Eager, S., Matencio Escolar, A., & Alcaraz Calero, J. M. (2020). An analysis of multicast inefficiencies in multi-tenant MEC infrastructures for 5G networks. In *2020 IEEE International Conference on Communications Workshops (ICC Workshops)* (IEEE Conference Proceedings). IEEE. https://doi.org/10.1109/ICCWorkshops49005.2020.9145140

An Analysis of Multicast Inefficiencies in Multi-Tenant MEC Infrastructures for 5G Networks

Steve Eager
School of Computing, Engineering and Physical Sciences
University of the West of Scotland
Paisley, UK
steve.eager@uws.ac.uk

Antonio Matencio Escolar
School of Computing, Engineering and Physical Sciences
University of the West of Scotland
Paisley, UK
antonio.matencio@uws.ac.uk

Jose M. Alcaraz Calero
School of Computing, Engineering and Physical Sciences
University of the West of Scotland
Paisley, UK
jose.alcaraz-calero@uws.ac.uk

***Abstract*—Communication and data networks require the efficient and timely delivery of packets with little or negligible latency in the transmission. This is especially important of 5G mobile networks where mission critical communications and applications are reliant on fast and efficient communications. It has long been understood that using multicasting, as a method of sending multiple packets simultaneously, over a medium will improve the efficiency of a communication. This is achieved by only transmitting multicast packets once to multiple recipients who will then receive the (multicast) packets. This paper identifies the inherent inefficiencies of multicasting when having to encapsulate packets in overlay networks such as those in the Multi-Access/Mobile-Edge to Core 5G infrastructure networks and demonstrate such inefficiencies empirically.**

- ***Keywords—Multicast, Unicast, Encapsulation, Cloud, Multi-Tenant, Edge, Core, 5G.***

I. INTRODUCTION

The development of fifth generation and beyond (5G and B5G) telecommunications technologies is a significant driver towards the evolution of Cloud-based Computing technologies. Multi-Access Edge Computing, formally Mobile Edge Computing (MEC), is one such architecture [1]. MEC provides a platform for Telecommunication companies (Telcos) and network providers to invest and transform their services to run more efficiently to meet the specific demands of industry and business end-users [2].

It is projected that the cost needed to implement MEC infrastructures whilst meeting the additional challenges of reducing capital expenditure (CAPEX) and operating expenditure (OPEX) can be recouped within a ten-year period [3]. Implementing cloud-based architectures that utilize technologies such as multi-tenancy, virtualization and Software Defined Networks (SDN), to efficiently control access to cloud-based applications and services, will contribute to achieving these targets [3].

There will be a requirement, during the implementation period, for end-users to continue to connect seamlessly to networks whilst telecom providers implement changes to achieve their CAPEX and OPEX reduction targets. This imposes the use of overlay networks using encapsulation protocols such as Network Virtualization using Generic Routing Encapsulation (NVGRE), Virtual eXtensible LAN (VxLAN) and Stateless Transport Tunneling (STT), to support multi-tenancy isolation of physical resources. The use of such overlay networks presents many challenges to achieve the efficient transmission of converged voice and data communications. One such challenge is related to the distribution of IP Multicast traffic over overlay networks. Current overlay networks, used in multi-tenant architectures, do not provide efficient support for Multicast traffic. The problem is even more apparent when we focus on the Edge-to-Core network segment where, to provide multi-tenancy isolation [3], valuable bandwidth is often unavoidably wasted by overlay traffic due to inefficiencies in the transmission protocol of this type of Multicast traffic. The optimal transmission method of Multicast traffic is to send a single stream to implement a Point-to-Multipoint connection [4].

The main motivation of this research work is focused on providing a detailed analysis about the reasons why Multicast efficiency is compromised in these types of novel multi-tenant infrastructures in order to share with the community, the root causes of these inefficiencies.

II. RELATED WORK

Surprisingly, the number of research papers addressing Multicast traffic in modern overlay networks such as the Edge-to-Core segment of Multi-tenant MEC networks is almost non-existent even though many applications, which are in operation throughout the world today, rely upon or implement Multicast traffic as a means to efficiently deliver data to multiple destinations. For example, applications and Internet services such as Netflix, Skype, Internet Protocol Television (IPTV), Video-on-Demand (VoD), Massively Multiplayer Online Games (MMOG) as well as end-user collaboration tools all use or can use Multicasting as a method for communication and data distribution. Specifically, we have found that there is very little evidence of research in the areas of Multicast over Unicast, Multicast over Multicast and Unicast over Multicast overlay networks in the Edge-to-Core segment of Multi-tenant MEC infrastructure networks.

We discovered that most of the current research taking place involving Multicast traffic is used in support of research around Software Defined Networks (SDNs) [5] and other (Multicasting) research focuses on solving the problem of end-users connected to Unicast only environments that are not able to receive Multicast data. This is an area highlighted by the Internet Engineering Task Force (IETF) who proposed Automatic Multicast Tunneling (AMT) [6] as a mechanism to enable network devices, in a Unicast-only network environment, to be able to receive Multicast traffic (224.x.x.x/4) even in the absence of end-to-end Multicast connectivity [7].

Additionally, some other research contributions are focusing on techniques to transmit Broadcast, Unknown

Unicast, and Multicast (BUM) multi-destination traffic to end devices connected to common Layer 2 broadcast domains in the overlay network. This research used Multicasting as a method of handling multi-destination traffic more efficiently across a VxLAN overlay network [8]. Other papers highlight concerns over the capabilities of many current physical infrastructures (routers and switches) to support Multicast packets where Multicast groups can contain thousands of participants. This is a fundamental reason why Data Centers are reluctant to implement Multicast communications and why the architectures are evolving towards scalable, virtualized environments [9].

This confirmed that, although research on Multicasting is taking place, it is not addressing the problems inherent in the Edge-to-Core segment of multi-tenant MEC infrastructures where we are conducting our research and analysis. The lack of investigation in the analysis of encapsulated Multicast traffic in overlay networks together with the relevance of this type of traffic has been the main motivation for this research work. Our research will focus on investigating the impact in performance of Multicast traffic in the Edge-to-Core segment of multi-tenant MEC network infrastructures.

Experiments will be performed using the current *de facto* protocols that are used to create virtualized and isolated environments for multi-tenancy MEC networks. This will allow us to investigate, Unicast and Multicast IP communications. The protocols are: VxLAN, NVGRE, STT and GENEVE.

Virtual eXtensible LAN (VxLAN) is a protocol that encapsulates existing L2 frames using a VxLAN/UDP 24-bit tunnel identifier header field (MAC-IP-UDP-VxLAN). The MAC-in-UDP encapsulation creates a logical tunnel that extends the L2 network over an L4 network. It is frequently used in high volume, multi-tenancy and cloud computing architectures where L2 segments are geographically disparate. It provides isolation of logical segments with only the hosts and applications on the same logical network, able to communicate. The overlay network can support Multicasting and other BUM traffic when the carrier network does not support point-to-multipoint traffic. It can also support common tunneling protocol security features such as authentication and encryption [9].

Network Virtualization using Generic Routing Encapsulation (NVGRE) is comparable to VxLAN, but with the ability to carry various non-Ethernet type payloads. The main difference to that of VxLAN is the IP/GRE header (MAC-IP-GRE). Similar to other encapsulation protocols, the VNI field is used to provide a mapping to support Multicasting and other BUM traffic when the carrier network does not support point-to-multipoint connections [9].

Generic Network Virtualization Encapsulation (GENEVE) aims to support the capabilities of VxLAN, NVGRE and STT but address the limitations of changing control-plane specifications. It achieves this by having no defined control-plane information or specifications, relying on other methods, to manage Virtual Tunnel Endpoint (VTEP) connections. This, control-plane independence or pure tunnel format, allows current and future applications to generate packets that are sent to pre-configured destination VTEPs as Unicast or Multicast using UDP as the transport mechanism (MAC-IP-UDP-GENEVE) [9].

Stateless Transport Tunneling (STT) is an encapsulation protocol with the same goals as the previous encapsulation protocols; to extend Layer 2 traffic over scalable IP network infrastructures. It differs from VxLAN and GENEVE by using TCP as its transport protocol method (MAC-IP-TCP-STT) and incorporates a 64-bit tunnel identifier header field. Performance tests have been impressive because of STT's use of the Network Adapter's TCP Segmentation Offload (TSO) engine which has been reported by VMWare as providing a throughput performance that is comparable with non-encapsulated traffic [10]. However, problems can arise when traversing internetwork devices, stateful firewalls and load balancers, with packets being dropped due to the use of a modified TCP header. Although the (STT) TCP header appears to be identical to a normal TCP header, the STT pseudo TCP header must be processed entirely differently as it does not function in the way that a normal TCP header functions. STT is a connectionless protocol that does not require standard TCP functions such as three-way handshakes to establish connections, acknowledgments (ACKs) or retransmissions. Hardware must be able to process the pseudo TCP part of the STT protocol to maximize performance and avoid transmission problems [11].

Table 1 provides an analysis of the protocols previously described against different IP traffic transmission scenarios, which is the basis of our research. The first column refers to the definition of the scenarios where the inner traffic is encapsulated over the outer traffic and transport method. This presents four possible scenarios for each protocol, from the combination of Unicast and Multicast transmissions in both inner and outer networks. For each protocol (columns 2 to 5) there are three values based on a binary YES/NO option to illustrate the following;

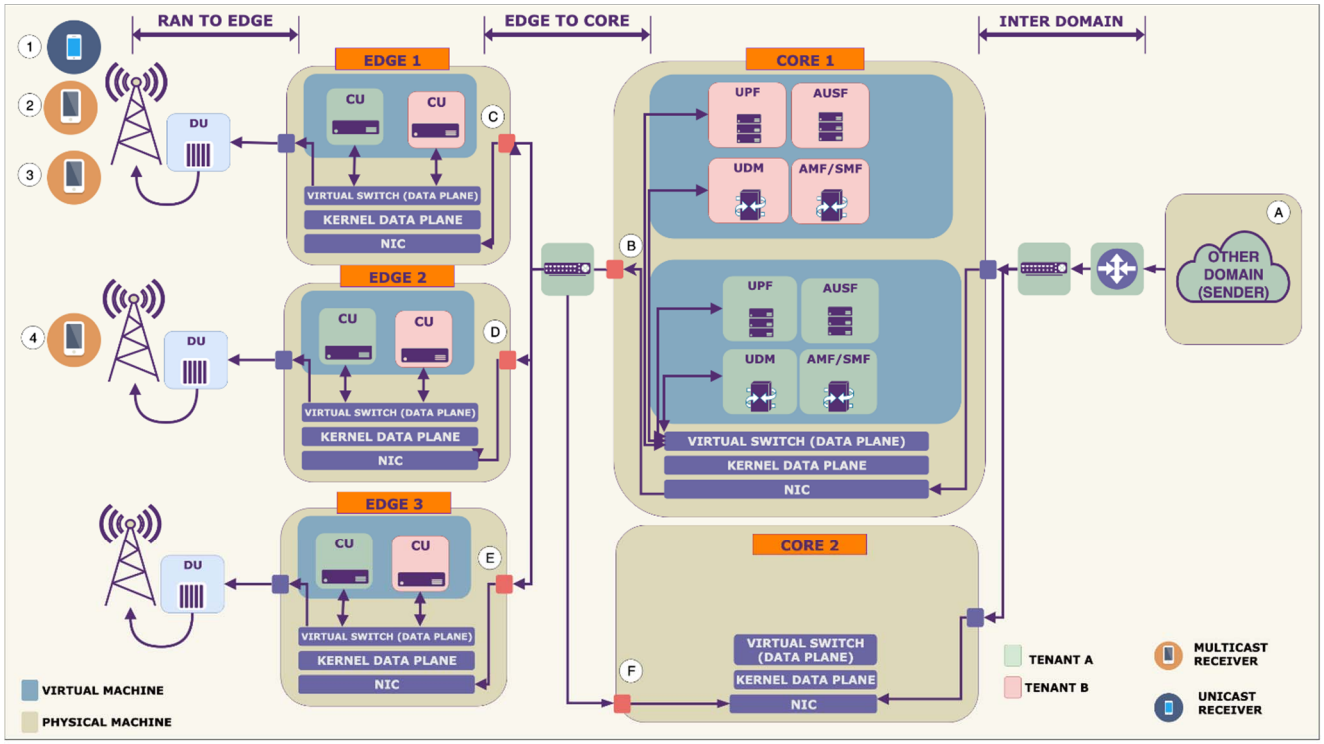
- The capabilities of the protocol to provide support in each of the scenarios based on the (protocol) specification.
- The capabilities of the protocol to provide support for the Linux kernel in each of the scenarios.
- The capabilities of the protocol to provide support in OpenVSwitch.

Notice that the main drawback presented, is that the NVGRE protocol does not provide support for establishing Multicast GRE tunnels, supporting only point-to-point (Unicast) connections. The rest of the protocols can be used with any combination of Unicast and Multicast arrangements.

III. MULT-TENANT MEC ARCHITECTURE

This section provides the reader with an overview of the architecture where this investigation has been carried out in order to contextualize the problem to be addressed.

Figure 1. Overview Multi-tenant MEC Architecture to provide 5G Connectivity



MEC provides a means to reduce network congestion and improve performance by providing simultaneous access to information, services and applications provided by "the cloud" but at the network edge. This, in effect, will reduce latency and provide high bandwidths and data rates by processing data closer to the end user. Figure 1 depicts an overview of a Multi-Access/Mobile Edge Computing (MEC) architecture where the reader can see network segments clearly defined.

The Core network segment is a traditional cloud computing infrastructure Data Center whereas the Edge network segments are an extension of the cloud computing infrastructure that characterizes a geographically distributed infrastructure where there are points of presence (PoP) closer to the final-users, i.e. in the last mile of a 5G infrastructure or a broadband network. MEC makes use of Commercial-Off-The-Shelf (COTS) hardware in both Edge and Core segments to reduce infrastructure costs and implements the use of multi-tenancy to reduce both capital and operational costs.

Notice that multi-tenancy creates different virtual infrastructures, running over the same physical infrastructure to share computational resources whilst maintaining isolation between them. This has been depicted in Figure 1 using different colour schemes to represent different tenants. Also, there are Inter-domain network segments used to interconnect the rest of autonomous system to the network. One of the scenarios associated with MEC architecture is the deployment of 5G Radio Access Network (RAN) segments on each of the edges of the network to allow users to have 5G connectivity. This scenario has been used in this contribution as an example of an application running on the MEC infrastructure. Thus, Figure 1 also shows a set of virtual machines used to deploy 5G architecture inside the MEC.

The virtual machines deployed in the Core Network represent the main 5G architectural functionalities: Authentication Server Function (AUSF), Unified Data Management (UDM), Access and Mobility management Function (AMF), Session Management Function (SMF) and User Plane Function (UPF). The scope of this paper prohibits the inclusion of an in-depth explanation, but if the reader is interested, Kim *et al.*, in a paper entitled 3GPP SA2 architecture and functions for 5G mobile communication system, provide a comprehensive description of the different architectural functionalities available in the 5G architecture [12]. The edge of the network is allocating the Centralized Units (CUs), used in 5G architecture to control the radio links in order to provide connectivity to the 5G

Table 1. Analysis of Multicast Capabilities of the Main Existing Protocols to Implement Overlay Network in Data Centers

Inner Network over Outer Network	VxLAN (L2/L4)	NVGRE (L2/L3)	GENEVE (L2/L4)	STT (L2/L4)
Unicast/Unicast	YES/YES/YES	YES/YES/YES	YES/YES/YES	YES/NO/YES
Multicast/Unicast*	YES/YES/YES	YES/YES/YES	YES/YES/YES	YES/ NO/YES
Unicast/Multicast*	YES/YES/NO	NO/NO/NO	YES/NO/NO	YES/ NO/NO
Multicast/Multicast*	YES/YES/NO	NO/NO/NO	YES/NO/NO	YES/ NO/NO
Transport Protocol	UDP	IP	UDP	TCP

users. All the virtual machines running on the physical machines are interconnected using Linux bridges. In the figure, we can see 3 different Edge physical machines (Edge 1-3) and 2 different Core physical machines (Core 1-2). All of them, are participating in the creation of the overlay network, with the exception of Core 2, which is included for illustration purposes.

IV. Problem Analysis

Let us assume that a sender (Other Domain) located at the border of our administrative domain wants to send traffic to the UEs available in Figure 1. The sender A will send Unicast traffic only to the UE 1 and Multicast traffic, simultaneously, to the UEs 2, 3 and 4 to fulfil the requirements. Thus, two different approaches can be used to deliver these different types of traffic within the overlay networks used to implement tenant isolation: Multicast and Unicast delivery. The expected behaviour would be that Unicast traffic is received in Edge 1 in order to deliver this traffic to UE 1 and that Multicast traffic is received in both Edges 1 and 2 in order to deliver this traffic to the UEs 2, 3 and 4, respectively.

Table 2 provides an analysis of the behaviour of the different delivery methods analysed, i.e. VxLAN and GENEVE since NVGRE does not support Multicast and STT is not supported by the Linux kernel. The different delivery methods are available in the first column. The rest of the columns represent the nodes depicted in Figure 1. Each cell in the table has three values, YES/NO/--, indicating the presence or not of traffic or – if it is not supported. The first one represents the expected (optimal) behaviour whereas the second one represents the behaviour that is in fact happening when using Linux kernel encapsulation. The third value is the behaviour when using OpenVSwitch. Thus, NO/NO/NO is interpreted as the packet is not expected to be received and, is in fact not being received either when using Linux kernel or OpenVSwitch.

Firstly, Unicast over Unicast (see row A in Table 2) traffic shows the expected behaviour, which in turn, confirms the suitability of this type of traffic delivery and the continuous optimizations developed by industry to achieve it.

Secondly, Unicast traffic is sent from location A to the UE 1 using Multicast (see row C in Table 2). Ideally, the traffic should be received only in Core 1 and Edge 1 since this is the data path involved in the communication between Sender and UE 1. The reasons why, ideally, traffic should not be forwarded at locations D, E and F are based upon the following;

- Core 2 does not have any UPF and thus does not participate in the overlay network.
- Edge 3 has a CU but, does not have any users.
- Edge 2 has both CU and users, but it is not the destination address of the Unicast traffic.

The reality is that encapsulated ARP traffic is sent to all the network devices, even to those that are not part of the overlay network and in principle should not receive them, causing a privacy concern (see *1 in Table 2). The operation of ARP requires devices to send ARP requests and replies using Unicast traffic rather than continuing to use Multicast traffic to optimize delivery.

Thirdly, when Multicast traffic is sent over Unicast (see row B in Table 2), it needs to be sent to all the different devices connected to the overlay network and is sent as multiple Unicast transmissions. This fact requires *n-times* replication of the associated Multicast traffic and generates a significant waste of precious bandwidth and processing resources which is the main motivation for transmitting using Multicast. Even worse, the replicated traffic will arrive at those computers that are not expecting to receive this traffic (E and F) when using OpenVSwitch. This replication generates a significant inefficiency in the delivery of Multicast traffic over Unicast tunnels due to the waste in bandwidth (inefficiency labelled as *2 in Table 2). It also generates traffic to receivers that are not supposed to receive it, again, creating a potential privacy concern (inefficiency labelled as *3 in Table 2). Finally, when Multicast traffic is sent over Multicast (see row D in Table 2), the reader would expect an optimized delivery method. However, this is not the case; all the devices involved in the overlay network will receive the traffic even if they do not contain any users that are subscribed to the Multicast channel (E). As the reader can see from the analysis, there are various reasons why there is a significant lack of support for an efficient delivery method for Multicast traffic in overlay networks. To share this analysis with the scientific community, has been the main motivation for this contribution.

V. VALIDATION

A. Analytical Results

Prior to the experiment taking place, a number of theoretical assumptions were discussed to ascertain the expected results and to provide a datum to compare against the actual results later on. Our initial thoughts were that Multicast (point-to-multipoint) traffic will be more efficient, with less bandwidth being consumed compared to Unicast (point-to-point) transmissions. We also decided that to validate our experiments, the experiments will be performed three times under identical conditions to ensure that the data packets being sent are comparable with the data packets and overhead being received at each Core, Edge, Multicast and Unicast UE device. The data packets being sent refer to the inner, encapsulated traffic. Other assumptions were to ensure that all Multicast group hosts were required to have a unique Multicast MAC address and any unnecessary network traffic and services were prohibited on our testbed network to maximize efficiency and to validate the test environment. From the MEC Architecture diagram (figure 1), the data path is considered to be from A (Sender) via the Core, Core to Edge and Edge to RAN, in order to reach each of the UEs connected to the RAN. Primarily, an investigation and analysis of Multicast and Unicast transmissions in the Edge-to-Core segment of Multi-tenant MEC infrastructures has been carried out, providing us with the expected behavior

Table 2. Analysis of Delivery Methods in 5G Multi-tenant Overlay Networks using Architecture depicted in Figure 1.

Inner Network over Outer Network	B (CORE 1)	C (EDGE 1)	D (EDGE 2)	E (EDGE 3)	F (CORE 2)
(A) Unicast/Unicast	YES/YES/YES	YES/YES/YES	NO/NO/NO	NO/NO/NO	NO/NO/NO
(B) Multicast/Unicast	YES/YES/YES	YES/--/YES	YES/--/YES(*2)	NO/--/YES (*3)	NO/--/YES (*3)
(C) Unicast/Multicast	YES/YES/--	YES/YES/--	NO/YES/-- (*1)	NO/YES/-- (*1)	NO/YES/-- (*1)
(D) Multicast/Multicast	YES/YES/--	YES/YES/--	YES/YES/--	NO/YES/-- (*3)	NO/YES/-- (*1)

about the efficiency of each Unicast and Multicast transmission protocol analyzed. Later, these results will be used as a comparison to the empirical results achieved.

Table 3 has a similar structure to Table 2 with the addition of all the different encapsulation protocols analyzed in the experiment. In fact, each cell of Table 3 has three values, but this time they are numerical values. X/Y/Z, indicating, namely, X: expected behavior, Y: Empirical Results Achieved using Linux Kernel, Z: Empirical Results Achieved using OpenVSwitch.

B. Testbed

All the empirical data gathered from the experiments to demonstrate the inefficiencies in Multicast delivery have been executed on a computer with the following hardware;

Specifications: Dell T5810 with an Intel Xeon E5-2630 v4 CPU, 10 cores with hyper-threading, 32GB RAM, 512 GB SSD, 10 Gbps NIC. This computer runs a virtualized infrastructure matching exactly the 5G architecture presented in Figure 1. The virtual infrastructure is using the two different technologies under investigation, Linux Kernel and OpenVSwitch, as virtual switches to implement the different overlay networks being investigated.

The following excerpts of code are example of how the Edge (and the Core) OVS switches located in the virtual computers are configured to create VxLAN tunnels using both Unicast and Multicast delivery methods, respectively.

```
sudo ovs-vsctl add-port br0 vxlan1 -- set
interface vxlan1 type=vxlan
options:remote_ip=10.0.0.1

sudo ovs-vsctl add-port br0 vxlan1 -- set
interface vxlan1 type=vxlan
options:group=224.1.1.1
```

The following excerpt of code is an example of how the Linux kernel switches are configured to create VxLAN tunnels using both Unicast and Multicast, respectively.

```
ip link add vxlan1 type vxlan id 1 remote 10.0.0.1
dstport 0 dev ens3
ip link add br0 type bridge
ip link set vxlan1 master br0
ip link set vxlan1 up
ip link set br0 up

ip link add vxlan1 type vxlan id 1 group 224.1.1.1
dstport 0 dev ens3

ip link add br0 type bridge
ip link set vxlan1 master br0
ip link set vxlan1 up
ip link set br0 up
```

Once the tunnels are configured, 1 GB of traffic is sent from location A to the Unicast address of UE1 for testing the transmission of Unicast traffic. 1 GB of traffic has also been sent from A to the Multicast address where UE2, UE3 and UE4 are subscribed, in order to test the transmission of the Multicast traffic.

C. Empirical Results

Table 3 shows the results from the experiments performed using the different encapsulation protocols (as described earlier in the paper) using the corresponding Multicast and Unicast combinations. Empirical results shown in Table 3 validate the analysis of the behaviour of the delivery methods represented in Table 2. In fact, the reasons for the inefficiencies annotated as 1-3 in Table 2 (previously explained in section IV), corroborate the inefficiencies achieved in the empirical validation. The annotations in Table 2 (1-3) match the numbering of the annotations in Table 3. It is worth noting that the STT protocol is not supported in the Linux kernel (see *4 in Table 3). This is the reason why there is a lack of empirical data for the experiment using this protocol. Also, STT on OpenVSwitch does not provide support for tunnels connected using Multicast (see *5 in Table 3). This is the reason why there is a lack of empirical data for the experiment of this type of traffic when using STT tunnels. In terms of bandwidth consumption, over the 10 Gb/s link, it is worth mentioning the benefits of Multicasting as a delivery method since the sender is sending 1 GB of data and the total data received by all the nodes is expected to be 2 GB of data for (Unicast over Unicast) and 3 GB for the other use cases analysed as indicated in the column *"Total Data Received by All Nodes"* shown in Table 3. For the case of Unicast over Unicast, (see row A in Table 3), the reader can see how the empirical results are close to the expected results, which in turn, validates the efficiency of this kind of delivery. The difference between the expected results and the empirically

Table 3. Analytical and Empirical Results using a Testbed as depicted in Figure 1.

Inner Network over Outer Network	Protocols	A (Sent) GBytes (X/Y/Z)	B (Core 1) GBytes (X/Y/Z)	C (Edge 1) GBytes (X/Y/Z)	D (Edge 2) GBytes (X/Y/Z)	E (Edge 3) GBytes (X/Y/Z)	F (Core 2) GBytes (X/Y/Z)	Total Data Received by All Nodes
(A) Unicast/ Unicast	VxLAN	1.00/1.00/1.00	1.00/1.05/1.05	1.00/1.05/1.05	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00	2.00/2.10/2.10
	GRE	1.00/1.00/1.00	1.00/1.04/1.04	1.00/1.04/1.04	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00	2.00/2.08/2.08
	GENEVE	1.00/1.00/1.00	1.00/1.05/1.05	1.00/1.05/1.05	0.00/0.00/0.00	0.00/0.00/0.00	0.00/0.00/0.00	2.00/2.08/2.10
	STT (*4)	1.00/-----/1.00	1.00/-----/1.07	1.00/-----/1.07	0.00/-----/0.00	0.00/------/0.00	0.00/------/0.00	2.00/-----/2.14
(B) Multicast/ Unicast	VxLAN	1.00/1.00/1.00	1.00/-----/1.05	1.00/-----/1.05	1.00/-----/4.20 (*2)	0.00/-----/1.05 (*3)	0.00/------/1.05 (*3)	3.00/-----/8.40
	GRE	1.00/1.00/1.00	1.00/-----/1.04	1.00/-----/1.04	1.00/-----/4.15 (*2)	0.00/-----/1.04 (*3)	0.00/------/1.04 (*3)	3.00/-----/8.29
	GENEVE	1.00/1.00/1.00	1.00/-----/1.04	1.00/-----/1.04	1.00/-----/4.20 (*2)	0.00/-----/1.04 (*3)	0.00/------/1.04 (*3)	3.00/-----/8.36
	STT (*4)	1.00/-----/1.00	1.00/-----/1.07	1.00/-----/1.07	1.00/-----/4.28 (*2)	0.00/-----/1.07 (*3)	0.00/------/1.07 (*3)	3.00/-----/8.56
(C) Unicast/ Multicast	VxLAN	1.00/1.00/1.00	1.00/1.05/------	1.00/1.05/-----	0.00/1.05/-----	0.00/1.05/----- (*1)	0.00/1.05/------ (*1)	3.00/5.25/-----
	GRE	1.00/1.00/1.00	1.00/1.04/------	1.00/1.04/-----	0.00/1.04/-----	0.00/1.04/----- (*1)	0.00/1.04/------ (*1)	3.00/5.16/-----
	GENEVE	1.00/1.00/1.00	1.00/-----/------	1.00/-----/-----	0.00/-----/-----	0.00/-----/----- (*1)	0.00/------/----- (*1)	3.00/-----/-----
	STT (*4) (*5)	1.00/-----/-----	1.00/-----/------	1.00/-----/-----	0.00/-----/-----	0.00/-----/----- (*1)	0.00/------/----- (*1)	3.00/-----/-----
(D) Multicast/ Multicast	VxLAN	1.00/1.00/1.00	1.00/1.05/-----	1.00/1.05/-----	1.00/1.05/-----	0.00/1.05/----- (*3)	0.00/1.05/------ (*1)	3.00/5.25/-----
	GRE	1.00/1.00/1.00	1.00/1.04/-----	1.00/1.04/-----	1.00/1.04/-----	0.00/1.04/----- (*3)	0.00/1.04/------ (*1)	3.00/5.16/-----
	GENEVE	1.00/1.00/1.00	1.00/-----/-----	1.00/-----/-----	1.00/-----/-----	0.00/------/---- (*3)	0.00/------/----- (*1)	3.00/-----/-----
	STT (*4) (*5)	1.00/-----/-----	1.00/-----/-----	1.00/-----/-----	1.00/-----/-----	0.00/------/---- (*3)	0.00/------/----- (*1)	3.00/-----/-----

achieved ones is mainly due to the overhead associated with the creation of new network headers when creating the encapsulation tunnels. In this respect, the encapsulation protocol that offers the best performance in terms of efficiency of overhead is Generic Routing Encapsulation (GRE).

When analyzing the other data, the results are completely different from what was expected. On the one hand, the use of Multicast to connect overlay tunnels (see rows C and D in Table 3) is, surprisingly, lacking support in OpenVSwitch and when implemented in the Linux kernel, it shows this method to be completely inefficient with around 5 GB of data being received, i.e. 72-75% more than expected. On the other hand, the use of Unicast tunnels to deliver Multicast traffic is even less efficient, not being supported in the Linux Kernel and only implemented in OpenVSwitch using Unicast replication. As a result, the data received is around 8.5 GB, i.e. 176-185% more than expected and is clearly a symptom of potential bottlenecks, having implications in both scalability and security. Another concern is that this inefficiency could also be misinterpreted, by displaying similar characteristics to that of an attack vector to achieve a denial of service (DoS).

These empirical results validate the findings about the current inefficiencies associated with Multicast traffic when using overlay networks. The results also highlight the lack of efficient delivery methods and support for Multicast overlay networks as well as the potential security issues associated to this kind of traffic.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented an analysis of the different delivery methods of traffic using overlay networks. The analysis has demonstrated, by empirical evaluation, the inefficiencies in the delivery of Multicast traffic in overlay networks, showing inefficiencies of up to 176% in overhead using the best-case encapsulation protocol. It has also demonstrated inefficiencies in the delivery of any traffic using Multicast to interconnect overlay networks, showing inefficiencies of 72% in overhead in the best-case scenario. This paper has demonstrated that GRE is the best performer in terms of efficiency of overhead from those protocols analyzed. As a future work, the authors plan to focus on the design and implementation of novel protocols and architecture to allow the efficient delivery of Multicast traffic in overlay networks.

VII. ACKNOWLEDGMENT

This work has been part funded by the European Commission Horizon 2020 under Grant Agreement Number H2020-ICT-2016-2/761913 (SliceNet: End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks).

VIII. REFERENCES

[1] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li and A. Ranjan, "www.etsi.org," June 2018. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf. [Accessed June 2019].

[2] S. Redana and A. Kaloxylos, "5G PPP Architecture Working Group - View on 5G Architecture," July 2016. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf. [Accessed March 2019].

[3] M. Fallgren and K. Manolakis, "5G PPP Automotive Working Group - A study on 5G V2X Deployment," February 2018. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/02/5G-PPP-Automotive-WG-White-Paper_Feb.2018.pdf.

[4] Cisco Systems, "IP Multicast Technology Overview," 18 April 2002. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html. [Accessed October 2018].

[5] A. Zainab, A. Imtiaz and I. Hussain, "Multicasting in software defined networks: A comprehensive survey," *Journal of Network and Computer Applications,* vol. 104, pp. 61-77, 2018.

[6] G. Bumgardner, "Internet Engineering Task Force, Request for Comments 7450," February 2015. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7450.txt. [Accessed October 2018].

[7] A. Ferdous and W. J. Atwood, "An Architecture for a Secured Tunnel in the Automatic Multicast Tunneling (AMT) Environment," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2016.

[8] E. F. Naranjo and D. S. C. Gustavo, "Underlay and Overlay Networks: The approach to solve addressing and segmentation problems in the new networking era," IEEE, 2017.

[9] O. Komolafe, "IP Multicast in Virtualized Data Centers: Challenges and Opportunities," in *2017 IFIP/IEEE International Symposium on Integrated Network Management (IM2017)*, Lisbon, Portugal, 2017.

[10] T. Koponen and e. al, "Network virtualization in multi-tenant datacenters," VMware, Palo Alto, USA, 2013.

[11] R. Kawashima and H. Matsuo, "Accelerating the Performance of Software Tunneling Using a Receive Offload-Aware Novel L4 Protocol," *IEICE Transactions on Communications,* Vols. E98-B, no. 11, pp. 2180-2189, 2015.

[12] K. Junseok, K. Dongmyoung and C. Sunghyun, "3GPP SA2 architecture and functions for 5G mobile communicationsystem," Elsevier, 2017.