Entity Title Architecture Pilot: Scaling Out the Deployment of a Clean Slate SDN Based Network at a Telecom Operator

Luiz Cláudio Theodoro,
Pedro Henrique A. D. de Melo

Federal University of Uberlândia
Uberlândia, MG, Brazil
Email: {luiz.theodoro, pedro.damaso}@ufu.br

Rogério de Freitas Ribeiro,
Flavio de Oliveira Silva,
Pedro Frosi Rosa

Federal University of Uberlândia
Uberlândia, MG, Brazil
Email: {rogeriofr, flavio, pfrosi}@ufu.br

Alex Vaz Mendes,
João Henrique de Souza Pereira

Innovation, Research and Development
Algar Telecom
Uberlândia, MG, Brazil
Email: {alexvaz, joaohs}@algartelecom.com.br

***Abstract*—The clean-slate approach to new network architectures, named Future Internet Architectures, is a response from the research community to the challenges that the Internet architecture faces today, such as mobility. One major issue in this area is the use of large scale production networks to deploy and test new network architectures. This work extends a previous one and deploys the clean-slate Entity Title Architecture (ETArch) on a production network of a telecom operator. By using Virtual Tunnel (VTun) as an overlay, it was possible to scale out the deployment and connect several customers in different cities. ETArch Pilot shows the feasibility to move forward toward future Internet deployment in order to bring new services and applications to customers.**

***Keywords*–SDN; ETArch; Network Architecture; VTun; Deployment.**

I. INTRODUCTION

As the Internet has become fundamental to a huge volume of worldwide activities, there is a need to refine the architecture proposed since the beginning of its operation. Throughout decades of intense use, the protocols which have always supported this huge global network became inadequate regarding new challenges.

The growing demand surrounding multimedia traffic has surprised the most optimistic predictions. Dealing with this volume of media has not been an easy task. Criticism involving information security, a major concern of today's society, has also been widely questioned in relation to the current Internet architecture. Mobility, which is a central requirement for users, is constrained by the current Internet technologies.

Trying to reach solutions, researchers around the world have been proposing new architecture models, new protocols by using a clean-slate approach or the evolution the current ones considering the same network architecture [1]–[5].

To evaluate new proposals, validation in an environment close to the one in the real world is crucial to verify performance, restrictions and benefits when compared to the current network architecture. However, this evaluation is really complicated to be conducted on real production networks considering security aspects and also possible out of service situations.

This work extends a previous one [6] and its goal is to scale out the deployment of a Software-Defined Networking (SDN) based clean-slate network architecture, named Entity Title Architecture (ETArch), in a real network managed by a telecom operator, namely Algar Telecom.

To support a growing number of users and bypass the different access technologies, a tunneling approach, named Virtual Tunnels (VTun), was used. By using VTun, it was possible to connect several Algar Telecom customers located in different cities. Thus, it was possible to deploy a clean-slate network architecture over an operational production network.

This work is organized as follows: Section II presents related work. Section III presents some ETArch basic concepts. Section IV describes the scale out of ETArch deployment at the current network of a telecom operator. Section V describes the experiments conducted and presents the results of the work, and finally, Section VI presents some concluding remarks and forthcoming works.

II. RELATED WORK

Several researches involving SDN and Networking Function Virtualization (NFV) are currently going on in the world and they effectively intend to show the implementation viability in specific scenarios. SDN is applicable in both academic and commercial areas and is effectively viewed as one of the most promising proposals for the networks of the future. NFV, in turn, has gained increasing acceptance as more and more works are being published.

One of the lines of research that challenge researchers is the actual use of these proposals in the world of telecommunications. Specifically, the problem of scalability has directed some work and reflects the effort to make the results extrapolate the boundaries of research labs.

Works such as [7] worried about the ever-increasing adoption of Wi-Fi networks, NFV-fostered middle-boxes, and the avalanche of Internet of Things (IoT)-based devices. For this, it proposes an SDN framework which allows abstracting the Media Access Control (MAC) layer and also the orchestration of WiFi networks. In addition, they present a new architecture named OpenSDWN, exploring the benefits of SDN and NFV for home and enterprise WiFi networks. It presents the design and implementation of a novel WiFi-SDN approach that exploits locality in SDN control plane operations for scalability reasons.

Another important contribution is given by [8] with the R-SDN. It points out that few IT organizations have applied

SDN to their networks. One of the challenges that hinder SDN development lies in the scalability of the control plane. It further states that programmatic interface from a centralized control plane of SDN can meet requirements such as flexibility and manageability, but scalability is questionable. It then proposes a new way of designing the SDN control plan, named R-SDN whose core idea is recursion. This idea originates from network virtualization. At the beginning, it virtualizes the global network as a single logical circle. After that, several local logical circles are derived from the global circle. Finally, it derives the localized circles, layer by layer, until it sees physical switches. This top-down abstraction view can enhance scalability of SDN network.

Many examples of SDN technologies are being applied to commercial cloud services supplied for commercial carrier networks. The use of computing resources on network is becoming active in the Internet and private networks. Openflow is drawing attention as a method to control network virtualization for the cloud computing services and other carrier services. [9] took advantage of the NTT Communications (NTT.com) which aggressively promotes OpenFlow/SDN technology Research and Development. As one of the board members of ONF, NTT watches the trend of NFV closely, as well. This work shows some issues, for instance, limitation of Virtual Lan (VLAN) IDs, intermediate switches MAC address table explosion, large network overhead, inefficient network usage and others. These problems are categorized mainly into scalability, hardware flow table limitation, network stability and operability. SDN is expected to resolve these problems by integrating each independent service network into one physical network, for example within a datacenter, and by reducing cost and delivery lead-time by configuring each virtualized network for each service. With the intention to deploy OpenFlow to commercial networks, this work points out that it is important to consider not only replacing existing network and equipment, but, also, to consider the impact to services and operation. It is necessary to promote investigation of OpenFlow/SDN from these perspectives. The problem with scalability is evaluated together with other issues.

One example is the work published by IBM [10] where challenges related to cross-cloud live migration could not yet be reached. Thus, with the intention of getting an efficient solution, Virtual Wire is proposed, given that this is a system whose providers into the cloud can offer connect and disconnect services which are by far easier to be managed, when compared to virtual network forwarding.

In this context, cloud providers must manage the associated control which specifies how the packets are routed inside a virtual network. For example, providers can implement a distributed virtual switch or logical control embedded in a network controller defined by the software. For this purpose, the Virtual Wire system matches each Virtual Network Interface Card (vNIC) with a point-to-point network tunnel. VNICs belong to Virtual Machines (VMs) that either are implemented in servers or network forwarding components at the level of users like routers and switches. Network users can build complex virtual networks connecting pairs of vNICs together.

By using a tunneling approach, Layer 2 endpoints are made available through Layer 3 network tunnels. In this way, a vNIC is sent through a layer 2 frame, by including an associated endpoint which encapsulates the entire packet (MAC header and VLAN tags in an User Datagram Protocol (UDP) packet). The Internet Protocol (IP) address and socket port number correspond to the physical network address of the endpoint manager. After receiving a package, the endpoint manager parses the headers, examines the ID connector, and then sends the packages to the endpoint destination.

Topics such as Tunneling, SDN and Virtual Switches usually appear in a considerable amount of research developed around the world. Bingham Liu [11] proposes the usage of a virtual switch (Open vSwitch) with the help of General Packet Radio Service (GPRS) Tunneling Protocol (GTP) to explore the SDN evolution in the design of the core of a mobile network by using cloud based computing. This approach has a common idea with our work, the deployment of a clean-slate network architecture in a real operator environment.

Another work [12] focuses precisely on the possible improvements for the SDN applications concerning the infrastructure of mobile networks, by showing current LTE structure, discusses how to simplify the network control and new services offering. In this case, the controller takes functions such as monitoring, Quality of Service (QoS), access control policies, virtual operators, and others, that is, the goal is that it takes the functions of some mobile network elements and the infrastructure's vision. Switches with the capacity to become control local agents are also part of the architecture, since the controller cannot be available to fast answers for local events. Furthermore, they have functions such as daily checking of the traffic counters and change of the queue priorities according to some limit.

This work, besides implementing some concepts of the SDN in a physical network, uses an actual telecom network, accessed by thousands of customers, and shows the flexibility and the simplicity of the ETArch architecture in this real environment, if compared to the regular network infrastructure.

III. Entity Title Architecture (ETArch)

The architecture of the Internet is not able to meet the requirements of current applications such as mobility, security, QoS. There are several research initiatives toward future Internet architecture [13]. One initiative is the Entity Title Architecture (ETArch), that was initially proposed by our research group. This section presents an overview of ETArch main concepts.

ETArch has a natural match with SDN, since both share the concept that the control plane is separated from the data plane. The ETArch prototype is being created in an incremental way and, currently, researchers from several universities are working with ETArch in order to add an extension to the architecture in order to satisfy several requirements from current applications, such as mobility [14], multicast [15], QoS [16] and routing [17].

An entity is everything which has the capacity to communicate and, this way, the entities can be hosts, smartphones, Network Elements (NE), users, applications, sensors and so on.

Another central concept is the Title, which is a unique identifier independent from the network topology [18]. ETArch uses the title to identify the entities. One Title can also be seen as a credential that can be used to relate the security features [19].

At ETArch, the communication happens by using the Workspace. The Workspace is a logical bus which enables the communication among the entities. Entities attach to a workspace in order to participate in a communication domain.

In ETArch, the Domain Title System (DTS) [20] represents the control plane of the network. Before starting the communication, an entity must register itself at the DTS. The DTS keeps the information about the entities, their titles and Workspaces. The DTS is a distributed system composed by Domain Title System Agents (DTSAs). Each DTSA is capable to control the NEs and is aware of the NE graph. The DTSA is responsible to the communication with other DTSAs. The DTSAs uses Openflow to control the NEs.

IV. ETArch Pilot Scale Out

This work proposes the scale out of the deployment of ETArch by using a real telecom network and their customers, which are geographically distributed in the operator's network. To accomplish this, it was necessary to establish a Layer 2 connection between customers and to have an fully OpenFlow capable infrastructure. Since this last condition was not satisfied in the operator infrastructure, then a tunneling technique was used.

By using the tunnels it was possible to connect geographically distributed clients over the operator's infrastructure. At each physical location, a software based OpenFlow switch was used. Each one of these switches was controlled by the DTSA then creating the conditions to deploy ETArch. Figure 2 shows, in a general way, the protocol tunneling technique that allowed the communication.

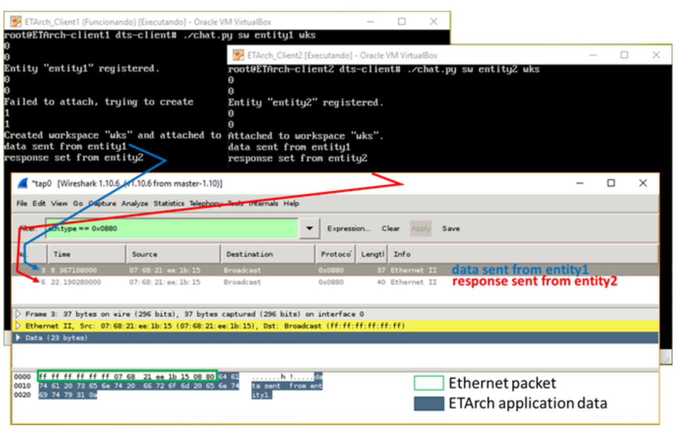

Figure 1. Wireshark capture of ETArch primitives between application instances.

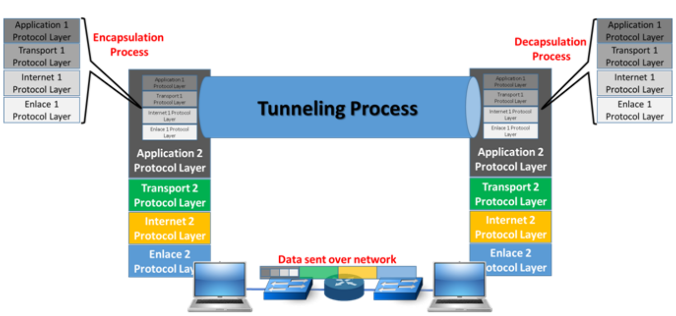

Figure 2. Two hosts communicating with each other using the tunneling technique.

The tunneling concept is commonly used in the computer networking area. Some examples are: Internet Control Message Protocol (ICMP) Tunneling [21], Secure Shell (SSH) Tunneling [22], Generic Routing Encapsulation (GRE) Tunneling [23] and Internet Protocol Security (IPsec) Tunneling [24].

To achieve our objectives, the tunneling technique was based in the VTun software [25]. The Virtual Tunnels (VTun) work in a client/server mode, and are capable to accomplish a point to point connection between the involved hosts. VTun offers a series of functions, like data compression/cryptography, connections access control, besides the bandwidth control. The supported tunneling encapsulation modes by the VTun are: IP Tunnel, Ethernet Tunnel, Serial Tunnel and Pipe Tunnel. This work used the Ethernet Tunnel.

For the traffic to be tunneled by the VTun, one host needs to act as a server, opening a socket in the system and listening to the port 5000. When a client connects to this service, one virtual interface is created in the operating system. This virtual interface is the *access bridge* to the created tunnel.

Any packet sent to that interface will be encapsulated by the Transmission Control Protocol (TCP)/IP protocol stack, and then, will travel through the tunnel to the host connected on the other side of the tunnel, where the decapsulation process will occur, extracting the original packet, which was encapsulated before it enters the tunnel. Figure 3 shows the process of Ethernet tunneling traffic from the ETArch architecture between two hosts connected to the Internet.

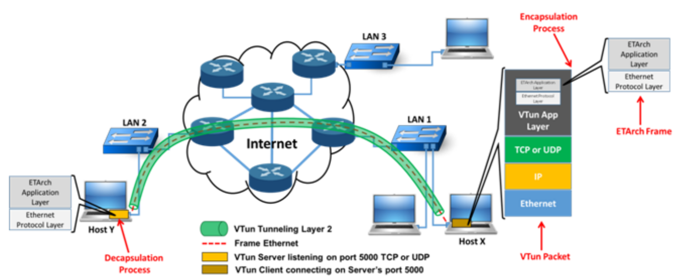

Figure 3. Remote hosts exchanging Ethernet traffic through the Internet.

Compared to our previous work [6], the relevant improvement occurred in the simplification of the tunneling process that occurred due to the use of VTun instead of GRE. That evolution brought to us a major simplicity since the VTun offers less complexity to accomplish the scale out of ETArch deployment. When using GRE, it is necessary to configure the client's modem in *bridge mode*.

By using VTun to produce the tunnel, it was possible to do a smooth deployment and use of ETArch based applications. With this approach, there was no need to change the customer's modem operation mode and in this case the only requirement to use ETArch based applications was to be a customer of the telecom operator. It happens because when the VTun is started, one virtual interface is created in the operating system, which is then responsible to carry the traffic through that interface and also for the created tunnel.

V. ETArch Pilot Experimental Evaluation

To conduct the tests, VTun was used to create the virtual tunnels between the hosts, allowing ETArch based applications to send data to all the hosts connected to the same workspace. Below, it is presented the description of the environment used

 ISBN: 978-1-61208-546-3

to perform the initial evaluation, and then, the scale out the ETARch deployment:

- The network operated by Algar Telecom was used to provide the connectivity between several users which are in fact, their customers;
- Ubuntu 14.04 operating system was used in all the computers used in the deployment, in the machine responsible hosting the DTSA, the VTun Server and in the customer's machines to execute the ETArch based application;
- The machines that acted as a VTun client, established a virtual tunnel to the VTun server, which in turn was listening to the port TCP 5000.
- At the VTun server host, Destination Network Address Translation (DNAT) was configured in the modem, redirecting all the traffic destined to the routable IP and port 5000 to the internal IP and keeping the same destination port number.
- For the machine that acted as the DTSA, DNAT was configured in the modem, redirecting the traffic destined to the routable IP and port 6633 to the internal IP while keeping the same destination port number.

Several tests were made and we classified them in two scenarios: Scenario One and Scenario Two. In Scenario one, two hosts have established one tunnel to communicate to each other, one acting as a VTun client and the other as a VTun server. In this scenario, each host also has the role to act as an OpenFlow switch. Each switch was controlled and programmed by the DTSA. The DTSA was responsible to create and modify the flows in order to provide the communication by using the Workspace. This workspace was used to support the chat application used during the test.

Figure 4 shows the detailed topology used in the tests defined in the Scenario One, describing the participating peers and their communications. After the analysis of the results from the tests conducted in the Scenario One, it was noticed that with more than two clients, the approach of establishing a tunnel between each one of the clients in the ETArch architecture would become painful in terms of configuration and troubleshooting. One of the major issues would be the necessity of the Network Address Translation (NAT) configuration in the customer's modems that run the VTun in server mode, which would restrict the participation of clients connected in mobile networks, like 3G and 4G. This restriction would occur because the clients that need to run the VTun in server mode would need additional applications on their cellular phones to create the NAT configurations.

To overcame these issues, Scenario Two was created. In this scenario, the topology was changed to create a concentration tunnel host. The function of this machine was to host the VTun in server mode and to receive all the client's connections from the ETArch architecture. In this scenario, only the VTun concentrator machine was acting like an OpenFlow switch. This made the configuration and troubleshooting process easier since there is only one OpenFlow switch in the topology. On the other side, in Scenario One, it would be necessary to have n OpenFlow switches, one for each connected client. The topology for the tests conducted under Scenario Two is described in detail by in Figure 5.

In order to verify the overhead based on packet capture in the test environment, it was possible to identify that the VTUN encapsulation process caused an overhead of about 50.4%, equivalent to 56 bytes in each packet generated by the chat application. This overhead of 56 bytes is the result of the subtraction of 111 bytes (Vtun packet) from 55 bytes (chat application packet that contains the text "message of the chat application ETArch"). Figure 6 shows the overhead of the VTun tunneling versus the packet size. The overhead is in a range between 82.35% to 3.73% considering a 1500 Bytes packet.

Figure 7 represents a capture of the VTUN packet, highlighting the VTUN encapsulation data and the packet data of the chat applications in the ETArch environment.

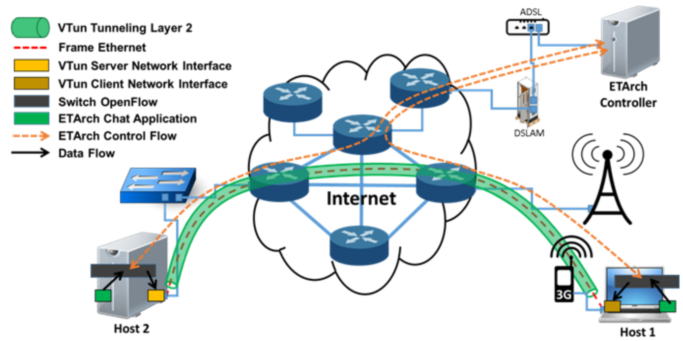

Figure 4. Scenario One - Tunnel between two hosts.

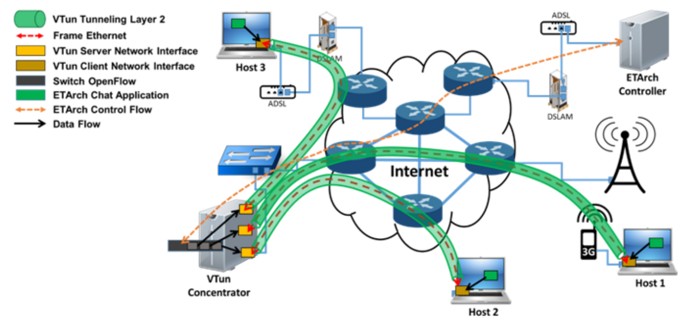

Figure 5. Scenario Two - Tunnel between multiple hosts.

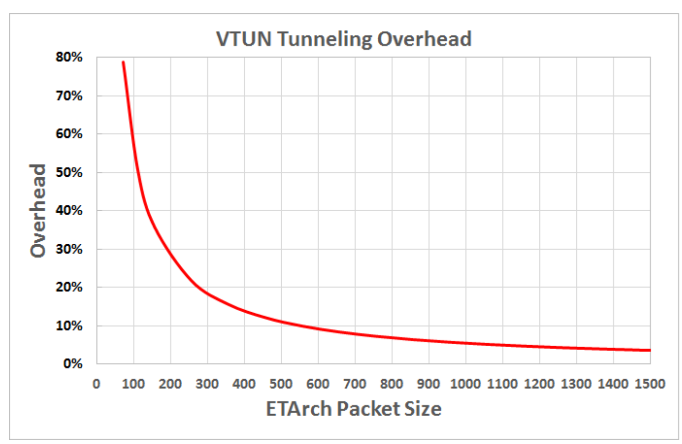

Figure 6. VTun Tunneling Overhead versus Packet Size

Figure 8 shows the result of the package obtained after the decapsulation process is carried out by the VTUN process.

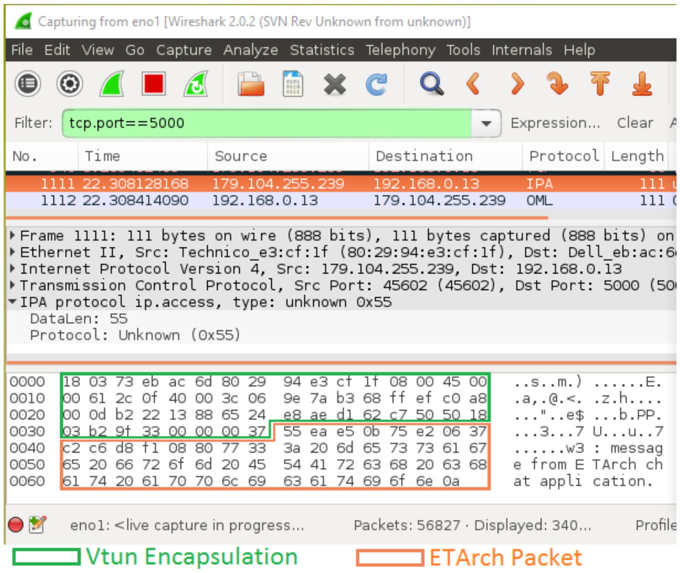

Figure 7. VTUN packet received at the server's network interface before the ETArch packet unpacking process.

After removing the 3 layers of the TCP/IP architecture (media access control, network, and transport) the VTUN process delivers on the virtual interface (tap0) of the server, only the chat application package, as it had been generated in the source.

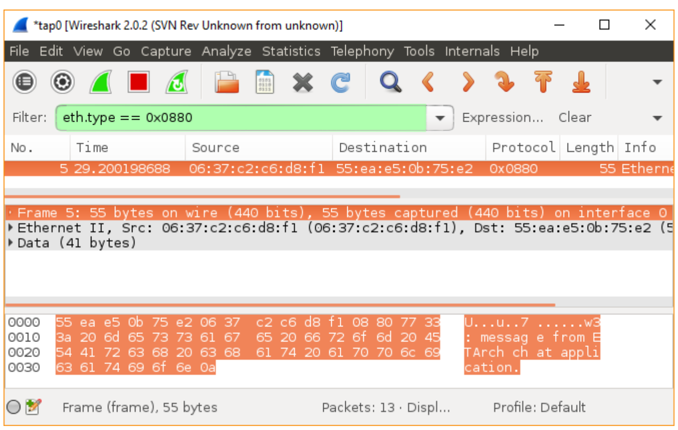

Figure 8. VTUN packet received at the server's network interface before the ETArch packet unpacking process.

Due to the second scenario being developed, it is possible to extend the number of entities in an unlimited geographical area. For this reason, we made a test with more than 40 users connected, distributed in a radius of 600 kilometers from the city of Uberlândia, according to Figure 9 which shows the localization of the machines performed by the chat clients. Most of the users are located in the city of Uberlândia as can be seen partially in Figure 10.

VI. Concluding Remarks and Future Work

This work scaled out the deployment of a clean-slate SDN based network architecture, named ETArch, in the real infrastructure of a network operator, named ALGAR Telecom, with little intervention in the customer environment.

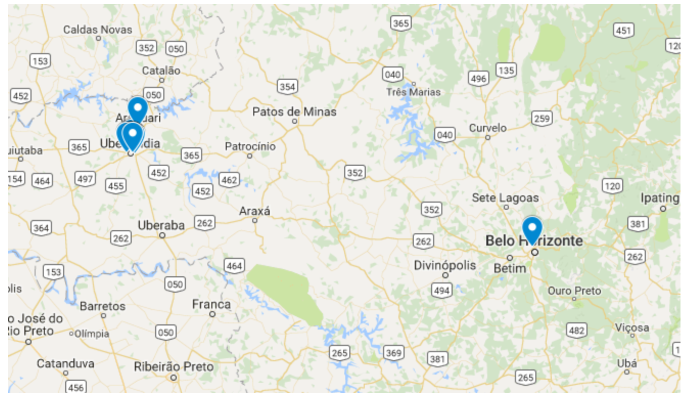

Figure 9. Geographical distribution - second scenario with maximum distance of 600 km

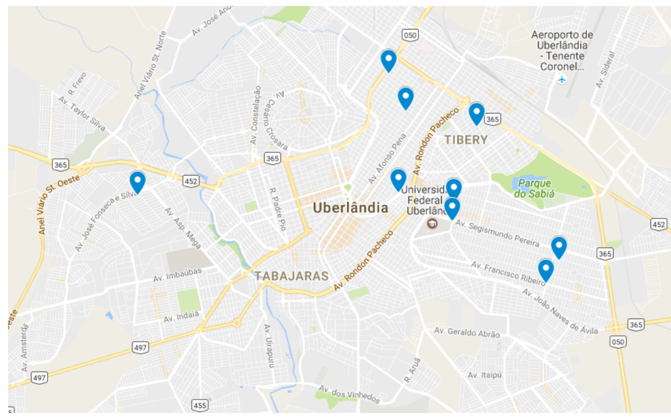

Figure 10. Geographical distribution - second scenario only in Uberlândia

By using VTun to support the tunneling process and to tackle the interconnection issues with the infrastructure, it was possible to use an ETArch based application by several customers located in different cities inside the operator coverage area. The chat application uses natives ETArch's capabilities to support multicast and mobility.

The use of VTun, when compared to the previous approach based on GRE, allowed an easier configuration on each host machine where the application was installed and the use of a VTun server inside the operator infrastructure enabled the scale out of the number of customers that could benefit from the new capabilities provided by ETArch.

The experimental evaluation demonstrated that the overhead imposed by the VTun is in a range between 82.35% to 3.73% of the packet size, by considering a chat application. Applications with greater packet sizes will present less overhead.

As future work, we plan to deploy over the operator network other ETArch based applications, such as a video streaming application which would provide video multicast in a seamless way.

The work demonstrates the feasibility to deploy new network architectures in parallel with current ones and go towards future Internet deployment.

ACKNOWLEDGMENTS

This work has been partially funded by the Brazilian agencies: CAPES, CNPq and FAPEMIG and also by PROPP/UFU. We also would like to thank ALGAR Telecom for the support and partnership on this work.

REFERENCES

[1] F. Bronzino, K. Nagaraja, I. Seskar, and D. Raychaudhuri, "Network service abstractions for a mobility-centric future internet architecture," in *Proceedings of the eighth ACM international workshop on Mobility in the evolving internet architecture*. ACM, 2013, pp. 5–10.

[2] H. Dongsu et al., "Xia: Efficient support for evolvable internetworking." in *NSDI*, vol. 12, 2012, pp. 23–23.

[3] V. Jacobson et al., "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[4] T. Anderson et al., "A brief overview of the nebula future internet architecture," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 81–86, 2014.

[5] Y. Wang, I. Matta, F. Esposito, and J. Day, "Introducing protorina: a prototype for programming recursive-networking policies," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 129–131, 2014.

[6] L. Claudio et al., "Entity Title Architecture Pilot: Deploying a Clean Slate SDN Based Network at a Telecom Operator," pp. 144–149, Jun. 2015, [retrieved: Mar, 2017]. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=aict_2015_7_40_10152

[7] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, "Opensdwn: Programmatic control over home and enterprise wifi," in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*. ACM, 2015, p. 16.

[8] W. Dai, G. Shou, Y. Hu, Z. Guo, and J. Liu, "Extending sdn network with recursive architecture," in *Wireless Personal Multimedia Communications (WPMC), 2014 International Symposium on*. IEEE, 2014, pp. 491–496.

[9] S. Yoichi, I. Fukuda, and T. Fujita, "Deployment of openflow/sdn technologies to carrier services," *IEICE transactions on communications*, vol. 96, no. 12, pp. 2946–2952, 2013.

[10] D. Williams, H. Jamjoom, Z. Jiang, and H. Weatherspoon, "Virtualwires for live migrating virtual networks across clouds," Technical Report RC25378, IBM, Tech. Rep., 2013.

[11] B. Liu, "Software defined networking and tunneling for mobile networks," Master's thesis, kTH Royal Institute of Technology, 7 2013.

[12] L. E. Li, Z. M. Mao, and J. Rexford, "Toward Software-Defined Cellular Networks," in *2012 European Workshop on Software Defined Networking*, Oct. 2012, pp. 7–12.

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

[14] C. Guimaraes et al., "IEEE 802.21-enabled Entity Title Architecture for handover optimization," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2014, pp. 2671–2676.

[15] M. Amaral Gonçalves et al., "Multicast traffic aggregation through entity title model," pp. 175–180, Jul. 2014, [retrieved: Mar, 2017]. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=aict_2014_7_40_10177

[16] J. Castillo et al., "Additions to the ETArch control plane to support multimedia QoS-guaranteed content transport over OpenFlow-enabled SDN future internet systems," in *Globecom Workshops (GC Wkshps), 2014*, Dec. 2014, pp. 172–177.

[17] N. Vieira de Souza Neto et al., "Control Plane Routing Protocol for the Entity Title Architecture," pp. 185–190, Apr. 2015, [retrieved: Mar, 2017]. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=icn_2015_7_40_30210

[18] J. de Souza Pereira, F. de Oliveira Silva, E. Filho, S. Kofuji, and P. Rosa, "Title model ontology for future internet networks," in *The Future Internet*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6656, pp. 103–114.

[19] ITU, *Security architecture for Open Systems Interconnection for CCITT applications*, International Telecommunications Union, Mar. 1991.

[20] F. Oliveira Silva et al., "On the analysis of multicast traffic over the entity title architecture," in *2012 18th IEEE International Conference on Networks (ICON)*, Dec 2012, pp. 30–35.

[21] J. Postel, "Internet control message protocol," RFC 792 (Standard), Internet Engineering Task Force, September 1981, [retrieved: Mar, 2017]. [Online]. Available: http://www.ietf.org/rfc/rfc792.txt

[22] T. Ylonen and C. Lonvick, "The secure shell (ssh) protocol architecture," RFC 4251 (Proposed Standard), Internet Engineering Task Force, January 2006, [retrieved: Mar, 2017]. [Online]. Available: http://www.ietf.org/rfc/rfc4251.txt

[23] S. Hanks, T. Li, D. Farinacci, and P. Traina, "Generic routing encapsulation (gre)," RFC 1701 (Informational), Internet Engineering Task Force, October 1994, [retrieved: Mar, 2017]. [Online]. Available: http://www.ietf.org/rfc/rfc1701.txt

[24] S. Blake et al., "An Architecture for Differentiated Service," RFC 2475 (Informational), Internet Engineering Task Force, December 1998, [retrieved: Mar, 2017]. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[25] VTUN, "VTun - Virtual Tunnels over TCP/IP networks," Sep. 2016, [retrieved: Mar, 2017]. [Online]. Available: http://vtun.sourceforge.net/

