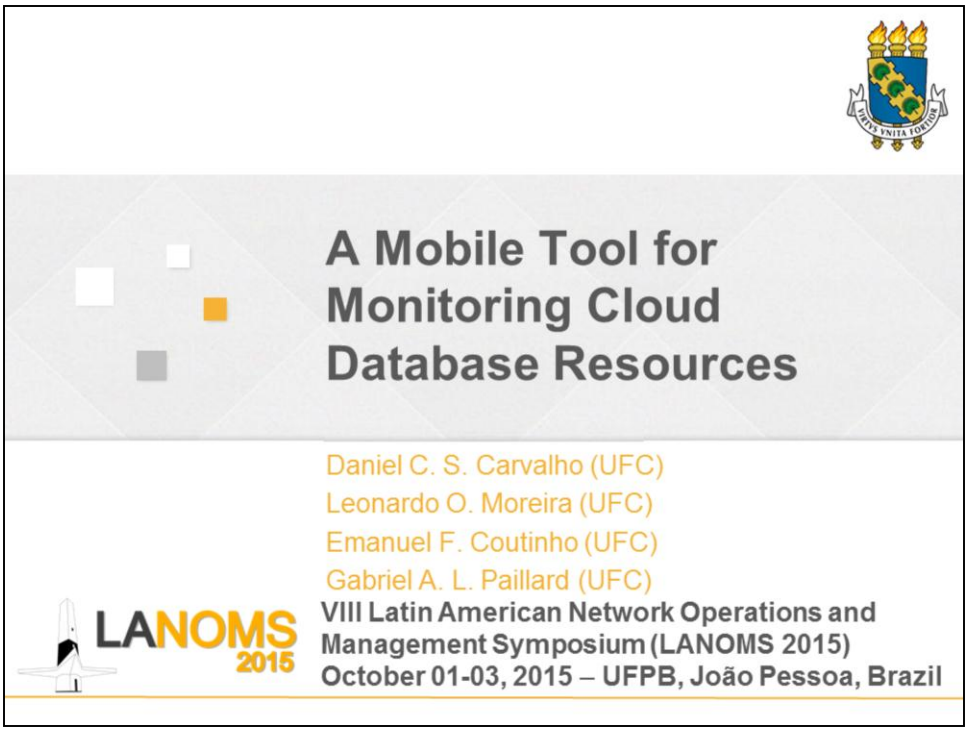
[4] Working Group 13 of Technical Committee 57, *Energy Management System Application Program Interface (EMS API) Common Information Model (CIM)*, International Standard IEC 61970-301, Geneva, Switzerland, International Electrotechnical Commission, 2003.

[5] IEEE Power and Energy Society. *IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010) IEEE Standard for Electric Power Systems Communications Distributed Network Protocol (DNP3)*. 2012.

[6] G. Ravikumar, S. A. Khaparde, & Y. Pradeep. (2013, July). *CIM oriented database for topology processing and integration of power system applications*. In Power and Energy Society General Meeting (PES), IEEE (pp. 1-5). 2013.

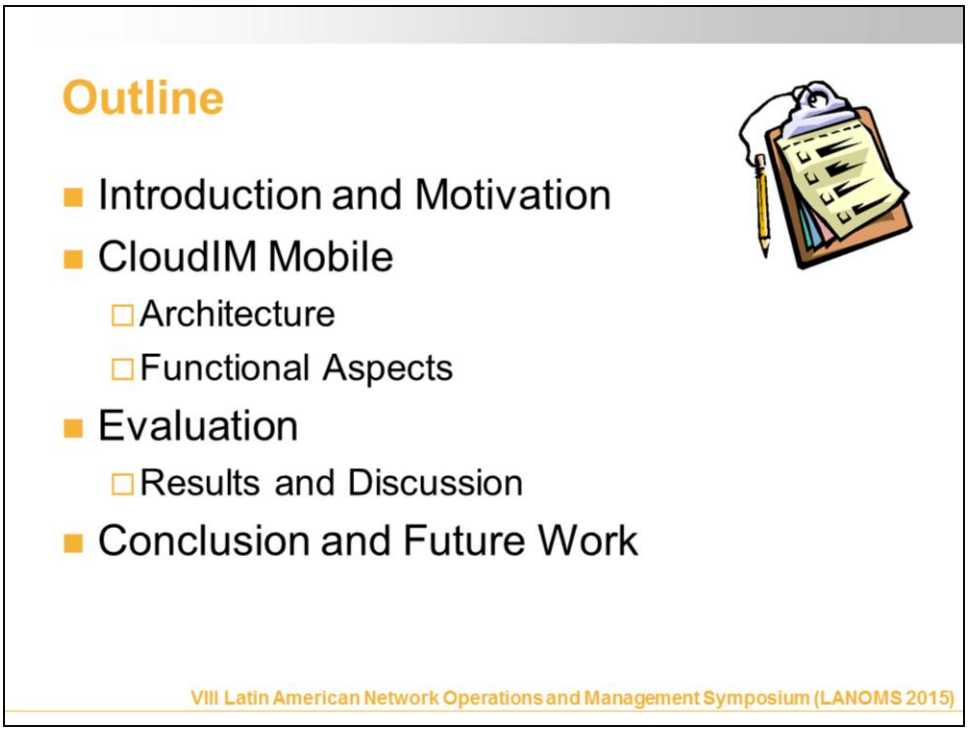
[7] J. Wu and N. Schulz, Overview of cim-oriented database design and data exchanging in power system applications, in Power Symposium, 2005. Proceedings of the 37th Annual North American, Oct. 2005, pp. 16 20.

[8] T. Kostic, O. Preiss, and C. Frei *Understanding and Using the IEC 61850: A Case for Meta-Modelling*, 2005, Comput. Stand. Interfaces 27, 6 (June 2005), 679-695.

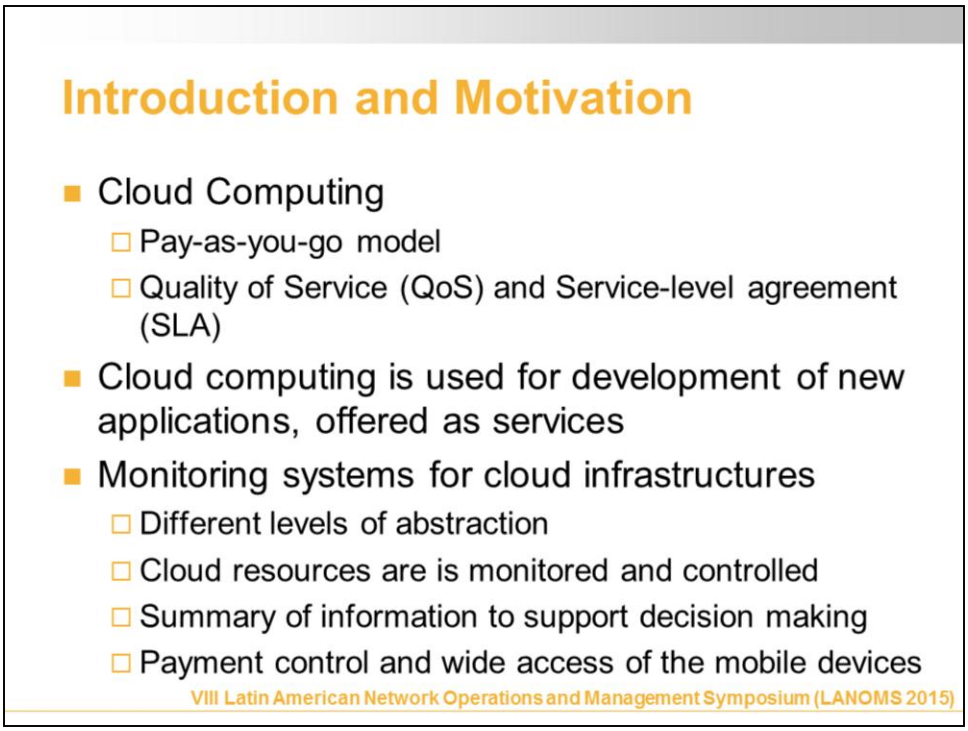


Keywords: Mobility management, Quality of Service, Cloud Databases

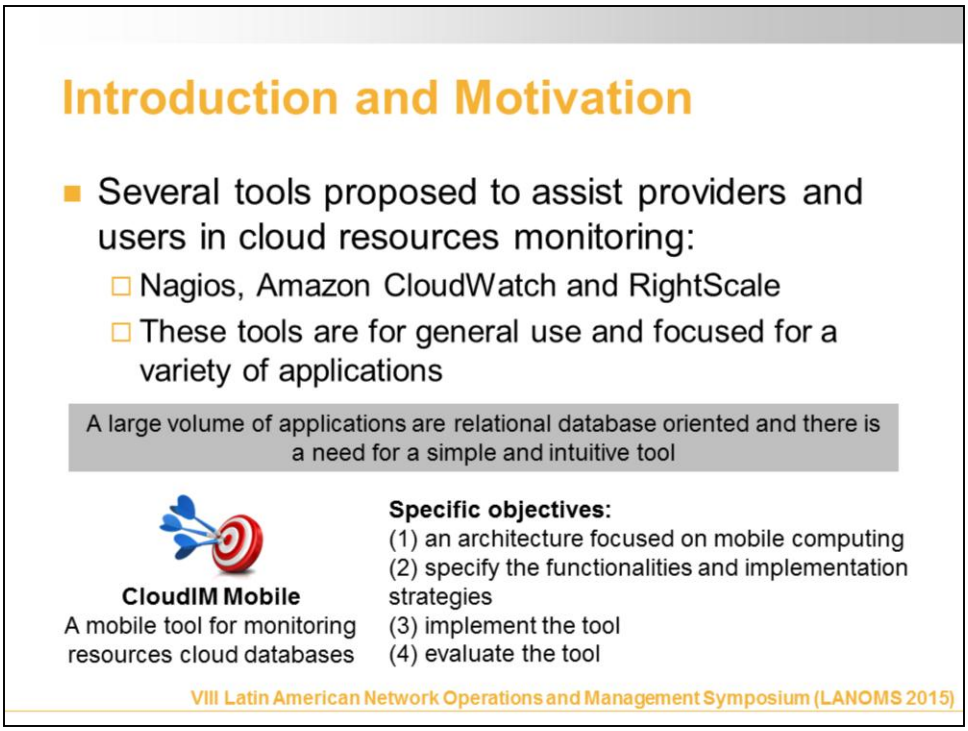
Abstract: Cloud computing is a paradigm for the use of computational resources, where the computational stack is offered as a service. Many applications are data-driven, and the cloud providers use strategies to consolidate multiple databases within a single DBMS, encouraging the sharing of resources. In the cloud, users pay for the use of infrastructure, and this use should be controlled to not generate financial impacts. And at the same time, it is necessary to check whether the provider is meeting the specified SLA. Therefore, it is important to use a tool that monitors under the perspective of cloud computing applications that use databases. This work aims to develop a tool to mobile devices that monitors a cloud infrastructure intended to database applications.



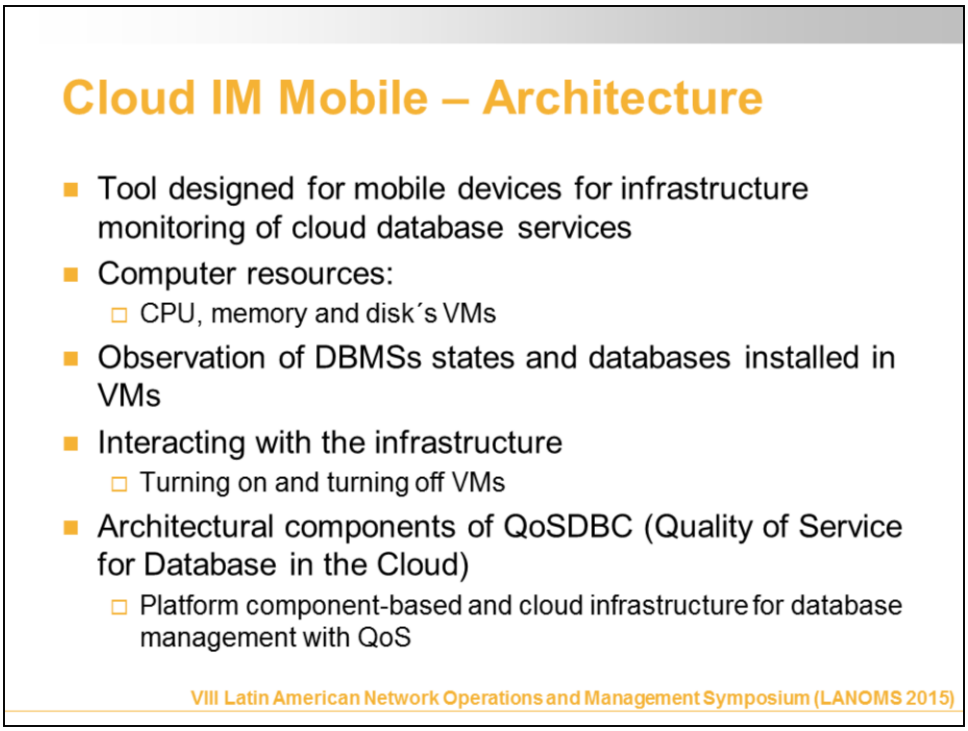
This paper describes the CloudIM tool. Initially, the introduction and motivation of the work will be presented, followed by the description of the tool highlighting its architecture and functional aspects. Hereafter, an evaluation of the tool will be presented, with the results and discussion. Finally, we ended the work with the conclusions and future directions.



Cloud computing is now a well-established paradigm to use computing resources, whereby hardware and software platforms for the development of new applications are offered as services available remotely and on a global scale [1]. Computational cloud users give up an own computing infrastructure to arrange it by services offered by third parties (cloud providers), delegating responsibilities and assuming costs proportional to the amount of used resources (pay-as-you-go model). Users aim at reduce the financial costs while use the resources needed for their applications. Providers aim at improve the use of resources by virtualization and multitenant. Providers and cloud computing users control and optimize the use of resources through monitoring systems [2]. Monitoring is performed at some level of abstraction appropriate for the service type, such as storage, processing, bandwidth, interference between applications, etc. However, the use of cloud resources is monitored and controlled, enabling transparency for stakeholders. In the context of this work, stakeholders refer to cloud members, including providers, developers applications for these environments, and the service end user, summarizing information to support decision making. To ensure quality of service (QoS), it can be used an approach based on service-level agreement (SLA) [3]. The SLA provides information on availability, functionality and performance levels, or other attributes of the service such billing and penalties for violation of these levels.

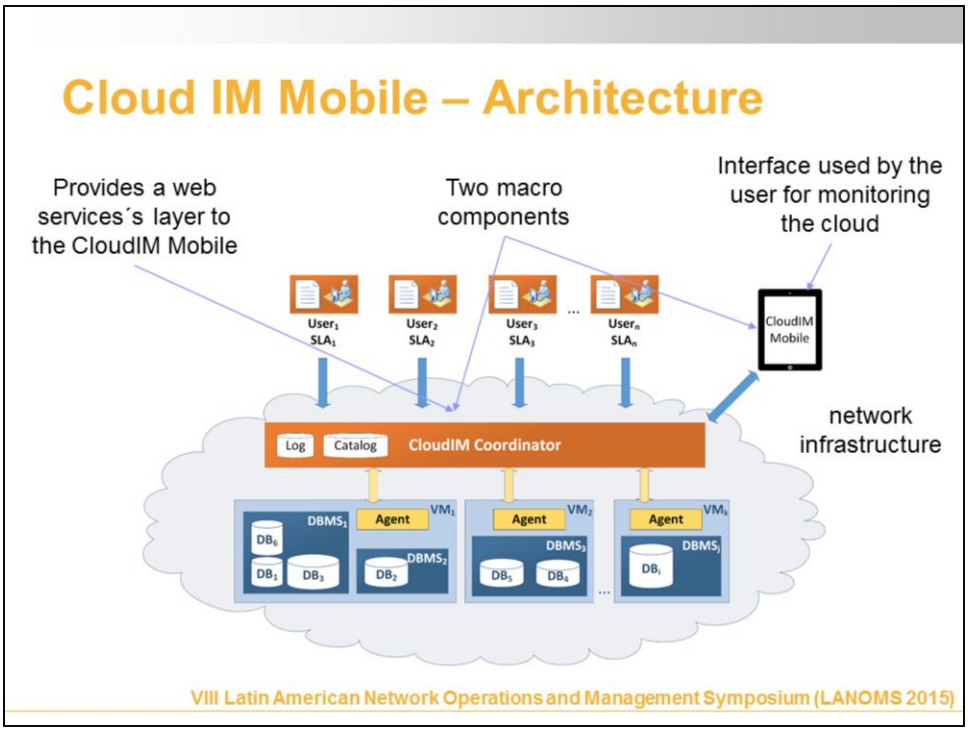


As the cloud uses a payment model based on its use, it becomes necessary the adoption of monitoring tools, not only for the provider, but also for users who deploy their applications in computational clouds. Several tools (e.g. Nagios, Amazon CloudWatch and RightScale) were proposed to assist providers and users to monitor cloud's resources. However, these tools are for general use and focused for a variety of applications, which make it fairly complex to inexperienced users. As a large volume of applications are relational database oriented [4], there is a need for a simple and intuitive tool for the monitoring of clouds from the perspective of less technical user's profile. In addition, a tool oriented for mobile devices enable a wide access due to mobility and aggregate interactivity on applications in this context. This article presents a mobile tool for monitoring resources cloud databases, called CloudIM Mobile. Some experiments about the user experience based on the use of the tool were conducted to validate the proposed work. To achieve the overall goal of this work, we design four specific objectives: (i) developing an architecture, focused on mobile computing, with the main purpose to serve as a structural base implementation; (ii) specify the functionalities and implementation strategies of each architectural component; (iii) implement the tool to show in practice monitoring cloud data services; and (iv) evaluate our proposal under the user viewpoint software satisfaction.

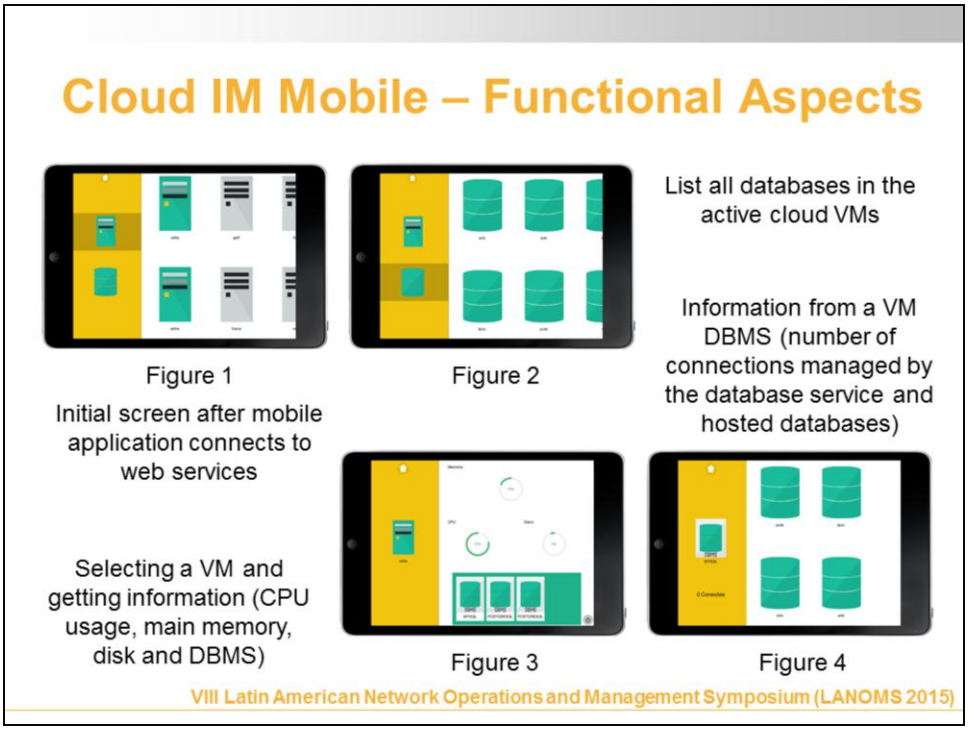


The CloudIM Mobile is a tool designed for mobile devices to monitor cloud database services. In this context, the tool monitors computer resources such as CPU, memory and disk´s VMs (Virtual Machines). Moreover, it becomes necessary the observation of the states of a DBMS (DBMS refers to a general class of data storage, including non-relational systems) and databases that are installed in VMs. Another important feature is that the tool beyond monitors, can interact with the infrastructure, turning on and turning off VMs.

The CloudIM Mobile uses the architectural components of QoSDBC (Quality of Service for Database in the Cloud) [5]. The QoSDBC is a component-based platform and cloud infrastructure for database management with QoS.



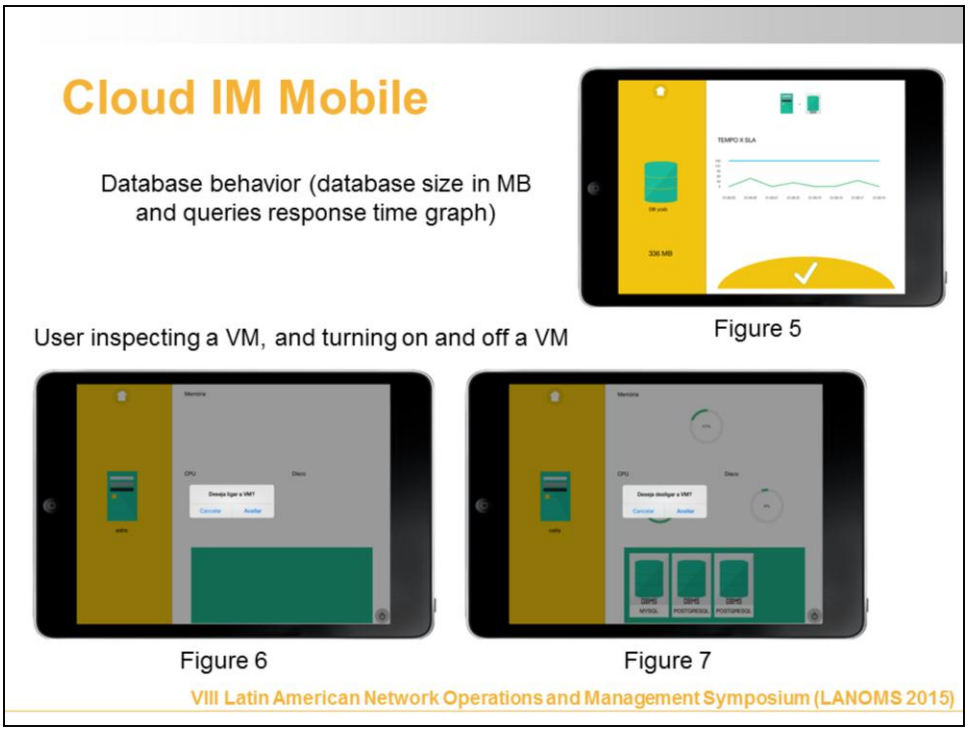
The CloudIM Mobile uses the components Log, Catalog and Agent, which are QoSDBC system resources monitoring. The CloudIM Coordinator component is responsible for providing a web services´s layer to the CloudIM Mobile. It is worth noting that the CloudIM Mobile is the interface used for cloud monitoring. Inside each VM active in the cloud infrastructure, there is an installed agent program. Once a VM is started, this agent runs in the background and stay monitoring the computational resources of the VM, beyond the installed DBMS and databases that are in each DBMS. Monitoring information of active and inactive VMs are arranged in the component catalog. This component also has information on which DBMSs are installed, which are the databases and their SLAs. When queries are executed and there are changes in the number of connections's DBMSs, the catalog is also updated. Information on the queries response times are available on component Log. The interaction with the cloud infrastructure is made through the invocation of services Web deployed in the CloudIM Coordinator. It is worth noting that the user´s CloudIM Mobile may be responsible to perform infrastructure management. However, considering the characteristic of payment based on use, the cloud contractor can make use of CloudIM Mobile to perform the monitoring of computing resources which are being used, and generate interventions or cloud settings.



To guide the tool explanation, we will describe the main screens and theirs main features. Figure 1 displays an initial screen after the mobile application connects to the web services deployed in the CloudIM Coordinator. Furthermore, there is an option to list all existing databases in the active cloud VMs, i.e., all databases from all VMs that are set on. This option can be observed in Figure 2.

Figure 3 shows what can be seen when it is selected a VM. If the VM is on, the user can get the information about CPU usage, main memory, disk and which the DBMSs are installed in that VM. In addition, there is an option for the user to turn off a VM connected or turn on a VM (if it is turned off).

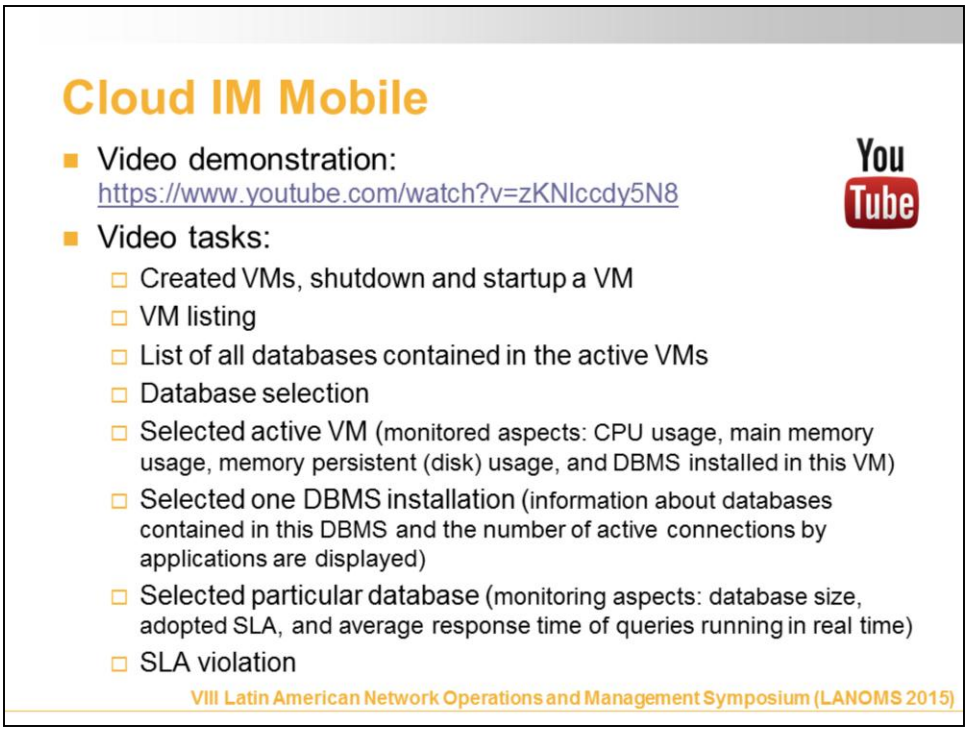
Figure 4 highlights the information obtained when a DBMS of a VM is selected. Among the information displayed are included: number of connections that the database service is managing and all databases that are hosted in the DBMS.



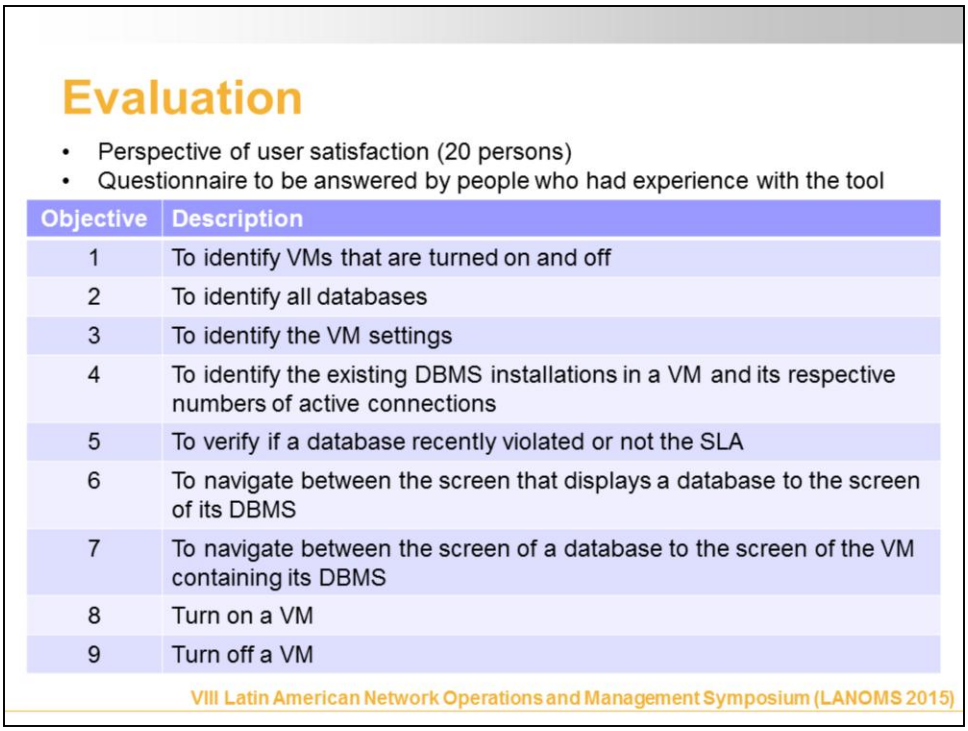
From Figure 5 it can be selected the database and to observe its behavior. Figure 5 shows the information obtained after selecting a database. As information obtained, we can highlight the database size in megabytes and a graph based on queries response time, that allows the user to check if the load running in the database is violating or not the SLA.

Figure 6 can be achieved when the user are inspecting a VM. If the VM is off, the user can turn it on, and when QoSDBC monitoring service sends the VM is ready, this information is propagated to the CloudIM Mobile.

If the VM is on, the user can turn off using the interface from Figure 7. This option to turn off a VM is propagated to the services of CloudIM Coordinator which sends a signal to the agent to turn off the machine. At the end of the procedure, the mobile interface is updated.



For a better understanding of the features and viewing execution of the CloudIM Mobile, a video showing its functional aspects is available in https://www.youtube.com/watch?v=zKNlccdy5N8. In the video, the CloudIM Mobile is running in a simulator (iOS Simulator - iPad, version iOS 7.1). To guide the demonstration we created some VMs and some of them were turned off to show their states on the application initial screen. After the VM listing, it was shown a list of all databases that are contained in the active VMs. One of the databases was selected to demonstrate the inspection function and the state of a database viewing. One of the active VMs was chosen to display the monitored aspects of each VM, such as CPU usage, main memory usage, memory persistent (disk) usage, and DBMS installed in this VM. Finally, it was shown when selected one of DBMS installations, information about databases contained in this DBMS and the number of active connections by applications are displayed. By selecting a particular database, we can see the monitoring aspects of the use of this database, such as the database size, the adopted SLA, and average response time of queries that are running in real time. On the same screen, it was forced an increase workload in the database to demonstrate a SLA violation, when the workload exceeds the accorded SLA. Finally, the video shows how to shut down a VM from the state of the VM screen. A new active list of VMs shows the selected VM turned off. After that, a VM has been chosen to demonstrate the connecting VM feature.

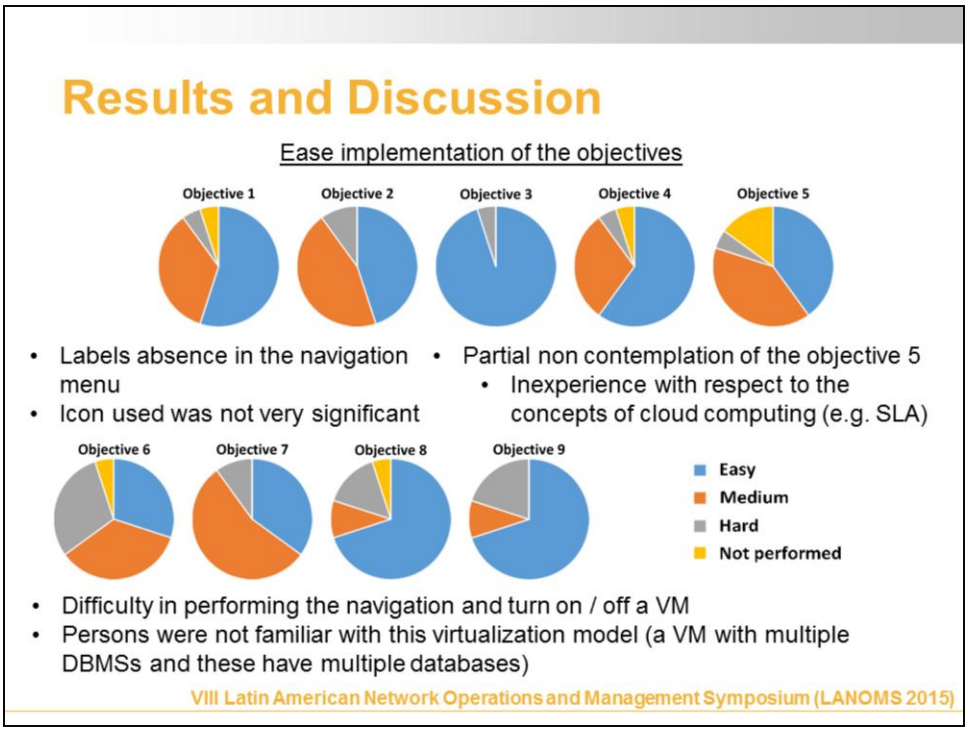


The CloudIM evaluation strategy was under the perspective of user satisfaction. We choose this evaluation approach because the goal is to verify if the mobile application meets the ease of use requirements, intuitive interface, and if the available information are sufficient for using the application and all its features. For this, it was elaborated a questionnaire to be answered by people who had experience with the tool. To guide the evaluation, some objectives were designed to be performed by respondents:

Objectives: (1) To identify VMs that are turned on and off; (2) To identify all databases; (3) To identify the VM settings; (4) To identify the existing DBMS installations in a VM and its respective numbers of active connections.

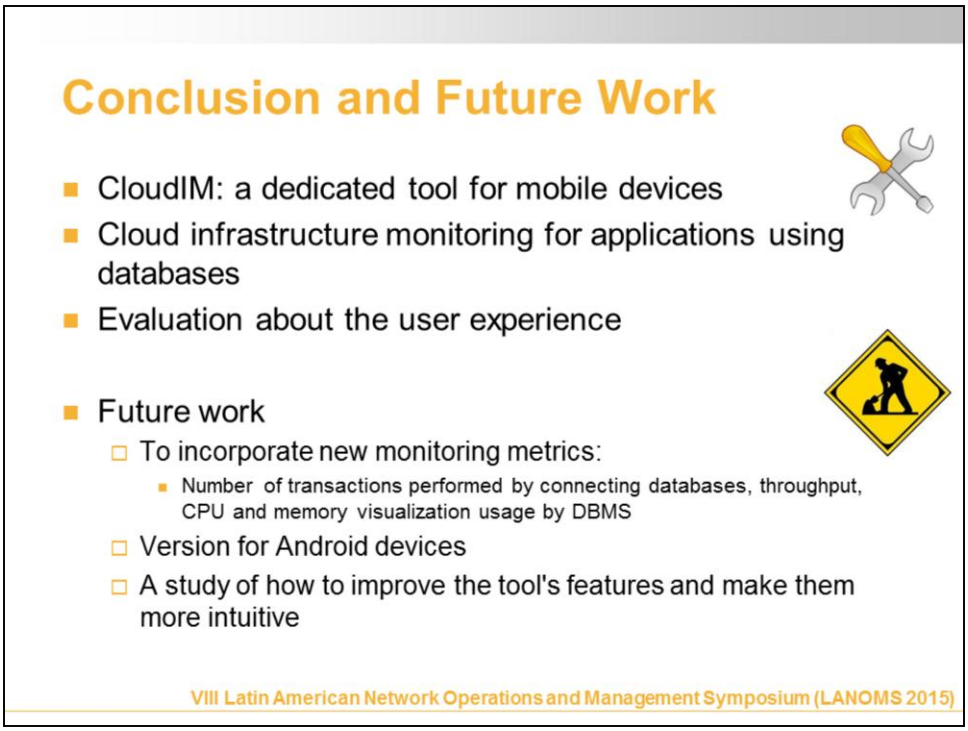
Objectives: (5) To verify if a database recently violated or not the SLA; (6) To navigate between the screen that displays a database to the screen of its DBMS; (7) To navigate between the screen of a database to the screen of the VM containing its DBMS; (8) Turn on a VM; (9) Turn off a VM.

All these objectives were observed, performed and answered according to four options: easy, medium, difficult and unfulfilled. The amount of 20 persons answered the questionnaire. Among these people, 11 are undergraduate and 9 are graduated. All persons have experience with tablets and/or smartphones. Furthermore, 18 person said they had some experience with languages programming.



The figure highlights the results achieved from the fulfillment of the objectives 1 to 5. From the results it can be observed that generally prevails the ease fulfillment of the objectives. However, there were some instances of trouble while running the second objective. One reported cause was the labels absence in the navigation menu. In addition, the used icon was not very significant as previously thought. It was also observed partial occurrence of non contemplation of the objective 5. In this case we can see the reason is inexperience of respondents with respect to the concepts of cloud computing, for example, many of them did not know what SLA was.

In the objectives 6 to 9 we identified an ease fulfillment of objectives. However, there was a greater difficulty in performing the navigation and turn on / off a VM. Regarding navigation purposes, respondents who experienced difficulties reported they were not familiar with this virtualization model, where a VM can have multiple DBMSs and these have multiple databases. An important suggestion is to explain the navigation hierarchy with labels. Regarding turn on / off a VM goals, respondents who experienced difficulties found that the icon was not very expressive and its position was inadequate. It was given as a suggestion to use an icon with another color and in a more central area of the screen.



This work presented CloudIM, a dedicated tool for mobile devices that monitors a cloud infrastructure for applications using databases, making use of a specified and implemented architecture. As future directions, we intend to incorporate new monitoring metrics, such as number of transactions performed by connecting databases, throughput, CPU and memory visualization usage by DBMS. We also aim to implement a version for Android. In future versions, a study will be performed of how to improve the tool's features and make them more intuitive. Due the cloud payment characteristics, an interesting feature would be the incorporation of prediction techniques based on workloads in the context of computing clouds. In this sense, the forecast techniques could be adapted as an additional feature in the proposed tool presented in this work.

REFERENCES

[1] Sousa, F. R. C., Moreira, L. O., and Machado, J. C. (2009). Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. In: ERCEMAPI 2009.

[2] Coutinho, E. F., Sousa, F. R., Gomes, D. G., and Souza, J. N. (2013). Elasticidade em computação na nuvem: Uma abordagem sistemática. In: SBRC 2013.

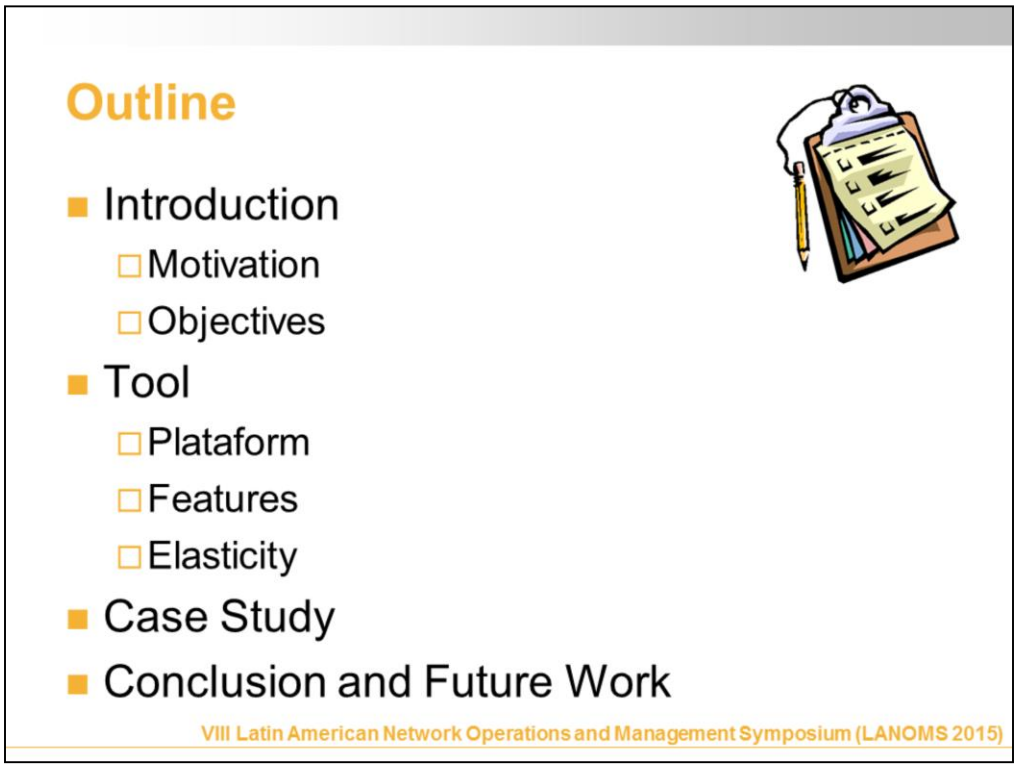
[3] Sousa, F. R. C., Moreira, L. O., and Machado, J. C. (2011). Sladb: Acordo de nível de serviço para banco de dados em nuvem. In SBBD 2011.

[4] Moreira, L. O., Sousa, F. R. C., and Machado, J. C. (2012). Analisando o desempenho de banco de dados multi-inquilino em nuvem. In: SBBD 2012.

[5] Sousa, F. R. C., Moreira, L. O., Santos, G. A. C., and Machado, J. C. (2012). Quality of service for database in the cloud. In: CLOSER 2012.

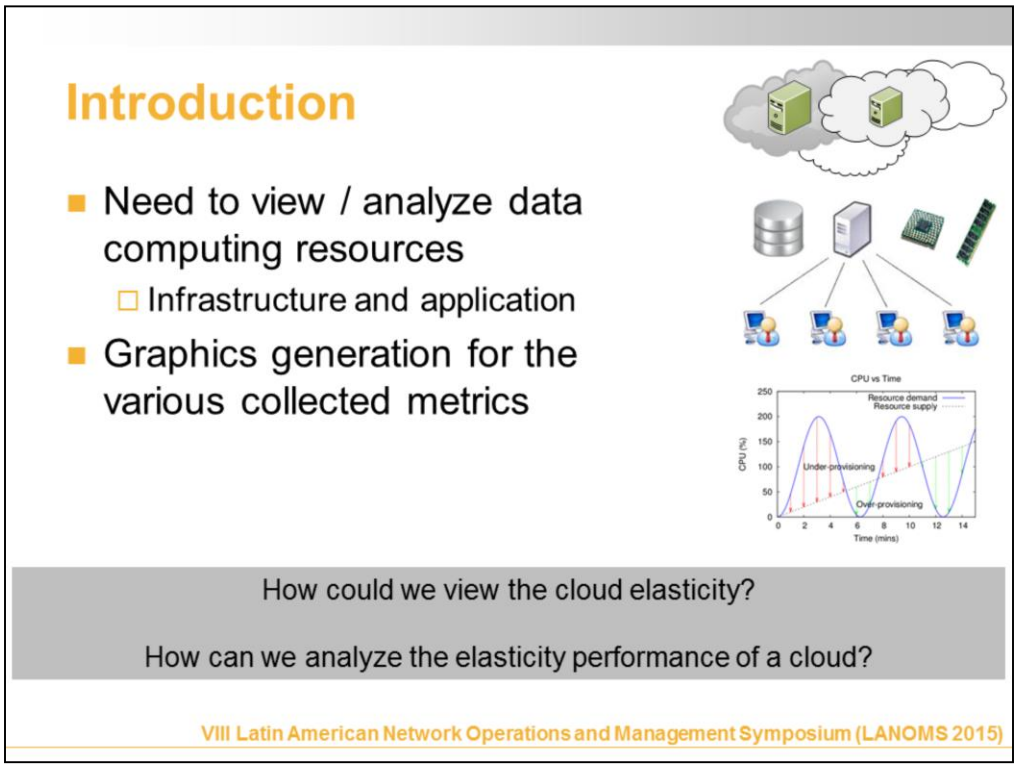
Keywords: Cloud Computing; Data Visualization; Tool; Elasticity.

Abstract: The use of cloud computing environments currently has been widely reported and used by service providers, enterprise users and academy. Therefore, computational clouds are often hired for their resources, and the promised service guarantee. One way to analyze if the contracted service is being fulfilled is monitoring resources and applications. Elasticity is one of the main characteristics of cloud computing, and usually it is associated to the use of computing resources. This work aims to present a tool for monitoring and visualization of computational cloud resources data, such as CPU usage, and highlighting the elasticity visualization.



Outline

In the next sections of this paper we presented the motivation and tool objectives, its features with the description of metrics, and types of available graphics. A case study with a performance analysis of a computational cloud is presented, focusing in cloud elasticity. Finally, conclusions and future work are discussed.



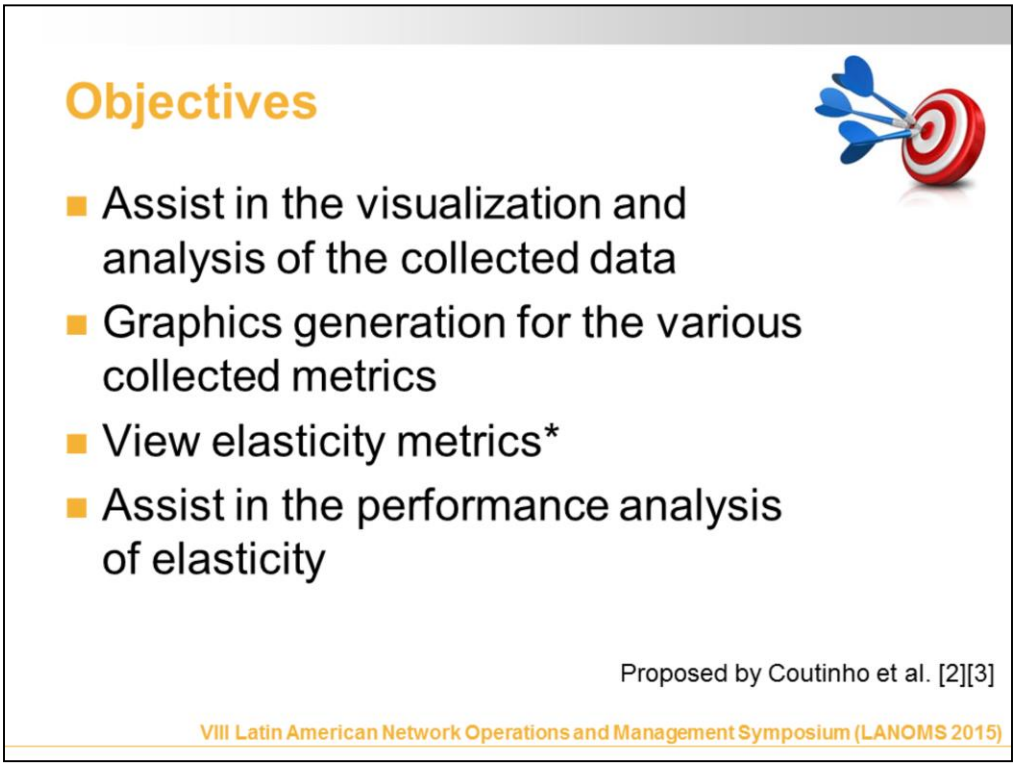
Introduction

Currently, the use of cloud computing environments has been widely disseminated and used by service providers, enterprise users, and academy. Thus, computational clouds are often hired for their resources, and by the affiance of a quality of service. One way to analyze if the contracted service is being fulfilled is monitoring resources and applications of the cloud.

In this context, the elasticity emerges as one of the main characteristics of cloud computing. Often it is associated with the use of computing resources of the cloud.

Regarding the elasticity, some issues are still complicated to answer, such as: (i) How could we view the cloud elasticity?; and (ii) How can we analyze the elasticity performance of a cloud?

Therefore, this work aims to present a tool for monitoring and visualization of computational cloud resources and application data, such as CPU usage and request response time, and highlighting the elasticity viewing.

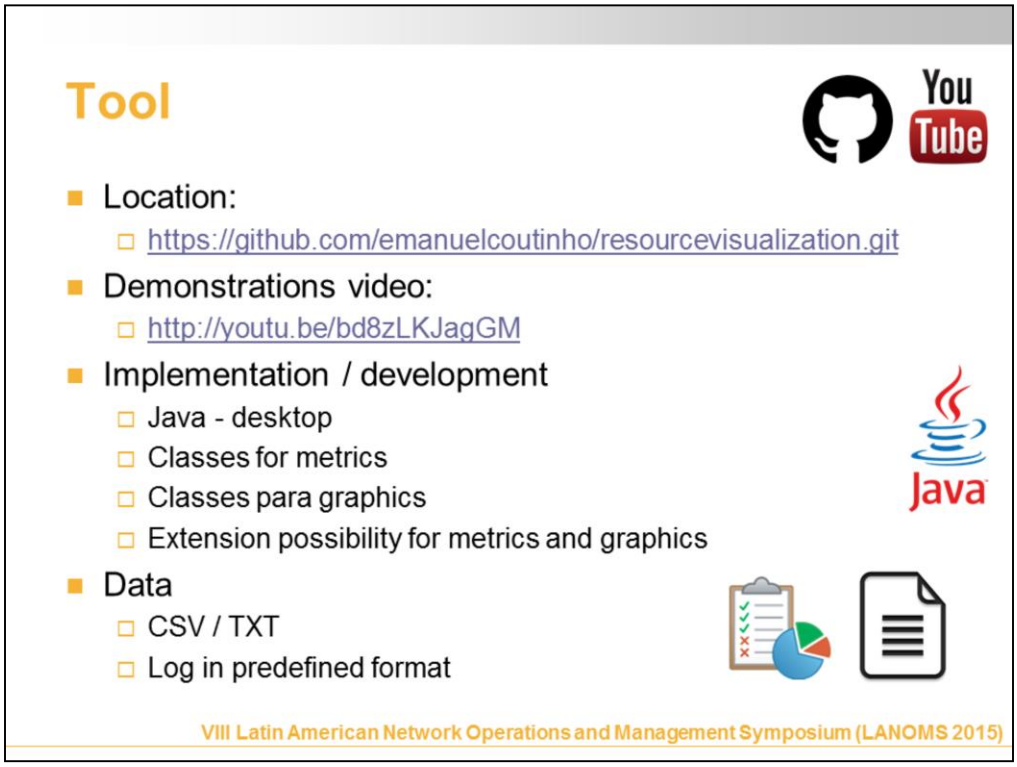


Objective

The main objective of this paper is to present a tool to aid in the visualization and analysis of data collected from a computational cloud.

This tool enables the generation of graphics for the various metrics collected from the resources and applications of the environment.

The elasticity metrics proposed in [2] and [3] can also be showed by the tool, and thus assist in the elasticity analysis.



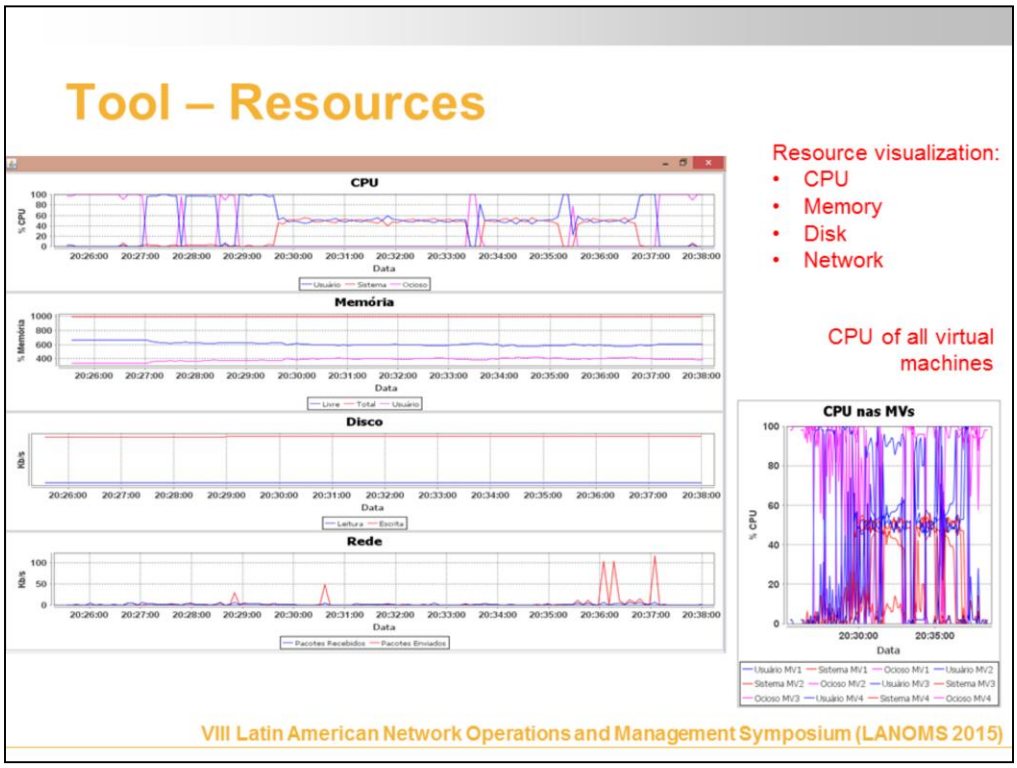
Tool

The application was developed with Java programming language. It is based on text files reading (CSV or TXT) with the information collected from the environment. These files record environment data, such as CPU usage and memory, and application data such as request response time. The generation of new metrics and graphics can be done by adding new Java classes incorporated in the tool. Further details of the tool are available in [6].

Some cited metrics could be individually showed per virtual machine, allowing the users to select which virtual machine they want to view the data. Additionally, the user can check some metrics in a parallel way. For all virtual machines it is possible to view in parallel CPU, memory, network, disk and response time.

Allocation of virtual machines and average CPU usage metrics are consolidated data, so they are from the environment as a whole, as well as the elasticity metrics (Elasticity of Resources [2] and Elasticity of Demand [3]).

The tool is available in a Github repository (https://github.com/emanuelcoutinho/resourcevisualization.git), and a demonstration video is available on YouTube (http://youtu.be/bd8zLKJagGM).



Tool – Resources

Some computational resources can be viewed through the tool. These metrics can be monitored by tool panels, where the user selects which resource he wants to view / monitor and in which panel.

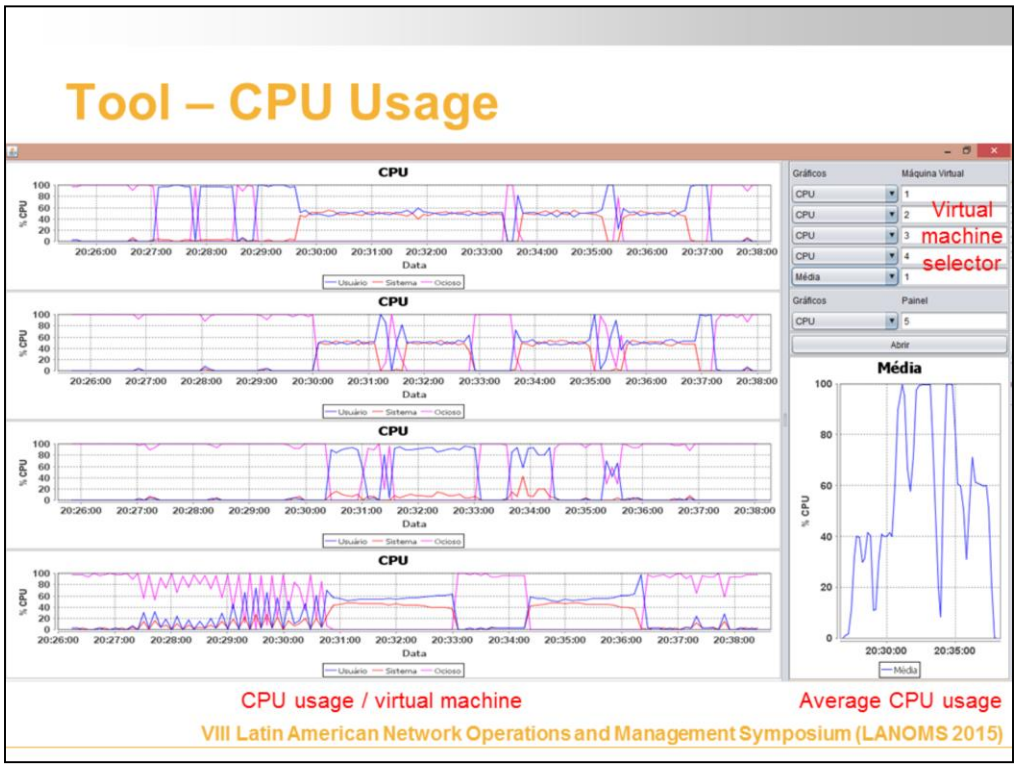
The resources that can be viewed are:

i) CPU – user CPU usage percentage, system CPU usage percentage, and idle CPU usage percentage. Ranging from 0% to 100%.

ii) Memory – the amount of free memory, the amount of total memory, and the amount of memory used by the user. Ranging from 0 up to the amount of available memory in Gbytes.

iii) Network – received packets and sent packets. In KB / s.

iv) Disk – amount of read data and amount of written data. In KB / s.



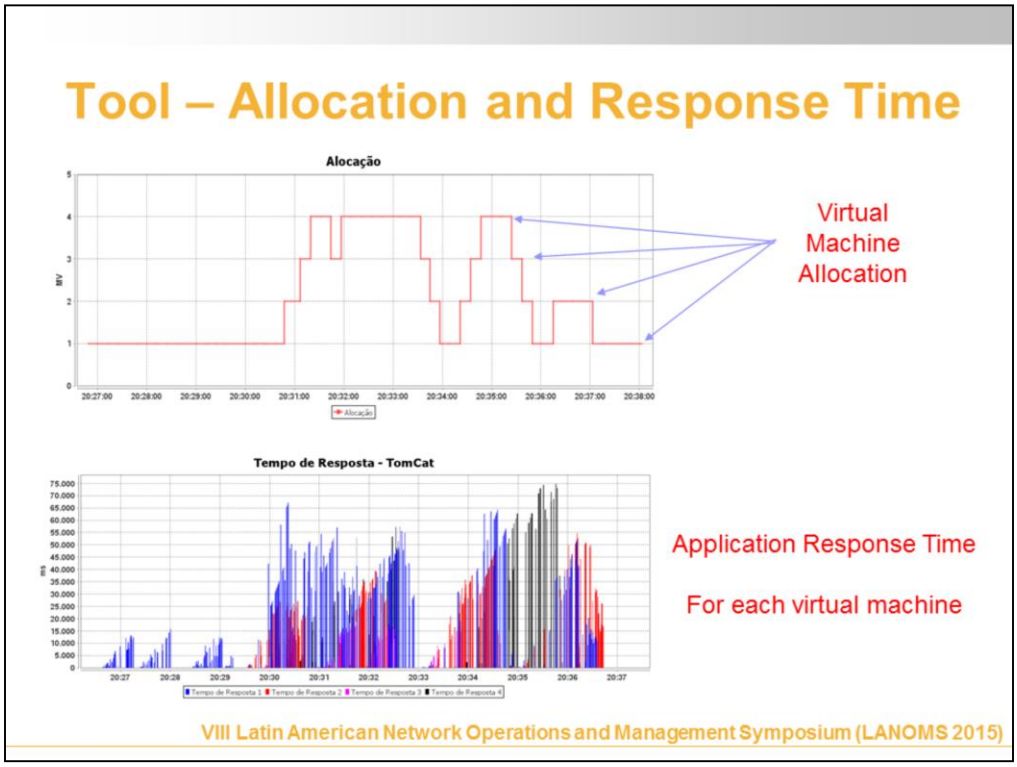
Tool – CPU Usage

One of the features of the tool is to enable the visualization of resource in a individual manner per virtual machine.

Another manner of visualization is in an overlapped manner with all the virtual machines on the same display panel.

In addition, the user can view the average CPU usage of all virtual machines in the same graphic, in percentages.

The image shows on the right panels the percentage of CPU usage in each of the four virtual machines of the example, and in the right panel the average CPU usage of all the virtual machines in percentage values.

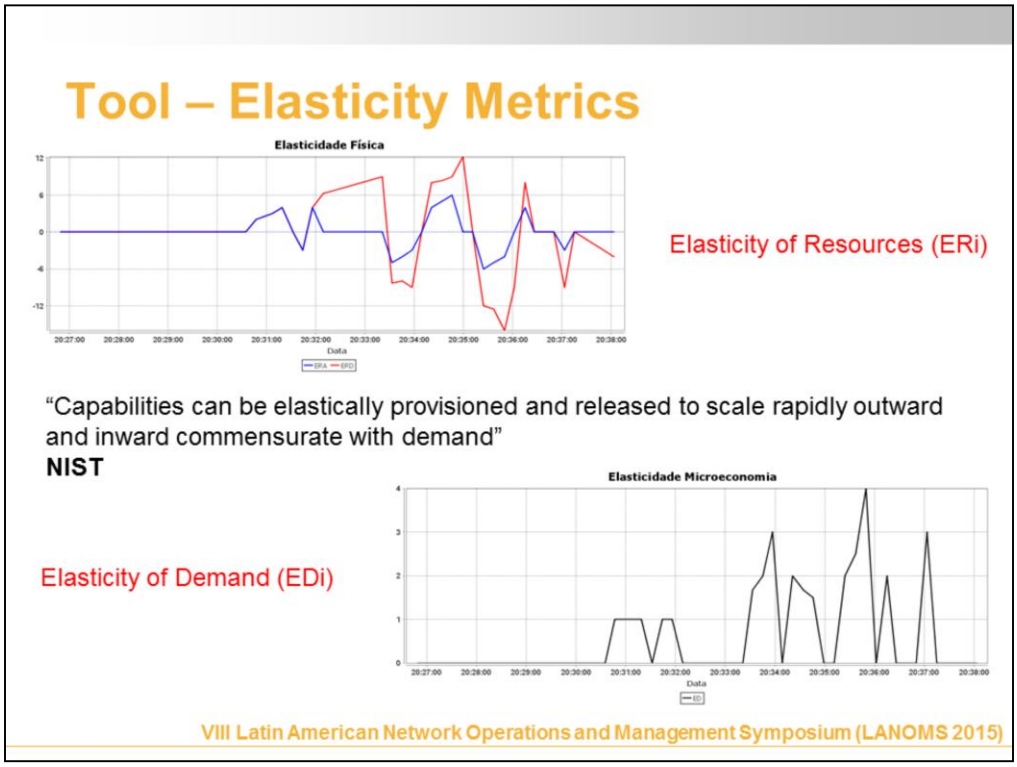


Tool – Allocation and Response Time

Two of the tool features that aid in the resources and applications visualization and analysis of cloud computing are the allocation of virtual machines graphics and the response time of the requests graphics, which are generated by the workloads applied to the environment in a dynamic manner.

The allocation graphic enables the visualization of virtual machines allocated over time, as the need for adding or removing resources. In general, load balancing mechanisms can add or remove virtual machines as some predefined elasticity strategy.

The application response time graphic shows the duration of requests in milliseconds (ms). These requests are generated through the workloads applied to the cloud and can be generated from benchmarks, microbenchmarks or cloud users. This graphic can be viewed by individual virtual machine or all virtual machines overlapping.



Tool – Elasticity

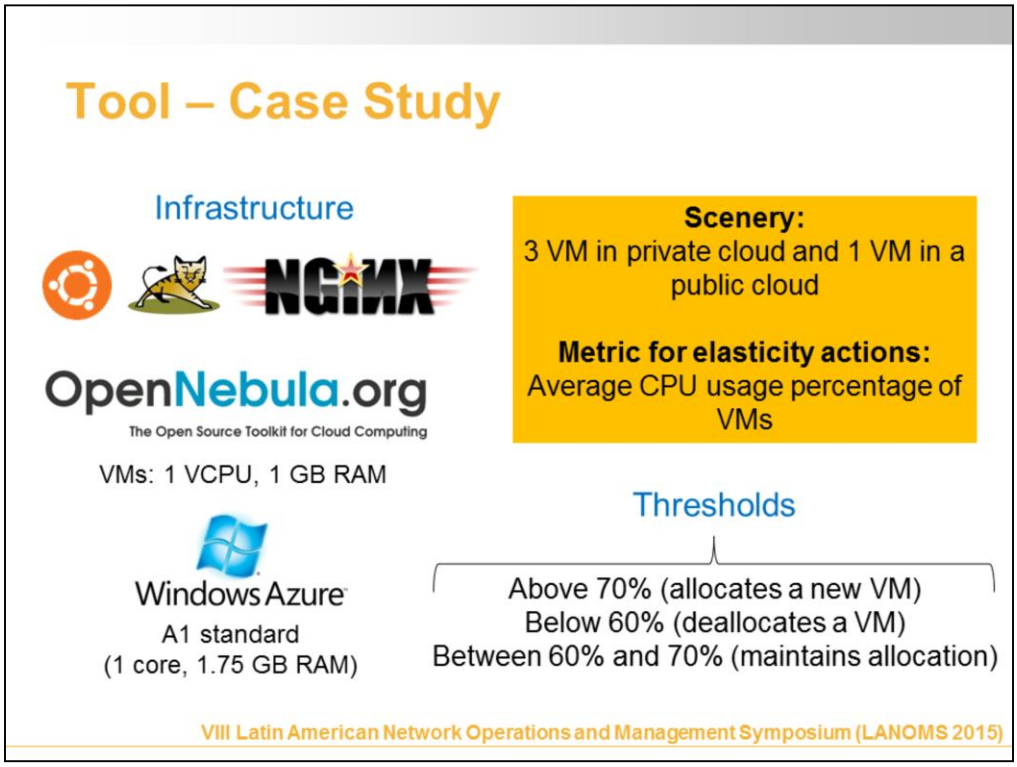
According to the National Institute of Standards and Technology (NIST) capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time [1].

Two metrics for elasticity can be viewed in the tool: Elasticity of Resources and Elasticity of Demand.

The Elasticity of Resources (ER) is based on Physics concepts [2]. It has two versions: Elasticity of Demanded Resources (EDRi) and Elasticity of Allocated Resources (EARi).

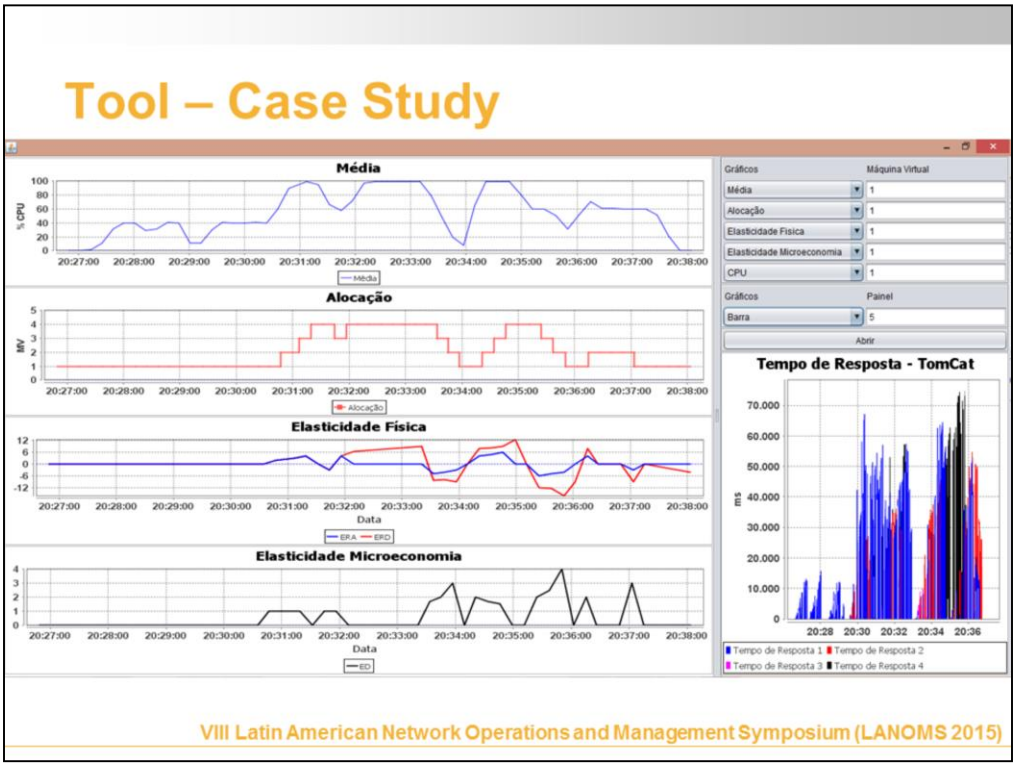
The Elasticity of Demand (EDi) is based on Microeconomics concepts [3].

The tool enables the graphics visualization of defined elasticity metrics and thus evaluate the performance of cloud computing elasticity.



Tool – Case Study

The architecture of this experiment was based on [4] and [5]. We built a private cloud with OpenNebula 3.8, virtual machines with 1 VCPU and 1 GB of RAM memory. For public cloud, we used Microsoft Azure, with A1 Standard type instance (1 core and 1.75 GB of RAM memory). The experiment used four virtual machines, Apache Tomcat as web container, HTTPERF tool as workload generator, and NGINX as load balancer. Workloads were generated directly on load balancer's virtual machine, coming from web browsers and HTTPERF. Thus, requests could be distributed among the other virtual machines and to emulate competition for resources, quite common in cloud computing environments. The objective of the experiment is to verify the performance of the environment when a virtual machine from a public cloud was used together with a private cloud to maintain the SLA. To trigger elasticity actions, we used the percentage average of virtual machines CPU usage. The thresholds used for performing elasticity actions were: above 70% (allocates a new virtual machine), below 60% (deallocates a virtual machine), and between 60% and 70% (maintains allocation). This value was calculated as the average of the last 10 samples of CPU usage in the virtual machines. The sampling interval was set in 1 second. We used an horizontal elasticity strategy, where new virtual machines instances are added to the load balancer when resources are needed, and removed if no longer needed.



Tool – Case Study

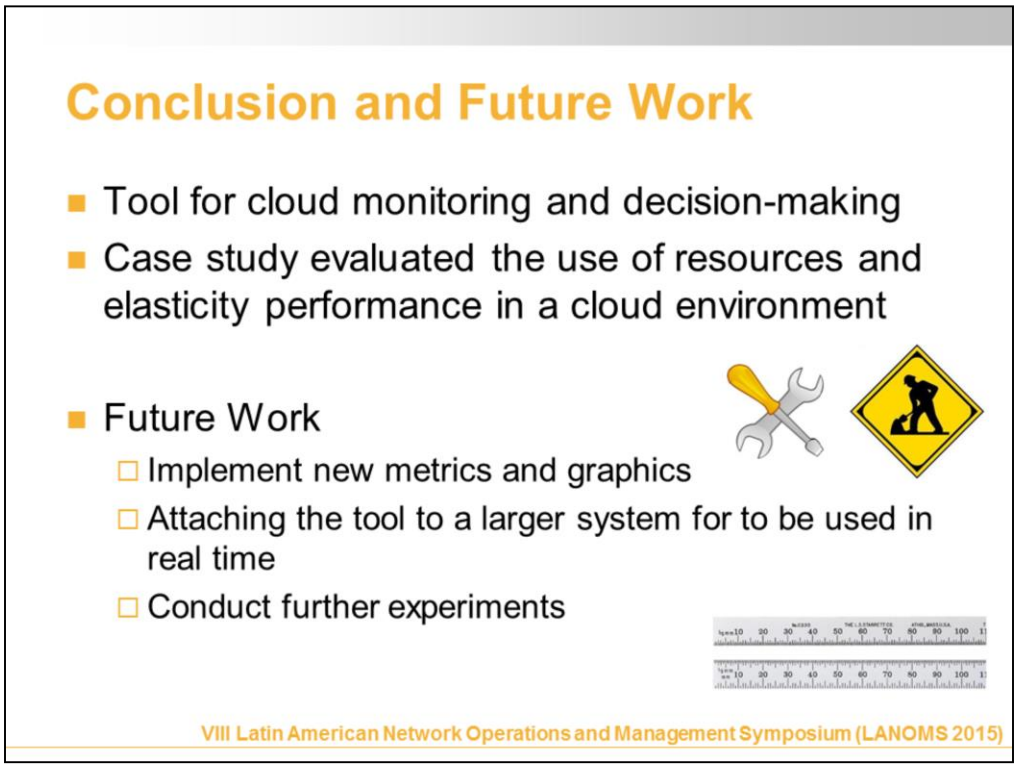
The tool on the left panels displays for the experiment the average CPU usage of all virtual machines, the allocation of virtual machines as they are needed, the elasticity of resources, and the elasticity of demand.

The right panel displays the response time of the requests that occurred on each virtual machine (in parallel).

Viewing the graphics points of higher CPU usage, we identified a greater allocation of virtual machines, and consequent reduction in CPU usage.

As the workload is applied, variations occur in CPU usage, resulting in more or less virtual machines being used. This behavior can also be viewed by the elasticity graphics, with peaks in the moment of higher use of resources.

Finally, the response time reached high values when the environment was in maximum allocation, and it did not have more resources to increase capacity.



Conclusion and Future Work

This work presented a tool for viewing and monitoring of infrastructure and applications resources of a cloud. The tool enables metrics and graphics view, which can be added programmatically. The case study analyzed the elasticity, CPU usage, allocation and requests response time to evaluate the cloud performance. As future work we intend to expand the tool with more metrics and graphics, and perform a usability study.

References

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” National Institute of Standards and Technology, vol. 53, no. 6, p. 50, 2009.

[2] E. Coutinho, P. Rego, D. G. Gomes, and J. D. Souza, “Uma Métrica para Avaliação da Elasticidade em Computação em Nuvem Baseada em Conceitos da Física”, WCGA2014, 2014.

[3] E. Coutinho, D. G. Gomes, and J. D. Souza, “On Applying Microeconomics Concepts to Cloud Elasticity Evaluation”, WCGA2015, 2015.

[4] E. Coutinho, D. G. Gomes, and J. D. Souza, “Uma proposta de arquitetura autonômica para elasticidade em computação em nuvem”, WOSIDA2014, 2014.

[5] E. Coutinho, D. G. Gomes, and J. D. Souza, “Elasticity provisioning in hybrid computational clouds based on autonomic computing concepts”, WOSIDA2015, 2015.

[6] E. Coutinho, “FOLE: Um Framework Conceitual para Avaliação de Desempenho da Elasticidade em Ambientes de Computação em Nuvem”, Doctoral Thesis, Mestrado e Doutorado em Ciência da Computação (MDCC), Federal University of Ceara (UFC), 2014.

Keywords: SDN, applications, repository, controller

Abstract: The growth in the number of SDN applications generates a demand for improve management. The Software Package Manager for SDN Applications (SPM) makes it possible to manage applications in SDN controllers. The SPM tool sets a model for the application software packaging SDN aiming host them on a Web repository, thus the SPM tool API promotes the interface between repository and controller, which allows you to automate the installation, updating and removal of these applications on the network.

Outline

- Motivation
- Contribution
- SPM
  - Proposal
  - Packing Model
  - SPM Diagrams
  - SPM's Syntax Commands
- Conclusions

In the following sections, we describe an overview of the SPM tool in addition to the elements that make up the solution and diagrams to assist their reading.

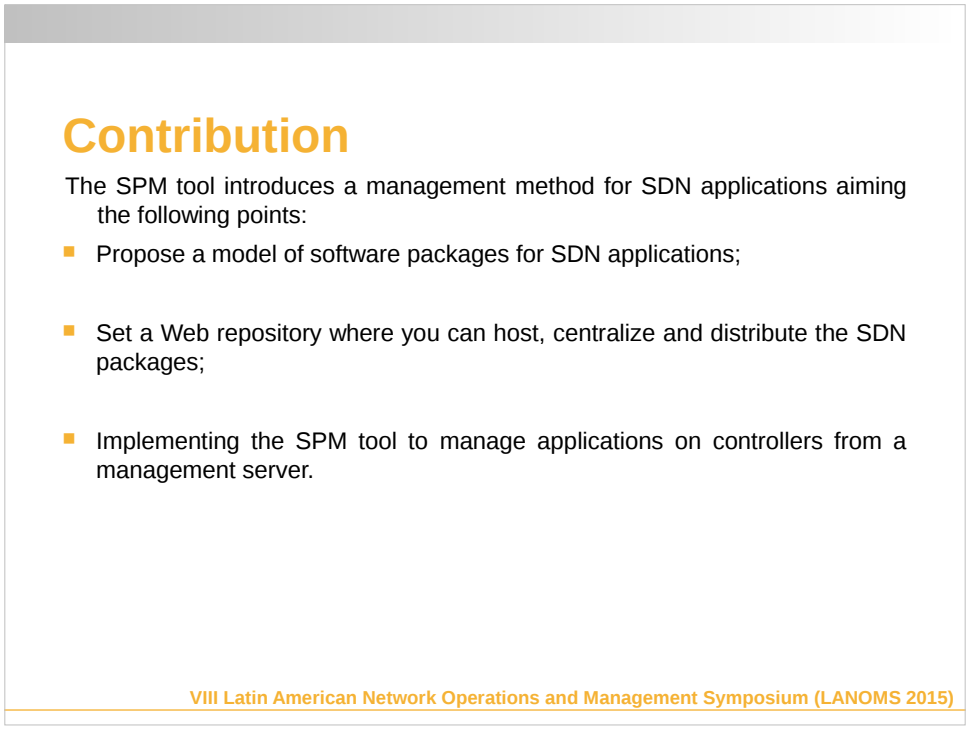
Motivation

- Currently, there are no open tools that provide applications management for SDN controllers;
- Similarly we do not find a defined model for packaging and delivering these applications;
- Therefore, there is a need of a tool that allows:
  - Optimize the control path SDN through manages the controllers and their application modules;
  - Hide of the SDN administrators the complexity in getting applications for controllers;
  - Automate the application delivery process by centralizing it in a Web repository.

In the context of SDN, control plans may have more than one controller and each controller can run different applications, this scenario makes the environment difficult to manager when the number of elements increases[1].

By using controllers of different architectures, SDN administrator faces problems to manage such applications [2], since every controller implements them according to their own methods.

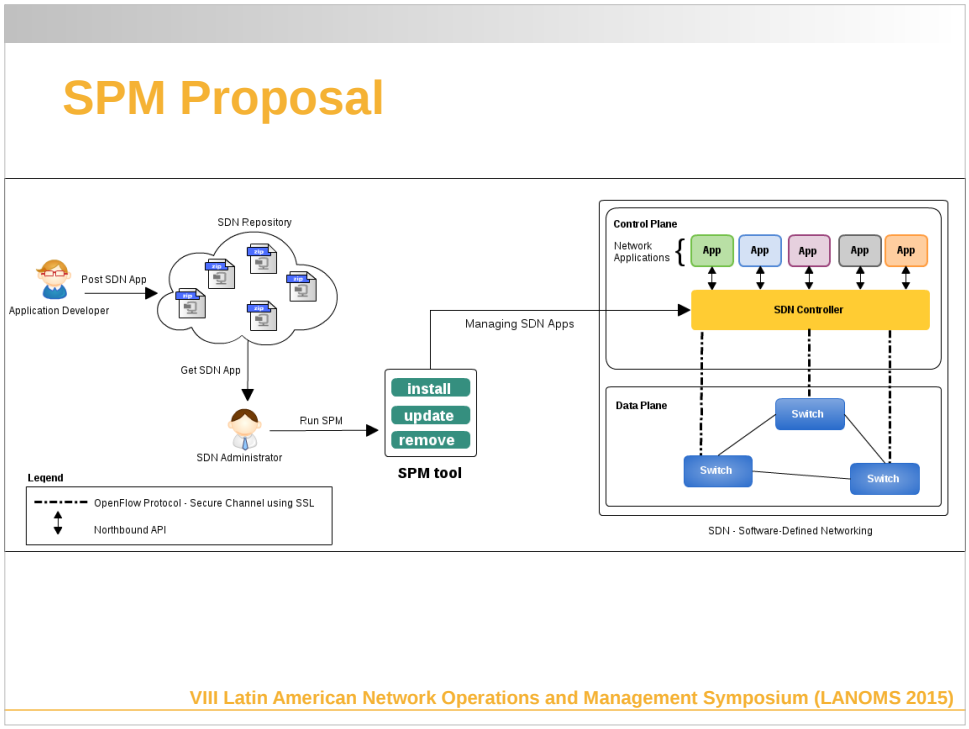
To solve these issues, a tool is required to include in its components the organization of controllers and applications, ensuring practical methods to manage them and allow a transparent manner to install, update or remove applications.



This work aims to define a tool that simplifies the management of software packages containing SDN applications. For this it is necessary to model the contents of these packages SDN applications in order to facilitate its management with the adoption of SPM tool.

As a solution to this problem we present an open tool to bridge the gap around application management in SDN, call: Software Package Manager for SDN Applications (SPM). This tool defines a packaging method for SDN applications whose goal is to address the following points:

- Propose the necessary content to create a software package design for SDN applications;
- Define a Web repository [3] where you can host, centralize and distribute standardized packages SDN applications;
- Implement the SPM tool and its database used to manage applications in SDN controller from the Web repository, and can install, update or remove this controller applications[4].



In this figure we can analyze the event flow of the SPM tool, acting as an interface between the Web repository and the SDN controller. This scenario can be divided as follows:

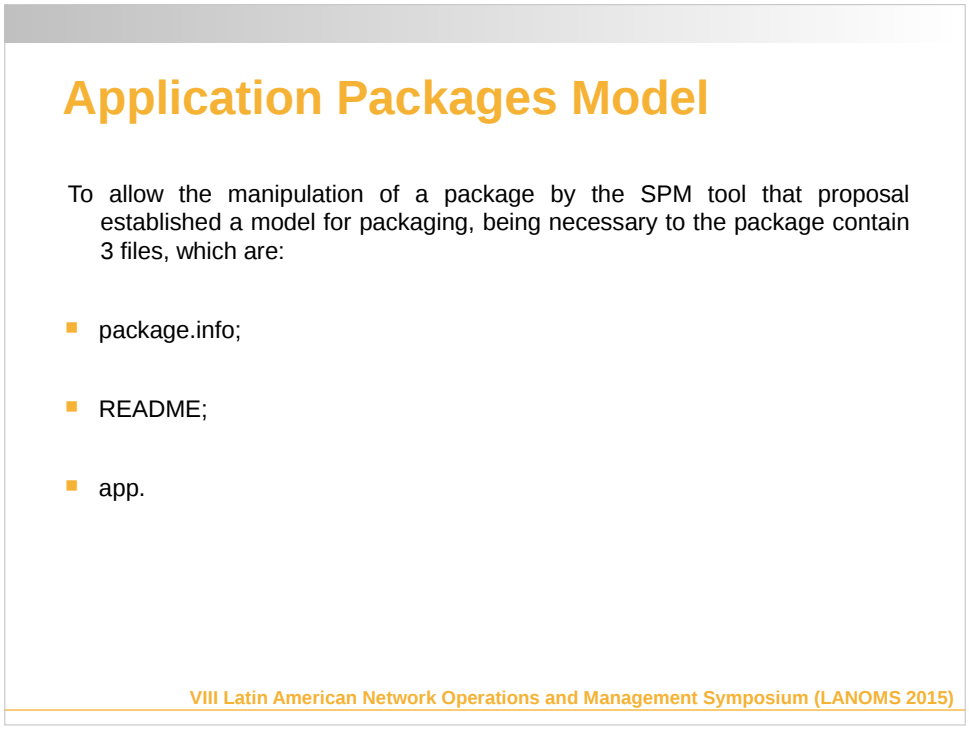
SDN Repository: Web repository where they are hosted software packages for SDN applications, they are distributed in ZIP compression following the proposed model;

Application Developer: Responsible for developing the SDN applications and send them to SDN Repository;

SDN Administrator: Responsible for managing applications on the controller using the SPM tool;

SPM tool: command line tool developed in Python and manages the application packages on the server used as controller;

Controller: main component of SDN where it operates the SPM tool managing the applications.



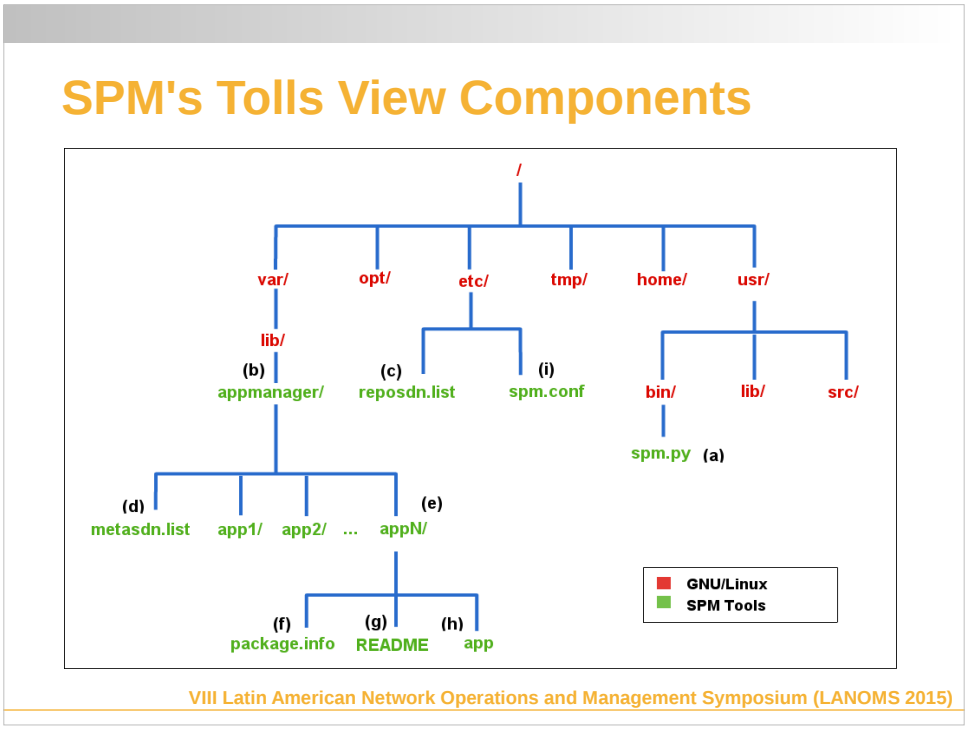
The specification of the contents of a software package for SDN applications is fundamental for the distribution and its management can be automated.

For better detailing its contents, the same will be described below:

- **package.info**: This file is structured using the JSON format, where 11 field record values representing the metadata for an application package. These metadata are: Name-app, app-Version, Maintainer, Packet-size-dependences app, Name-controller, Version-controller, Architecture, Type, Description and Copyright;

- **README**: This file is the application's documentation, with it SDN Administrator can get information on configuration, installation manual about using the application, information about the distribution license, contact the distributor or developer, in addition to guidance on how to report bugs in the application and additional information;

- **app**: the implemented application. A binary for compiled languages and the source code for the interpreted.



The following components make up the solution: :

installspm.sh: this script is the installer tool, being responsible for making the GNU/Linux system compatible with the SPM;

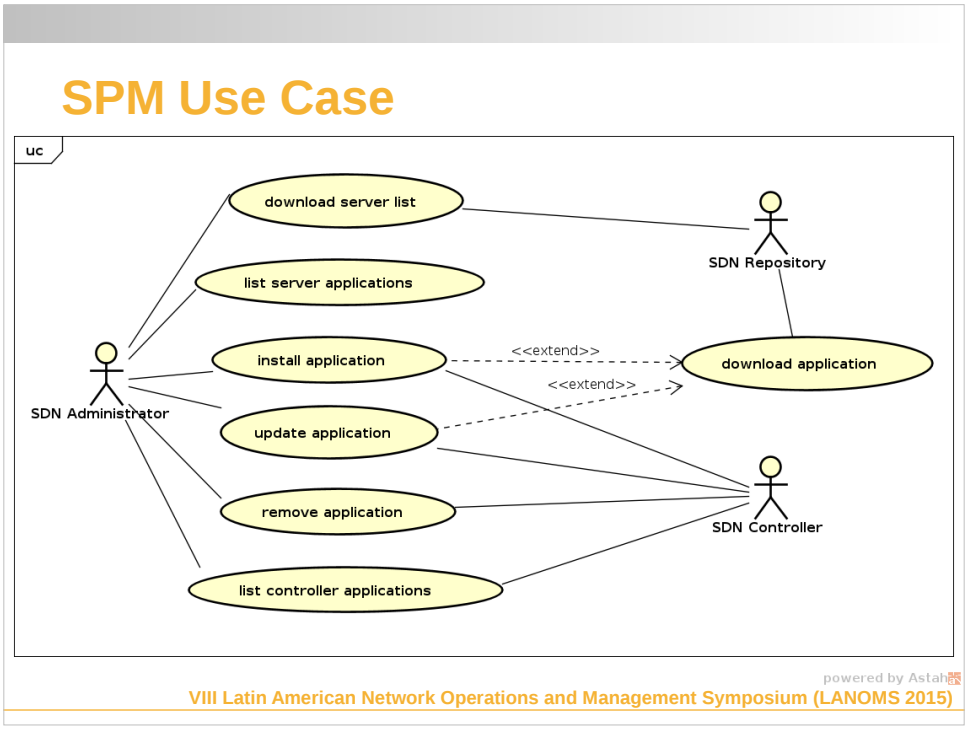
appmanager/: this directory is kept as local repository of SDN, that is where SDN's package applications are stored;

reposdn.list: Web repository application list, it is presented in a structured way so that it is compatible with the tool database;

metasdn.list: list of applications installed on the controller;

spm.conf: It is the tool database, it is organized by attributes that designate the controllers of SDN, their types, their paths, names and an configuration with the Web repository;

spm.py: It is the main file where the features are implemented.

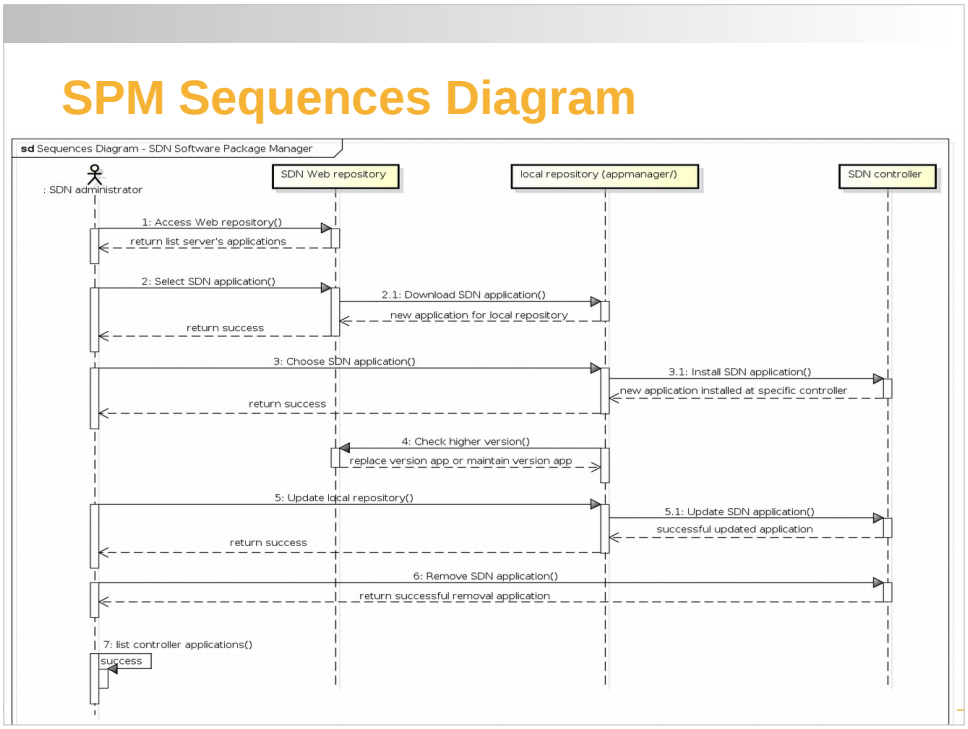


With this diagram use case, we can highlight logically organizing events that include the use of commands provided by the SPM tool.

The featured actors represent the SDN administrator who is responsible for using the proposed tool as the access interface and thus interact with the other actors who are SDN Repository and SDN Controller.

The SDN administrator has through the use of resources this solution options to download an application package and choose to install a new application or keep it updated in the local repository to your SDN, or even remove the application of any of its controlling .

In addition to these main functions are highlighted the implementation of secondary functions that help the SDN administrator in decision-making, with the SPM tool can list the Web applications repository list or applications from one controller to choose.



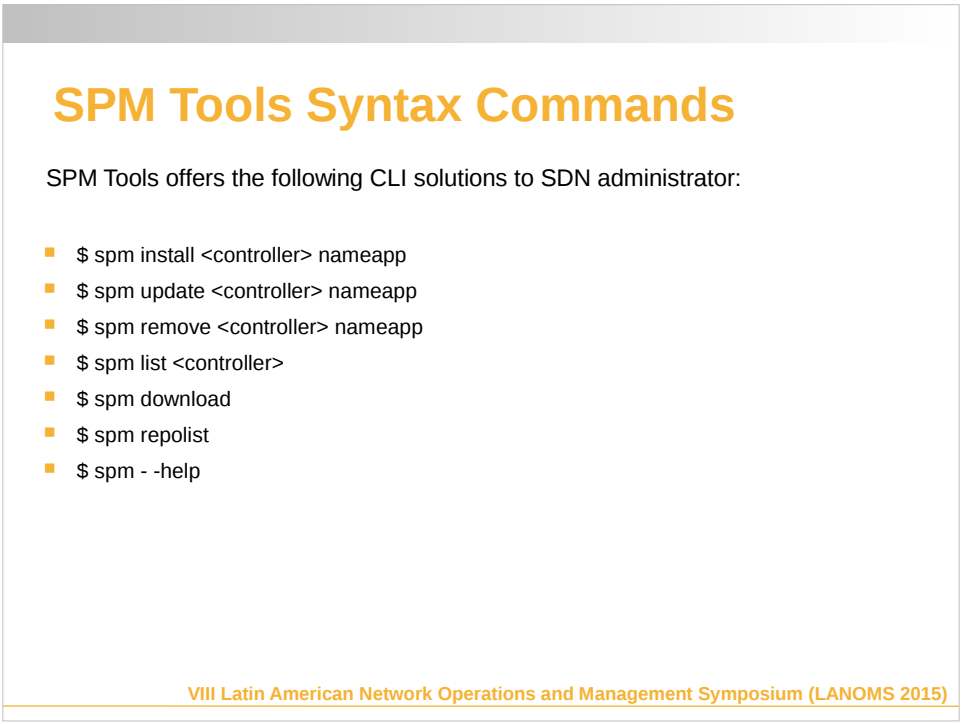
The sequence diagram shows chronological order of the actions performed by the Administrator within the context of the use case discussed earlier, the behavior of various objects that make this solution as well as the exchange of messages between them.

The SDN administrator start interacts with the Web repository, where the lifetime of this first event is closed upon receipt a list of applications called reposdn.list, it organizes the applications, the supported controller and the package version .

The application selection event return as success the download an application package to the local repository of SDN. After downloading a package, it will be kept at the local repository for eventual installation on another controller.

The local repository remains synchronized with the Web repository, so that the application package will be updated whenever a new version is available. Thus when using the update command will have as successfully updating an application in a given controller.

The SPM tool implements and you can view the start of the event where you remove an application, which returns as successfully removing a controller application. Finally the SDN administrator will have the list controller applications option, that return the installed applications in the control plane of its SDN.



The SPM tool provides a CLI (Command Line Interface) wherewith the SDN administrator can manage packages of the applications used on the network.

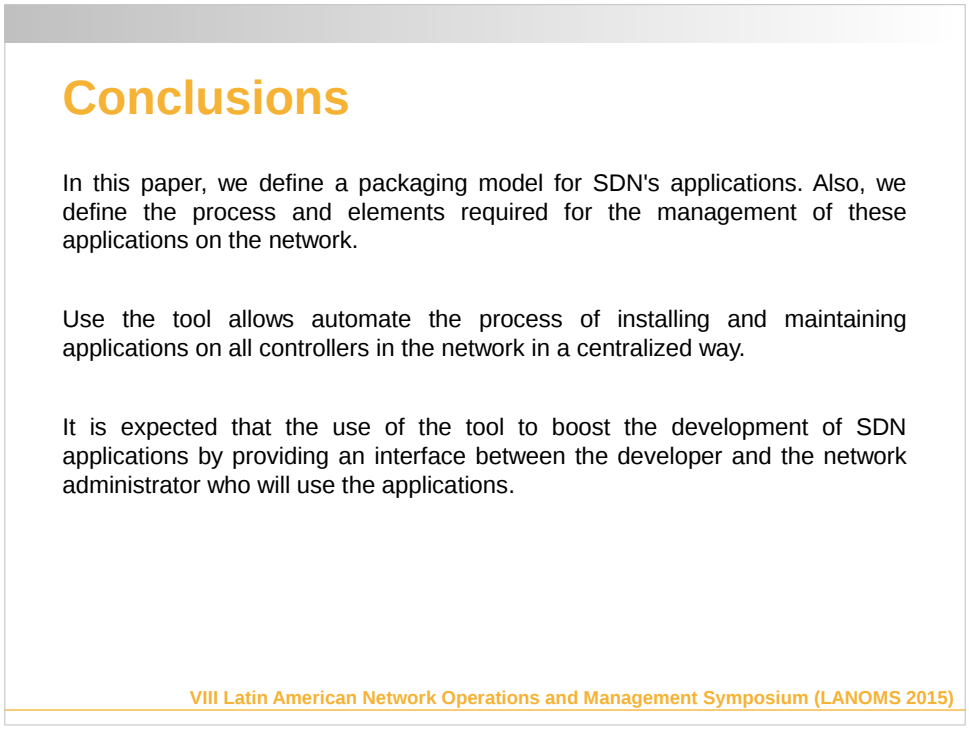
Following the defined syntax, the SDN administrator has access to commands that allow organize and easily identify the applications installed in the SDN controller.

The friendly environment is the great advantage of this proposal, it can be seen when using the ***install*** command, where it is possible to parameterize the name of an application present in the web repository and the controller name where it should be installed.

The ***update*** and ***remove*** commands require the same parameters, but the application is already installed on the controller specified as a parameter.

Additionally the following commands are available: ***download*** (which returns a list of the applications available in the repository SDN). Also, using the command ***repolist*** it is possible to obtain the list of applications contained in the Web repository, as well as, the controller and its respective version.

In turn, the ***<controller> list*** (with which you can analyze the applications installed on a specific controller) and the ***-help*** command that gives brief instructions on how to use the syntax of the commands.



For future work, we intend to support a bigger number of controllers and provide the infrastructure for the Web Repository for SDN applications.

The authors would like to thank the SERPRO for the financial support.

References

[1] H. Kim and N. Feamster, “Improving network management with software defined networking,” *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114-119, 2013.

[2] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, “Are we ready for sdn? Implementation challenges for software-defined networks,” *Communications Magazine*, Vol. 51, no. 7, 2013.

[3] Apple 2015, App Store, Avaliable at: http://www.apple.com/itunes/charts/paid-apps/. [Online; accessed in May 21, 2015]

[4] die.net, “yum(8)- Linux man page,” http://linux.die.net/man/8/rpm, [Online; accessed in April 21, 2015]