Monitoring Next-Generation Networks:
A Flow-Based Approach

Master of Science Thesis

by

Richard Johannes Hofstede

Date: July 20, 2011
Committee: dr. ir. Aiko Pras
Idilio Drago, M.Sc.
dr. Anna Sperotto
dr. Ramin Sadre
Institution: University of Twente, Enschede, The Netherlands
Faculty: Electrical Engineering, Mathematics and Computer Science (EWI)
Chair: Design and Analysis of Communication Systems (DACS)
# Contents

1 Introduction 3  
  1.1 Assignment 4  
  1.2 Research Topics 6  
  1.3 Thesis Organization 7  

A AIMS 2011 Paper 10  
B EUNICE 2011 Paper 23  
C JNSM Paper 36  
D TERENA TNC 2011 Poster 50  
E Next-Generation Ethernet Monitoring Press Release 52
Chapter 1

Introduction

This thesis describes the process and bundles the results of the author’s Master assignment work. In the summer of 2010, the EWI examination committee of the University of Twente approved a special study program, proposed by the author, for obtaining a M.Sc. degree. This program consisted of a scientific element and a more industry-oriented element, allowing the author to find out the preferred direction for a career after obtaining the M.Sc. degree. It was arranged that the author would become a member of the DACS group, in order to work together with Ph.D. candidates and scientific staff.

Many results that were obtained in the period from September 2010 to July 2011 were published at conferences, by means of papers and a poster. Other results were presented at research meetings of the Dutch academic backbone network provider SURFnet [1] (i.e. results of the industry-oriented part of this assignment), or at colloquia at the DACS group. Due to the special nature of this Master assignment, the structure of this thesis is not common. As such, this thesis provides an overview of the author’s Master assignment by motivating the scientific publications and shortly discussing the material which was not published (yet). The general conclusions of this assignment consist of the individual conclusions of the attached papers.

The remainder of this chapter will describe the Master assignment, outline the research topics, followed by an overview of the organization of this thesis.
1.1 Assignment

The general goal of a Master assignment is to demonstrate the merit of a M.Sc. qualification. Although a special study program for the 3rd and 4th semester has been approved by the EWI examination committee, a Master assignment always needs to adhere to a number of criteria for graduation assignments, as defined in the *Studiegids Master CS* of the EWI faculty at the University of Twente ([2]). In order to meet those key criteria, it was arranged that several scientific papers would have to be written, instead of a more traditional M.Sc. thesis. At the moment of writing this thesis, two papers have already been approved for publication in conference proceedings and one of the papers has been presented. In the following, the quality of this Master assignment will be demonstrated by describing the activities undertaken to fulfill each of the key criteria in [2].

- **Clearly formulate a problem statement.** This criterion has been satisfied by defining research questions, research approaches and contributions of the papers to be written during this Master assignment. The two directions in the research projects (*i.e.* more scientific and industry-oriented) and the available projects/funding at the DACS group, resulted in three paper topics, for which all the mentioned paper attributes needed to be defined.

- **Identify relevant literature.** After formulating a problem statement by means of research questions, an in-depth literature survey needs to be done. It turned out that problem statements needed to be modified, after the relevant literature was studied. All papers included in this work contain a dedicated *Related Work* section, outlining the state-of-the-art work in the same field.

- **Draw up a work plan.** As soon as the contents of the papers were defined, relevant conferences with matching key topics were chosen. Since every conference has registration/submission deadlines, these deadlines automatically became the internal deadlines for the work related to the papers.

- **Adjust problem statement and work schedule in accordance with interim evaluations.** Regular (interim) evaluations were present during this graduation assignment. Based on those evaluations, problem statements, approaches and work plans were adjusted, in order to improve the work quality or to be more efficient with respect to
the available time. Besides the evaluations by the graduation committee, also conference paper/poster reviewers provided feedback on the delivered work. This feedback helped to improve the work and gave a good indication of whether the problem statements and proposed contributions were of interest for the research community.

- **Analyze different possible solutions and motivate a choice between them.** Each of the papers in this thesis contains a dedicated section with related works, against which the papers need to be judged for relevance. This implies that the author should be able to motivate why the related works do not solve the problem outlined in the paper. A good example of analyzing different possible solutions is demonstrated in the paper in Appendix B. Several related technologies are outlined to find out whether, or until which extent, a certain problem can be solved.

- **Communicate the research and design activities both written and in presentations.** To satisfy this requirement, the research outcomes were documented in scientific papers. These papers were submitted to, and presented at several conferences and workshops in our field. In addition, several smaller presentations were done at the DACS group for colleagues, in order to inform them of the progress or to make them acquainted with a new research area or topic.

- **Show the ability of reflection on the problem, on the research/design approach, on the solution and on ones own performance.** Based on new findings and comments, problem statements, approaches and possible solutions need to be adjusted. Reflection of the author is required for realizing the need for adjustments. With respect to ones own performance, feedback from others is required. The various conference presentations were always reflected afterwards, in order to improve presentation skills.

- **Demonstrate creativity and the ability to work independently.** During the various meetings with Ph.D. candidates (who shared their office with the author) and supervisors, new ideas and insights were contributed. Research directions were sometimes adjusted, in order to incorporate new ideas from the author. Higher-quality work and synergy were the result of the various cooperations. However, the author also needed to work independently during the research process, especially during the actual research and paper writing process.
1.2 Research Topics

This thesis covers research topics in the field of network management, in particular fault and performance management, combined with high-speed traffic monitoring. The following list outlines those topics, together with a short description of their contents and interrelation.

- **Next-Generation Ethernet.** Ethernet has been the major link-layer technology in local area networks (LANs) for many years. Several updates to the original standard were standardized, which make Ethernet suitable for use in transport networks. In general, when Ethernet is used in such environments, it is referred to as Carrier Ethernet or Metropolitan Ethernet. SURFnet, the Dutch academic backbone network operator is currently investigating the use of Carrier Ethernet technology in their Next-Generation Network [3]. The contribution of this work in the context of this research program is:
  1. Survey on the state-of-the-art Ethernet OAM management standards.
  2. Evaluation of the use of IPFIX for Next-Generation Ethernet monitoring.

- **IP Flow Information Export (IPFIX).** IPFIX is a standardized flow monitoring technology [4][5], based on Cisco’s NetFlow v9 [6]. The major difference between the two monitoring technologies is the flow key definition flexibility offered by IPFIX. As a result of that flexibility, IPFIX can be used for monitoring Ethernet networks, for instance (instead of only IP networks). Due to the possible deployment of Next-Generation Ethernet by SURFnet, IPFIX could be a suitable flow monitoring solution. Several use cases for such a deployment will be identified in this thesis.

- **Probe configuration & deployment.** A more practical research topic during the Master assignment was the configuration and deployment of a (flow monitoring) probe. The UT acquired specialized equipment from the Czech university spin-off company INVEA-TECH [7], offering an early-deployment solution for IPFIX monitoring at the Ethernet-layer (for use in Carrier Ethernet networks). During this Master assignment, practical experience with the configuration of such equipment has been gained, resulting in a thorough understanding of the related technologies.
1.3 Thesis Organization

The remainder to this thesis is organized as follows:

- **Appendix A**: Paper published in the proceedings of the *5th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2011)* [8].

- **Appendix B**: Paper published in the proceedings of the *17th EUNICE Open European Summer School (EUNICE 2011)* [9].

- **Appendix C**: Draft paper to be submitted to the *Journal of Network and Systems Management (JNSM)*.

- **Appendix D**: Poster presented at the *TERENA TNC 2011* conference.

- **Appendix E**: Press release on Next-Generation Ethernet monitoring, which was published together with INVEA-TECH.
Acknowledgements

This work concludes all the work performed between September 2010 and July 2011. It was a special year, due to the special contents of this Master assignment and the opportunity to share the office with Ph.D. candidates. Of course, this assignment could not have been completed without the help of others. At first, I would like to thank the members of my graduation committee: Anna, Idilio, Ramin and Aiko. The various meetings and brainstorming we had, together with the time you shared with me doing research, were of a great value to me. Second, I would like to thank my office mates - Anja, Idilio, Rafael and Giovane - for the place at your office, the nice talks and fun we had during this year. At the third place, I would like to thank all the other colleagues at the DACS group for their openness and for giving me the feeling to be welcome in the team.

Besides the thanks to my colleagues, I would like to show my special thanks to my girlfriend, Kimberley, for her understanding, joy and endless support during my whole study. She provided me the right support at the right moment, and came up with great ideas at the moments in which I needed a short break during my work. Also my family, and especially my parents Peter and Maartje, and my sister Kim, deserve my sincere thanks for their interest, support and encouragement during the years of my Bachelor and Master studies.

Rick Hofstede
Neede, July 2011
Bibliography


Appendix A

AIMS 2011 Paper

This appendix includes the final version of the conference paper *Carrier Ethernet OAM: An Overview and Comparison to IP OAM*. The paper relates, as the title already implies, to the following topics:

- **Carrier Ethernet**, an acronym for the use of Ethernet in a service-provider or backbone environment.

- **Ethernet OAM**, a pair of standards for adding fault and performance management facilities to Ethernet.

- **Comparison of Ethernet OAM and IP OAM**, in order to verify whether functionality is duplicated at both the IP- and Ethernet-layer, or not.

| Name: | 5th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2011) |
| Venue: | INRIA, Nancy, France |
| Date: | June 14 - 17, 2011 |
| Presentation date: | June 16, 2011 |

Table A.1: Conference information
Carrier Ethernet OAM: An Overview and Comparison to IP OAM

Rick Hofstede, Idilio Drago, Giovane C.M. Moura, Aiko Pras
University of Twente, The Netherlands
r.j.hofstede@student.utwente.nl, {i.drago,g.c.m.moura,a.pras}@utwente.nl

Abstract. Ethernet has evolved from a local area to a wide area network technology. When it is used in a service provider environment, it has more complex requirements, which demand a set of management techniques for the Ethernet layer. Ethernet OAM comprises a set of management techniques for Carrier (or Metropolitan) Ethernet networks. Carrier Ethernet devices often have IP connectivity for management purposes, which might be used for IP OAM as an alternative management solution to Ethernet OAM. This paper provides an overview of Carrier Ethernet technology and evaluates whether, and until which extent, IP-based protocols can replace Ethernet OAM in Carrier Ethernet networks.

Keywords: Network Management, Carrier Ethernet, Metropolitan Ethernet, Provider Backbone Bridging, Ethernet OAM, IP OAM

1 Introduction

Ethernet has been the number one link-layer technology in local area networks (LANs) for a long time [1]. To make Ethernet suitable to be used in larger networks, such as metropolitan and wide area networks (MANs and WANs respectively), it was extended to provide high availability, quality of service, secure communication, and superior scalability [1]. Carrier Ethernet1 allows service providers to offer connectivity at the Ethernet level, in contrast to the IP level. Some backbone network operators are already considering a deployment of Ethernet in their Next-Generation Network [2].

With the deployment of Carrier Ethernet services it is required to manage Ethernet in an end-to-end manner, besides managing it on the link-layer. To achieve this, IEEE and ITU-T have standardized two Operations, Administration & Maintenance (OAM) protocol suites by means of IEEE 802.1ag [3] and ITU-T T.1731 [4], respectively. Ethernet OAM offers end-to-end fault and performance management over Ethernet Virtual Connections (EVCs), which are logical connections between various customer sites [5].

From a functional point of view, the Ethernet OAM standards seem to be very similar to IP OAM protocols, such as Ping and Traceroute. Assuming that

1When Ethernet technology is used in large-scale (e.g. service provider) networks, it is commonly referred to as ‘Carrier Ethernet’ or ‘Metropolitan Ethernet’.
Most Carrier Ethernet devices support IP OAM, one could wonder whether OAM functionality is now duplicated at both the link and network layer. Moreover, if this is the case, could IP OAM functionality be used as a replacement for Ethernet OAM? Since IP OAM has been used already for a long time, operators have more experience with it than with Ethernet OAM. This lead us to the following research questions:

1. What exactly is Carrier Ethernet and which functionality does it provide?
2. How does Ethernet OAM functionality compare to IP OAM, and, more specifically, can IP-based protocols in Carrier Ethernet networks provide the same functionality as comparable Ethernet OAM management techniques?

Since many papers have already been published on the topic of Ethernet OAM, we will first review those that focus on using alternative solutions for Ethernet OAM in Section 2. Section 3 describes the evolution of Ethernet towards a WAN protocol, especially for deployment in service provider environments. The Ethernet OAM standards for managing Carrier Ethernet networks are discussed in Section 4. Although several works state that IP OAM is not (entirely) suited to manage a Carrier Ethernet [1] [5], a clear explanation or motivation is not provided. Section 5 will fill this gap by defining an Ethernet OAM deployment scenario, by means of which an IP-based approach will be analyzed. Finally, we close this paper in Section 6, where we draw our conclusions and future work.

2 Related Work

Many studies have already been performed on the usage of Ethernet OAM for WANs. McFarland et al. [1] state that enterprise networks typically have straightforward topologies and that IP-based protocols such as SNMP, ICMP, Ping and Traceroute will suffice for management. However, it will not be suitable for managing service provider networks, carrying thousands of services for different customers. Motivations for the unsuitability of IP are not given.

Indukuri goes beyond IP-based protocols for managing Ethernet networks [6], by outlining the use of IP Ping, MPLS LSP Ping, Bidirectional Forwarding Detection (BFD) and especially Ethernet OAM for Virtual Circuit Connectivity Verification (VCCV). It is desirable for metropolitan and especially critical networks to have a fast and accurate fault detection mechanism. Such a sub-50 ms detection and restoration facility is provided by BFD and Ethernet OAM. The author concludes, however, that the choice for a VCCV mechanism should not only depend on technical decisions, but also on the underlying transport infrastructure. In the case that a virtual circuit is constructed on top of an Ethernet network (e.g. an Ethernet Virtual Connection (EVC)) and end-to-end management should be performed, it is wise to use Ethernet OAM in order to avoid the need for translation layers between different network layers. However, this work does not outline why IP-based management techniques do not suffice for the management of Carrier Ethernet networks.
3 Carrier Ethernet Evolution

During its evolution from a LAN technology to a MAN and WAN technology, Ethernet was extended to support customer traffic separation, quality of service (QoS) and, most importantly, a greater number of MAC addresses (of customers, among others) in the forwarding tables of switches [7]. The frames of the various Ethernet standards are depicted in Figure 1. ‘Ethertype’ and ‘Frame Check Sequence’ fields are left out for the sake of space. The evolution of Ethernet has been standardized by the IEEE in several standards, starting with IEEE 802.1Q [8]. This standard adds a VLAN tag to an Ethernet frame, right after the source and destination MAC addresses, by means of which the forwarding plane can be partitioned into logical segments.

In the same year as IEEE 802.1Q was standardized, an amendment was defined in IEEE 802.1ad [9], also known as Provider Bridging (PB). We assume this standard and all following ones to be Carrier Ethernet standards. The IEEE 802.1Q VLAN tag contains VLAN IDs of 12 bits, supporting up to 4094 VLANs. Although this number of VLANs will be enough for most LANs, it will not suffice for large service provider environments. To overcome this scalability problem [7], IEEE 802.1ad defines VLAN tag stacking, allowing service providers to insert an additional VLAN tag of 12 bits in an already tagged frame. This ‘S-VID’ VLAN tag is only used inside the service provider domain and is inserted in front of the initial VLAN tag, which is now referred to ‘Customer VID’ (C-VID).

IEEE 802.1ah [10], also known as Provider Backbone Bridging (PBB), allows a strict separation between customer and service provider domains by encapsulating customer traffic. This is achieved by inserting a new Ethernet header in front of the existing one, including a new backbone VLAN tag (B-VID) and a new ‘Service Instance ID’ (I-SID) field. The latter can be considered an extended VLAN ID, used to identify customer instances inside the operator network. By considering the entire PB frame as payload and inserting a new Ethernet header in front of it, a completely isolated address space is used inside the Ethernet backbone network. The result is a drastically reduced complexity/size of the forwarding tables in backbone nodes, since only backbone node addresses and backbone VIDs are needed for switching.

**Fig. 1.** Evolution of Ethernet frames
The most recent Carrier Ethernet standard is IEEE 802.1Qay [11]. It uses the same frames as IEEE 801.ah (PBB), but it adds traffic engineering capabilities and related rapid protection against failures [12]. IEEE 802.1Qay is therefore referred to as Provider Backbone Bridging with Traffic Engineering (PBB-TE). It adds support for static, traffic-engineered paths by replacing the use of the spanning tree protocol (STP) by an external method. Besides disabling STP, also broadcasting and MAC address learning are disabled [13]. Broadcast traffic and traffic for unknown destinations are discarded by the edge nodes of the network.

In the next section we present Ethernet OAM, by means of which fault and performance management have been added to Ethernet.

4 Ethernet OAM

End-to-end OAM has been added to Ethernet by means of IEEE 802.1ag & ITU-T Y.1731. While these standards had different focal areas when work on them started, IEEE 802.1ag is nowadays considered a subset of ITU-T Y.1731. Both standards cover fault management, while performance management is solely covered by ITU-T Y.1731. Fault management can be used for detecting and isolating faults in a network, just as notifying about faults. Performance management allows to measure throughput, delay, etc. This will help to verify and prove service performance against a Service-Level Agreement (SLA) [14], for instance.

Before the Ethernet OAM management techniques can be discussed, some terminology needs to be described. Maintenance End-Points (MEPs) are actively managed components, which are positioned at Maintenance Domain (MD) boundaries. Interconnected MEPs are called a Maintenance Entity (ME). A Maintenance Entity Group (MEG)\(^2\) can include several MEs, depending on the topology: for point-to-point Ethernet connections, a MEG contains a single ME. In a multipoint setup, a MEG consists of several MEs. Inside a ME, and thus between MEPs, one or more Maintenance Intermediate-Points (MIPs) can be placed. MIPs only react to OAM flows, while MEPs initiate and terminate them. In order to have management hierarchies, OAM levels can be defined to run OAM mechanisms completely separated. These concepts will be highlighted again in Section 5.1, where a deployment scenario for Ethernet OAM will be discussed.

The remainder of this section will focus on the most commonly known/used Ethernet OAM management techniques.

4.1 Continuity Check

Continuity Check (CC) can be used to detect interruptions in connectivity (and thus continuity) between end points (MEPs) in an Ethernet network. This is accomplished by transmitting ‘heart-beat’ messages between MEPs, which are forwarded by MIPs. By doing so in a periodic manner, connectivity can be verified. MEPs exchange CC messages with the other MEP inside the same ME, and at each administrative OAM level.

\(^2\)This terminology is based on ITU-T Y.1731. In IEEE 802.1ag, a MEG is called a Maintenance Association (MA).
4.2 Loopback

Loopback (LB) provides a way to transmit request/response messages, in order to verify bi-directional connectivity with another MEP or MIP. Upon reception of a LB message, a response message is returned towards the requester. In contrast to Continuity Check, which sends messages in a periodic fashion, LB messages are typically initiated by operator command, although nodes can be configured to transmit LB messages in a periodic fashion as well.

4.3 Link Trace

Link Trace (LT) can be used to isolate faults in Ethernet networks. MEPs send out LT messages on a particular ME, in order to identify the connectivity and relationships with remote MEPs and MIPs. While a LT can only be initiated by a MEP, all MIPs and MEPs downstream the path towards a destination MEP at the same OAM level will respond to it.

4.4 Alarm Indication Signal

Alarm Indication Signal (AIS) provides a method for notifying operators about a network anomaly. As soon as a MIP detects a failure at its OAM level, it will send out an AIS message towards the reachable peer MEPs of the same ME. After the MEPs receive the AIS from the MIPs, they will send out a multicast AIS message in the upstream direction of a fault, at the next most superior OAM level and on every service provider VLAN affected by the failure. AIS is not supported by IEEE 802.1ag.

4.5 Loss Measurement

Loss Measurement (LM) offers a way for operators to determine the amount of frame loss in an Ethernet network, over an EVC for instance. More precisely, it is the ratio between undelivered OAM frames and the total number of OAM frames transmitted during a specific time interval.

ITU-T Y.1731 defines two types of LM:

1. Single-Ended. LM messages are transmitted to another MEP, which includes transmission and reception frame counts in its response message. In this case, only the LM initiator is able to derive frame loss from the counters (since it does not include its local counters in the initial LM message).

2. Dual-Ended. Continuity Check messages are used to carry frame transmission and reception counters. In contrast to the single-ended approach, this approach allows all MEPs inside a ME to derive frame loss, instead of only the initiating node.
4.6 Delay Measurement

Delay Measurement (DM) can be used for measuring delay in a Carrier Ethernet network. The unit of measurement is the round trip delay of a frame, measured from its first transmitted bit, until the reception of its last bit. Since a DM frame needs to be sent back to its originating node, LB messages are used.

Two types of DM can be identified:

1. One-way measurement. An initiating MEP includes a transmission timestamp in the Ethernet frame. The destination node will capture the frame reception timestamp, and compare both timestamps. As a consequence, the clocks of the sending and receiving nodes need to be synchronized.

2. Two-way measurement. In contrast to the one-way measurement, this DM type does not require clock synchronization. The initiating node still includes a timestamp in the Ethernet frame. After the destination node performs a loopback on the frame, the initiating node will receive the frame again. On reception, this node will capture the reception timestamp. Finally, the difference between the timestamps can be calculated.

5 Comparison to an IP-Based Approach

The Ethernet OAM management techniques discussed in the previous section are, from a functional point of view, very similar to IP-based management techniques. A few examples:

- Ethernet OAM Continuity Check resembles a uni-directional IP Ping.
- Ethernet OAM Loopback verifies connectivity with a MEP or MIP, by performing a loopback on a frame. This is similar to IP Ping.
- Ethernet OAM Link Trace offers comparable functionality as IP Traceroute. Both techniques allow to trace a path between nodes through a network. Instead of using a ‘time-to-live’ (TTL) field in a frame header, MEPs/MIPs pass LT messages downstream the path towards a destination node.
- Ethernet OAM Alarm Indication Signal is able to send out notifications to the reachable MEPs of Maintenance Entity (ME). SNMP allows the transmission of traps/notifications to a SNMP manager as well.
- Ethernet OAM Loss Measurement & Delay Measurement offer similar features as certain SNMP Management Information Bases (MIBs) do.

As we outlined in the Introduction, we would like to verify whether it is possible at all, or until which extent, to manage Carrier Ethernet networks with IP-based protocols. In order to do so, IP needs to be supported on top of the Ethernet infrastructure. Although this seems to contrast the principle of having a pure Ethernet network, most network devices already have an IP interface for management purposes, in order to support a Web interface, Telnet, SSH, syslog or SNMP, for instance. Since it might not be desirable to have a full IP infrastructure on top of an Ethernet backbone network, we assume that IP will not be used for routing purposes and that all managed devices take part in a management VLAN. This has the following consequences:
1. Only devices inside the same management VLAN and IP domain are reachable. However, it can be advantageous for end-to-end EVC management to have nodes reachable from outside a specific domain, for instance.

2. The IP TTL field value is not lowered on Ethernet network hop transition.

3. Since frames destined to a node (identified by a B-MAC and B-VID) for which no path has been defined will be discarded by ingress Backbone Edge Bridges (BEBs), paths from a management node to all managed devices need to be defined.

In order to verify whether an IP-based approach could be used to manage a Carrier Ethernet network, we defined a typical deployment scenario for a Carrier Ethernet, in which Ethernet OAM could be deployed. It will be used to analyze an IP-based approach for managing a Carrier Ethernet network.

5.1 Deployment Scenario

Figure 2 shows our deployment scenario for a Carrier Ethernet network. Four customer sites are shown, belonging to two different customers (A and B). Customers can be end customers, service providers, operators, and access or aggregation networks [15]. Both customers acquired an EVC between their two sites. The result is a (virtual) one-hop connection between the customer-facing switch ports of the edge devices. The transport network considered here is based on IEEE 802.1Qay (PBB-TE).

A customer site is connected to a BEB, which consists of two components [15]:

1. **I-Component**: maps S-VIDs to I-SIDs (Instance IDs) and adds a PBB header with/without a B-VID.
2. **B-Component**: maps I-SIDs to B-VIDs and adds a B-VID to the PBB header (or the whole PBB header in case the I-Component has not done so).

Fig. 2. Deployment scenario of a PBB-TE network with Ethernet OAM

\[^{1}\text{ Whether an I-Component inserts PBB header or not depends on vendor implementation.}\]
I-Components are used for bridging in the customer space, based on customer MAC addresses and S-VIDs. B-Components are used for bridging in the provider domain based on B-MAC addresses and B-VIDs [15]. The I-Component is often called Customer Premises Equipment (CPE). A CPE is the last hop between a service provider network and a customer’s equipment [16]. The two components of a BEB can be either in one or in two devices. In Figure 2, the components of the left BEB are in separate devices. EVCs are established between two CPEs.

The next hop after the first BEB is one of the Backbone Core Bridges (BCB), depending on the B-VID onto which the frame’s I-SID (Instance ID) got mapped. Although a PBB network is considered here, the BCBs consider the frames as normal (VLAN-tagged) Ethernet frames. This is because the first three fields of the Ethernet header are the same for PB, PBB and IEEE 802.1Q, as shown in Figure 1. After the BCBs, the (BEB) egress switch of the backbone network is the next hop. The I-Component and B-Component are now packed into one device and perform the same tasks as the first BEB, but in reverse order.

Section 4 discussed how Carrier Ethernet networks could be managed by using either IEEE 802.1ag or ITU-T Y.1731. These standards require managed nodes to be either a Maintenance End-Point (MEP) or Maintenance Intermediate-Point (MIP). In our deployment scenario, (B-Components of the) BEBs are assigned the role of MEP and BCBs the role of MIP. Three Maintenance Entities (MEs) can be identified, one for each path through the network, so that each MEP is taking part in three MEs. In this work we consider only the operator’s OAM mechanisms and a single OAM level.

The remainder of this section discusses the use of IP-based protocols in place of Ethernet OAM management techniques.

5.2 Continuity Check & Loopback

To automate Ping message transmission and to make it easier for an operator to handle this, a ‘Remote Ping’ MIB, which is part of the DISMAN (short for ‘Distributed Management’) framework, could be used. It lets a ‘Local host’ command a ‘Remote host’ to perform a Ping to a ‘Target host’. Assuming that all nodes are reachable by the ‘Remote host’ and have a ‘Remote Ping’ implementation, a network operator could issue a Ping request from and to each node inside the same VLAN.

Compared to Ethernet OAM, the following advantages can be identified when using IP Ping:

- **When using IP Ping, both source and destination nodes can detect a failure.** Request messages can be sent in a periodic fashion, and a destination node must be configured to expect them in that fashion as well. When a reception timeout occurs, a faulty link or device can be assumed. This is not possible with Ethernet OAM CC, which is uni-directional by definition.

- **All nodes between which a path exists in a PBB-TE network, can exchange IP Ping messages by means of the DISMAN framework.** If such a path does not exist, frames will be dropped by the edge switches. The same is done for
broadcast traffic [13]. Ethernet OAM CC and LB can only be initiated and terminated by MEPs. Therefore, the set of nodes reachable by IP Ping can be larger than the amount of nodes reachable by Ethernet OAM.

Besides these advantages, also several disadvantages can be identified:

- **Without network-layer routing, the set of manageable nodes is restricted to a single IP/management domain.** In contrast, Ethernet OAM can be performed in an end-to-end manner over an EVC for monitoring a single service, spanning multiple domains. It is not possible for a service provider to inject (IP) packets into an EVC to verify its functioning in an end-to-end manner.

- **It is hard to ensure that IP Ping takes the path of a particular EVC, to ensure its connectivity.** When customer data arrives at an ingress BEB, it is mapped onto a B-VID, which takes a predefined path through the network. It is hard to ensure that the management VLAN uses exactly the same path.

- **A per-customer/EVC granularity requires several translation steps.** Since Ethernet OAM allows the verification of an EVC, it is immediately clear which customers are affected by a fault. More knowledge about the network is required with IP-based protocols, in order to derive the same information. Operators will need to know onto which VLAN the customer traffic is mapped. This involves active cooperation with other parties, since operators normally do not know how others mapped customer traffic onto the VLANs.

Customers can also be involved in monitoring an EVC by using IP Ping. They will, however, see the EVC service as a one-hop path. As such, they will be able to detect a problem on the EVC, but without being able to isolate it.

The deployment scenario discussed before can also be made more complex, by considering multiple operator networks between the customer sites. Customers will still see the provider domain as a one-hop connection. As soon as a customer detects a problem without Ethernet OAM, it is up to the service provider to find out which operator network causes the fault. Assuming that the path of the operator’s management VLAN is the same as the customer’s EVC, fault isolation with IP Ping is possible. With Ethernet OAM however, several (external) nodes could be configured as MIPs, so that a service provider could isolate a fault directly, even if it is located in another administrative domain.

5.3 **Link Trace**

In a similar way as we discussed the replacement of Continuity Check and Loopback by IP Ping, we assume the use of IP Traceroute as a replacement for Link Trace (LT). By adjusting the value in the ‘time-to-live’ (TTL) field of the IP header, the path through the network can be traced. Since IP is only used inside the management VLAN and not used for routing purposes, intermediate nodes towards a destination will never modify the TTL field value. All traces will then consist of one-hop connections and link tracing by using IP Traceroute will therefore never work in our deployment scenario.

---

4We assume the service provider network here to consist of at least two operator networks.
5.4 Alarm Indication Signal

Ethernet OAM allows the transmission of fault notifications by means of an Alarm Indication Signal (AIS). In IP-based networks, SNMP traps can be used for the transmission of notifications from agents to managers. Several default traps have been defined, such as ‘linkup’ and ‘linkdown’. When a node inside the network detects a failure, it could send out a trap to an SNMP manager.

SNMP as a replacement for AIS offers several advantages:

- **SNMP traps can be sent out to an arbitrary set of SNMP managers inside the management VLAN, while AIS can only be sent out to MEPs.**
- **SNMP offers more flexibility in defining trap structures, by allowing the definition of custom (‘enterprise-specific’) traps.** An arbitrary set of variables can be included in a SNMP trap. Also different traps can be sent for different purposes, while Ethernet OAM AIS has a fixed structure.

Besides these advantages, several disadvantages can be identified. At first, AIS can be multicasted on each S-VLAN affected by the failure automatically. This is also possible with IP-based solutions, but this requires more overhead in deriving the EVCs/customers affected by a failure. Second, SNMP traps can only be sent out inside a single IP/management domain.

5.5 Loss Measurement

Ethernet OAM Loss Measurement (LM) calculates the frame loss between two MEPs, by comparing the difference between OAM frame transmission and reception counters at the MEPs of a particular ME. By means of the RMON-MIB [17], SNMP manages Ethernet interface counters, such as ‘etherStatsPkts’. Although this counter keeps track of the sum of ingoing and outgoing frames, it is possible to define an ‘enterprise-specific’ MIB which manages these counters individually. Some MIBs exist for this purpose, such as a ‘Round Trip Time Monitoring’ (RTTMON) MIB and ‘Service Assurance Agent’ of Cisco.

The use of SNMP for LM offers several advantages:

- **Arbitrary values can be retrieved, depending on the used MIB.** The IF-MIB and RMON-MIB offer a rich set of counters and other interface statistics. Ethernet OAM LM only allows OAM frame counters to be retrieved.
- **SNMP PDUs can be sent to an arbitrary set of SNMP managers inside the management VLAN.** If all nodes have the IF-MIB/RMON-MIB deployed, a SNMP manager can retrieve the counter values from each of these nodes. Ethernet OAM LM only allows MEPs to calculate loss on a path.

Ethernet OAM LM measures OAM frame loss between MEPs inside a single ME. As such, frames coming from nodes outside the ME are not considered. To do the same with SNMP, an ‘enterprise-specific’ MIB would be needed to differentiate between frame sources or types for measuring frame loss between two network end points. Besides that, SNMP PDUs can only transmitted inside a single management/IP domain. This has been discussed in Section 5.2.
5.6 Delay Measurement

The use of IP Ping for managing Ethernet networks has been discussed before. This protocol provides round trip delay measurements together with its results. Although Ethernet OAM DM offers some sophisticated ways to compensate for processing times at end nodes, round trip delays can be measured by using IP Ping inside the management VLAN as well. This results in the same advantages and disadvantages as described before in Section 5.2. Besides that, several SNMP MIBs have been defined for the purpose of delay measurement, such as Cisco’s RTTMON MIB, as discussed in the previous subsection.

6 Conclusions

This paper presented an overview of the various Carrier Ethernet standards and the related Ethernet OAM mechanisms. By considering a specific deployment scenario for a Carrier Ethernet operator network, an IP-based approach for managing these networks has been analyzed.

In the first section of this paper, two research questions were addressed:

1. What exactly is Carrier Ethernet and which functionality does it provide?
   Compared to the initial Ethernet standard for LANs, especially scalability improvements have been added to Ethernet. This allowed Ethernet to deal better with a greater number of MAC address in wide area networks. Due to this network scale increase, management of Ethernet became much more important than before. As a result, a set of management techniques was defined, in order to manage Ethernet on the Ethernet layer. When Ethernet is used in large-scale networks and manageable by using Ethernet OAM, it is commonly referred to as ‘Carrier Ethernet’ or ‘Metropolitan Ethernet’.

2. How does Ethernet OAM functionality compare to IP OAM, and, more specifically, can IP-based protocols in Carrier Ethernet networks provide the same functionality as comparable Ethernet OAM management techniques?
   From a functional point of view, the various Ethernet OAM management techniques appeared to be very similar to IP OAM protocols, such as Ping, Traceroute and SNMP. For a single operator domain, most IP-based protocols discussed in this paper are able to provide similar functionality, as their Ethernet OAM ‘counterparts’. IP Traceroute is the only protocol that turned out not to be functional at all. Besides that, the scope of an IP-based approach is limited to a single management/IP domain, since network layer routing was not considered in our deployment scenario. Consequently, IP-based protocols are not deployable in an end-to-end fashion over an Ethernet Virtual Connection, which makes it impossible to verify the end-to-end service offered to a customer. IP-based protocols are therefore not a suitable replacement for Ethernet OAM in operator environments.

As future work, a multi-domain Carrier Ethernet deployment could be investigated. Besides that, the use of other OAM techniques for Carrier Ethernet networks, such as MPLS OAM or SDH OAM, could be investigated.
Acknowledgements

This research work has been supported by SURFnet’s GigaPort3 project for Next-Generation Networks and the EU FP7-257513 UniverSelf Collaborative Project. Special thanks to Mark Prins from TNO-ICT for his valuable contribution to the research.

References

3. IEEE: 802.1ag: Connectivity Fault Management (December 2007)
8. IEEE: 802.1Q: Virtual Bridged Local Area Networks (May 2006)
9. IEEE: 802.1ad: Virtual Bridged Local Area Networks - Amendment 4: Provider Bridges (May 2006)
10. IEEE: 802.1ah: Virtual Bridged Local Area Networks - Amendment 7: Provider Backbone Bridges (August 2008)
Appendix B

EUNICE 2011 Paper

This appendix includes the final version of the conference paper Flow Monitoring Experiences at the Ethernet-Layer. The paper focuses on use cases for flow monitoring at the Ethernet-layer, which was not possible with existing technologies such as Cisco’s NetFlow. IPFIX - a recent standard for which offers the flexibility to assign Ethernet key fields to flow definitions, among others - could only be deployed in a specialized setup, which was available at the University of Twente during the Master assignment.

| Name: | 17th EUNICE Open European Summer School (EUNICE 2011) |
| Venue: | Technische Universität Dresden, Dresden, Germany |
| Web page: | http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_elektrotechnik_und_informationstechnik/ifn/eunice |
| Date: | September 5 - 7, 2011 |
| Presentation date: | September 2011 |

Table B.1: Conference information
Flow Monitoring Experiences at the Ethernet-Layer

Rick Hofstede, Idilio Drago, Anna Sperotto, Aiko Pras

University of Twente
Centre for Telematics and Information Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Design and Analysis of Communications Systems (DACS)
Enschede, The Netherlands
r.j.hofstede@student.utwente.nl, {i.drago,a.sperotto,a.pras}@utwente.nl

Abstract. Flow monitoring is a scalable technology for providing summaries of network activity. Being deployed at the IP-layer, it uses fixed flow definitions, based on fields of the IP-layer and higher layers. Since several backbone network operators are considering the deployment of (Carrier) Ethernet in their Next-Generation Network, flow monitoring should also evolve in that direction. In order to do flow monitoring at the Ethernet-layer, Ethernet header fields need to be considered in flow definitions. IPFIX provides the flexibility to change the definition of flows, incorporating information from several layers in the network (including non-IP fields). The deployment of IPFIX is still at an early stage, which means that use cases for Ethernet-layer monitoring are not well known yet. This paper provides an overview of the usability of IPFIX at the Ethernet-layer and presents several use cases in which Ethernet-layer flow monitoring provides new insights and different views on a network.

Keywords: Network management, flow monitoring, IPFIX, Carrier Ethernet

1 Introduction

The huge amount of traffic in high-speed networks requires scalable approaches for network monitoring. Flow\textsuperscript{1} monitoring is a feasible solution in such networks. It provides aggregated network data, resulting in a summary of network activities at a certain network layer. This can increase the visibility of the network behaviour by, for example, showing hosts and applications that are generating specific traffic. The main advantage of flow-based approaches is that they overcome the scalability problems of packet-level captures, where all traffic must be exported. For high-speed network connections (10 Gbps and higher), packet-level monitoring is not feasible, or could lead to severe performance problems of probing equipment.

\textsuperscript{1}We consider a flow as “a set of packets passing by an observation point in a network during a certain time interval and having a set of common properties” [14].
Cisco’s NetFlow [3] is currently the major network flow export technology. It aggregates packets into flows if they share the same values in their key fields. Non-key fields are not considered in the definition of a flow. NetFlow version 5 (v5), which is still the most-used protocol for flow export, provides flow data at the IP-layer with a fixed flow definition. As such, neither flow key fields (such as source/destination IP, source/destination port, protocol etc.), nor non-key fields (such as packet and octet counters) can be changed. NetFlow version 9 (v9) was proposed by Cisco to overcome this limitation, allowing flow export records to be specified freely by means of templates. IPFIX (IP Flow Information Export) [4] is an effort by the IETF (Internet Engineering Task Force) to create a standard protocol for collecting and exporting flows. Cisco’s NetFlow v9 was used as the basis for the IPFIX specification [11]. One of the most distinctive characteristics of IPFIX is that it allows flows to contain information from several layers, including the Ethernet-layer.

Since several backbone network operators are considering the deployment of Carrier Ethernet\(^2\) in their Next-Generation Network [17], monitoring at the IP-layer is not a suitable solution anymore. IPFIX, however, could be used for that purpose. Due to the fact that the deployment of IPFIX is still in an early stage, the applicability of the protocol for Ethernet monitoring is not well known yet. In this context, this paper investigates several use cases, answering the following research question:

*What are the advantages of flow monitoring at the Ethernet-layer, compared to IP-layer flow monitoring?*

In order to answer this question, the University of Twente (UT) acquired two specialised, early-deployment probes (i.e. dedicated flow export devices) from INVEA-TECH\(^3\). This equipment provides a means to define flows based on Ethernet-header fields. Before deploying the equipment in a Carrier Ethernet (i.e. service-provider) network, it was tested in the UT’s 802.1Q-based Ethernet network, which carries 110 Virtual LANs (VLANs). This paper presents an overview of the IPFIX prototype equipment specially adopted for this research and several use cases identified during the testing phase.

This paper is organised as follows: the IPFIX architecture and its deployment at the Ethernet-layer are discussed in Section 2. The fact that no suitable IPFIX software is available in the market had severe impact on the design of the early-deployment IPFIX equipment. Details on that will be provided in Section 3. After that, Section 4 describes the exported Ethernet flow data, together with four identified use cases. Although some other monitoring technologies exist for monitoring a network at the Ethernet-layer, IPFIX offers several advantages. Section 5 will focus on related technologies, by comparing them to IPFIX. Finally, we close this paper in Section 6, where we draw our conclusions and future work.

\(^2\)When Ethernet technology is used in large-scale (e.g. service-provider) networks, it is commonly referred to as ‘Carrier Ethernet’ or ‘Metropolitan Ethernet’.

\(^3\)INVEA-TECH is a university spin-off company from Brno, Czech Republic.
2 IPFIX at the Ethernet-Layer

IPFIX is a flow export protocol, based on the principles of NetFlow v9. Its architecture is defined in [15]. According to the standard, an IPFIX Device hosts at least one Exporting Process and eventually Observation Points and Metering Processes. An Observation Point is a location where packets are collected from the network by a Metering Process. Each pair formed by an Observation Point and a Metering Process belongs to a unique Observation Domain. The Exporting Process is the entity responsible for exporting flow records to Collectors. The tasks of Collectors are 1) the interpretation of IPFIX messages from different Observation Domains and 2) the storage of control information (e.g. flow definitions) and flow records received from an IPFIX Device. The IPFIX Device architecture is depicted in Figure 1.

Fig. 1. IPFIX Device architecture

The main tasks performed by the Metering Process are depicted in Figure 2. After packets are captured at an Observation Point and timestamped, packets can be sampled (i.e. selected for processing within a stream of packets) or filtered. The IPFIX standard, however, does not specify any techniques for that. The packets that qualify for flow processing are passed to the next stage, where flows are either created or updated. When using IPFIX for Ethernet monitoring, those tasks are still the same, although the Metering Process will deal with complete Ethernet frames, instead of IP packets.

Fig. 2. IPFIX packet selection criteria
As said in Section 1, IPFIX allows to change the key fields of a flow [21]. Moreover, it allows flow definitions to consist of other fields than which are present in the IP-based definition of 5 and up to 7 IP packet attributes. Among those are fields from the Ethernet-layer, for instance. The possible (key and non-key) fields are maintained by the IANA (Internet Assigned Numbers Authority) and are called IPFIX Information Elements [20]. Due to the fact that IPFIX can also export non-IP flows, the list of Information Elements (IEs) is much larger than the list of possible fields for NetFlow. An overview of the most elementary Information Elements for the Ethernet-layer is presented in Table 1. More information about that can be found at IANA Web site [20], and IEEE standards [7], [8] and [9].

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceMacAddress</td>
<td>Source MAC address</td>
</tr>
<tr>
<td>destinationMacAddress</td>
<td>Destination MAC address</td>
</tr>
<tr>
<td>ethernetPayloadLength</td>
<td>MAC client data size (including any padding)</td>
</tr>
<tr>
<td>ethernetType</td>
<td>Ethernet type field, which identifies the type of payload in the Ethernet frame</td>
</tr>
<tr>
<td>dot1qVlanId</td>
<td>IEEE 802.1Q VLAN identifier. In case of a QinQ or 802.1ad frame, it represents the VLAN tag in the service-provider domain</td>
</tr>
<tr>
<td>dot1qPriority</td>
<td>IEEE 802 user priority. In case of a QinQ or 802.1ad frame, it represents the user priority in the service-provider domain</td>
</tr>
<tr>
<td>dot1qCustomerVlanId</td>
<td>In case of a QinQ or 802.1ad frame, it represents the VLAN tag in the customer domain</td>
</tr>
<tr>
<td>dot1qCustomerPriority</td>
<td>In case of a QinQ or 802.1ad frame, it represents the user priority in the customer domain</td>
</tr>
<tr>
<td>ethernetHeaderLength</td>
<td>IEEE 802 frame header size. It is the difference between the total frame size and the MAC client data size</td>
</tr>
<tr>
<td>metroEvcID</td>
<td>Ethernet Virtual Connection (EVC) ID, which uniquely identifies an EVC in a Carrier Ethernet network</td>
</tr>
<tr>
<td>metroEvcType</td>
<td>Represents the type of service provided by an Ethernet Virtual Connection</td>
</tr>
</tbody>
</table>

Table 1. IPFIX Information Elements for Ethernet

Although several Information Elements in Table 1 are only relevant in Carrier Ethernet networks, some are also valid in regular (i.e. non-Carrier) Ethernet networks. The fields related to customer frames (dot1qCustomerVlanId, for example) and Ethernet Virtual Connections (metroEvcID, for example) are the most important exceptions: they provide more insights into the customer traffic and, therefore, are essential for monitoring Ethernet transport networks.

The standard 5-tuple consists of the following fields: source and destination IP addresses, source and destination ports, and transport protocol. The other two common key fields are the type of service (TOS) and the input interface.
3 IPFIX Device Prototype

In order to answer the research question listed in Section 1, the UT acquired two INVEA-TECH FlowMon Probes [10]. This equipment is specialised in flow export (by means of NetFlow v5/v9/IPFIX) in high-speed networks (up to 10 Gbps), and uses an easily extensible software platform. A special Ethernet-plugin was developed by INVEA-TECH for the UT, in order to provide an IPFIX Device prototype, able to collect Information Elements from the Ethernet-layer.

Instead of reimplementing the Metering Process to follow the IPFIX architecture (as described in Section 2), this prototype stores Ethernet data in the IPv6 fields of NetFlow v9 records. In other words, the device is exporting NetFlow v9 packets, but uses IPv6 fields to store the Ethernet data. The complete mapping from IPFIX Information Elements to NetFlow v9 fields is listed in Table 2 and 3. In these tables, the left column refers to the original NetFlow v9 field, while the right column refers to the IPFIX Information Element exported by the Ethernet-plugin. The presented approach has the following advantages:

1. An early-deployment of IPFIX for Ethernet-layer monitoring could be made, because existing (IP-layer) flow processing algorithms (e.g. hash tables in the flow cache) could be reused. Besides that, no suitable IPFIX Collectors are available yet. By using NetFlow v9 packets, the existing NetFlow Collectors can be used. As an example, nfcapd, part of the nfdump tools suite [12], can be used as a Collector to store NetFlow records on stable storage.

2. Several existing monitoring tools, which support NetFlow v9, can be used to analyse the exported flow data. In some cases, however, small corrections are necessary. For example, nfdump, which normally allows to display flow data and to perform aggregations, will not be able to interpret all fields exported by the IPFIX Device prototype correctly. This is because IPv6 fields are used to store non-IPv6 data. However, it is possible to overcome certain incompatibilities, by extending nfdump. An example of this is shown in Table 4. It shows nfdump, combined with an utility from INVEA-TECH to adapt its standard output to Ethernet-layer data, in order to print flows to the terminal. Several columns, such as Destination MAC and Priority, have been left out of the table, for the sake of space.

\begin{table}[h]
\centering
\begin{tabular}{ll}
srcIPv6 & sourceMacAddress \\
srcPort & dot1qVlanId \\
dstIPv6 & destinationMacAddress \\
dstPort & ethernetType \\
l3.proto & 0 (unused) \\
l4.proto & 0 (unused) \\
port.in & probe port ID \\
\end{tabular}
\caption{Ethernet-plugin key fields}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{ll}
srcAS & ethernetHeaderLength \\
dstAS & ethernetPayloadLength \\
ToS & dot1qPriority \\
TCP flags & dot1qCustomerPriority \\
port.out & dot1qCustomerVlanId \\
flow start & first frame seen \\
flow end & last frame seen \\
packets & frames \\
bytes & bytes \\
\end{tabular}
\caption{Ethernet-plugin non-key fields}
\end{table}
Table 4. `nfdump` output showing Ethernet data

<table>
<thead>
<tr>
<th>Start time</th>
<th>Src MAC address</th>
<th>Type</th>
<th>VLAN</th>
<th>EHL</th>
<th>EPL</th>
<th>Frames</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-04-03</td>
<td>23:57:29.275</td>
<td>0x0800</td>
<td>161</td>
<td>14</td>
<td>56</td>
<td>99</td>
<td>11238</td>
</tr>
<tr>
<td>2011-04-03</td>
<td>23:57:29.529</td>
<td>0x0800</td>
<td>103</td>
<td>14</td>
<td>62</td>
<td>47</td>
<td>7456</td>
</tr>
<tr>
<td>2011-04-03</td>
<td>23:57:31.792</td>
<td>0x86DD</td>
<td>103</td>
<td>14</td>
<td>443</td>
<td>39</td>
<td>16355</td>
</tr>
<tr>
<td>2011-04-03</td>
<td>23:57:32.659</td>
<td>0x0806</td>
<td>103</td>
<td>14</td>
<td>50</td>
<td>16</td>
<td>1024</td>
</tr>
<tr>
<td>2011-04-03</td>
<td>23:57:34.440</td>
<td>0x0806</td>
<td>103</td>
<td>14</td>
<td>50</td>
<td>5</td>
<td>320</td>
</tr>
</tbody>
</table>

EHL – Ethernet Header Length
EPL – Ethernet Payload Length

4 Results

The previous sections have made clear that monitoring at the Ethernet-layer by means of IPFIX is a completely new area in the network management community. This is especially true when it comes to hands-on experience. After the two INVEA-TECH FlowMon Probes had been installed in the campus network of the UT, several tests have been performed. This section will highlight several aspects of the obtained hands-on experience, in the fields of traffic profiling, misconfiguration detection and device misbehaviour detection.

4.1 Traffic profiling

Traffic profiling is the process of exploring active traffic types in a network. The IPFIX Device prototype allows to do that at the Ethernet-layer. This gives a completely different view on the network, since all active layer-2 protocols\(^5\) can be monitored. Besides all protocols that we expected to see active in our campus network - such as Novell IPX, Link-Layer Discovery Protocol (LLDP), and Address Resolution Protocol (ARP) - we have discovered other less common protocols. Among them are DECnet Phase IV protocols, Cisco WLAN Context Control Protocol and Multi-Protocol Label Switching (MPLS) Unicast. Since these protocols do not operate on top of IP, NetFlow would not have been able to identify them.

Having Ethernet flow data allows to compare the amount of flows, packets and octets that were exchanged by the active layer-2 protocols. One of the most striking results obtained was the difference in the traffic behaviour of IPv4 and IPv6 (shown in Figure 3). Note that traffic profiling for IPv4 and IPv6 could also have been done by using NetFlow, although the higher data aggregation level of Ethernet flow data makes profiling much faster and easier.

Over a period of 24 hours, the amount of IPv4 flows was almost equal to the amount of IPv6 flows on the campus network. However, the amount of octets generated within 24 hours by IPv4 was roughly 40 times as high as the amount of IPv6 octets.\(^6\)

\(^5\)The set of protocols operating directly on top of Ethernet, such as IP and ARP.

\(^6\)The set of protocols operating directly on top of Ethernet, such as IP and ARP.
of octets generated by IPv6 (25 TB and 600 GB, respectively). Although most machines have a dual-stack setup nowadays (to support both IPv4 and IPv6), most of the traffic carrying user payload is sent over IPv4. This behaviour can be clearly identified in Figure 3. Even though the amount of IPv6 flows starts to increase significantly after noon on the capturing day, the amount of octets exchanged remains low. One of the reasons for that is the Neighbour Discovery Protocol (NDP), which is part of IPv6 (and ICMPv6). As such, flows caused by NDP will be counted as IPv6 flows. For IPv4, neighbour discovery is handled by ARP. ARP operates directly on top of Ethernet, which is therefore not counted as an IPv4 flow.

4.2 Misconfiguration detection

Both main routers at the edge of the UT campus network support the DECnet Phase IV protocol suite for management purposes. Since these protocols are not used anymore, their interfaces should have been disabled for security reasons. One of the active layer-2 protocols on the network, however, belongs to the DECnet Phase IV protocol suite. We discovered this traffic by identifying the corresponding ethertype. Besides that, the flow behaviour shows a clear periodicity, which is shown in Figure 4. The network managers found out that the DECnet interface on one of the routers was not properly disabled. Without Ethernet-layer monitoring, this misconfiguration could not have been detected.
4.3 Device misbehaviour detection (1)

While profiling the network on the UT campus, two unknown ethtypes were detected: 0x8259 and 0x0A59. The IANA maintains a list of registered ethtypes [19], but the discovered ethtypes were not present on that list. After transforming these hexadecimal values into decimal IP addresses, the IP subnet prefixes used by UT (130.89/16 and 10.89/16) are obtained.

Packet-level capturing at various points in the network allowed us to identify the generator of these Ethernet frames: a data centre switch of a major network device vendor (operating with beta firmware) had a bug in its IGMP Snooping functionality, resulting in mangled packets. As such, the switch was putting the first two octets of IP addresses (extracted from the Ethernet payload) inside the ethtypes field. The consequence of this is that Ethernet frames were partly overwritten, making them corrupt and useless.

4.4 Device misbehaviour detection (2)

During our experiments, a campus host with a malfunctioning network driver (for hardware firewalling) caused severe problems to the UT’s campus network. The host generated a huge amount of ARP messages, resulting in a degraded network performance. This is depicted in Figure 5. While ARP normally generates 2 million octets per minute on average (as shown in Figure 5(a)), it generated around 35 million octets per minute at the moment the host started sending malicious data (Figure 5(b)). Since it is not possible to monitor ARP traffic with normal NetFlow technology, it would not have been possible to detect this issue without the use of IPFIX.
5 Related Work

As mentioned in Section 2, the predecessor of IPFIX is NetFlow v9. Some of the use cases presented (i.e., misconfiguration and device misbehaviour) would not have been possible with NetFlow. NetFlow uses fixed flow keys, which do not contain Ethernet fields. IPFIX, however, offers flexible flow keys (by means of Information Elements), which allows to monitor a network at the Ethernet-layer.

A complementary protocol to IPFIX is PSAMP (Packet Sampling) [5]. According to [5], “the main difference between IPFIX and PSAMP is that IPFIX addresses the export of Flow Records, whereas PSAMP addresses the export of packet records”. The two protocols share a part of their architectures, which is depicted in Figure 6.

The IPFIX architecture consists of two stages, namely 1) packet processing, and 2) flow processing. The first stage is identical in the IPFIX and PSAMP architectures. When a packet header is captured and timestamped, it is passed to the packet selection process. In this step, packets can be sampled or filtered. After that, the next step depends on the considered protocol:

1. If IPFIX is used, packets reach the flow processing stage, in which they are mapped to flows. This means that either an existing flow record is updated, or a new flow record is created. The final step is to export the flows.
2. If PSAMP is used, packet reports are exported, instead of flow records. These reports can be seen as a special IPFIX record, containing the information about a single packet.

With PSAMP, it would not have been possible to do traffic profiling (as discussed in Section 4.1) as precise as with IPFIX. The reason for that is the sampling, which is done by PSAMP by definition. Although it is possible to mathematically compensate for sampling [6], this process is not straightforward. The data presented in this paper, however, is always unsampled.

While IPFIX is an IETF-standard, also industry technologies exist for network monitoring. One of them is sFlow [16], which uses packet sampling (but not by means of PSAMP) for exporting network data. Just as IPFIX, it offers
a network monitoring solution at the Ethernet-layer. Some differences can be identified when comparing sFlow to IPFIX:

- sFlow is usually available on a dedicated hardware chip in a network device, while IPFIX usually shares a hardware and a software solution. The advantage of a complete hardware-based approach is that the CPU and memory of the device are preserved for other tasks (such as routing and switching).
- sFlow uses packet sampling by definition. Although this saves hardware resources of the network device, the resulting data set is a subset of the actual network traffic. On the other hand, IPFIX allows to collect unsampled flow data, resulting in a more complete overview of the traffic. Moreover, even though IPFIX is used without sampling, it is still scalable. Because of the packet sampling used by sFlow, traffic profiling cannot be done with the same precision as with an IPFIX Device, just as it is the case for PSAMP.

*tcpdump* [18] is a packet-level traffic capturing and analysis tool. It uses PCAP (Packet Capture) for capturing packets on a medium and eventually to write them to files. Due to the limited bandwidth available in machines for writing data to stable storage, capturing network traffic in high-speed networks (*e.g.* 10 Gbps) causes severe performance problems to systems. A consequence of these performance problems is that packets will be dropped by the kernel of the operating system, resulting in incomplete traces. For these reasons, making packet-level captures in high-speed networks, and especially in transport (service-provider) networks, is not a suitable solution.

NeTraMet (Network Traffic Meter) [2] is another approach to flow monitoring and an open-source implementation of the IETF Meter MIB [13]. Within NeTraMet, *rule sets* are used to specify the information fields that should be gathered from the network traffic [1]. As a consequence, these *rules* can also be used for specifying which flows are filtered. NeTraMet is a software-based solution, which uses PCAP (just as *tcpdump*). Therefore, NeTraMet is not suitable for monitoring high-speed network links (*e.g.* 10Gbps), for the same reasons as *tcpdump.*
6 Conclusions

Flow monitoring is a scalable technology for monitoring traffic in high-speed networks. Until recently, it was mainly deployed at the IP-layer, providing a summary of network traffic based on IP and TCP/UDP fields. When it comes to flow monitoring at the Ethernet-layer, as it is needed for Carrier Ethernet networks, another technology is required. IPFIX is a suitable solution for that, since it allows to define flow keys based on Ethernet fields. The protocol, however, is still in an early-deployment phase and little hands-on experience has been gathered. The IPFIX Device prototype acquired by the UT has been tested in a campus network, in order to answer the research question risen in Section 1:

What are the advantages of flow monitoring at the Ethernet-layer, compared to IP-layer flow monitoring?

Several use cases were presented, in which Ethernet-layer monitoring provides new insights into the traffic patterns inside the UT’s campus network. These use cases ranged from detecting misconfigurations to detecting device misbehaviour. The discussed related monitoring technologies would not allow to do them with the same simplicity and precision as IPFIX. The major advantage of flow monitoring at the Ethernet-layer is the ability to monitor all active protocols that operate directly on top of Ethernet. Among them are protocols, such as ARP for IPv4, which are essential for IP-based communications. Besides helping to understand how much data these protocols generate and how this amount depends on the number of active hosts, Ethernet-layer flow monitoring can assist network managers in detecting anomalies and debugging problems.

Although Ethernet-layer monitoring provides new insights into the traffic transiting within a network, we think that the implementation discussed in this paper will never be able to replace standard NetFlow. The main reason for this is that the IPFIX Device prototype solely provides Ethernet-layer data (i.e. it supports only Ethernet-based IPFIX Information Elements). Besides that, using NetFlow v9 for carrying Ethernet data is just a temporary solution. In a fully implemented and compatible IPFIX Device, which will become available in the future, it will be possible to add IP-based Information Elements to flow definitions, resulting in a more complete overview of the traffic.

As future work we consider to investigate the detection of more anomaly types, by means of Ethernet flow data. This can be done in two directions: 1) Investigating anomalies which cannot be detected by NetFlow, and 2) investigating how IP-layer anomalies reflect to Ethernet flow data.

Acknowledgements

This research work has been supported by SURFnet’s GigaPort3 project for Next-Generation Networks, the IOP GenCom project Service Optimisation and Quality (SeQual), and the EU FP7-257513 UniverSelf Collaborative Project. Special thanks to Jeroen van Ingen Schenau from the University of Twente for his valuable contribution to the research.
References

7. Institute of Electrical and Electronics Engineers: Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE Standard 802.3 (December 2005)
8. Institute of Electrical and Electronics Engineers: Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks. IEEE Standard 802.1Q (May 2006)
9. Institute of Electrical and Electronics Engineers: Virtual Bridged Local Area Networks - Amendment 4: Provider Bridges. IEEE Standard 802.1ad (May 2006)
18. TCPDUMP/LIBPCAP: (July 2011), http://www.tcpdump.org
19. The Internet Assigned Numbers Authority (IANA): Ether Types (April 2010), http://www.iana.org/assignments/ethernet-numbers
Appendix C

JNSM Paper

This appendix includes a draft version of the journal paper *Observing Remote Service Health Using Network Flows*.

**Table C.1: Journal information**

<table>
<thead>
<tr>
<th>Name</th>
<th>Journal of Network and Systems Management (JNSM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web page</td>
<td><a href="http://www.springer.com/computer/communication+--networks/journal/10922">http://www.springer.com/computer/communication+--networks/journal/10922</a></td>
</tr>
<tr>
<td>Submission date</td>
<td>August/September 2011</td>
</tr>
</tbody>
</table>
Observing Remote Service Health Using Network Flows

Idilio Drago, Rick Hofstede, Ramin Sadre, Aiko Pras

Received: date/ Accepted: date

Abstract Keywords Service Monitoring · Passive Monitoring · Measurement Inaccuracies · DDoS Attack · NetFlow

1 Introduction

Research questions:
1. Can we verify the availability of remote services using flow data?
2. Which hurdles need be overcome before monitoring remote service health?

2 “Anonymous” against PayPal

Anonymous is a group of Internet activists that gained notoriety claiming to support Internet freedom. The group became widely visible in the mainstream media during the reactions that followed the release of USA embassy diplomatic cables by the website WikiLeaks [16]. WikiLeaks started releasing those cables on November 28, 2010. In reaction, WikiLeaks' website became target of Distributed Denial of Services Attacks (DDoS). At the same time, some companies decided to stop doing business with WikiLeaks by, for example, closing accounts used by the website to receive donations.

The group Anonymous reacted setting up Distributed Denial of Services (DDoS) attacks against services offered by those companies. These DDoS attacks, named Operation Payback, were able to disrupt the website of companies like MasterCard and Visa for some hours in December 2010. On the 9th of December 2010, Anonymous' target was PayPal, in a retaliation against PayPal’s decision of canceling WikiLeaks' account. Latter in the same day, PayPal has posted a message in its official blog admitting that the company was targeted in a DDoS attack. However, the company declared that no service disruptions were registered, and

I. Drago · R. Hofstede · R. Sadre · A. Pras
University of Twente, The Netherlands
E-mail: {i.drago,r.j.hofstede,r.sadre,a.pras}@utwente.nl
that only some few clients could have experienced slower response times than in normal conditions [9]. No more information about the impacts of these attacks on PayPal’s services have been published after that. A complete description about the events that followed the release of the USA cables and the *Operation Payback* is presented in [12] and [8].

In this paper, the attacks against PayPal are used as a case study. We would like to verify PayPal’s statement about the health of its services, using the information that is typically available at Internet providers. If that would be possible, Internet providers would be able to identify if such an attack is impacting their own users, without setting up special (active) probes for each situation (e.g. remote destination). In this scenario, network traces are the only source of information, since neither PayPal’s infrastructure nor the attackers’ machines can be directly monitored. In the following sections, more details about *Anonymous*’ attacks are given. First, the tool used during the attacks, called LOIC (Low Orbit Ion Cannon), is described. After that, the expected traces created by LOIC when applying the tool against PayPal web services are described. During the *Operation Payback*, other methods of attacks and variations of LOIC have also been employed - we close this section presenting some of those variations.

2.1 The LOIC Tool

The tool used by *Anonymous* during the *Operation Payback* is a modified version of an open source software, originally developed by Praetox Technologies [11], to perform stress testing against servers. The tool, written in C#, consists of a simple interface that allows the user to type the address of the target server, and to setup some few parameters, like the number of concurrent threads, the message to be sent to the target server, and the method of test. The tool offers three methods, directly related to the implemented protocols: UDP, TCP or HTTP/TCP. The version used by the attackers adds an extra option, which makes the tool to obtain all parameters automatically from a remote IRC (Internet Relay Chat) server. When using this modified version, the user is voluntarily joining a botnet.

An example of a command string sent by one IRC server that was coordinating the attacks is depicted in Figure 1. We have captured this command in the afternoon of 9 December 2010 (GMT), by setting up an ordinary IRC client to listen to a channel used by the attackers to command LOIC.

```
#loic :!azor default targethost=api.paypal.com subsite=/ speed=3
   threads=15 method=tcp wait=false message=Good_night,_paypal
```

Fig. 1: IRC message containing a command string sent to LOIC in December 9, 2010

At that moment, the attackers were announcing the use of 15 threads and running the TCP attack. Since the source code of the modified version of LOIC is also available, we were able to analyse the expected network traces generated by this command. Figure 2 depicts part of the source code of the TCP attack.
implemented in LOIC. This code fragment was invoked by each of the threads created according the command shown in Figure 1.

```java
while (IsFlooding) {
    Socket socket = null;
    if (Protocol == 1) {
        socket = new Socket(...);
        [...]
        socket.Connect(RHost);
        [...]
        try {
            while (IsFlooding) {
                socket.Send(buf);
                [...]
                if (Delay > 0)
                    System.Threading.Thread.Sleep(Delay);
            }
        }
        catch { }
    }
}
```

Fig. 2: Source code fragment of the TCP attack implemented in LOIC

Some LOIC parameters influence the number of packets generated by the tool. In the example of Figure 1, speed is directly mapped into the variable Delay shown in Figure 2 (lines 12−13) - the tool was set to wait 3 milliseconds after sending each message to the server. In addition, LOIC is using standard libraries of C# (class `socket`) to establish connections and to send messages to the target. In this way, packet retransmissions and connection timeouts are completely controlled by the operating system. Moreover, as we have already shown in [12], LOIC does not try to protect the identity of the attackers, for instance by spoofing IP addresses or using virtual connections to (infected) third-party machines. All these nuances make it clear that the tool is not adequate for a DDoS, both from the perspective of the attacker’s privacy and (specially) from the perspective of the attack effectiveness. These aspects will play a fundamental role in our analyses in the following sections.

2.2 TCP Attack Signature

As it will be shown in the next sections, the TCP method was the one used most of the time during the attacks against PayPal. Before presenting traces supporting that, the expected signature generated by this method is summarised in this section. In order to understand the flows created by the TCP attack from LOIC, the attacked services need to be explained. Two PayPal’s sites have been targeted on December 9: the company official blog, and PayPal’s web service API (Application Programming Interface). In this paper, only the attacks against the web service API will be analysed.

PayPal’s API consists of web services that allow third parties to use PayPal’s payment systems [10]. That means that final users do not have directly access to those services, and therefore a DDoS attack would affect primarily other companies offering PayPal as a payment method. PayPal providers both SOAP (Simple
Object Access Protocol) and NVP (Name-Value Pair) interfaces over HTTPS (Hypertext Transfer Protocol Secure), and both interfaces require authentication. Two methods are available: API signature or API certificate. In the first case, application credentials must be sent to the server in each HTTPS request. In the second case, a certificate issued by PayPal must be presented during the HTTPS handshake. The two authentication schemes are hosted in several IP addresses - during the attacks, only the addresses that requires API certificates (IPs attached to the domain `api.paypal.com`) were targeted. Since the attackers did not have valid certificates, as well as LOIC did not contain any code for dealing with HTTPS traffic, no requests generated by LOIC could pass the HTTPS certificate exchange.

In Figure 3, time sequence diagrams describing the TCP attack are depicted. In the right-hand side, the expected signature when the target is answering the requests is shown. In this case, after the initial TCP handshake, the web server waits for a command from the client in order to start the exchange of certificates. That command may come in several TCP packets, which will be reassembled at the server as soon as two bytes marking the end of line are received. While these bytes are not received, the server could make use of delayed acknowledgments to reduce the traffic of empty packets in the network. Because of that, the number of packets going to the server tends to be bigger than the number of packets coming back to the client.

![Fig. 3: Time sequence diagram of the TCP attack when the target is responding (right-hand side) and when the server is not responding (left-hand side)](image)

In Figure 3, the expected signature when the server does not answer LOIC is also shown. Because LOIC uses libraries that follow the normal operating system behavior for timeouts and retransmissions, there will be several attempts to establish the connection (TCP packets with `syn` flag on), with an increasing delay between each attempt. As an example, Microsoft Windows XP/Vista uses 3, 6 and 12 as first delays. Such behavior would reduce the amount traffic created by LOIC as soon as a target started to be overloaded, affecting the attack effectiveness. This also means that the effects of this attack can be diminished by blocking hosts generating suspicious traffic directly on the target server, since the bandwidth consumed by anomalous traffic will not be intensive in this case.

---

1 PayPal’s web servers hosting `api.paypal.com` announce Apache in their HTTP header variables
2.3 LOIC Variations

3 Estimating Attack Damage

Goal: verify the health of services:

– Not only if the service was completely down, but also if the capacity of answering was affected
– Three methods:
  – Verifying the traffic going and coming back to the service - this would indicate that the service was at least reacting to some attackers
  – Counting request and response flows: given the signatures in figure X, the numbers would be the same if the server was health and no requests were lost/block
  – Matching request/response flows and estimating statistics from that (e.g. latency)

Details about the data set:

– NetFlow V9 Traces collected at 2 Internet providers. First provider is sampling packet at a rate of 1/100. Second provider does not sample packets.
– Giving the possible privacy issues related to the intended analyses, complete data set containing full packet traces could not be obtained. For this research, data sets composed of network flow data (in Cisco NetFlow V9 format [2]) collected at 2 different Internet backbones were used. All data were filtered in advance by the Internet providers, in order to retain only flow records related to the attacks. The original IP addresses of client machines generating the traffic were anonymized.
– One important remark about the 2 collection points is that, normally, no traffic to PayPal web services is observed on the network, except for the attack day.
– There was 2 IPs providing api.paypal.com
– In the following section, we present a summary of the attack. Whenever stated the opposite, all pictures is depicting the average quantity per second (flows, packet, etc), after smoothing the data using a 5 minute moving average window.

3.1 Overall Traffic Traces

3.2 Counting Flows

3.3 Matching Requests and Responses

4 Monitoring Infrastructure Inaccuracies

The previous section presented two approaches for estimating attack damage of a remote service, which seem to be trivial at the first sight. However, several problems in the data collection process make this estimation more complex. The following problems can be identified:

1. **Flow separation caused by timeouts.** After a flow record is stored in the flow cache of a flow exporter, it is kept there until it is considered to be *expired.*
According to the NetFlow specification [2], a flow can be exported under one of the following conditions:

(a) The exporter detected the end of a flow. For example, if the FIN or RST flag is detected in a TCP connection, the flow record is exported.

(b) A flow has been inactive for a certain period of time. This inactivity is normally configurable on the flow exporter. This condition is commonly referred to as **inactive timeout**.

(c) Long-lasting flows should be exported in a regular basis. This timeout is normally configurable on the flow exporter. This condition is commonly referred to as **active timeout**.

(d) If the exporter experiences internal constraints, a flow may be forced to expire prematurely. This might be the case when the flow cache of an exporter is full. The effect is equivalent to variable **active timeouts**.
As long as flows are not exported for the last listed condition, the behavior of flow exporters should be completely deterministic. If a router or probe, however, is experiencing heavy traffic load (e.g. in case it is under attack), flows may be expired before being ended or a timeout has occurred. In that case, an exporter is not deterministic anymore, and the number of exported flows might become non-deterministic as well.

Coming back to the two approaches presented in Section 3, which do both rely on counting the number of ingress and egress flows in a data set, it is clear why these seemingly trivial approaches are not reliable under heavy-load circumstances, such as the attacks presented in Section 2. An example situation in which prematurely exported flows influence the number of ingress and egress flows, is shown in Figure 5. All depicted packets use the same flow definition, so a total of two flows would normally be exported (since flows are unidirectional in NetFlow, one flow is exported per packet direction). Nevertheless, due to an overload of the exporter’s flow cache during the attack, the first three packets might be exported as single packet flows, while the three returning packets are exported as a single flow consisting of three packets. When counting flows for estimating remote service availability in this case, it would seem that more requests were sent to the service, than were returned by the service. As a consequence, wrong assumptions would be made with respect to service availability, since the service might be completely available, although the data set gives the impression that the service was not able to answer all requests.

Figure 6 illustrates the effect of flow separation caused by timeouts on the number of flows. It shows a long-lasting (i.e. about 50 minutes) TCP flow, where the interconnected points in the graph represent the durations of individual flow records. In other words, a long-lasting TCP flows was separated into multiple smaller (NetFlow) flows. In the middle of the TCP flow, a flow record
with a significantly lower duration can be identified. Because TCP flows can be terminated by TCP’s FIN flag, for example, this behavior is not surprising. However, since all depicted flow records are caused by the same host, have the same source port number, and the flow continues after the potential termination, the flow can never have been terminated as such. In other words, the flow was not terminated by the source host, and the shorter duration should have been caused by premature export of the exporter.

Another interesting observation in Figure 6 is the fluctuation in the flow durations in general. The used flow exporter used an active timeout of 64 seconds. Due to a software thread, which is responsible for purging flows of the flow cache, the actual export of a flow can be delayed by at maximum 8 seconds. As such, an active flow being expired by an active timeout can have a duration between 64 and 72 seconds. This is clearly reflected in the figure. Note, however, that the exact duration of the flows is not deterministic. As a consequence, the number of resulting flows is not deterministic anymore, making the counting of flows unreliable.

![Fig. 6: Fluctuations in flow duration due to delayed flow export](image)

2. **Packet/flow loss in exporter.** Packet loss can occur before a packet is exported as part of a flow. Moreover, it can even get lost before arriving at the exporter. When a dedicated flow exporter/probe is used, traffic on a link needs to be duplicated. This can be done by either 1) using an electrical or optical splitter, or by 2) using a switched mirror port. The first option will result in an exact clone of the original data, but the second option can influence the obtained data set. According to [17], the resulting data set can be subject to timing differences, packet reordering and packet loss. As a consequence, the resulting flow data set will not be correct either, if the monitored data (going to the exporter/probe) is arriving in an incorrect manner.

After a packet arrived at a flow exporter, several problems can occur in the flow creation process. The most significant issues for packet loss inside an exporter are caused hash tables. Flow exporters usually use hash tables to store flow
records: by hashing the key fields of a flow, a (hopefully) unique index inside a table can be found, in order to store the flow record. Hash functions do always have a certain efficiency, which means that certain different flow keys will result in the same hash value (i.e. table index). Such a situation is called a collision. When collisions occur, the current flow cannot be stored in the flow cache. Several solutions can be identified for solving this issue. Cisco, for example, uses a separate (small) table for storing an alias index for a key, for which the index is already in use in the normal flow cache. This allows to handle a certain amount of hash collisions and at most one duplicate per hash value. If there are no free entries in the flow cache and an alias is already stored in the second table, the current packet is not considered for flow export. Cisco refers to this event as a flow learn failure [1]. The behavior of the flow learn failure rate of a Cisco flow exporter is depicted in Figure 7. It is clear that the failure rate is higher during working hours, than during nights, since more flows will be active during those periods. Note that the flow learn failure rate can be significantly reduced by using more aggressive values for flow expiration timers, in order to clean up the flow cache sooner after a flow has finished.

Fig. 7: Flow learn failures in a flow exporter

3. Packet loss at collector. Sadre et al. analyze the behavior of a flow monitoring application - which can be a flow collector, writing flow records to a disk - in an attack situation [13]. The described DDoS attack consisted of a huge amount of SYN packets. Since each SYN packet results in a new flow record, the flow cache of a flow exporter will be filled up rapidly. The authors show that the amount of flow records stored in a flow packet (i.e. a packet carrying flow records), being constant under normal circumstances, significantly increases when the exporter becomes heavily loaded. As a consequence, a flow collector will have to save more flow records to the disk in the same amount of time. Flow packets can get lost in such situations, especially when the buffer capacity of the application is small and limited.
4. **Packet loss and retransmissions of user data.** The previous subsections discussed flow data issues introduced during the flow creation process. It is, however, also possible that user data (behavior) is causing flow data artifacts. An example of this is packet loss, resulting in user data retransmissions. Figure 8 shows a situation, in which a FIN packet - used to close a connection and expire a flow - is lost. As a consequence, a new FIN packet will be transmitted. It is assumed that the flow exporter resides somewhere between the two communicating parties and that the exporter receives all transmitted packets. The problem is that the first FIN packet (which gets finally lost) will expire the actual flow, and the second FIN packet will create a new flow which is immediately expired as well. As a result, one extra flow is generated, which was originally not expected to happen.

![Diagram of TCP packets](image)

Fig. 8: Effect of user data retransmissions on flows

5 **Remote Topology Interference**

The firewall.

6 **Effects of Sampling**

We would like to show:

- what are the techniques to *invert flows* (get the original numbers) and how good this estimation is?
- How to conclude that some requests were not served within a certain confidence interval?

Don’t forget: TCP flags not exported for hardware-switched flows.

7 **Related Work**

We will organise the related work as follows:
1. Other works using flow counters to check health of remote services:

2. Other works about the impacts of measurement artifacts on NetFlow data:
   - Paxson shows some best practices for Internet measurement and a classification of common error sources: precision, accuracy, and misconception. Our work is aligned with that classification, and shows a practical example.
   - Trammell et. al. [15] describe problems related to the precision of timestamps in NetFlow V9 data. Those problems would also affect the calculation of Bi-Flow latency, as presented in our paper.
   - Sommer and Feldmann: Challenges in creating TCP connection summary from flow data.
   - Cunha et. al.

3. There are several works about inversion of sampled flow data:
   - Duffield [4] summarises methods to invert flow data, including some notes about bias in estimating number of flows and flow duration and some notes about flow record loss. However, as we have shown, those methods would not work properly in most of real data available in practise because of the measurement artifacts.
Some new flow export solutions, like biflow export of [3], would be used in our context, for example, to automatic matching flow directly at the exporter. However, they would also be affected by sampling, and maybe by new artifacts.

8 Conclusions

1. Measurement devices cannot be trusted blindly.
2. Remote network topology must be considered when deriving health of services.
3. Packet sampling can hide important events on the network.
4. Application (traffic type) needs to be known in advance in order to fully understand your flow data.

Acknowledgment

This work has been carried out in the context of the IOP GenCom project Service Optimization and Quality (SeQual), which is supported by the Dutch Ministry of Economic Affairs, Agriculture and Innovation via its agency Agentschap NL.

References


Appendix D

TERENA TNC 2011 Poster

This appendix includes the final version of the conference poster *Ethernet Flow Monitoring with IPFIX*. The poster summarizes the findings presented in the paper in Appendix B.

<table>
<thead>
<tr>
<th>Name:</th>
<th>TERENA TNC 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Venue:</td>
<td>Prague, Czech Republic</td>
</tr>
<tr>
<td>Date:</td>
<td>May 16 - 19, 2011</td>
</tr>
<tr>
<td>Presentation date:</td>
<td>May 17 - 18, 2011</td>
</tr>
</tbody>
</table>

Table D.1: Conference information
Ethernet Flow Monitoring with IPFIX

Rick Hofstede, Idilio Drago, Anna Sperotto, Aiko Pras
University of Twente, The Netherlands
r.j.hofstede@student.utwente.nl, {i.drago, a.sperotto, a.pras}@utwente.nl

This work has been supported by SURFnet’s "GigaPort 3" project for Next-Generation Networks.

1. What is Flow Monitoring?
Flow monitoring offers an aggregated view of network activity.


Flexible NetFlow/IPFIX: allows user selection of flow keys and records.

Future: use flow monitoring in Carrier Ethernet networks?

2. IPFIX at the Ethernet Layer
Network operators are considering deployments of Carrier Ethernet networks.

Monitoring at the Ethernet layer provides an overview of active layer 2+ protocols.

Our goal: evaluate the use of IPFIX for Next-Generation Ethernet (NGE) monitoring.

3. Probing Equipment
INVEA-Tech FlowMon Probes support IP flow export using NetFlow v5/v9/IPFIX. A special Ethernet-plugin was developed for exporting Ethernet flows.

Exported fields:
Start time, end time, source MAC address, destination MAC address, EtherType, sVLAN, cVLAN, sPriority, cPriority, header length, payload length, packets, octets

Two FlowMon Probes have been installed in our University’s campus network.

4. Case 1: Misconfiguration
DECnet Phase IV traffic was found in our network. It was used in previous years for router configuration, but it was not disabled until now.

5. Case 2: Security
A misbehaving host generated a big amount of ARP traffic, which can cause serious damage to the health of the network.

Normal ARP traffic on campus network:

Erroneous ARP traffic on campus network:

6. Case 3: Profiling
Although most IPv4 campus traffic is generated during working hours, IPv6 flows behave differently:

Many small flows are generated by IPv6, especially during evenings. What will be the effect on probing equipment, when more hosts make use of IPv6?

7. Conclusions
Ethernet flow monitoring provides new insights into active traffic types in a network.

Besides profiling network activity, it can support network managers in detecting misconfigurations and security issues. Other use cases will be investigated as future work.

This work has been supported by SURFnet’s 'GigaPort 3' project for Next-Generation Networks.
Appendix E

Next-Generation Ethernet Monitoring Press Release

This appendix includes a press release, published together with INVEA-TECH, on the use of their FlowMon Probe [7] platform for Next-Generation Ethernet monitoring. The press release is based on the paper and poster in Appendix B and Appendix D, respectively.
Next-Generation Ethernet Monitoring

Enschede (NL), Brno (CZE), 7. 7. 2011

Researchers from the University of Twente monitor Ethernet networks with FlowMon Probes - a solution of INVEA-TECH - and obtained interesting and surprising results, which are currently being published at major conferences.

The Design and Analysis of Communication Systems (DACS) research group from the University of Twente (The Netherlands), lead by professor Aiko Pras, focuses on research in the area of Next-Generation Ethernet monitoring. SURFnet, the Dutch academic backbone network operator, is currently investigating the deployment of a Next-Generation Ethernet network. FlowMon Probes from the Czech university spin-off company INVEA-TECH are used for practical measurements for the new SURFnet network.

INVEA-TECH prepared a special plugin for their FlowMon Probe platform, in order to support Ethernet-layer monitoring at the University of Twente. Thanks to this customized plugin, the probes monitor traffic at the Ethernet-layer and use a modified process of flow creation (source and destination MAC addresses, VLAN ID and Ethernet type are used as key-fields, among others; non key-fields statistics provided are Ethernet header length, Ethernet payload length etc.) The measured statistics are exported to IPFIX collectors for storage and analysis.

The new type of flow data offers a completely new insight into the network. While other flow monitoring technologies, such as NetFlow, provide flow data at the IP-layer (by means of IPv4 and IPv6), Ethernet-layer flow data provides an overview of all the traffic protocols operating on top of Ethernet. Among these protocols are Address Resolution Protocol (ARP), Link-Layer Discovery Protocol (LLDP), Novell IPX, DECnet Phase IV protocols, Spanning Tree Protocol (STP) and Wake-on-LAN Magic Packets.

“The special FlowMon Probe plugin from INVEA-TECH enables us to profile network traffic, but also to detect misconfigurations and security issues, which would not have been visible at the IP-layer. We are currently publishing first results of measurements at academic conferences,” said Aiko Pras, Associate Professor at the University of Twente.

Researchers of the University of Twente are joining INVEA-TECH’s community program, which will enable them to prepare their own FlowMon Probe plugins in the future and to use the probes for upcoming research experiments. “We are glad that experts from the University of Twente are joining our community program and look forward to continuing this close and very productive cooperation,” said Petr Springl, FlowMon Product Manager. “The FlowMon community program is open to every research and academic organization focused on network monitoring and security based on flow technology.” For more information about FlowMon Community program please contact us at www.invea-tech.com/contacts.

---

About the University of Twente:

The University of Twente (UT) is located in the Netherlands, and has some 3,300 scientists and other professionals working together on cutting-edge research, innovations with real-world
relevance and inspiring education for more than 9,000 students. The Design and Analysis of Communication Systems (DACS) group has around 20 researchers, who investigate the design and implementation of dependable networked systems, as well as methods and techniques to support the design and dimensioning of such systems, such that they are dependable, in all phases of their lifecycle. Important topics within DACS are network management and measurements, with a focus on flow-based techniques for the purpose of security and configuration management.

About INVEA-TECH:

INVEA-TECH develops and markets comprehensive network solutions for networks from 10 Mbps to 100 Gbps. The core idea of the company is to provide a complete range of innovative products and services for network security, network monitoring, traffic analysis and hardware-accelerated application development. The flagship product is FlowMon – a complete NetFlow/IPFIX monitoring and security solution which enables flow monitoring in each network and provides efficient network management, increased network security (NBA - network behavior analysis), LAN & WAN & Internet monitoring, users and application monitoring, data retention law fulfillment and more. The next products are FPGA boards with 10GE interfaces, the NetCOPE platform for rapid development of network applications or NIFIC - wire-speed filter for lawful interception, and others. For more information, visit www.invea-tech.com.