System and Network Engineering — RP2

Master System and Network Engineering
Universiteit van Amsterdam

Multicast support on the AMS-IX infrastructure

Attilla de Groot and Yuri Schaeffer
attilla.degroot@os3.nl
yuri.schaeffer@os3.nl

July 5, 2009

Abstract

On the current *Amsterdam Internet Exchange* (AMS-IX) platform multicast support is provided in a seperate VLAN. This prevents multicast traffic to be forwarded to every connected member. This year the AMS-IX will upgrade their platform to a MPLS/VPLS setup. This report describes how multicast can be integrated in the main ISP VLAN in the new setup. We have examined the use of PIM-snooping with respect to scalability, feature implementation, resource usage, and abuse.

Our experiments show that PIM-snooping is scalable and they answered our questions on feature implementation. Our abuse experiments show that enabling PIM-snooping at this time will severely comprise the stability of the platform. Until this is solved we recommend to not enable PIM-snooping in the AMS-IX platform.

Supervisor(s): Ariën Vijn, Martin Pels

Contents

1 Introduction

The *Amsterdam Internet Exchange* (AMS-IX) is one of the largest Internet exchanges in the world. AMS-IX provides a layer2 network for ISP's so that they can send IP traffic at a lower cost then trough a transit provider. On the platform a separate *Virtual LAN* (VLAN) is created for the use of multicast traffic. Customers that are connected to this VLAN will receive every multicast stream, even the ones that they did not subscribe to. On the platform multicast traffic is handled similar to broadcast traffic, thereby flooding it out off every port of the VLAN. This VLAN prevents every other customer to receive unwanted multicast traffic on their connection to the AMS-IX.

On local broadcast networks a client can send *Internet Group Message Protocol* (IGMP) to join and leave a multicast group. In a routed network, *Protocol Independent Multicast* (PIM) and *Multiprotocol BGP* (MBGP) are used to exchange multicast routing information. For inter domain multicast routing *Multicast Source Discovery Protocol* (MSDP) is used to manage multicast source information. This results in routers connected to the AMS-IX exchanging PIM messages in order to build a multicast tree.

In the summer of 2009 the AMS-IX is planning to change their Ethernet platform to *Multi Protocol Label Switching* (MPLS) in combination with *Virtual Private LAN Service* (VPLS). This will still provide a transparent Ethernet platform for all customers. The VPLS platform can provide multiple instances. To match the current setup a separate multicast instance will be created. However, multicast on a separate instance is undesirable, because it is inefficient, and creates an extra administrative burden. In addition it does not stimulate the use of multicast. The most desirable situation would be to allow multicast traffic on the ISP VPLS instance. This leads to the following research question:

> *How can multicast support be provided on a* VPLS *platform, such as implemented on the* AMS-IX, *in an efficient way regarding scalability, performance, and stability?*

Providing multicast support can be done by snooping PIM traffic. This way, only customers who are interested in multicast traffic, as indicated by the PIM messages, would receive the streams. The AMS-IX platform provides a peak throughput of approximately 675Gbit/s. If new features are not properly tested they can have a significant impact on the uptime of such a high availability infrastructure.

2 Amsterdam Internet Exchange

The AMS-IX is an association with more than three hundred members. Members are organisations such as Internet service providers, carriers and content providers. The AMS-IX provides its members with a platform to exchange Internet traffic with any other member (peering).

Each member determines its own peering policy and can contact other members on individual basis to exchange traffic. Usually these peerings do not involve financial transactions. In this respect the AMS-IX enables members to reduce costs of Internet traffic.

2.1 Current platform

Currently the AMS-IX provides connections at six datacenters, each *Point of Presence* (POP) is located in Amsterdam. In one or more POPs members can connect to the AMS-IX access switches. Each POP has a connection to both cores which are located at different sites, see figure 1.

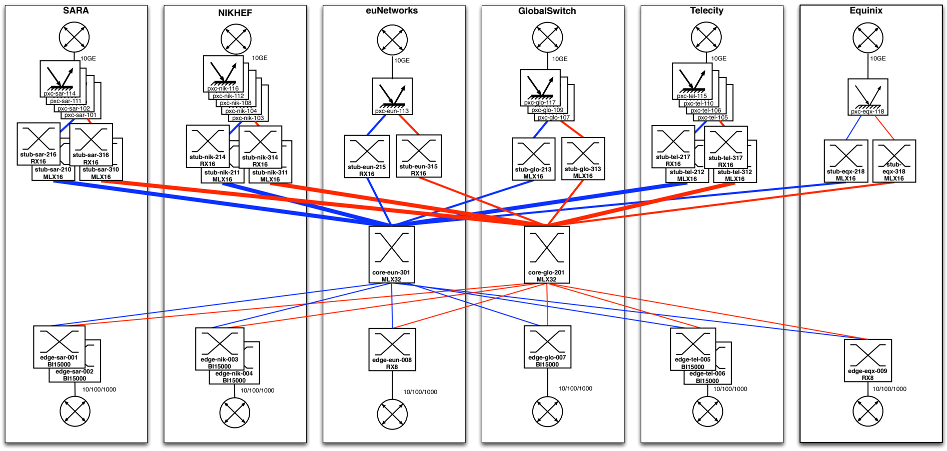

Figure 1: Schematic AMS-IX topology

Logically, the AMS-IX provides an ethernet platform for its members. Everyone can send frames to any other member (whom may or may not accept the packet). The AMS-IX does not participate explicitly in the routing of IP traffic.

In the current situation only one of the two core switches is used at a time. The state of both switches is constantly monitored, and the switch with the best connectivity is chosen as active. When a link fails on the active switch the other switch takes over. In this situation all traffic now goes through the new active switch, the old one goes completely idle.

This situation is not efficient nor scalable. Adding extra switches increases redundancy but reduces efficiency since it is not possible to balance the load over all switches.

2.2 MPLS/VPLS platform

Currently, to overcome limitations of the current platform, the AMS-IX is in transition to a new setup. As of August 2009 MPLS will be introduced to the platform. The transition will occur in a few steps to ensure failover in case of unexpected problems.

A frame enters a MPLS network through a *Label Edge Router* (LER). The LER performs a lookup in its IP table for a route to another LER and encapsulates the IP packet in an Ethernet frame and a MPLS header containing a specific label. Other devices receiving this frame can perform a lookup (in a much smaller table) and decide quickly where to switch the packet. The table has a mapping for incoming labels to outgoing, the switch pops the old label and pushes a new one for the next device. Because MPLS involves routing the platform is no longer is an ethernet domain.

VPLS can be used to define networks on top of MPLS. With VPLS the *original* ethernet frame is encapsulated within a new frame with a MPLS and VPLS header. This way a packet exits the last LER with its original frame, making the platform transparent for its members.

2.3 Multicast

Multicast is treated somewhat special on the AMS-IX. In the current situation the platform is one large broadcast domain, where the AMS-IX does no routing. When a multicast frame arrives, a pure layer2 switch has no choice but to broadcast the frame because it does not participate in the building of a multicast tree. Building such a tree requires knowledge of layer3.

This results in members receiving potentially large amounts of traffic they are not interested in. To dampen this undesired behaviour a separate VLAN is introduced for anyone interested in multicast. Multicast traffic received in this VLAN is thus broadcasted to anyone even remotely interested in multicast. This situation is far from optimal but with the current low multicast bandwidth usage it is acceptable.

However, this solution is not future proof. In case the usage of multicast increases to considerable bandwidth, this solution could potentially create a problem on the platform. It is also worth mentioning the extra administrative burden involved for members interested in multicast.

The AMS-IX and its members would benefit from a more scalable situation in respect to multicast. Because of the upcoming change on the platform our research will have to focus on multicast on VPLS.

3 Inter domain multicast routing

The AMS-IX platform is used to connect *Autonomous System* (AS). These autonomous systems send their IP traffic to other autonomous systems over the AMS-IX platform. The same holds for multicast traffic. Although multicast traffic is IP traffic, it is being replicated to all destinations on a broadcast network. Resulting in a one-to-many distribution from source to receivers.

As explained in the previous section, this result is not always an ideal situation. A multicast stream should only be received by devices who are interested in the stream. With multicast, several protocols are used to eventually provide the stream to an end host. In this section we explain the protocols used in the multicast process and how they relate to the AMS-IX platform.

3.1 IGMP

When a unicast IP packet is sent, the packet is routed over the Internet through one or more AS's until it arrives at its destination. A multicast stream is only sent if there is a client who is interested in the multicast stream (assuming *PIM Sparse Mode* (PIM-SM) is being used). For the sender to know that there are interested clients a *Reverse Path Tree* (RPT) has to be built from the destination to the source.

Building the RPT starts with the client sending IGMP [1] messages with the multicast groups it wants to join. If source specific multicast is used, the source of the multicast groups is included in the message. This message is sent to a link-local multicast address and received by an upstream router that will start sending the multicast stream or will send a PIM message to the next router towards the multicast source, as will be explained in the next section.

Although the AMS-IX platform is a large broadcast domain (in the current setup and a VPLS instance), IGMP packets will not be used on the platform since no multicast hosts, but only routers are connected to the AMS-IX. Multicast protocols used between Internet exchange customers are only PIM or inter domain routing protocols.

3.2 PIM

The PIM protocol is used to build a multicast distribution tree. There are several variations of the protocol, such as *PIM Dense Mode* (PIM-DM)[2], PIM-SM[3] and *PIM Source Specific Multicast* (PIM-SSM)[4].

In PIM dense mode a source builds a tree to every node in the network until a node sends a prune message. This protocol is effective when there are many subscribed nodes on the network, which makes the protocol inefficient to be used for inter domain multicast and will therefore not be discussed in this study.

PIM sparse mode and source specific mode are the opposite of dense mode. A source or any router in the tree will only send the multicast stream when there are members for that branche in the tree. PIM-SSM is a subset of the PIM-SM functionality. The difference between *Any Source Multicast* (ASM) provided with PIM-SM and *Source Specific Multicast* (SSM) is the lookup of the source. With ASM a client will send a join message only for an unspecified source with a multicastgroup (*,G), (source,multicastgroup tuple). The network then has to find the source trough a RPT. With PIM-SSM the node requesting the multicast stream specifies a source and multicastgroup for the stream (S,G), which is obtained through out-of-band communication such as HTTP.

Both PIM-SM and PIM-SSM are based on the IGMP and PIM protocols to build the multicast tree (multicast overview in figure 3). PIM-SSM is gaining popularity, because the implementation is simpler and implementing any source multicast does not have many advantages [5, *Since the bulk of the complexity is providing the least important functionality, the "ratio of annoyance" is disproportionately high in ASM*].

The router on the local subnet of the end node will receive an IGMP join message (either with or without source). Upon receiving, it will send a link-local PIM message towards the multicast source or forward the already receiving stream out of that interface 2.

The result of sending link local multicast PIM messages toward the source is that these messages are broadcasted over the AMS-IX platform to every connected customer. This provides the opportunity to sniff these packets on the AMS-IX platform and control the multicast streams that are requested.

3.2.1 PIM-snooping

PIM-snooping is a functionality which layer2 ethernet switches provide to control multicast traffic. When this function is enabled, multicast is not forwarded to every connected port. Forwarding to a port starts when a connected node sends a PIM-join message. This means that only routers that requested a multicast stream will receive it.

The AMS-IX platform, explained in the previous section, is one broadcast domain that would, without PIM-snooping, broadcast all the frames to every connected node. If a large volume of multicast is sent, this would overload ports that did not request the traffic.

As can be seen in figure 1, customers are connected to an access layer of switches at several POPs. In order to provide PIM-snooping for all the customers, the functionality should be enabled at every access switch in the used vpls instance.

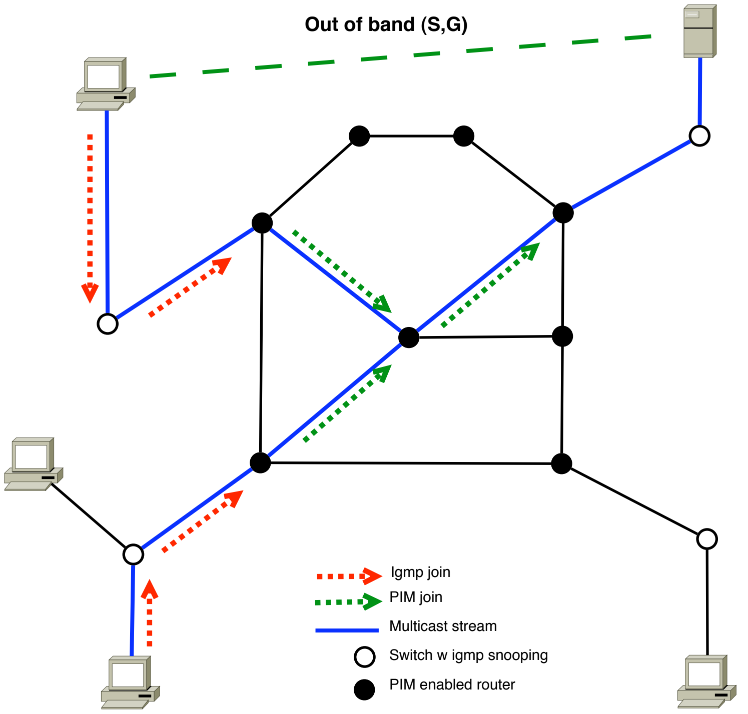

Figure 2: IGMP/PIM Multicast

3.2.2 PIM-proxy

In addition to the PIM-snooping feature, a PIM-proxy feature is available. Instead of copying PIM messages to the CPU and concurrently forwarding them to the destination, a PIM-proxy blocks, aggregates and transmits all pim messages towards the source at a specified interval. The result of a PIM-proxy is that routers connected to a switched infrastructure would receive less PIM messages and the information is aggregated with approximately 70 joins/prunes per message.

3.3 Multicast routing protocols

To send a PIM message towards the multicast source the route to the source has to be known, which does not different from normal unicast traffic. To build the RPT, the standard routing table can be used and the already available routing protocols to fill the routing table with the best or shortest routes, such as *Open Shortest Path First* (OSPF), *Routing Information Protocol version 2* (RIPv2), *Enhanced Interior Gateway Routing Protocol* (EIGRP), *Intermediate System to Intermediate System* (ISIS), or *Border Gateway Protocol* (BGP).

In case of multicast a separate table can be created to build the multicast RPT, by using routing protocol extensions such as *Distance Vector Multicast Routing Protocol* (DVMRP) [6] (no PIM-SSM support), *Multicast Open Shortest Path First* (MOSPF) [7], or MBGP [8]. The AMS-IX connects autonomous systems and BGP is the default inter-domain routing protocol. If a routing protocol is used to build a separate table for a RPT, BGP will be the protocol used over the AMS-IX.

3.3.1 MSDP

In the any source multicast shared trees are used and the root of the tree is called a *Rendezvous Point* (RP). The multicast sources will send their stream to the RP which causes the traffic to be distributed to the subscribed nodes. However the address of the RP is configured (either static or automatically) on the first router of the multicast source (*Designated Router* (DR)). As this is only the case in a local AS, a RP in another AS is not aware of the multicast source and cannot provide the stream to the requesting DR.

To provide inter domain multicast routing, RPs have to setup a MSDP [9] session in order to exchange their multicast sources. This allows the DR from one domain to forward join messages to the RP in the remote AS and discover the source for the multicast stream and start the normal PIM-join process.

An MSDP session is a TCP unicast session between two RP's in separate AS. For the AMS-IX platform this is just another TCP session and is not used to provide multicast support, since the normal PIM-join process still crosses the exchange.

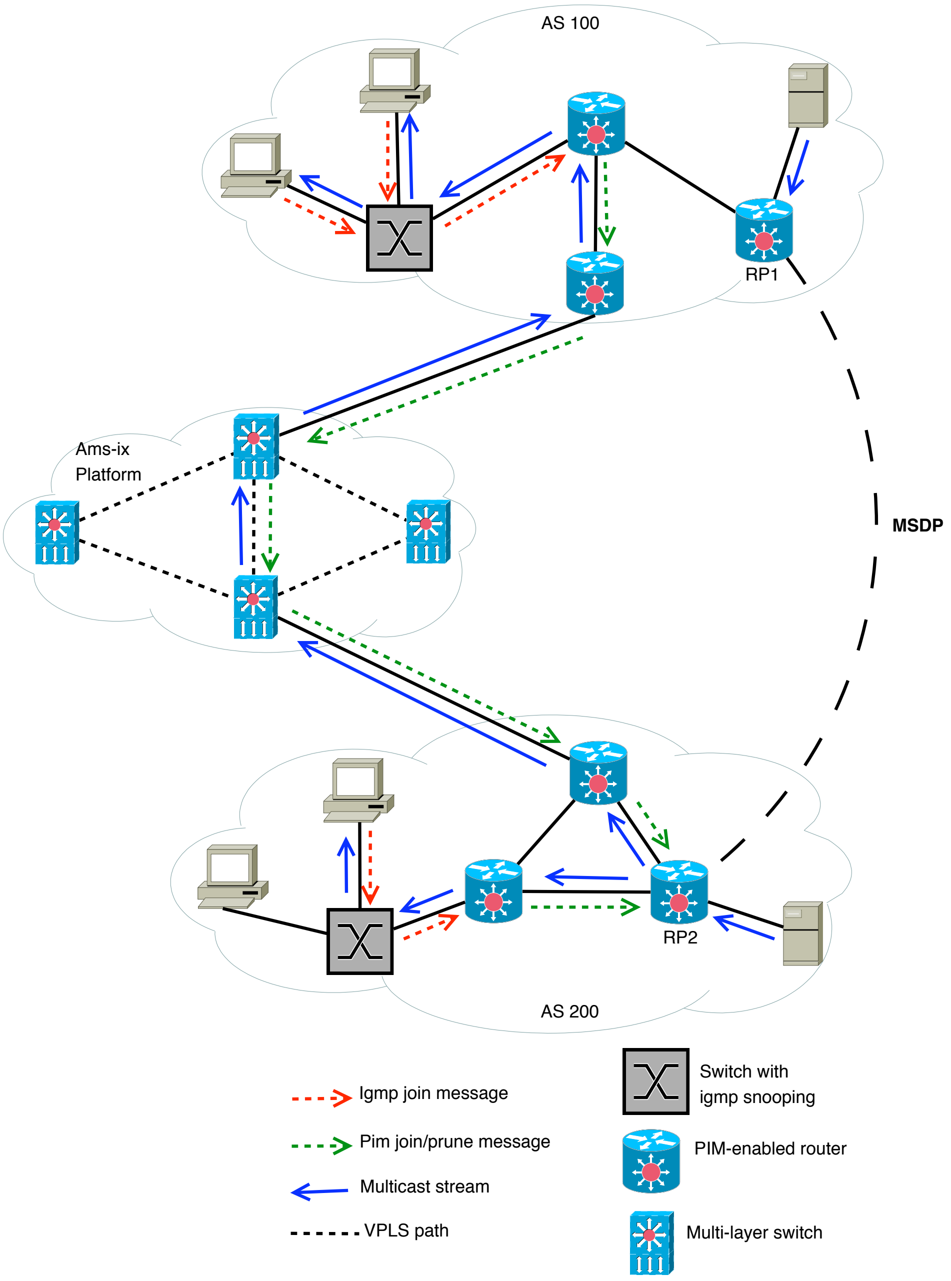

Figure 3: Inter-domain multicast routing

4 Research questions for multicast support

In the previous sections we explained the AMS-IX platform and how multicast can be supported. The AMS-IX as an Internet exchange connects different autonomous systems. As such we only have to deal with the PIM protocol as it is the de facto standard for intra-domain and inter-domain multicast [10].

Implementing PIM-snooping on the platform is a simple task of enabling the functionality on the appropriate switches and vlan or vpls instance. Enabling a new functionality in a platform as large as the AMS-IX should be researched to prevent implementation errors. To this end we enable the PIM-snooping feature in the AMS-IX lab environment and run several tests to answer specific research questions. In this section we will discuss the possible implementation problems and research questions.

4.1 Traffic replication

Multicast is efficient because traffic is only replicated where absolutely necessary, in contrast to unicast where the same stream for multiple recipients travel largely the same path. To optimise efficiency the replication should occur as deep in the tree as possible. The AMS-IX platform would benefit if multicast traffic is forwarded only to a *Provider Edge* (PE) that has one or more interested routers connected to it.

> *Where in a* VPLS *environment are multicast packets being replicated? What effect does this have on scalability?*
>
> *Does* PIM*-snooping use the* VPLS *path information to direct the multicast streams only to the source* PE*.*

4.2 Address overlapping

A multicast packet for group X on IP level will be send to MAC address X'. Packets with this address can be recognised as multicast traffic at ethernet level. Equation 1 shows that the 23 least significant bits of the group address are used to construct the MAC address, the 25 most significant bits are fixed on `01005E`), and the operators are bitwise. This has the implication that 2^5 multicast addresses (the first 4 bits of multicast IP addresses are constant) map on a single MAC address.

$$
\begin{aligned}
IPv4 : MAC &\leftarrow \texttt{01005E000000}_{16} \vee (\textit{group} \wedge \texttt{7FFFFF}_{16}) \quad (1)\\
IPv6 : MAC &\leftarrow \texttt{333300000000}_{16} \vee (\textit{group} \wedge \texttt{FFFFFFFF}_{16}) \quad (2)
\end{aligned}
$$

Although the *Internet Protocol version 6* (IPv6) addressing scheme is different from *Internet Protocol version 4* (IPv4), the mapping between group address and MAC address still exists. With IPv6 addresses the 32 least significant bits of the address are mapped to the 32 least significant bits of the MAC address with static `3333` most significant bits, see Equation 2.

The consequence of this overlapping is that PEs could forward multicast groups to routers that are not interested in these groups.

> *Which metric does* PIM*-snooping on the* PE*s use for switching the multicast stream? Does this cause problems?*

4.3 ASM and SSM

There are two different modes of multicast. ASM, where multiple hosts can transmit to a multicast group simultaneously and SSM where only one fixed speaker is possible.

With ASM it is very well possible that two different speakers transmit over two different PEs for the same multicast group. A router connected to another PE could be interested in just one of the two speakers.

> *What is the difference between* ASM *and* SSM *in respect to the* AMS-IX*?*

4.4 Resource usage

The PIM-snooping functionality requires a switch to copy PIM messages to the CPU. The CPU processes the data and writes the appropriate data to the cam [11]. It should be measured how much CPU time is needed to process the PIM data and if this interferes with other switch functionality. Is PIM-snooping handled as control traffic and thus competing resources with routing protocols that are used in the AMS-IX platform? In the implementation of pim-snooping can the number of multicast packets that are sent to the CPU be limited? How does this limitation affect on PIM-snooping?

> *Which problems can be expected when enabling* PIM*-snooping on the* AMS-IX *in terms of routing, load, performance and availability.*

4.5 Abuse traffic

It is interesting to see how the switches handle correct PIM and multicast traffic. Intentions are not always good and implementations rarely perfect. We like to examine how the switches behave in case of correct but unexpected traffic as well in case of incorrect traffic.

> *What is the performance impact on the switches when* PIM*-snooping is enabled in terms of latency?*

> *What will happen when a* PE *receives an excessive amount of join messages?*
>
> *How do the switches react to unexpected* PIM *messages?*

4.6 Other protocols

Next to standard unicast traffic, other protocols that utilise multicast can be used on a network such as IPv6 neighbour discovery, OSPF or *Cisco Discovery Protocol* (CDP). Since the implementation of PIM-snooping will block multicast traffic by default, we would like to investigate whether other multicast routing protocols are affected by PIM-snooping as well.

> *Does the* PIM*-snooping feature have an effect on other multicast protocols such as Internet Control Message Protocol version 6 (*ICMP*v6) neighbour discovery,* OSPF *or* CDP?

5 Experiments in AMS-IX environment

In the previous chapter we explained problems that can be expected when implementing multicast support in the AMS-IX vpls platform and based our research questions on these problems. To solve or explain the research questions, several tests have to be conducted.

AMS-IX provided us with a test setup (shown in figure 4) in their lab environment that is based on their new VPLS platform. This setup has four PEs and two core switches. For our tests a traffic generator is available, which can generate custom ethernet frames with custom IP packets and payload. We can use the generator to send a high volume multicast stream or some static PIM messages. To send custom PIM messages we wrote our own Python script [12] that runs on the Linux machine.

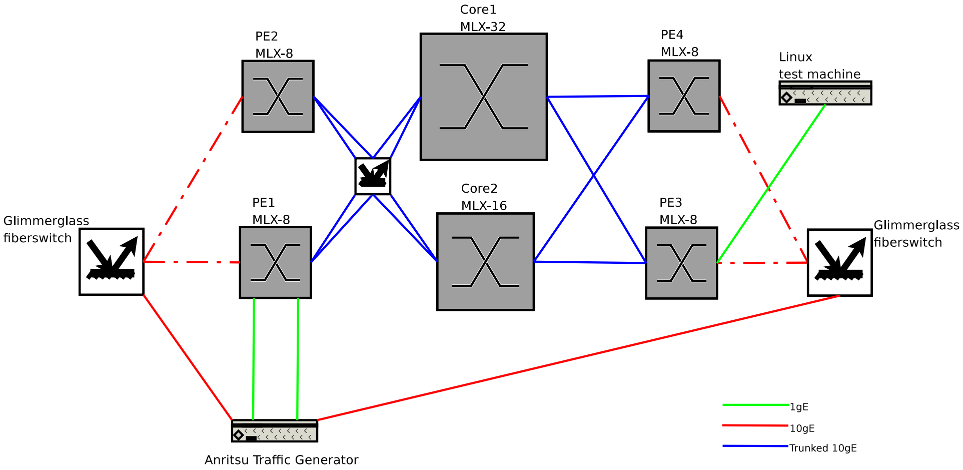

Figure 4: AMS-IX lab network

The Anritsu traffic generator is an appliance which is able to send traffic and receive/capture it. By using the Glimmerglass optical crossconnect, we can send traffic through the Anritsu and capture/analyse traffic on the links between the PEs and core switches.

5.1 PIM-join/prune message format

For most of our experiments we had to send custom PIM-join/prune messages. Therefore it is useful to know what such a packet looks like, figure 5 shows the format of a PIM-join/prune message.

In the header a PIM message has an upstream neighbour address, which is the host responsible for forwarding the message. The header also contains a field with the number of multicast groups listed in this single message. After the header this number of groups appear. A group has an

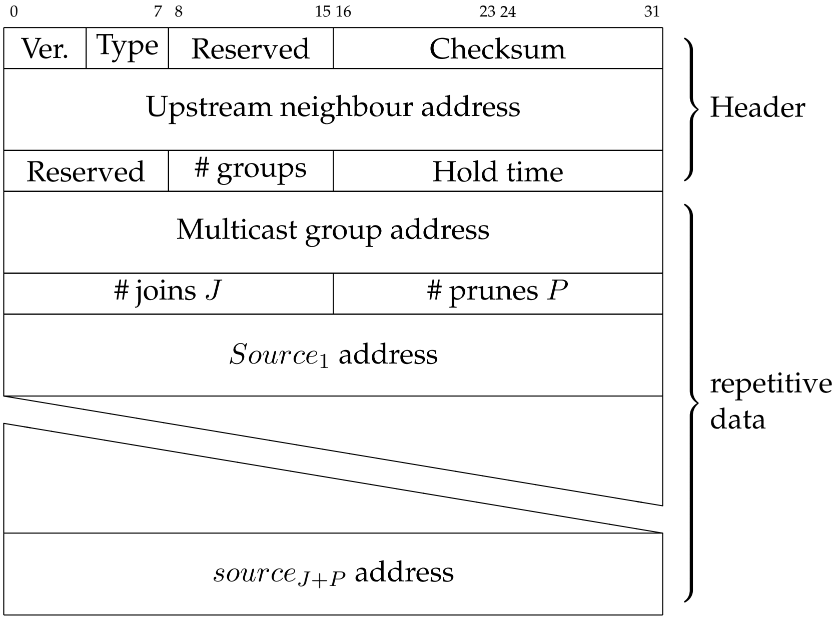

Figure 5: PIM-join/prune message format

address, a mask and some other fields depending on the address family. For each of those groups is stated how many joins and prunes for different sources will follow. The sources, formatted in a similar fashion as the groups, are sorted so that all the joins proceed the prunes. There is no explicit separator between join and prune records and they look the same. For detailed description of the message format we would like to refer to the appropriate *Request for Comment* (RFC) [3]

5.2 Traffic replication in a VPLS platform

As explained in section 2, a VPLS platform is built from a mesh of logical point-to-point links. The functionality is the same as a standard VLAN. We would like to know how a multicast stream is replicated by the switches. E.g. will PE_4 replicate the traffic to all other PEs despite the lack of interested receivers connected to that PE? Or when PE_4 has to send a multicast stream to both PE_1 and PE_2 will PE_4 send one stream and will this stream be replicated by the core?

5.2.1 Setup

In this experiment we send a multicast stream into the vpls instance through PE_2. With the traffic generator connected to PE_1 and the Linux machine we will send join messages for `(10.0.235.1,225.1.1.1)`.

This setup should result in PE_1 registering two interested ports and PE_4 registering one interested port for the multicast stream. The join messages are broadcasted on the VPLS instance and snooped by PE_2. This test should show how these join messages are registered and how the traffic is forwarded over the VPLS network.

To test our conclusion we captured traffic between PE_2 and $Core_1$. This should give us a view of the encapsulated multicast streams in mpls headers, and show us whether the core is participating in the multicast replication and where the traffic is replicated.

5.2.2 Observations

The setup indeed resulted in PE_4 registering one interested port for multicast group 225.1.1.1. It seems that the PE is already aware of the source PE as can be seen by the upstream PE address.

PE4 registered join messages

```
PE4(config-mpls-vpls-multicast)#show ip multicast vpls 601
L2mdb port type: R-router port, V2-igmp v2, V3-igmp v3, P_SG-pim sg, P_G-pim g
VPLS ID 601
IP multicast snooping is running - Passive
IP pimsm snooping is running
IP igmp operating version  - 2 (74s)
  Number of Multicast Groups: 1

  1    Group: 225.1.1.1
       Ports: 2/1 vlan 601 type  P_G (1s)
       1    Source: (10.0.235.1,  TNNL peer 192.168.99.2) FID 0x8009 mvid 1
            SG group ports: 2/1(2/1) vlan 601 type  P_G (0s)
```

The same result for PE_1, but for two interested ports.

PE1 registered join messages

```
PE1(config)#sh ip multicast vpls 601
L2mdb port type: R-router port, V2-igmp v2, V3-igmp v3, P_SG-pim sg, P_G-pim g
VPLS ID 601
IP multicast snooping is running - Passive
IP pimsm snooping is running
IP igmp operating version  - 2 (241s)
  Number of Multicast Groups: 1

  1    Group: 225.1.1.1
       Ports: 4/44 vlan 601 type  P_G (3s)
              4/43 vlan 601 type  P_G (24s)
       1    Source: (10.0.235.1,  TNNL peer 192.168.99.2) FID 0x8980 mvid 1
            SG group ports: 4/44(4/44) vlan 601 type  P_G (0s)
                            4/43(4/43) vlan 601 type  P_G (0s)
```

At switch PE_2 we see two interested ports, but instead of ethernet interfaces it registers the VPLS peers. It can be concluded that a multicast stream is then replicated to each downstream PE and not for every client, thereby building a tree over the VPLS tunnels.

PE2 registered join messages

```
PE4#show ip multicast vpls 601
L2mdb port type: R-router port, V2-igmp v2, V3-igmp v3, P_SG-pim sg, P_G-pim g
VPLS ID 601
IP multicast snooping is running - Passive
IP pimsm snooping is running
IP igmp operating version  - 2 (129s)
  Number of Multicast Groups: 1

  1    Group: 225.1.1.1
       Ports: TNNL peer 192.168.99.1 type  P_G (13s)
              TNNL peer 192.168.99.3 type  P_G (1s)
       1    Source: (10.0.235.1,  1/1) FID 0x800d mvid  1
            SG group ports: TNNL peer 192.168.99.1 VC Label 000f0000 R Label
                              0000040e Port 8/1 type  P_G (0s)
                            TNNL peer 192.168.99.3 VC Label 000f0000 R Label
                              0000040a Port 8/1 type  P_G (0s)
```

When we look at the captured data between PE_2 and $Core_1$ we indeed see two encapsulated multicast streams with different mpls id's, instead of three. We can conclude that in a VPLS network such as the AMS-IX a multicast stream has to be replicated for each PE by the source PE. We can also conclude that the core does not participate in replicating the multicast traffic, since the core sees the multicast traffic as unicast traffic because of the mpls encapsulation.

From a network design perspective this is the ideal situation, since a core switch should only be forwarding traffic and leave routing to the edges. For optimal link usage it would benefit if the cores participated in multicast replication so that a PE would only have to send one multicast stream over its up link. However compared to unicast, this benefit is insignificant.

Encapsulated multicast packet 1

```
Frame 1 (1360 bytes on wire, 1360 bytes captured)
Ethernet II, Src: FoundryN_9d:e4:50 (00:12:f2:9d:e4:50), Dst: FoundryN_ea:20:30 (00:12:f2:ea:20:30)
Multi Protocol Label Switching Header, Label: 1040, Exp: 0, S: 0, TTL: 255
    MPLS Label: 1040
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 255
Multi Protocol Label Switching Header, Label: 983040, Exp: 0, S: 1, TTL: 255
    MPLS Label: 983040
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 255
Ethernet II, Src: 3com_01:00:00 (00:01:02:01:00:00), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01)
Internet Protocol, Src: 10.0.235.1 (10.0.235.1), Dst: 225.1.1.1 (225.1.1.1)
Data (1300 bytes)
```

Encapsulated multicast packet 2

```
Frame 2 (1360 bytes on wire, 1360 bytes captured)
Ethernet II, Src: FoundryN_9d:e4:50 (00:12:f2:9d:e4:50), Dst: FoundryN_ea:20:30 (00:12:f2:ea:20:30)
Multi Protocol Label Switching Header, Label: 1039, Exp: 0, S: 0, TTL: 255
    MPLS Label: 1039
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 255
Multi Protocol Label Switching Header, Label: 983040, Exp: 0, S: 1, TTL: 255
    MPLS Label: 983040
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 255
Ethernet II, Src: 3com_01:00:00 (00:01:02:01:00:00), Dst: IPv4mcast_01:01:01 (01:00:5e:01:01:01)
Internet Protocol, Src: 10.0.235.1 (10.0.235.1), Dst: 225.1.1.1 (225.1.1.1)
Data (1300 bytes)
```

5.3 Layer2 multicast address overlapping

We tested how the switches handle multiple unrelated multicast streams which are send toward a different group address but map onto the same MAC-address. E.g. groups 231.130.4.53 and 227.2.4.53 both map to MAC-address 01:00:5E:02:04:35. The streams will have this address as destination on layer2. A pure layer2 switch cannot tell both streams apart.

5.3.1 Setup

Through PE_2 we send two multicast streams: M_1 for group address 225.1.1.1 and M_2 for group address 225.129.1.1. The traffic generator is used to monitor between this PE and the core. Every 30 seconds we send PIM join messages for M_1 from the testing machine. From the traffic generator to PE_1 we send joins for M_1 and M_2 from its first interface and for M_2 for its second interface.

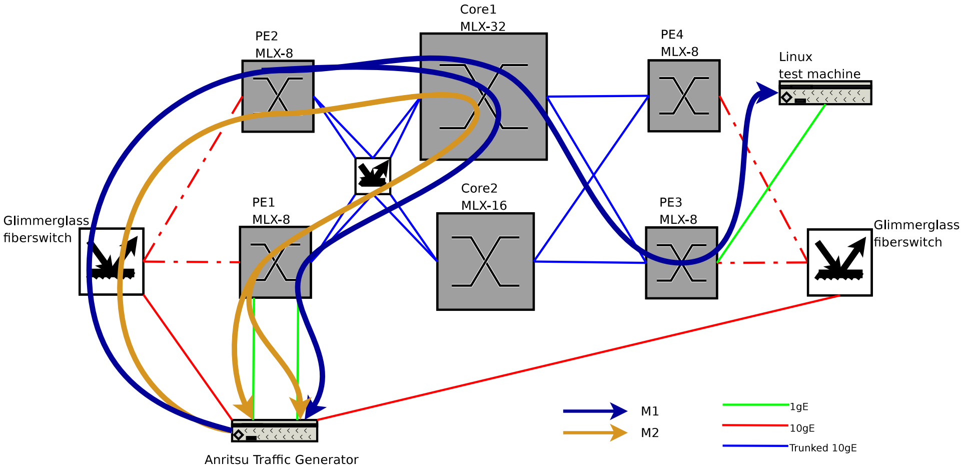

Figure 6: Multicast stream data flow

5.3.2 Observations

Figure 6 shows where we observe multicast traffic in the test setup. Between PE_2 and $core_1$ we see three multicast streams: twice M_1 and M_2 once. In total three streams arrive at the traffic generator. This means PE_1 is replicating traffic for M_2, which is correct behaviour. This also means in order to process multicast traffic, a switch also uses information from layer3.

5.4 Any source versus source specific multicast

Multicast knows two distribution modes ASM and SSM. In SSM an interested host explicitly asks for a number of specific sources in a group. This means a host must know exactly who is sending to a group in order to decide whom to listen to. An exterior mechanism must be in place to discover new sources.

In the case of ASM a host simple requests all traffic send to a particular multicast group. Discovering sources is a function of the network. This shifts administration from the receiver and possibly some central point to the network.

5.4.1 PIM

In terms of regular multicast packets there is no difference for an Internet exchange as these packets all have a normal source and destination. A PIM message however can have (S,G) tuples for SSM and (*,G) tuples for ASM.

In case of SSM a snooping switch should keep track of all joins sources per multicast group. This requires potentially much more memory in the switch. Below we see entries for both SSM and ASM when only one join/prune message is sent, i.e. no actual multicast stream goes over the network.

Multicast group entry SSM

```
1    Group: 225.1.195.67
       1    Source: (10.0.194.67,  TNNL peer 0.0.0.0) FID 0xffff mvid none
            SG group ports:
       2    Source: (10.0.194.68,  TNNL peer 0.0.0.0) FID 0xffff mvid none
            SG group ports:
```

Multicast group entry ASM

```
1    Group: 225.1.237.155
      Ports: TNNL peer 192.168.99.3 type  P_G (0s)
```

In case of ASM/SSM we can conclude that for the AMS-IX the only difference are the extra entries used for the sources of a multicast stream.

5.5 Platform performance with PIM-snooping

When enabling PIM-snooping, each PIM message is copied to the CPU and processed. In a platform such as the AMS-IX a large quantity of PIM messages are expected. Therefor we have to measure what the impact of enabling the feature is on routing, memory and cpu.

5.6 Memory usage

When PIM messages are snooped, entries are stored in RAM. By offering a large number of groups we found that the switch can hold up to 70285 unique ASM groups. SSM obviously needs more memory to store the groups as well as the sources. A switch can hold 11788 groups at a time. It is interesting to see that when increasing the number of sources per group does not change this maximum.

When a multicast packet arrives at an interface, the hardware will lookup its destinations in the *Content Addressable Memory* (CAM). When not available, the Traffic Manager (processor to handle inter blade communication) looks for an entry in the RAM and will update the CAM accordingly. If the entry is not available at all the multicast packet is dropped. At the time of testing the switches had 4096 CAM entries available for PIM-snooping. This partitioning can be changed in the switch's configuration.

During our research two AMS-IX employees tested the new platform in the USA. When asked about these limitations a Foundry engineer revealed that currently only 2000 cam entries are available for multicast . This means that only this amount of concurrent multicast streams can be supported, independent of the amount the software can handle.

5.6.1 CPU usage

Due to the maximum limit of entries in the cam we were not able to perform a full cpu test. We only examned the cpu usage when adding and renewing a maximum of 2000 multicast groups. We found that each (S,G) tuple is one entry in the cam table. Thus for ASM every source takes up one entry.

Our test shows that when adding 2000 groups, either ASM or SSM, the cpu usage stays at 0%. We did encounter flapping *Label Switched Path* (LSP) links during the tests. It seems that multicast has an effect on the routing in the network. However, at this time we are not able to confirm this.

Vpls link flap

```
Jun 24 15:25:14 Curr Task: mpls
Jun 24 15:25:14 call stk: 0x0821903c 0x08b985dc 0x08b980d8 0x08c8e2b8 0x08c8dd40\
    0x08c5c3fc 0x08c5c17c 0x08cbdc00 0x08c26fdc
Jun 24 15:25:14 *LDP dump: Send to 192.168.99.4 tcp len 18 (TCP handle a000000a:216ffca8)
0001000e c0a86301 00000201 00040000 693f
Jun 24 15:25:14 *LDP dump: Rcvd from 192.168.99.2 tcp len 18 (TCP handle a0000007:216ff9d8)
0001000e c0a86302 00000201 00040000 694d
Jun 24 15:25:15 Warn: MPLS PROBLEM or EXCEPTION trap, sys logged
Jun 24 15:25:15 **** PROBLEM    0x2102 - 167  (   1) **** at 0: 2:16
The first ERO subobject in a received PATH message does not correspond
to an address or interface on this LSR.
Session tunnel ID         = 0X0009
Session extended ID       = 0XC0A86304
Session destination addr  = 0XC0A86302
```

5.7 Effect of misbehaving traffic

Knowing how much load the platform can handle and what effects are to be expected is very important when introducing a new technique. Unfortunately the world is rough and unpolished and we cannot afford to ignore unexpected traffic. Either a failing machine, implementation errors or bad intentions should be handled correctly.

5.7.1 Mask length

In a PIM message IP addresses (for groups, sources, etcetera) are coupled with a *Variable Length Subnet Mask* (VLSM). For IPv4 this should always be /32 and for IPv6 /128.

We are interested in how the switch will handle packets with other subnet masks, especially those lower than default values. In other situations the subnet mask represents a range of addresses. For this purpose we slightly crippled the PIM join/prune messages.

We tested for IPv4 with some lower and higher values than 32 for the group address and source addresses. We have found that the entries are gracefully accepted by the switch. The debug output does not show that the mask field is faulty in our messages and seems to ignore the field completely.

A wrong value in this field has no effect what so ever (as long as the checksum of the packet is correct) and we see no room for abuse. This behaviour is slightly out of line with the RFC [3, *"A router SHOULD ignore any message received with any other mask length"*].

5.7.2 Data at end of message

An other scenario is a PIM message with some extra random data at the end of it. The message we crafted is valid in a sense that the checksum is correct for the whole message and all data claimed in the header are available. After this data we insert some extra nonsense, e.g. 8 random chosen bytes.

The switch appears to interpret the message correctly. There is no sign that it tries to read the appended bytes nor does it discard the message as invalid. The RFC makes no statements over desired behaviour in cases like this.

5.7.3 Joins for unicast addresses

When snooping is disabled, a switch should handle multicast the same as broadcast traffic and forward the traffic to all other interfaces. With snooping enabled the switch must use layer3 information to forward traffic.

How will the switch behave in an ambiguous situation? For this experiment we forge a join message containing one group and one source to join. The group address should be class D, since it makes no sense for a router or snooping switch to accept otherwise. Our message however contains a unicast address for the group address and is sent towards one of our switches.

When looking at the multicast state of the switch as well as the debugging output we see that the message gets handled as if nothing is wrong.

The faulty group address is accepted and the switches state updated accordingly.

Unicast group address

```
1    Group: 145.100.104.19
     1    Source: (10.0.235.1,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
```

At this time it is unknown how the switch will handle an IP packet where the destination IP address is mapped to a multicast MAC address. Depending on the implementation the switch may or may not forward the packets to interested listeners.

Apart from having yet another way to fill the switch's memory with nonsense we consider the risk for this problem to be low. It is probably hard to convince all routers in between to actually map a multicast mac address instead of doing a regular arp lookup. If it does work it is unlikely that other traffic on the platform is affected.

Additionally, a class D address as a source address (which should always be a unicast address) is also accepted by our switch.

5.7.4 Number of joins per group

In the PIM message header there are fields to enumerate the amount of groups and the amount of sources per group for that message, see section 5.1.

We first try sending a message without any groups at all. Next we send one group without any sources. The debugging output show that both of these messages are handled correctly by the switch. The packets are being processed normally, with the exception that there is just nothing to do for the message.

It gets more interesting when our message contains one group with one source to join but the header claims it contains more (see figure 7). The below output shows the result on the switch when sending one message claiming to carry six sources for the group but is actually only holding source *10.0.235.1*. We observe that the additional sources do not relate to any information in our message and appear to be random. It is likely that the snooping code does not check boundaries and tries to read the entries consecutive after the message from memory.

Bogus sources SSM

```
1    Group: 225.1.1.1
     Ports: TNNL peer 192.168.99.3 type  P_G (2s)
     1    Source: (127.0.0.5,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
     2    Source: (221.180.0.7,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
     3    Source: (221.180.32.0,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
     4    Source: (10.0.235.1,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
     5    Source: (0.4.0.0,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
     6    Source: (221.217.0.0,  TNNL peer 0.0.0.0) FID 0xffff mvid none
          SG group ports:
```

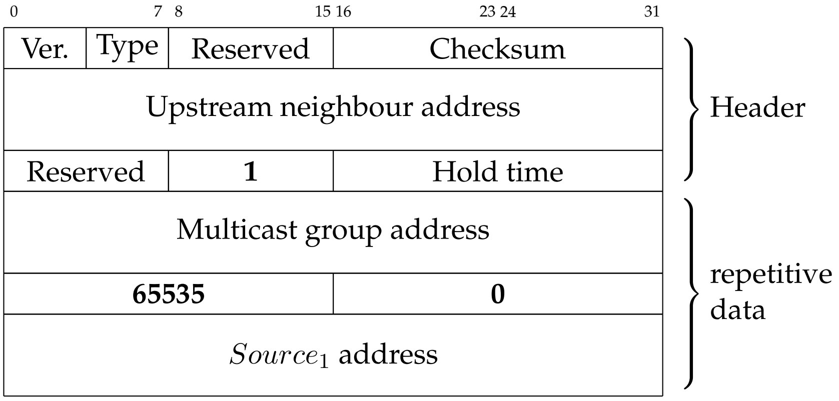

Figure 7: Forged PIM-join/prune message

When we alter the message in such a way that it claims to have 65535 joins for the group. The connected switch and all other PEs in the vpls instance will turn unresponsive for a few seconds and then reliably crash and reboot without any warning output on the console.

MLX8 crash

```
Exception Type 0000 (Soft Check - Timeout), mcast
0000f030(msr)
00000000(dar)
00000000(dsisr)
084427c8(pc)
08442858(lr)
08442858
08442ac4
08442e48
08443460
08443830
0827c214
0873bc80
0873c038
08731ed0
08649810
08649a18
08966ecc
000043f0
End of Trace

NetIron XMR/MLX Boot Code Version 3.5.00
...
Enter 'a' to stop at memory test
Enter 'b' to stop at boot monitor
```

One possible reason for this crash is that the switch runs out of memory when it tries to store this many joined sources. However when we do the same experiment with prunes instead of joins the switch reacts exactly the same. In theory, a processed prune should not take any memory but only potentially free some.

A likely explanation is that during the processing of the message the code tries to read a piece of memory which simply does not exist or which it is not allowed to read.

5.8 Multicast routing protocols

If PIM-snooping is enabled multicast traffic is blocked until a connected router sends a join message for a specific group. However there are other protocols that also use multicast, but do not use PIM to start sending multicast packets, for example OSPF or IPv6 neighbour discovery.

When doing our experiments we noticed that packets with a destination for the well-known multicast addresses (224.0.1.0/24) are not blocked. This means that protocols such as OSPF that use these addresses are not affected by the PIM-snooping feature.

During our experiments other IPv6 and arp experiments were done in the same lab environment which show that enabling PIM-snooping does not cause any problems for the IPv6 neighbour discovery. We also tested the effect on the proprietary CDP. This protocol sends its messages to the destination mac address `01:00:0c:cc:cc:cc`, but we did not notice any effect on these messages.

It should be noted that to our knowledge there is no documentation available on blocked traffic. And that we are not able to test all proprietary protocols out there. However this does not cause problems for the AMS-IX platform, because only specific protocols are allowed on the network.

6 Conclusion

Reviewing our research question from Section 1.

> *How can multicast support be provided on a* VPLS *platform, such as is implemented on the* AMS-IX, *in an efficient way regarding scalability, performance, and stability?*

Multicast support in the AMS-IX VPLS platform can be provided by using the PIM-snooping functionality.

In respect to scalability we can conclude that the snooping functionality uses the vpls path information to deliver multicast streams instead of broadcasting the streams to every PE in the network. This results in a situation where only PEs with interested routers connected will receive the multicast stream. In order to create a more efficient delivery of multicast traffic the core switches should also participate in the vpls instance. Which would create a shorter distribution tree. However to let the core switches participate the architecture of the platform has to be changed. In our opinion the benefits of the core participating in the multicast delivery do not warrant the change in architecture.

The limit of 2000 group entries in hardware resource usage does not form any problems with respect to performance. When conducting our performance experiment resource usage was too low to measure. This means that for performance we can conclude that enabling pim-snooping with a limit of 2000 group entries does not affect performance.

Stability of the platform will be degraded when the PIM-snooping feature will be enabled at this time. Our experiments show that a flood of pim messages or a malformed message can trigger memory exhaustion. This causes the switch to turn unresponsive and reboot. The discovered bugs have already been sent to Brocade and are currently being researched. A workaround (limiting the number of pim messages being accepted) is scheduled for the next release.

6.1 Recommendations

When PIM-snooping is not enabled in a VPLS setup, multicast traffic will not be load balanced over aggregated links. However, the amount of multicast traffic at this time does not warrant the implementation of a feature that compromises the stability of the platform. Thus, we recommend that until Brocade has fixed the bug that causes the memory exhaustion or implements a work-around, PIM-snooping is not enabled and multicast traffic is placed in a seperate VLAN.

In addition to the PIM-snooping feature, PIM-proxy can be enabled. Enabling this feature prevents every connected router from receiving different

PIM-messages. In doing so, original messages are blocked, aggregated and new pim messages are sent at a specified interval. This results in the AMS-IX infrastructure becoming part of the multicast tree and might place an extra administrative burden on the AMS-IX. Also in our opinion, an Internet exchange should not interfere with traffic from its customers.

Currently, multicast is hardly used on the AMS-IX platform. For now the maximum of 2000 multicast groups will suffice. However, nobody can predict the development of services used on the Internet. Supporting better multicast on the AMS-IX platform might even stimulate multicast usage. For more scalability in the platform, Brocade should implement a solution to support more multicast groups.

6.2 Further research

It should not be a major problem that our tests focus more on IPv4 than IPv6, since they only differ in addressing. From a multicast perspective it should not cause any problems. However, our tests show that there is a difference between using VPLS/MPLS or just plain Ethernet. Testing with IPv6 could be useful further research.

Brocade will look into the limit of 2000 multicast groups by mapping multiple software entries to one hardware entry. However 2000 groups still remains an unrealistic number, compared to the 2^{28} available multicast groups. Will the solution from Brocade scale for Internet exchanges?

Acknowledgements

We would like to thank the following people for their help while conducting this research project.

- Ariën Vijn, Martin Pels, Elisa Jasinka, and Niels Bakker, for their guidance throughout the research project.
- AMS-IX, for giving us the opportunity to do this research.

References

[1] H. Holbrook *et al.*, Using internet group management protocol version 3 (igmpv3) and multicast listener discovery protocol version 2 (mldv2) for source-specific multicast, 2007, `http://tools.ietf.org/html/rfc4604`.

[2] A. Adams *et al.*, Protocol independent multicast - dense mode, 2005, `http://tools.ietf.org/html/rfc3973`.

[3] B. Fenner *et al.*, Protocol independent multicast - sparse mode (pim-sm), 2006, `http://tools.ietf.org/html/rfc4601`.

[4] H. Holbrook and B. Cain, Source-specific multicast for ip, 2006, `http://tools.ietf.org/html/rfc4607`.

[5] B. M. Edwards and B. Wright, *Interdomain Multicast Routing: Practical Juniper Networks and Cisco Systems Solutions* (Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002), Foreword By-Stewart, John W.

[6] D. Waitzman *et al.*, Distance vector multicast routing protocol, 1988, `http://tools.ietf.org/html/rfc1075`.

[7] J. Moy, Multicast extensions to ospf, 1994, `http://tools.ietf.org/html/rfc1584`.

[8] Y. Rekhter *et al.*, Multiprotocol extensions for bgp-4, 2007, `http://tools.ietf.org/html/rfc4760`.

[9] D. Meyer *et al.*, Multicast source discovery protocol (msdp), 2003, `http://tools.ietf.org/html/rfc3618`.

[10] A. Farrel, *The Internet and Its Protocols: A Comparative Approach (The Morgan Kaufmann Series in Networking)* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004).

[11] Content-addressable memory, 00:34, 20 May 2009, `http://en.wikipedia.org/w/index.php?title=Content-addressable_memory&oldid=291070135`.

[12] Y. Schaeffer and A. de Groot, Protocol independent multicast injection script, 2009, `http://www.attilla.nl/os3/pim_inject.zip`.

Acronyms

AMS-IX *Amsterdam Internet Exchange*

AS *Autonomous System*

ASM *Any Source Multicast*

BGP *Border Gateway Protocol*

CAM *Content Addressable Memory*

CDP *Cisco Discovery Protocol*

DR *Designated Router*

DVMRP *Distance Vector Multicast Routing Protocol*

EIGRP *Enhanced Interior Gateway Routing Protocol*

ICMP *Internet Control Message Protocol*

ICMPv6 *Internet Control Message Protocol version 6*

IGMP *Internet Group Message Protocol*

IPv4 *Internet Protocol version 4*

IPv6 *Internet Protocol version 6*

ISIS *Intermediate System to Intermediate System*

LER *Label Edge Router*

LSP *Label Switched Path*

MBGP *Multiprotocol BGP*

MOSPF *Multicast Open Shortest Path First*

MPLS *Multi Protocol Label Switching*

MSDP *Multicast Source Discovery Protocol*

OSPF *Open Shortest Path First*

PE *Provider Edge*

PIM *Protocol Independent Multicast*

PIM-DM *PIM Dense Mode*

PIM-SM *PIM Sparse Mode*

PIM-SSM *PIM Source Specific Multicast*

POP *Point of Presence*

RFC *Request for Comment*

RIPv2 *Routing Information Protocol version 2*

RP *Rendezvous Point*

RPT *Reverse Path Tree*

SSM *Source Specific Multicast*

VLAN *Virtual LAN*

VLSM *Variable Length Subnet Mask*

VPLS *Virtual Private LAN Service*