Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 497 – 501

Procedia Computer Science

International Workshop on Applications of Software-Defined Networking in Cloud Computing (SDNCC)

MAC Based Dynamic VLAN Tagging with OpenFlow for WLAN Access Networks

Marc Koerner[a], Odej Kao[a]

[a]*Technische Universitaet Berlin, Einsteinufer 17, 10587 Berlin, Germany*

Abstract

Many network device vendors are providing a vendor specific VLAN based access solutions for WLAN clients. This applications allows network operators to specify WLAN devices which automatically fall into their department specific networks ans allows them to access their local resources like e.g. printers. The configuration of these VLAN mappings is usually manufacturer specific and depends also on the local VLAN policies. However, the presented OpenFlow approach on the other hand presents a solution to encapsulate this functionality as network application. Thus, an architecture, implementation, and evaluation is presented in order to demonstrate that this particular functionality can be easily realized in an OpenFlow network application.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Software Defined Networking; OpenFlow; VLAN tagging

1. Introduction

Software-defined networking (SDN)[1] is continuously evolving nowadays networks and network applications. Researchers and network operators are focused on integrating traditional network solutions on this innovative technology. One of the main challenges are applications for productive environments, so vendors are more and more promoting SDN applications and products for enterprise networks. SDN provides a basic opportunity to evolve these networks in general, since all packets are now processed by a centralized management entity. This mechanism provides the opportunity for several new network applications covering all network management functions from a loop free path deployment up to the virtual local area network (VLAN) separation.

VLANs are an useful tool to fragment a network into different segments and reduce the size of a broadcast domain. Moreover, they are an opportunity to enforce security and control client access within the network. Usually the network operation control (NOC) department in a company or university is using this to segregate the different departments. Moreover, an additional MAC to VLAN mapping mechanism is utilized in order to embed mobile clients

* Corresponding author. Tel.: +49-30-314-78583; fax: +49-30-314-21060.
E-mail address: marc.koerner@tu-berlin.de

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.077

into their department network independently where the access the WLAN infrastructure. Therefor usually the NOC operator just adds the mapping information to authentication, authorization and accounting (AAA) system. Thus, if a known user connects his traffic will be VLAN tagged and routed within the network into his associated department network.

The remaining paper is structured as follows. Section 2 explains the background of this work. In sec. 3 is briefly explained how the proposed MAC to VLAN mapping concept is used and how it is working. In contrast, in sec. 5 a detialed evaluation based on Mininet is presented. Finally, in sec. 6 the concept and evaluation is concluded.

2. Background

The SDN paradigm offers by default a separation of the control- and data-plane for packet forwarding elements. It builds a logical separation between the packet forwarding hardware and the control logic. The concept further extends this control approach by introducing a centralized control entity called controller or network operating system (NOS). It follows a similar approach like the host based operating systems, where the business logic runs on top. This is an excellent opportunity to encapsulate all network functionalities in virtual network functions (NFV).

MAC-based dynamic VLAN tagging is a procedure where the switch usually detects the VLAN affiliation of a host which gets connected to one of its ports. The VLAN affiliation is detected based on the hosts MAC address. Cisco implementation uses a mechanism called VLAN Membership policy Server (VMPS). This is basically a database which stores the MAC / VLAN mapping information entered by the NOC operator. If a new device is connected to a switch port of the access network switches, the regarding switch queries the VMPS for the concerning MAC address of the new host. Thus, the VMPS browses its database using the obtained MAC address for the corresponding VLAN ID. Afterwards the switch adapt its internal port configuration according the new VLAN assignment. This flexibility makes dynamic VLAN the ideal solution for networks whose composition often changed[2] .

3. Architecture

As introduced in the previous section, there exist proprietary hardware solution which provide dynamic VLAN tagging. However, in the remaining paper an open source solution is introduced providing the same features on community OpenFlow[3,4] hardware.

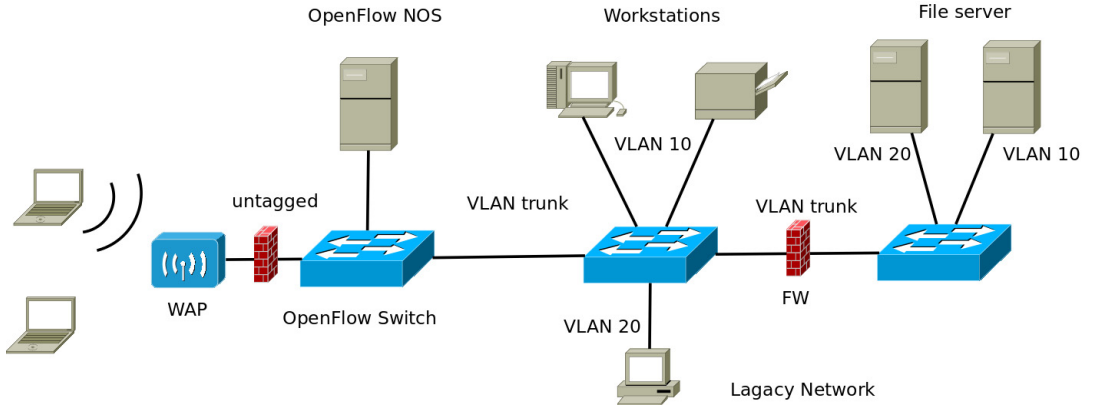

Fig. 1. Typicall campus network department based use-case

Figure 1 illustrates the proposed OpenFlow based solution for a typical campus network. It shows a network basically composed out of two legacy switches and OpenFlow switch behind an wireless access point (WAP). This is a typical SDN-legacy hybrid network. The legacy switches use port-based VLANs to separate workstations, printers and file servers on a department level. These devices do not change frequently and have a fixed VLAN mapping. In contrast to the mobile systems, such as laptops. These systems change often their locations within the university area.

However, mobile laptop based workstations should be connected to their department VLAN in order to access their usual services and hardware. Therefore, the presented prototype uses OpenFlow switches behind the WAP to manage the dynamic VLAN tagging to redirect the mobile workstation's network traffic into the regarding VLAN. Thus, the implemented MAC based VLAN tagging network function (NF) is covering this particular lookup process and adds or strips the VLAN tags. This NF was implemented as a network application based on the Floodlight internal Java API. The only requirement is an OpenFlow enabled WAP aggregation switch.

4. Implementation

The implementation is composed out of two applications, a helper tool, and the NF which is containing the business logic. The helper tool is a small java application which was developed in order to stores the client MAC / VLAN mapping information within an XML file. Moreover, in order to realized the MAC-based dynamic VLAN tagging a Java based NF was implemented using the Floodlight internal java API. This NF makes use of OpenFlow protocol utilizing floodlight to cover the VLAN tagging and un-tagging process directly behind the WAP.

5. Evaluation

This section deals with the evaluation of the implemented NF and the environment used for the evaluation. Figure 2 shows the evaluation network which was generated by deploying two Mininet[5] instances. Each Mininet instance is running in its own virtual machine. The first instance is using an Open vSwitch in OpenFlow mode with an external controller, which is hosted on the physical host. The OpenFlow enabled switch is connected with three three virtual hosts. The dynamic VLAN tagging will later on be processed on this switch using the instructions coming from the implemented NF running on the Floodlight controller. In the second Mininet instance a legacy network is created consisting of an Open vSwitch operating in a regular MAC learning switch mode with VLAN tagged ports, as shown in figure 2. This switch is also connected with three virtual hosts. Moreover, both Mininet instance are connect to each other by using the gateway function and a GRE tunnel. This connection is used for VLAN tagged traffic to the trunk port of instance two.

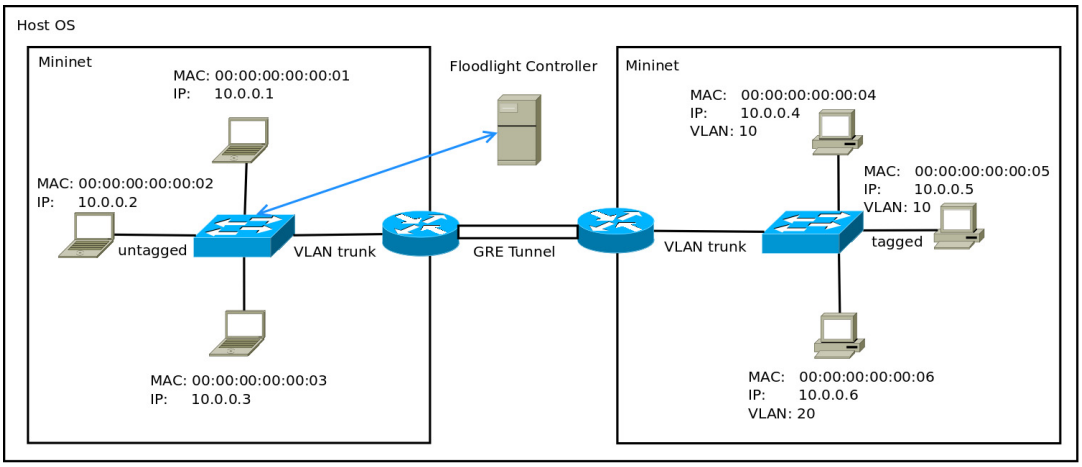

Fig. 2. Evaluation network topology with two Mininet instances

The following subsections will demonstrated that the untagged internal and untagged to VLAN communication is working using the introduced OpenFlow approach in combination with a usual hybrid network. However, the used (sub)network IP range is (10.0.0.0/24). This IP configuration was chosen for both VLAN configurations in order to demonstrate that communication between systems with different MAC/VLAN mappings (h1 / vlan-id = 10, h3 / vlan-id = 20) does not work independently of the shared IP range.

5.1. Untagged WLAN network communication

In this first evaluation scenario an ICMP request from host 1 (10.0.0.1) to host 2 (10.0.0.2) is used to demonstrate the ”wireless” internal untagged communication.

```
 cookie=0x0, duration=42.085s, table=0, n_packets=5, n_bytes=378, idle_timeou
t-60, priority-1000,dl_src-00:00:00:00:00:02,dl_dst-00:00:00:00:00:01 actions
=output:1
```

```
 cookie=0x0, duration=41.088s, table=0, n_packets=4, n_bytes=336, idle_timeou
t-60, priority-1000,dl_src-00:00:00:00:00:01,dl_dst-00:00:00:00:00:02 actions
=output:2
```

Fig. 3. Flow entries within the flow table: ping from h1 to h2 (untagged)

By using the Open vSwitch control tool and running the following command *sudo ovs-ofctl dump flows s1* it is possible to obtain the flow table entries and stats form the OpenFlow switch. It lists the matching fields, statistics like number of matched packets as well as byte length, and the actions associated with the entry. The entries are look like a normal L2 forwarding entries.

5.2. WLAN untagged to VLAN communication

To evaluate the MAC to VLAN mapping an ICMP request is send form host 1 (10.0.0.1) to host 4 (10.0.0.4). Figure 4 shows the flow entries that are created. When an ICMP request arrives, firts of all the NF which is in charge of the VLAN mapping advises the OpenFlow switch to send the ARP request unchanged to all switch ports (port 2) which are not part of a trunk link connection. After this, it creates a flow entry (red border)for the VLAN mapping in order to resolve the layer three network address of a host with a VLAN connection. This is processed by forwarding the ARP requests from host 1. Therefore it takes the MAC address of host 1 and the usual L2 Broadcast address (ff: ff: ff: ff: ff: ff) with the data link layer source and destination address as match fields. Afterwards, it adds the corresponding MAC to VLAN tag and sends it again with the appropriate VLAN ID on the port with the trunk link.

```
 cookie=0x0, duration=42.147s, table=0, n_packets=25, n_bytes=1050, idle_time
out=60, priority=1000,dl_src=00:00:00:00:00:01,dl_dst=ff:ff:ff:ff:ff:ff actio
ns=output:2,push_vlan:0x8100,set_field:4106->vlan_vid,output:4
```

```
 cookie=0x0, duration=12.314s, table=0, n_packets=10, n_bytes=924, idle_timeo
ut=60, priority=1000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04 action
s=push_vlan:0x8100,set_field:4106->vlan_vid,output:4
```

```
 cookie=0x0, duration=13.184s, table=0, n_packets=11, n_bytes=1010, idle_time
out=60, priority=1000,dl_vlan=10,dl_src=00:00:00:00:00:04,dl_dst=00:00:00:00:
00:01 actions=pop_vlan,output:1
```

Fig. 4. Flow entries within the flow table: ping from h1 to h4 (tagged)

When the controller and therewith implemented NF is seeing the reply, it has all necessary information to deploys the other two flow entries for the data exchange. Thus, the gray and yellow bordered boxes are the flow entries responsible for passing the ARP Response and the following ICMP packets.

Again, the first flow (red border) is for forwarding of the ARP requests. The second flow (gray border), is responsible for forwarding the ICMP packets from host 1 to host 4. This flow is not only forwarding the packets, it is also tagging them with the regarding VLAN tag. Afterwards, the tagged packet is forwarded via the trunk port (output: 4). The third flow (orange border) is responsible for forwarding the ARP response packets and the ICMP response

packets. Since they have an untagged destination address the flow entry come with an remove/strip (Pop vlan) VLAN action. Subsequently, the packet is be forwarded to switch port 1 (Output: 1).

In this paragraph some of the utilized OpenFlow instructions are briefly explained exemplary. Thus, there are different instructions. So the instruction corresponding to the push vlan tag is to modify the Ethernet header (0x8100) an eventually add the VLAN tag to the Ethernet frame. The hexadecimal value 0x8100 defines an VLAN tagged Ethernet type. The output instruction output enforces the switch to send out the packet on the corresponding physical port (output: 1). The instruction pop vlan instruction corresponds to the removal of a VLAN tags. Thus, the match field dl vlan must be checked to ensure that the Packet has an actual VLAN tag.

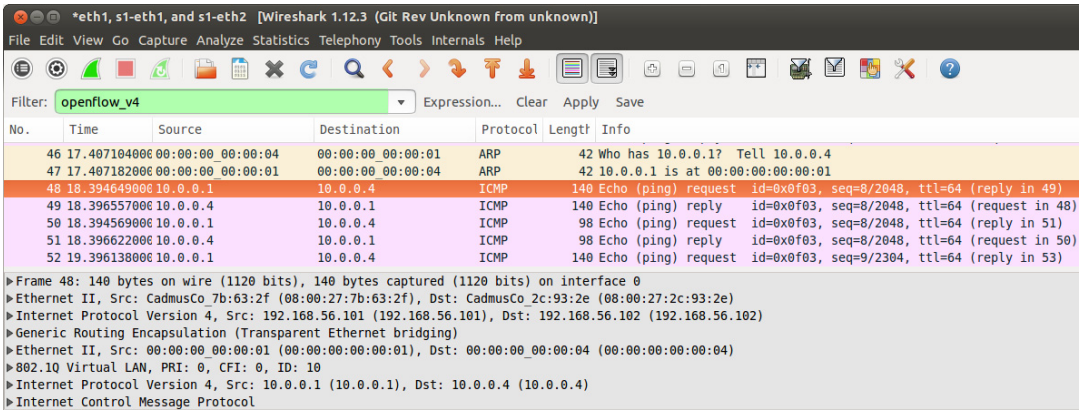

Fig. 5. Wireshark dump: ping from h1 to h4 (tagged)

Again, figure 5 shows a packet dump collected with wireshark during the ICMP request between host 1 and host 4. It demonstrates that the packets were successfully VLAN tagged by the implemented NF and a stable communication has been established.

5.3. The evaluation in a nutshell

The described approach is working for the evaluated scenarios and the operation of the MAC-based dynamic VLAN tagging was verified. Mininet is a reasonable emulator to evaluated the proposed infrastructure. It was investigated that the created flows are processing as expected. In addition, the Wireshark capture is confirming the observations of the flow tables and the resulting assumptions. Thereby a before and after picture of the structure of the packets was proven. This means the dynamic VLAN tagging was evaluated and verified successfully.

6. Conclusion

The introduced prototype is a full functional proof of concept work, which demonstrates the variety of SDN applications. It was developed and evaluated within the scope of SDN application for campus and enterprise networks. The introduce application demonstrates the variety of common SDN applications and can be easily implemented on nearly ever available controller platform, since it utilizes a trivial OpenFlow mechanism.

References

1. ONF, . Software-defined networking: The new norm for networks. 2012.
2. Jungbluth, T., Donner, A.. Dynamische vlans abgeschirmte work- grouploesung mit hoher flexibilitaet. 2009.
3. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., et al. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 2008;.
4. Consortium, T.O.. Openflow switch specification. 2012. `http://www.openflow.org/wp/documents/`.
5. Team, M.. Mininet. 2014. `http://mininet.org/`.