Flow monitoring experiences at the ethernet-layer

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[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.

[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]. The most distinctive characteristics of IPFIX are the flexibility to change the key fields of a flow, and the possibility to include 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.

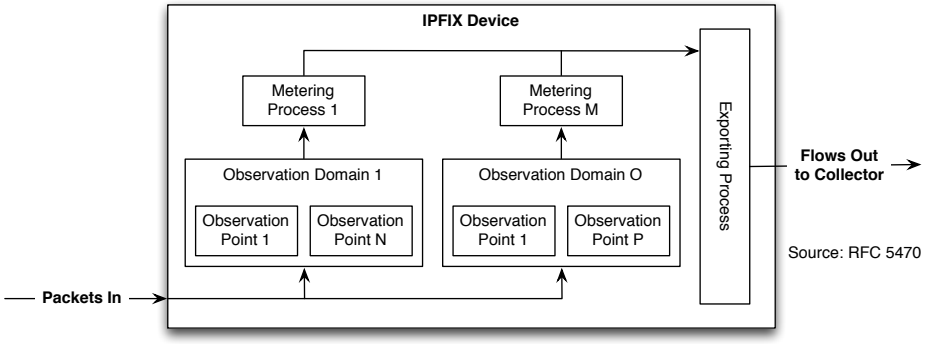


Fig. 1. *IPFIX Device* architecture

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.

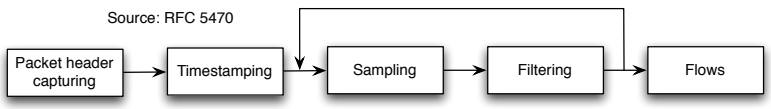


Fig. 2. IPFIX packet selection criteria

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[4]. 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].

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

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 import exceptions: they provide more insights into the customer traffic and, therefore, are essential for monitoring Ethernet transport networks.

[4]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.

srcIPv6	sourceMacAddress
dstIPv6	destinationMacAddress
srcPort	dot1qVlanId
dstPort	ethernetType
l3.proto	0 (unused)
l4.proto	0 (unused)
port.in	probe port ID

Table 2. Ethernet-plugin key fields

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

Table 3. Ethernet-plugin non-key fields

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.

Start time	Src MAC address	Type	VLAN	EHL	EPL	Frames	Bytes
2011-04-03 23:57:29.275	00:25:B3:1F:F3:0A	0x0800	161	14	56	99	11238
2011-04-03 23:57:29.529	00:0B:60:AA:80:00	0x0800	103	14	62	47	7456
2011-04-03 23:57:31.792	00:23:5A:C3:C9:7D	0x86DD	103	14	443	39	16355
2011-04-03 23:57:32.659	C8:0A:A9:F0:E3:4A	0x0806	103	14	50	16	1024
2011-04-03 23:57:34.440	00:00:0C:07:AC:00	0x0806	103	14	50	5	320

EHL - Ethernet Header Length
EPL - Ethernet Payload Length

Table 4. *nfdump* output showing Ethernet data

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.

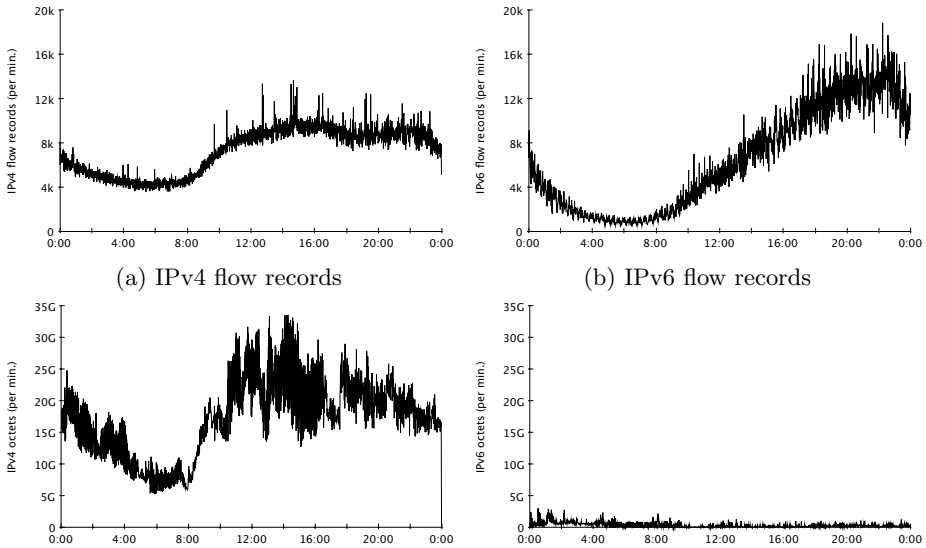
4.1 Traffic profiling

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

[5]The set of protocols operating directly on top of Ethernet, such as IP and ARP.



(a) IPv4 flow records (b) IPv6 flow records

(c) IPv4 octets (d) IPv6 octets

Fig. 3. Traffic profiling for IPv4 and IPv6

of octets generated by IPv6 (25 TB and 600GB, 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.

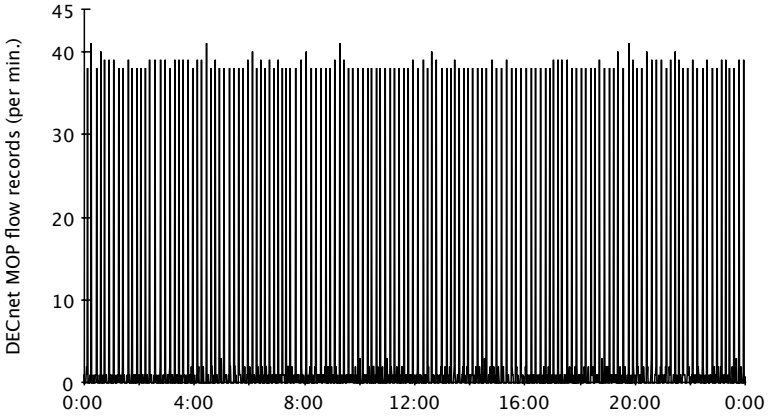


Fig. 4. DECnet Maintenance Operation Protocol (MOP) flow records

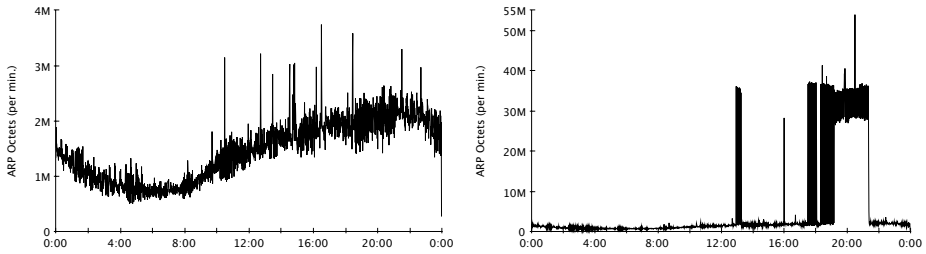
4.3 Device misbehaviour detection (1)

While profiling the network on the UT campus, two unknown *ethertypes* were detected: 0x8259 and 0x0A59. The IANA maintains a list of registered *ethertypes* [19], but the discovered *ethertypes* 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 *ethertypes* 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.



(a) ARP octets on March 28, 2011 (b) ARP octets on April 4, 2011

Fig. 5. Misbehaving host becomes security threat

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

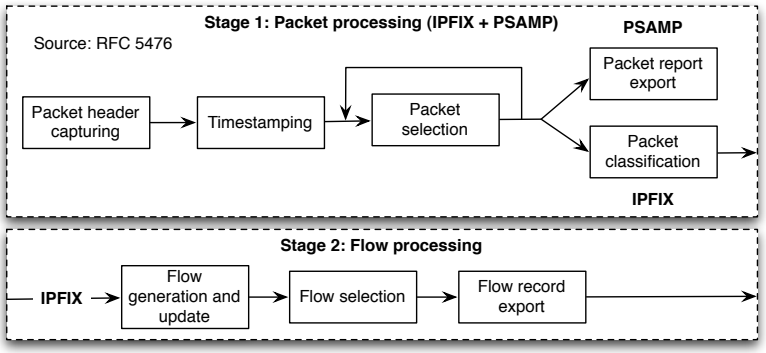


Fig. 6. IPFIX and PSAMP architectures

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

1. Brownlee, N.: Traffic Flow Measurement: Meter MIB. RFC 2720 (Informational) (October 1999), http://www.ietf.org/rfc/rfc2720.txt
2. Brownlee, N.: NeTraMet & NeMaC Reference Manual, Version 4.3 (June 1999), http://www.caida.org/tools/measurement/netramet/download/ntm43.pdf
3. Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational) (October 2004), http://www.ietf.org/rfc/rfc3954.txt
4. Claise, B.: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101 (Standards Track) (January 2008), http://www.ietf.org/rfc/rfc5101.txt
5. Claise, B., Johnson, A., Quittek, J.: Packet Sampling (PSAMP) Protocol Specifications. RFC 5476 (Standards Track) (March 2009), http://www.ietf.org/rfc/rfc5476.txt
6. Duffield, N., Lund, C., Thorup, M.: Properties and prediction of flow statistics from sampled packet streams. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. pp. 159–171. IMW '02, ACM, New York, NY, USA (2002)
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)
10. INVEA-TECH: FlowMon Probe (July 2011), http://www.invea-tech.com/products-and-services/flowmon/flowmon-probes
11. Leinen, S.: Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX). RFC 3955 (Informational) (October 2004), http://www.ietf.org/rfc/rfc3955.txt
12. NfDump: (July 2011), http://nfdump.sourceforge.net
13. Poortinga, R., van de Meent, R., Pras, A.: Analysing campus traffic using the meter-MIB. In: Proceedings of Passive and Active Measurement Workshop. pp. 192–201 (2002)
14. Quittek, J., Zseby, T., Claise, B., Zander, S.: Requirements for IP Flow Information Export (IPFIX). RFC 3917 (Informational) (October 2004), http://www.ietf.org/rfc/rfc3917.txt
15. Sadasivan, G., Brownlee, N., Claise, B., Quittek, J.: Architecture for IP Flow Information Export. RFC 5470 (Informational) (March 2009), http://www.ietf.org/rfc/rfc5470.txt
16. sFlow: Making the Network Visible (July 2011), http://www.sflow.org
17. SURFnet: GigaPort3 and SURFnet7 (July 2011), http://www.surfnet.nl/en/innovatieprogramma's/gigaport3/pages/default.aspx
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
20. The Internet Assigned Numbers Authority (IANA): IP Flow Information Export (IPFIX) Information Elements (July 2011), http://www.iana.org/assignments/ipfix/ipfix.xml
21. Trammell, B., Boschi, E.: An Introduction to IP Flow Information Export (IPFIX). IEEE Communications Magazine 49, Issue 4, 89–95 (April 2011)