OpenFlow Enlightenment

Extending lightpaths through the campus network

Diederik Vandenvenne
10436251

Tjebbe Vlieg
5647185

Master thesis

Master System and Network Engineering

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisors
Marijke Kaat
Ronald van der Pol

SURFnet

July 24, 2013

Abstract

Lightpaths are a service offered by SURFnet to provide users with a private network link between two locations, which is typically characterized by a high bandwidth and low jitter, packet loss and latency. Although lightpaths can be set up dynamically, the path through the campus network has still proven to be problematic in terms of usability. In this research paper various scenarios of this problem are investigated: we examine campus networks that consist of fully layer two VLAN switching; layer three routing; and MPLS switching. For all of these scenarios we propose possible solutions and subsequently describe their feasibility in the light of both the OpenFlow specification[1] and the capabilities of the physical switches that were used in our testbed.[2] We show that although the specification supports most of the proposed solutions, albeit optionally, the implementation on the Pica8 P3290 switch is still incomplete and therefore not capable of running the solutions.

keywords: OpenFlow, lightpaths, bandwidth-on-demand.

[1] Openflow v1.2

[2] Pica8 P3290

Contents

1 Introduction

Since 2007, SURFnet (the dutch NREN[3]) offers a service called dynamic 'SURFlightpaths'[1] to its customers. Lightpaths provide researchers with a direct, secure and fast connection from one end point in SURFnet its network to another end point. The introduction of dynamic lightpaths gave researchers the opportunity to automatically allocate a lightpath within seconds of time. However, there are still obstacles in making lightpaths more accessible to a wide range of end users. Those obstacles do not lie in SURFnet its backbone network, but rather in connecting the end user to the end point of the lightpath. Even though this problem is situated in the campus network, and thus falls under administrative responsibility of the campus network administration, an easy and dynamic solution would be desirable for SURFnet nonetheless, as this could have a positive impact on the amount of lightpath users.

OpenFlow, a relatively new open standard which enables researchers to test experimental protocols on production networks[2], could provide the functionality and flexibility to implement solutions for the aforementioned problem of connecting end users to their reserved lightpath. OpenFlow gives new possibilities for switching – and even routing – policies than the conventional destination MAC address switching that is currently in use. A powerful feature of OpenFlow is its capability to install switching rules (called 'flows') on the fly. Multiple fields in packet headers can be used for the purpose of matching packets onto flows, such as source and destination MAC address, IP addresses, TCP port numbers, but also VLAN or MPLS tags. These functionalities could form a powerful toolkit possibly able to solve lots of networking problems.

The goal of this research study is to investigate whether the use of OpenFlow is viable to the problem of connecting end users to their lightpaths, and whether its current implementation – on the Pica8 P3290 switch[3] – is mature enough to provide those solutions. The research question is therefore formulated as follows:

Given the wide variety of campus networks, what solutions exist to provide end users with fast and easy access to lightpaths in a dynamic and secure way, using OpenFlow on Pica8 P3290 switches?

This research paper is structured as follows: section 2 covers related work; in section 3 the problem space is described; in section 5 an overview of the

[3]National research and education network

proposed solution is given; section 4 describes the approach of this research study; from section 6 to 8 overviews of various types of campus networks along with applicable solutions are given; and finally, in section 9 conclusions are drawn.

2 Related Work

Rolf Biesbroek[4], in collaboration with Richa Malhotra[5] and Pieter-Tjerk de Boer[6], have done research on extending lightpaths to the user its desktop. Although this research study, which was started in 2012, is not yet finished, an extended abstract[4] as well as a presentation of their work can be downloaded from the website of the Terena Networking Conference 2013[7]. In their research project they present an overview of various techniques that can be used to extend lightpaths through the campus network. The main focus of their research project is on traffic characteristics in terms of latency, jitter, packet loss and guaranteed bandwidth of the extended lightpath, as well as the implications on 'normal' campus network traffic, illustrated by tests ran on the campus network of the University of Twente.

However, as their main concern is performance, Biesbroek et al. do not pay much attention to usability of the resulting system, although the configuration of end to end connectivity of lightpaths has proven to be still far from a trivial task. Our research project aims at bridging this gap by providing dynamic solutions from the end of the lightpath to the user its desktop.

Another aspect that does not get any attention in the aforementioned work is security. When a lightpath is connected to a campus network, the requesting party does not want other users of the network to be able to access the lightpath. Usage should be limited exclusively to authorized persons. In this research study ways of enabling this level of security will also be explored.

A research study that investigates the current implementation of OpenFlow with emphasis on its scalability, is performed by Michiel Appelman and Maikel de Boer, both master students at the University of Amsterdam at that time, in 2012[5]. In their report, they describe various experiments testing the performance and hardware characteristics of the Pica8 P3290 OpenFlow

[4]Masters student at the University of Twente
[5]Product manager for network services at SURFnet
[6]Associate professor at the University of Twente
[7]https://tnc2013.terena.org/core/presentation/45

switch. The main difference between the work done by Appelman and de Boer and this research study is their focus on hardware performance, whereas this study focuses on functionality of both the OpenFlow specification and its implementation on the Pica8 P3290 switch.

3 Problem space

3.1 SURFnet lightpaths

SURFnet lightpaths are private links in the SURFnet network connecting two locations. Characteristics of lightpaths are a guaranteed bandwidth, low jitter, packet loss and latency. Possible usages of lightpaths are the migration of big amounts of data or guaranteed delivery of critical traffic, e.g. astronomy data or remote surgery traffic flows.

When a user wants to use a lightpath, he first has to request a minimum of two virtual ports[8], which will be the end points of the requested lightpath. The request for a virtual port is done via a web interface[9] provided by SURFnet, and has to be approved by a contact person on the campus side. This procedure normally takes a few days, but has only to be done once, meaning that once a user has access to a virtual port, the network administrator does no longer need to bother with the reservation of lightpaths.

After the necessary virtual ports have been acquired, the user can continue to reserve a lightpath, using the same web interface as was used to request virtual ports. The user has to select two virtual ports, a desired bandwidth, a start and possibly an end time. If the lightpath request is legit, i.e. if the user is authorized and there is enough bandwidth available, the reservation is made and the lightpath is set up dynamically. Otherwise, if the lightpath reservation is not possible, the user is notified.

From SURFnet7[10] on (the latest generation SURFnet network, which is currently in development), lightpaths can be connected to the campus network by means of a single multiservice port. This port is not only used for lightpaths, also all the other traffic which does not belong to lightpath users is

[8] In SURFnet its latest network a virtual port corresponds in effect to a VLAN id in a multiservice port.

[9] https://dashboard.surfnet.nl/login.php

[10] http://www.surfnet.nl/en/Hybride_netwerk/surfnet7/Pages/default.aspx

going through this port. All services in such a multiservice port are distinguished by means of VLANs, so each lightpath will have its own VLAN tag.

3.2 Scope

The goal of this research study is twofold: first, methods for connecting end users with lightpaths through campus networks are investigated, and second, the viability of using OpenFlow for implementing these methods on hardware switches is investigated, providing a snapshot of the current maturity of a specific OpenFlow implementation.

As the development of methods for connecting end users with lightpaths is concerned, this study is limited to three classes of networks: layer two switching; layer three IP routing; and MPLS switching. Other types of campus networks, as far as they are not covered by the ones summarized above, are considered out of scope.

Al presented methods for extending the lightpath focus on functionality. The most important aspect of the problem is to let end users use their reserved lightpath from a location in the campus network, as that is where the biggest problems are now. For this reason, quality of service is considered out of scope.

Another problem in using lightpaths is the assignment of IP addresses to systems connected to the lightpath. If those systems also have to be able to communicate with other systems in the campus network, or with systems on the global internet, then the IP address range of use should be disjoint of the IP address ranges in the campus networks on either side of the lightpath, otherwise routing conflicts could occur. Although the assignment of PI addresses is an interesting problem that also needs to be solved, it will not be addressed in this research study.

As OpenFlow is still a new technology with a fast-paced release cycle, there exists little documentation on the hardware and its corresponding firmware. For this reason, experiments have to be performed to gain insight into the characteristics of OpenFlow hardware. The main focus of these experiments are, however, not on the performance of the hardware, as that has already been done by Appelman and de Boer[5], but rather on the functionality of the switches.

In this research paper only the current OpenFlow implementation on two

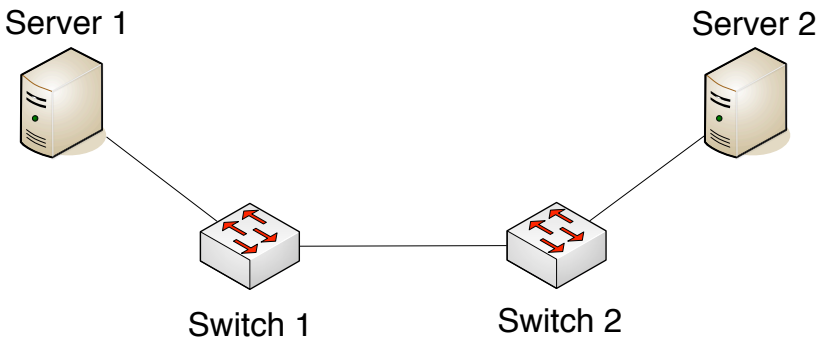

Figure 1: Overview of the testbed.

Pica8 P3290 switches with Open vSwitch firmware is investigated, so the results do net necessarily correspond to other brands of switches.

4 Approach

The research approach is both theoretical and practical. The key aspects that must be part of the solution are defined first. After a few common campus network architectures are chosen, a solution that meets the requirements is developed for each scenario. Subsequently, the support for the proposed solutions by the OpenFlow 1.2 standard is being checked. If the OpenFlow standard supports the solution, it is being investigated how the solution can be implemented within the testbed. Experiments are done to check if everything works as expected and to test if actions are performed in hardware or software.

4.1 Testbed

A testbed is built to be able to implement the solutions and to do some experiments. Figure 1 displays the testbed. It consists of two Linux based servers and two Pica8 P3290 OpenFlow switches. The Pica8 switches use a 1.6 version of the picos firmware (picos-1.6.1-3290-r9380) and run in Open vSwitch (OVS) mode. Open vSwitch version 1.9.90 is used and the flow table supports version 1.2 of the OpenFlow standard.

5 Overview of the proposed solution

As described before in section 1, the solution must meet a few requirements. It should be possible to implement the solution with as little changes to the current network as possible. After the implementation, the network administrator (or some other person) should only be involved in checking and approving the lightpath (virtual port) requests from end users. They should not be involved in the configuration of the individual dynamic lightpaths. The configuration of the dynamic lightpaths should be automated as much as possible and should be initiated by the end user.

To be able to use dynamic lightpaths within the SURFnet network, end users should first request a virtual port at the institutions where the devices that they want to connect with the lightpath are located. A virtual port is linked to a VLAN ID that is used to identify different lightpaths. After these requests are approved by the Bandwidth on Demand (BoD) administrator of each institution, the end users can set up their lightpaths dynamically through a web interface or API. A similar approach can be used to extend the lightpath through the campus network of a connected institution. The rest of this section will give a general overview of how this can be done with OpenFlow.

A few key aspects that should be part of the solution can be distinguished: Identification, authentication and authorization, a protocol to transport the packets through the campus network by some form of routing or switching and packet marking to be able to identify the packets of different lightpaths. In addition, there should be a way to let end users configure and schedule their lightpaths on their own without too much involvement of a network administrator or other person. Figure 2 gives a high level view on what changes are needed to the campus network to make it work.

One OpenFlow switch must be placed somewhere in front of the device that is connected to the optical network that will carry the lightpaths, which will be called the lightpath entrance. This OpenFlow switch must add the correct VLAN tag that corresponds to the virtual port that is associated with the lightpath of the end user. It must also support a protocol that is used to transport the packets through the campus network and some marking functionality to differentiate between the packets of the various lightpaths. One or more OpenFlow switches must also be placed in front of the computers of the end users that want to connect to their lightpath. These OpenFlow switches must support a way to identify, authenticate (optional) and authorize the traffic of an end user. Like the OpenFlow switch at the lightpath

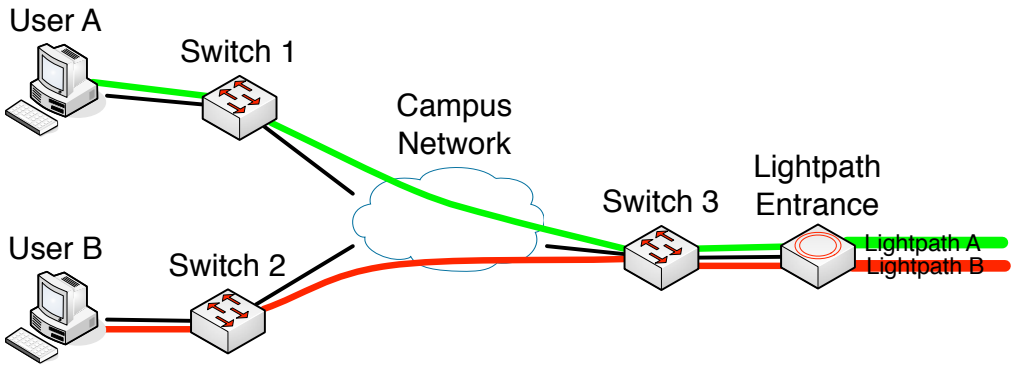

Figure 2: High level view on solution topology.

entrance, these switches must also support a protocol that is used to transport the packets through the campus network and some marking functionality to differentiate between the packets of the various lightpaths.

5.1 Identification, authentication and authorization

Because only the end user that requested the lightpath should be able to access it, some form of access control should be implemented. The simplest way to identify packets from an end user that must traverse a lightpath is based on the switch port they arrive on. When one trusts the physical security of the network, packets that enter a specific switch port on the OpenFlow switch at the end users side can be authorized by default to traverse the lightpath. No real authentication takes place in this case.

Another way to identify and authenticate packets is by checking their source MAC address or source IP address. However, because these addresses can be spoofed quite easily, relying on the source MAC or IP address alone is not a secure way to prevent unauthorized users to access the lightpaths. In addition, IP addresses of the private address space[6] could be used by the systems on both ends of the lightpath. These addresses may not be unique in the network and thereby not suitable as a source for authentication.

The IEEE 802.1X standard defines a port-based Network Access Control method based on EAP (Extensible Authentication Protocol). With IEEE 802.1X, the end user or connected device must authenticate first before access to the network is granted. After successful authentication, authorized packets are identified by the source MAC address. Although IEEE 802.1X

is vulnerable for MITM (Man In The Middle) and DoS (Denial of Service) attacks, it is more secure than the previous discussed methods. Besides being more secure, it is also a flexible method as end users are not bound to a specific switch port.

The IEEE 802.1X standard or a similar protocol designed specifically for the use with OpenFlow could be used as a relative secure method to prevent unauthorized access to the lightpaths while maintaining flexibility on the location the end user connects from. However, we are not aware of the existence of an IEEE 802.1X implementation (or a similar protocol) that can work well together with OpenFlow. For this reason we assume the simple switch port based scheme based on physical security without real authentication as the access control method.

The switch port the device is connected to and the virtual port at the lightpath entrance at the edge of the network are the two elements that should ultimately be connected. For proper security, it should be checked if the owner of the lightpath also 'owns' the switch port. If an end user could connect every switch port to it's lightpath, he could attack the attached device from the other end of the lightpath. This must of course be prevented. In this case, a manual assessment by the network administrator is necessary. When it is made certain that an end user 'owns' a switch port, he should be able to connect any of his lightpaths to this port without further intervention of the network administrator. The use of the IEEE 802.1X standard or a similar protocol avoids this manual check as the end user is not bound on a specific physical port but is authenticated based on one or more authentication factors such as a password or certificate.

5.2 Packet transport protocols

No campus network is the same. However, they all are primarily based on Ethernet switching and IP routing to transport packets to different places in the network. Another protocol that might be used in some places of the network is MPLS. When introducing OpenFlow switches in the network as described before, these OpenFlow switches must play along nicely with the current infrastructure. Dependent on the architecture, these switches and the OpenFlow controller should support a bunch of protocols to be able to communicate with the neighboring devices. One can think of ARP, the 802.1Q standard for VLAN tagging, routing protocols such as OSPF and LDP in case MPLS is used. Besides the support of these protocols, the OpenFlow switches and especially the OpenFlow controller must be manually

configured to integrate the OpenFlow devices in the network. Section 6 to Section 8 describe what protocols must be supported and what kind of initial configuration is needed to integrate the OpenFlow devices in different kind of network architectures.

5.3 Packet marking

As a packet arrives on one of the OpenFlow switches on an interface facing the campus network, the switch should somehow be able to know which lightpath it belongs to. As has been pointed out before, the OpenFlow switches know which lightpath belongs to which end user because there is a link made between the switch port of the end user and the virtual port of the lightpath at the lightpath entrance. However, this information is only locally available at either side of the network. There should be a way to contain this information in the packet to make it possible for the OpenFlow switches to identify the lightpath a packet belongs to. This is where packet marking comes into play and where the power and flexibility of OpenFlow should be visible. By adding headers or changing specific fields in the packet headers, the OpenFlow switches can communicate the identity of packets to each other. Care should be taken that only header fields that are not interpreted by the other devices in the campus network are used for this purpose.

Packet marking is not new. The best example of a situation where packet marking is used is QoS (Quality of Service). With QoS, the PCP (Priority Code Point) field in the VLAN header and the DS (Differentiated Services) field in the IP header are used to mark packets to be able to assign a different PHB (Per Hop Behavior). Because QoS might be implemented in the campus network, the PCP and DS fields would make a bad choice as an identifier for lightpaths.

Header fields that look more promising as lightpath identifiers are VLAN tags and MPLS labels because it is possible to attach more than one of these headers to a packet. In this way, the outer VLAN or MPLS header can be used to transport the packet through the network while the inner VLAN or MPLS header is used to identify the lightpath the packet belongs to. Without OpenFlow, the network administrator should configure the network devices manually to add and remove these headers with the assigned values. However, with OpenFlow the controller can assign a VLAN tag or MPLS label to a specific lightpath and configure the flows automatically without any intervention of the network administrator.

In section 6 to 8 will be explained what kind of headers can be used best in different scenarios.

6 Layer two VLAN network

In this section a practical implementation of the high level solution that was described in section 5 is described. First, a description of the type of campus network that the implementation is applicable for, in this case a layer two VLAN network, is discussed. Next, a more in depth description of the system is illustrated. After that it is pointed out to what extent the solution is supported by both the OpenFlow standard and by the Pica8 switch that was tested.

6.1 Network characteristics

The scenario that will be described in this section is that of a campus network in which a layer two connection from lightpath to end users could be possible. Such a network is subdivided in VLANs, with routers in the core taking care of traffic between hosts in separate VLANs. If two hosts are in the same VLAN both hosts can communicate directly with each other without any router swapping layer two headers.

An important characteristic of layer two switching is that Ethernet source and destination address normally remain unaltered, i.e. if there is no network address translation (NAT) in place, which we assume to be the case for the internal campus network. Another characteristic is that switches within the network do not inspect headers from higher layers than those of layer two. This means that it is possible to use any higher layer header of ones choice without affecting the switching through the network.

6.2 Proposed solution

There are three important problems that the solution has to solve. First, there is the problem of connectivity: how can the end user reach the lightpath entrance? The second problem is how to map specific users to their corresponding lightpaths. This problem is only applicable in case there are multiple users with distinct lightpaths. Finally, users should be prohibited from having access to each others systems and each others lightpaths.

As was already pointed out in section 5, at least two OpenFlow switches must be added in the campus network. The first switch must be placed near the lightpath entrance. This switch leads traffic from the campus to the appropriate lightpath, and subsequently traffic from a lightpath to its corresponding users. Another switch – or possibly multiple switches in case of users in different locations – is placed near the end users. Both switches must be connected to an OpenFlow controller which is responsible for installing all flows. This controller is connected to a web interface which functions as an interface to the controller and can be accessed by end users for the purpose of making lightpath reservations.

The connectivity problem is solved by configuring all OpenFlow switches to be on the same VLAN. This implies that they all are on the same broadcast domain. All packets intended for the other end of the lightpath and that traverse the OpenFlow switch near the end user are thus able to reach the OpenFlow switch near the lightpath entrance.

In order to correctly map users to their appropriate lightpaths and vice versa an MPLS header must be pushed on the packet by the OpenFlow switches. A second VLAN tag could also be used but it was clear from the start that multiple VLAN tags as in IEEE 802.1ad are not supported by the Pica8 switches. The concept stays the same though.

Two cases will be considered: traffic that comes from the end user and should be transported through the lightpath; and traffic that comes from the other end of the lightpath and should be forwarded to the end user. When a user sends a packet, its origin is recognized by the OpenFlow switch by means of the input port. It is assumed that only authorized users are connected to the configured ports. The switch then pushes an MPLS header in which the *label* field is set to a value that corresponds to a particular lightpath. After that, the packet is forwarded as normal. When the packet arrives at the OpenFlow switch near the lightpath, this switch checks the MPLS header. If it recognizes the MPLS label, it will pop the MPLS header and forward the packet to the appropriate lightpath by pushing its corresponding VLAN tag. If it does not recognize the MPLS label, it will drop the packet. The traffic from lightpath to end user is almost identical, except that the mapping between lightpath VLAN and MPLS label is made when packets enter the campus network. We will call this MPLS label tag the *lightpath identifier*.

The mapping of lightpath identifier to virtual port (i.e. VLAN of the multiservice port) is made through the web interface. There, each user must specify which virtual port – which maps to a lightpath given a reservation has been made – they want to connect to. The software, which is connected

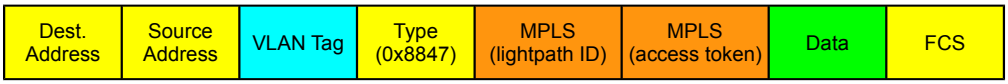

Figure 3: Layer two packet with additional MPLS tags.

to the OpenFlow controller, generates a lightpath identifier that is not yet in use. This mapping, that is stored in the controller, are used for the flows that are installed at the switches.

For the authorization problem another MPLS tag can be pushed. In this header the label field contains an access token which is set at all end points of the campus network for traffic entering the campus network from either lightpath or end user. The access token is a bit string set up by the controller and communicated over a secure channel (e.g. TLS) to the OpenFlow switches. The switches on the other side of the path check for this access token and drop packets containing no or incorrect values. In order to make this system somewhat more secure, the token must be refreshed every regular interval. It will however take more research in order to fully develop this mechanism, as there are still many considerations that have to be taken into account, e.g. synchronization between both switches or denial of service attacks by rewriting the access token. Figure 3 depicts what a packet would look like with the addition of the MPLS tags.

To reduce overhead, both the lightpath identifier and access token could be combined in only one tag. This way, there would only be one MPLS tag with one bit string which would be lightpath identifier and access token at the same time. Of course, this label should also be refreshed every interval.

The necessity of having a VLAN configured in between the OpenFlow switches makes this solution somewhat less dynamic, as this has to be done for each new user on a new location. However, one VLAN for all lightpaths would suffice as there is a mechanism in place to prevent users from accessing each others lightpaths. The main advantage of this solution is that the VLAN configuration has to be done only once for each user location, and that each subsequent lightpath reservation – possibly even reservations for other lightpaths – can be set up automatically.

6.3 Support by OpenFlow specification

Actions that the OpenFlow switches should perform in order to implement this solution are VLAN push, pop and modify. Furthermore, MPLS push, pop and modify are absolutely necessary for this solution to work.

All of these actions are listed in the OpenFlow v1.2 specifications[7]. Although all actions are specified as being optional, the implementation of VLAN push, pop and modify is suggested, *"to aid integration with existing networks"*[7]. For the MPLS push, pop and modify, this is not the case, as they are just optional.

6.4 Support by Pica8 switch

The testbed with the two Pica8 OpenFlow switches was used to test if the proposed solution also works in practice. Pushing and modifying VLAN tags worked flawlessly on the Pica8 switches. The experiments showed that VLAN tags were stripped automatically on access ports. This is maybe not expected behavior from a pure OpenFlow switch as this should be done by an action statement in a flow but in practice it gave no problems. Matching on VLAN tags also worked correctly.

The first experiments with adding, modifying, removing MPLS tags and matching on MPLS tags looked promising. The testbed was configured in such a way that MPLS packets arrived on the destination host which had a tcpdump session open. Because this server did not understand MPLS, the traffic was flowing one way. The result of this is that only ARP packets were sent in the direction of the destination host. Adding, modifying, removing and matching on MPLS labels worked fine with these ARP packets. However, other experiments with IPv4 and IPv6 traffic showed that only the first packet was labeled by the OpenFlow switch while subsequent packets were sent without an MPLS label. A few iperf tests showed no real bottleneck with adding, modifying and removing VLAN tags or MPLS labels.

It can be concluded that all actions were performed in hardware. The bottom line is that all needed MPLS related actions are supported by the Pica8 switches but because of a bug it did only work correctly for ARP traffic and for IPv4 and IPv6 traffic. It is thus not possible to successfully implement the proposed solution on the Pica8 P3290 switch.

7 MPLS enabled network

The second type of campus network that will be discussed is that of a campus network supporting MPLS. In an MPLS network, the forwarding of packets is performed by means of label switching. Label switching tables are constructed with the help of a routing protocol such as OSPF, which implies that every label switching router (LSR) must support such a protocol.

7.1 MPLS characteristics

A feature of MPLS is that it is allowed to push multiple MPLS headers on one packet, whereas the 802.1Q standard (VLAN) only allows one VLAN label, or two labels in case of Q-in-Q. Headers can be pushed, popped or swapped. For the label switching always the outermost header is used.

7.2 Proposed solution

Connecting both OpenFlow switches is not a straightforward process. Because we propose to add additional MPLS headers for the purpose of containing a lightpath identifier, and ingress MPLS routers normally do not accept packets that already have an MPLS tag, both OpenFlow switches must function as label edge routers (LER) for the path through the campus. In order to create label switched paths (LSPs) in between both OpenFlow switches, both switches must propagate a route through the network originating at themselves to which the LSP must lead. We propose to use the IP addresses that are configured on the campus facing interfaces of the OpenFlow switches, with a /32 prefix, for this purpose.

When the routes to both switches have been propagated, an LSP can be set up, for example by means of RSVP-TE[8]. All the switch now has to do to send the packet to the other switch is to push the correct MPLS label and transmit it on its appropriate interface. An advantage of using RSVP-TE is that besides configuring the route through the network also reservations regarding bandwidth can be made, i.e. if the network supports such quality of service.

As MPLS allows multiple MPLS headers stacked up on each other, the mapping of specific users to their corresponding lightpaths can be done by adding an additional MPLS tag containing a lightpath identifier on top of the header

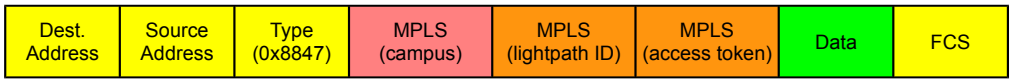

Figure 4: MPLS packet with additional MPLS tags.

that is used for switching through the network. The order of pushing the tags is important: first the tag containing the lightpath identifier must be pushed, after which the 'normal' MPLS tag is pushed. When a packet arrives at the OpenFlow switch near the lightpath entrance, the switch first pops the outermost switching MPLS tag and after that, it checks the MPLS tag with the lightpath identifier and forwards the packet to the corresponding lightpath or drops it in case of an unknown tag.

To solve the authorization problem, a second additional MPLS tag can be added to the packets. This header should contain the access token, as is described in both sections 5 and 6. The access token can also be combined with the lightpath identifier in only one additional MPLS header, as was the case in the solution for a layer two switched network. In figure 4 an example MPLS packet with additional MPLS tags for lightpath mapping and authorization is shown.

The necessity of supporting a routing protocol makes this solution much more complicated to implement, as the switch, and controller, must be able to handle a multitude of protocols, like ARP and the routing protocol in use. An advantage of this MPLS based solution is that RSVP-TE can be used to make bandwidth reservations, if the network would support that, leading to a better preservation of the lightpath characteristics.

7.3 Support by OpenFlow specification

Functionality that is necessary for implementation of this solution is MPLS pushing, popping and swapping. Also VLAN pushing, popping and modifying is needed for the OpenFlow switch near the lightpath entrance. As was already mentioned in section 6, the OpenFlow v1.2 specification supports these actions, although optionally.

The implementation of the routing protocol, as well as ARP, have to be fully implemented on the controller side, as the switches themselves are not capable of handling protocol requests and replies. It is feasible to let the controller handle these kind of protocols as they need not be processed on

line speed, and the loss in bandwidth and delay is thus not a big problem.

7.4 Support by Pica8 switch

As was already described in section 6, all needed MPLS related actions are supported by the Pica8 switches. However, because of a bug the MPLS label is not always correctly attached to packets. With ARP traffic it works fine but with IPv4 and IPv6 traffic only the first packet does get a label attached to it.

In this scenario it is also necessary to be able to attach more than one MPLS label. Pushing more than one MPLS label did not give any problems, except for the one already mentioned. There does not seem to be a clear limit on the amount of MPLS labels that can be pushed on a packet. Experiments show that it is possible to push more than 2000 MPLS labels on a single packet. However, the provided tools did have some trouble reading out the flow table at a certain point. This problem did occur with certain label values and amount of labels. However, adding an extra label solved the problem sometimes. A quick test with iperf showed that all actions were done in hardware, even when many labels were pushed.

8 Layer 3 routed network

The last scenario that will be discussed is that of a network in which a layer two connection between the two OpenFlow switches is not possible and also MPLS is not supported. In such networks packet forwarding is at least partly done by routers based on the information in their IP routing table. This has a lot of consequences for the kind of headers that can be used for lightpath ID marking and the solution in general, as will be explained in the rest of this section.

8.1 Network characteristics

When the hosts on both ends of the lightpath are separated by a router they are not located on the same subnet and are not able to communicate directly, unless some kind of tunneling is applied. A router should act as a default gateway for these hosts. Usually the routers are configured with a dynamic routing protocol such as OSPF to exchange routes with their neighbors. In

an OSI layer three network, all headers beneath the IP header are removed when a packet arrives on a router. This means that headers beneath the IP header cannot be used for marking with a lightpath identifier. The router attaches a new OSI layer two header when it has decided which interface to forward the packet through.

8.2 Proposed solution

The transport of the packets in this scenario could be done by normal IP routing. In this case, the OpenFlow switches should act as a router and thereby support protocols such as ARP and OSPF or another routing protocol that might be used in the campus network. This setup has however some disadvantages. At first, the software and the configuration of the OpenFlow controller gets much more complex than when normal layer two forwarding is used. The IP addresses for the lightpath end points should be carefully selected because these subnets should be routable and thus unique. All these elements require a good coordination between the end users and the network administrators of the networks on both ends of the lightpath. This makes automation of the creation of the lightpaths much harder. Secondly, it is hard to find a field in one of the headers that can be used to mark the packet. As has been told before, the DS field in the IP header is a bad choice because this might be used for QoS. Fields in headers below the IP header cannot be used because they are swapped on every router. This does not leave a lot of options open. As one might notice, there does not seem to be much benefit in using OpenFlow instead of normal routers.

A better solution for this scenario is the use of a GRE (Generic Routing Encapsulation) tunnel. With GRE, the original IP packet is encapsulated within another IP packet with a GRE header in between. A GRE tunnel could be configured between the two OpenFlow switches. According to the routers within the infrastructure, these OpenFlow switches are normal end points. The IP addresses of the end user devices at both ends of the lightpath are not visible for these routers. An optional extension to the GRE specification [9] allows the use of the Key and Sequence Number fields. According to RFC 2890 [9], the 4 byte Key field is *"intended to be used for identifying individual traffic flows within a tunnel."* This field is thus ideal for embedding a lightpath identifier. Figure 5 displays a general overview of this solution with GRE.

As within the other scenarios, the identification of the lightpath packets can be done based on the source port at the side of the device of the end

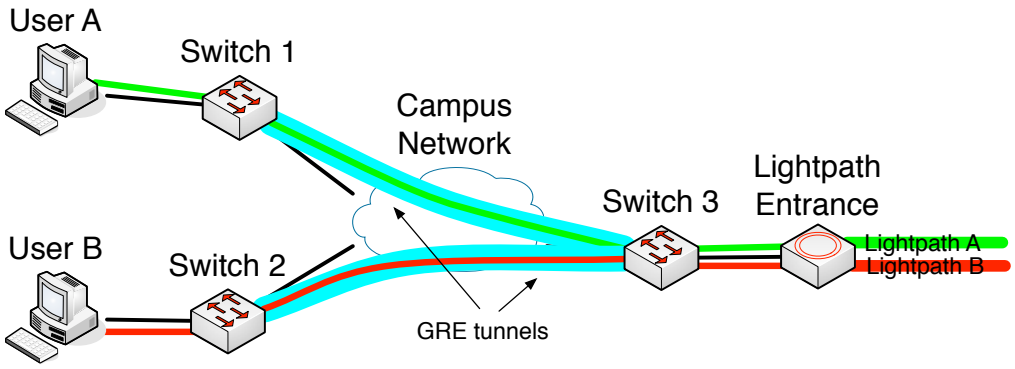

Figure 5: GRE solution overview.

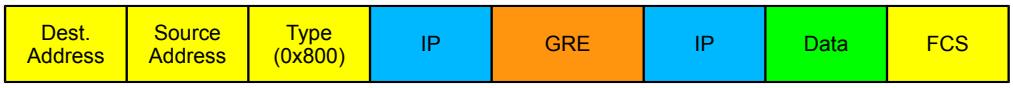

Figure 6: IP packet with additional GRE header.

user and based on a VLAN ID at the side of the lightpath entrance. The transport of the lightpath packets through the campus network is done by the intermediate routers based on the destination IP address. However, the destination IP address is in this case the IP address of the GRE tunnel endpoint at the other side and not the IP address of an end host. The Key field in the GRE header is used to embed the lightpath identifier. To automate the creation of the campus network extension of the lightpath, a unique value for the GRE Key field can be associated with a virtual port of an end user. The OpenFlow controller can automatically install the needed flows to add and match on the GRE Key field when an end user schedules his lightpath. Only the GRE tunnel itself must be manually configured on both OpenFlow switches, but this is a one time operation. Figure 6 shows an example IP packet with an additional GRE header.

8.3 Support by OpenFlow standard

Although GRE is not explicitly mentioned in the OpenFlow v1.2 standard[7], section 4.4 of it does describe 'logical ports', as being *"switch defined ports that don't correspond directly to a hardware interface of the switch"*. Logical ports are defined in the switch using other methods than OpenFlow. A use-case for a logical port is a tunnel end point, for example in the case of GRE.

OpenFlow can thus handle GRE tunnels in the form of logical ports but it cannot match on the GRE Key field, which is necessary for the solution discussed in this section.

8.4 Support by Pica8 switch

Although the OpenFlow standard does currently not fully support GRE, Open vSwitch does support the creation of GRE tunnels and is able to set and match the GRE Key field. The Pica8 switches used in the test environment do also support GRE according to the configuration guide[11] It is possible to create GRE tunnels and create flows to set and match the GRE Key field, although only the creation of the GRE tunnels is described in the configuration guide. However, in practice it did not work properly when multiple GRE Key values were used. Our experiments showed that all packets sent through the GRE tunnel were matched by the first flow that handled GRE packets, irrespective of the GRE Key values in the packets and the match statement of the flow. Due to time constraints we were not able to find the exact root cause of this behavior. Most likely the GRE Key value is not supported notwithstanding the fact that the flows that match on a GRE key were not rejected. An iperf test showed no bottleneck when using a GRE tunnel.

9 Conclusion

This paper described several scenarios and solutions to extend dynamic lightpaths with the help of OpenFlow. As a form of SDN, OpenFlow makes the network programmable, which is hard to do with conventional network hardware because they do not have an open API. With OpenFlow it should be possible for end users to schedule a lightpath from any location in the network. The OpenFlow controller can install flows in the OpenFlow switches that mark the packets of an end user automatically and send these packets through a lightpath while protecting the lightpath from unauthorized access.

Most of the features that are needed to make the proposed solutions work are supported by the OpenFlow version 1.2 standard. However, a lot of the features in the OpenFlow standard are marked as optional. Manufacturers are not obligated to implement these optional features to comply with the

[11] http://www.pica8.org/document/picos-1.6-ovs-configuration-guide.pdf

OpenFlow standard. Combined with the fact that there is little information available for the Open vSwitch mode of the Pica8 P3290 switches, this makes it hard to know which features are supported, how they work and how one can configure them. Having this insight is a prerequisite for a production environment.

The experiments that were performed in the testbed showed that not all supported OpenFlow features are implemented in the Pica8 P3290 switches. Besides this, some of the features that are implemented contain bugs that make it impossible to use them properly. From this we can conclude that, although OpenFlow seems a very promising standard, the implementation of OpenFlow in the Pica8 P3290 switches is not complete and mature enough to be able to use these switches to extend dynamic lightpaths through the campus network.

References

[1] *SURFlichtpaden*. URL: http://www.surfnet.nl/nl/Hybride_netwerk/SURFlichtpaden/Pages/lichtpaden.aspx.

[2] Nick McKeown et al. "OpenFlow: enabling innovation in campus networks". In: *SIGCOMM Comput. Commun. Rev.* 38.2 (Mar. 2008), pp. 69–74. ISSN: 0146-4833. DOI: 10.1145/1355734.1355746. URL: http://doi.acm.org/10.1145/1355734.1355746.

[3] *Pica8 Datasheet 48 x 1 Gbe p3290*. URL: http://www.pica8.com/documents/pica8-datasheet-48x1gbe-p3290-p3295.pdf.

[4] Rolf Biesbroek, Richa Malhotra, and Pieter-Tjerk de Boer. "Extending dynamic, on-demand lightpaths to the desktop – Solving the last-mile (extended abstract)". In: 2013.

[5] Michiel Appelman and Maikel de Boer. "Performance Analysis of OpenFlow Hardware". In: (2012).

[6] Y. Rekhter et al. *Address Allocation for Private Internets*. RFC 1918 (Best Current Practice). Updated by RFC 6761. Internet Engineering Task Force, Feb. 1996. URL: http://www.ietf.org/rfc/rfc1918.txt.

[7] *OpenFlow Switch Specification Version 1.2*. Dec. 2011.

[8] D. Awduche et al. *RSVP-TE: Extensions to RSVP for LSP Tunnels*. RFC 3209 (Proposed Standard). Updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711, 6780, 6790. Internet Engineering Task Force, Dec. 2001. URL: http://www.ietf.org/rfc/rfc3209.txt.

[9] G. Dommety. *Key and Sequence Number Extensions to GRE*. RFC 2890 (Proposed Standard). Internet Engineering Task Force, Sept. 2000. URL: http://www.ietf.org/rfc/rfc2890.txt.

A Test results

In this appendix, the results and findings of our practical experiments with the Pica P3290 OpenFlow switches are taken together in a table. This table is not a complete overview of every possible match or action statement.

A note about VLAN behavior on the Pica P3290 OpenFlow switches: The switches have port based VLAN's. Without the correct flows, no tag is added on trunk ports. This means that by default only local checking of the VLAN ID is done. When a non-tagged packet enters a trunk port, the default VLAN ID is used for that packet. When a VLAN tag matches the VLAN ID of the default VLAN of the outgoing trunk interface, the VLAN ID is removed.

Match / Action	**Support**	**Comments**
match in_port	ok	
match dl_vlan	ok	
match dl_vlan_pcp	ok	
match dl_src	ok	
match dl_dst	ok	
match dl_type	ok	
match nw_src	ok	
match nw_dst	ok	
match nw_proto	ok	
match tun_id	no	Experiments with multiple GRE tunnel ID's failed. Not sure if match or action failed.
action output	ok	
action drop	ok	
action mod_vlan_vid	ok	
action mod_vlan_pcp	ok	
action strip_vlan	yes/no	VLAN ID is stripped on access port. Works on trunk if VLAN ID matches default VLAN.
action push_vlan	no	mod_vlan_vid can be used.
action push_mpls	no	It works for ARP packets. Only first packet is labeled with IPv4 and IPv6 Pushing multiple labels works.
action mod_dl_src	ok	
action mod_dl_dst	ok	
action mod_nw_src	no	
action mod_nw_dst	no	
action mod_tp_src	no	
action mod_tp_dst	no	
action mod_nw_tos	no	
action set_tunnel	no	Experiments with multiple GRE tunnel ID's failed. Not sure if match or action failed.