A Distributed Passive Measurement Infrastructure

Patrik Arlos, Markus Fiedler, and Arne A. Nilsson

Blekinge Institute of Technology, School of Engineering,
Karlskrona, Sweden
{patrik.arlos,markus.fiedler,arne.nilsson}@bth.se

Abstract. In this paper we describe a distributed passive measurement infrastructure. Its goals are to reduce the cost and configuration effort per measurement. The infrastructure is scalable with regards to link speeds and measurement locations. A prototype is currently deployed at our university and a demo is online at http://inga.its.bth.se/projects/dpmi. The infrastructure differentiates between measurements and the analysis of measurements, this way the actual measurement equipment can focus on the practical issues of packet measurements. By using a modular approach the infrastructure can handle many different capturing devices. The infrastructure can also deal with the security and privacy aspects that might arise during measurements.

1 Introduction

Having access to relevant and up-to-date measurement data is a key issue for network analysis in order to allow for efficient Internet performance monitoring, evaluation and management. New applications keep appearing; user and protocol behaviour keep evolving; traffic mixes and characteristics are continuously changing, which implies that traffic traces may have a short span of relevance and new traces have to be collected quite regularly.

In order to give a holistic view of what is going on in the network, passive measurements have to be carried out at different places simultaneously. On this background, this paper proposes a passive measurement infrastructure, consisting of coordinated measurement points, arranged in measurement areas.

This structure allows for a efficient use of passive monitoring equipment in order to supply researchers and network managers with up-to-date and relevant data. The infrastructure is generic with regards to the capturing equipment, ranging from simple PCAP-based devices to high-end DAG cards and dedicated ASICs, in order to promote a large-scale deployment of measurement points.

The infrastructure, which currently is under deployment at our university, was designed with the following requirements in mind:

1. *Cost.* Access to measurement equipment should be shared among users, primarily for two reasons: First, as measurements get longer (for instance for detecting long-range dependent behaviour) a single measurement can tie

up a resource for days (possibly weeks). Second, high quality measurement equipment is expensive and should hence have a high rate of utilization.
2. *Ease of use.* The setup and control of measurements should be easy from the user's point of view. As the complexity of measurements grows, we should hide this complexity from the users as far as possible.
3. *Modularity.* The system should be modular, this to allow independent development of separate modules. With separate modules handling security, privacy and scalability (w.r.t. different link speeds as well as locations). Since we cannot predict all possible uses of the system, the system should be flexible to support different measurements as well as different measurement equipment.
4. *Safety and Security.* Measurement data should be distributed in a safe and secure manner, i.e. loss of measurement data should be avoided and access to the data restricted.

To solve these requirements we came up with an infrastructure consisting of three main components, *Measurement Point* (MP), *Consumer* and *Measurement Area* (MAr). The task of the MP is to do packet capturing, packet filtering, and distribute measurement data. The approach to the second design requirement was to use a system with a web interface. Through this interface users can add and remove their desired measurements. The MAr then handles the communication with the MPs. The cost for implementing this architecture is not very high, compared to a normal measurement setup you need two additional computers and an Ethernet switch of suitable speed, and this basic setup can grow as the requirements change.

There are several other monitoring and capturing systems available, here we describe only a few.

CoralReef [1] is a set of software components for passive network monitoring, it is available for many network technologies and computer architectures. The major difference between CoralReef and our infrastructure is that CoralReef does not separate the packet capturing and analysis as we do. Furthermore, the CoralReef trace format does not include location information as our does.

IPMON [2] is a general purpose measurement system for IP networks. IPMON is implemented and deployed by Sprint. IPMON separates capturing from analysis, similar to our infrastructure. On the other hand, the IPMONs store traces locally and transfer them over a dedicated link to a common data repository. The repository is then accessed by analyzers.

Gigascope [3] uses a similar approach as IPMON, by storing captured data locally at the capturer. This data is then copied, either in real time or during off-peak hour, to a data warehouse for analysis. It uses GSQL as an interface to access the data.

The IETF has (at least) two work groups that are relevant for this work; Packet Sampling (PSAMP) [4] and IP Flow Information Export (IPFIX) [5]. PSAMP works on defining a standard set of capabilities for network elements to sample subsets of packets by statistical and other methods. Recently an Internet draft was published [6], which describes a system at a higher level than our infrastructure, but they are very similar and our system could benefit by adjusting somewhat to the PSAMP notation. The IPFIX group is interesting since they deal with how to export measurement data from A to B, thus it is interesting with regards to consumers.

In Section 2 we will discuss the components and how they interact. This is followed by Section 3 where we describe how the system handles rules and filters. In Section 4 we discuss privacy and security related to the infrastructure. In Section 5 we describe two cases where the system has been deployed. In Section 6 we describe some of the ongoing and future work. And in Section 7 we conclude the paper.

2 Components

The three main components in the infrastructure will be described in the following subsections.

2.1 Measurement Point

In Figure 1 the components of a schematic MP are shown. This is the device that does the actual packet capturing. It is managed from a Measurement Area Controller (MArC) and transfers the captured data to consumers attached to the Measurement Area Network (MArN). The MP can either be a logical or a physical device. A logical MP is simply a program running on a host, whereas a physical MP could either use a dedicated computer or custom hardware in order to create high-speed high-performance MPs.

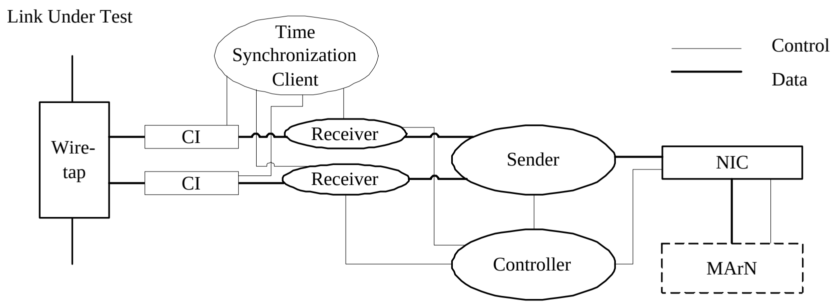


Fig. 1. Schematic overview of a MP.

A MP can tap one or more links; each link is tapped via a wiretap. For full-duplex Ethernets, a wiretap has two outputs, one for each direction. These are connected to separate capture interfaces (CI). A receiver listens to a CI and filters the packets according to the filter rules stated by the MArC. If the CI hasn't timestamped the packet the receiver will do so. The packets are then delivered to the sender, which is responsible for sending the captured packets to the appropriate consumers. Such a measurement frame can contain several packets, where the number of packets is controlled by the maximum transfer unit (MTU) of the MArN. Each MP also has a controller that is responsible for the configuration of the MP and the communication with the MArC. A time synchronization client (TSC) is used to keep all the MPs with in a MAr synchronized, which can be done using a dedicated device or a simple NTP server.

The filter rules used by the receiver specify, in addition to packet properties, a consumer and the amount of the packet to be captured (currently the upper limit is 96 bytes). For each frame that passes the filter, the MP attaches a capture header (Figure 2). In this header, we store a CI identifier, a MP identifier, a timestamp when the packet was captured (supporting an accuracy of picoseconds), the packet length, and the number of bytes that actually were captured. The filters are supplied to the MP from the MArC, and they will be discussed in Section 3. Once a packet matches a filter, it is stored in a buffer pending transmission. Once the buffer contents reaches a certain threshold the buffer is transmitted using Ethernet multicast. This way, it is simple to distribute frames to several consumers in one transmission. The duplication of data is done by the MArN. This approach will also reduce the probability of overloading the MArN, and hence preventing loss of measurement frames as far as possible. However, in order to detect frame loss each measurement frame is equipped with a sequence number that is checked by the consumer upon reception. If a measurement frame is lost it is up to the consumer to handle this particular loss and notify the MArC. Given this information the MArC can take actions to prevent future losses. Actions can be to alter filters as well as requesting additional switching resources inbetween the MPs and the Consumers. The current implementation only notifies the consumer "user", who has to take appropriate actions.

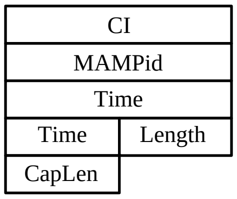


Fig. 2. Capture Header.

The capture header enables us to exactly pinpoint by which MP and on what link the frame was captured, which is vital information when trying to obtain spatial information about the network's behaviour. This also enables us to use several MPs to measure a single link, which is interesting when the measurement task of a link speed becomes too great for a single MP to handle. This would require a device that is capable of distributing the packets such that the wiretap feeds different MPs in a round robin approach.

2.2 Measurement Area

In Figure 3 an example of a MAr is shown. The MAr provides a common point of control for one or more MPs. It uses a dedicated network in between MPs and the MAr subsystems for reasons of performance and security. A MAr consists of the following subsystems: a MArC, a time synchronization device (TSD), a MArN and at least one consumer and one MP. The MArC is the central subsystem in a MA. It supplies the users with a GUI for setting up and controlling their measurements. It also manages the MPs by supplying filters and by keeping track of their status. The TSD supplies all the MPs in the MA with a common time and synchronization signal. It can utilize the existing Ethernet structure to the MPs, or it can utilize some other network to distribute the time signal.

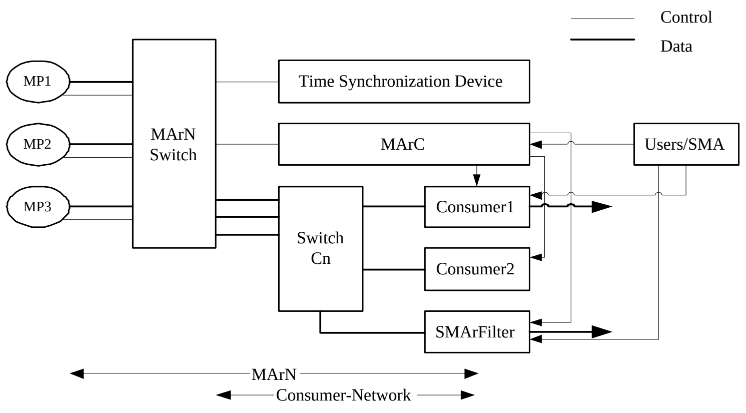


Fig. 3. Simple overview of a MA with three MPs, four consumers, one MArC and a time synchronization unit.

The capacity of the MArN should be such that it can handle the peak rate of the measured traffic. Assume that a MP monitors a 10Base-T link, with a frame rate of 800 fps where each frame is 1500 bytes long ($\approx$ 9.6 Mbps). From each frame we collect 96 bytes, add a capture header of 36 bytes and store the data in a measurement frame, see Figure 4. Given a MArN MTU of 1500, a measurement frame can contain 1480 bytes of measurement data, consisting of capture headers and frames, the remaining 20 bytes are used by a measurement header (MH). In the current example we can store 11 frames in each measurement frame ($11 * (36 + 96) = 1452 \leq 1480$ bytes), causing the MP to send only $800/11 \approx 72$ fps into the MArN, see Figure 5. If the monitored link would have a frame rate of 14000 fps, each frame would only be 85 bytes long ($\approx$ 9.6 Mbps), the measurement frame would contain 12 frames ($12*(36+85) = 1452 \leq 1480$ bytes), yielding a frame rate of $14000/12 \approx 1167$ fps. However, if the MArN MTU was 9000, the measurement frame could contain 74 frames, yielding a frame rate of 189 fps.

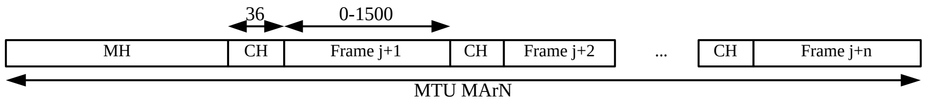


Fig. 4. Measurement frame encapsulation.

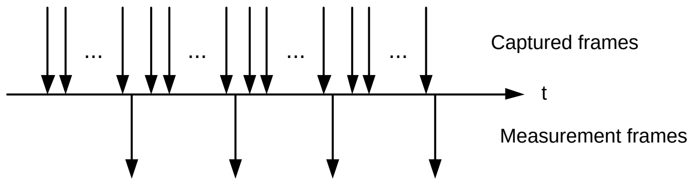


Fig. 5. After capturing N frames one measurement frame is sent from the MP.

A consumer that attaches to the MArN should not request more data than the link that it is attached to can handle. For instance a consumer C1 is the recipient of two measurement streams, S1 and S2, each generating 1272 measurement frames per second. As long as the total frame rate of S1 and S2 is less or equal to the capacity offered by link and switch there should be no problems, but if the consumer desires to get full frames it might run into problems quite fast, since the MP adds a capture header to each captured frame potentially generating more traffic than it captures. The current implementation addresses this problem by having a maximum capture size of 96 bytes. The MArC also provides the user with an estimation of the frame rate on the links that the MPs are monitoring, giving the user an indication of the amount of traffic that his consumer might receive.

The example in Figure 3 contains a consumer network (CN). It is placed on a separate switch to minimize processing required by the MArN, thus enabling additional consumers to be easily connected to the MArN, for instance new probes, analyzers etc. to be evaluated in parallel. If the number of consumers is low, the MArN switch might handle them directly, and no CN switch is necessary. This would be the normal setup, see Figure 6. In Figure 7 a minimal MAr is shown. In both cases the MPs are using a separate network for the time signal distribution.

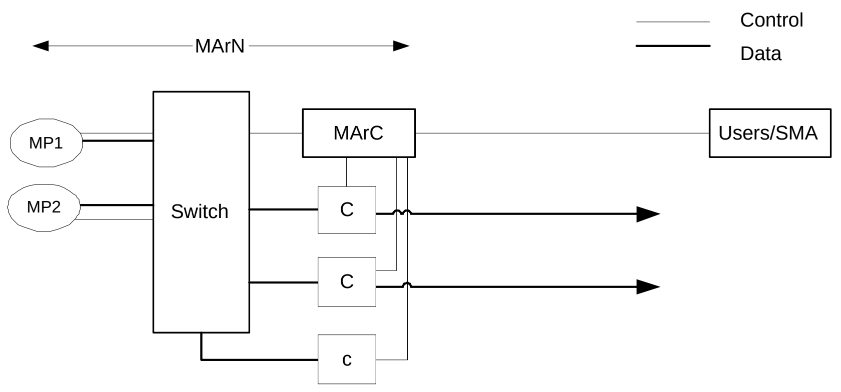


Fig. 6. Normal MAr

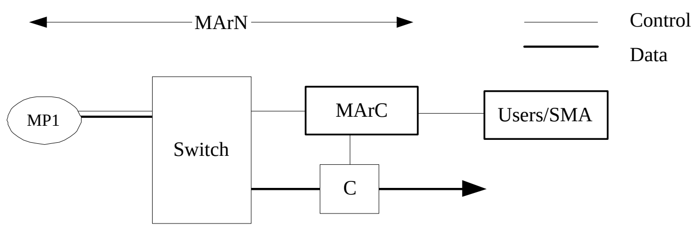


Fig. 7. Minimal MAr

2.3 Consumer

A consumer is a user-controlled device that accepts packets according to the format specified by the system. A consumer should filter the content of the measurement frame that it receives, since the MP merges multiple user requests some filters will capture packets that match several requests. Such a joint filter might not perfectly match the desired frame description; this is discussed in the following section.

3 Filters and Rules

A user supplies rules to the MArC. These rules describe *what* data the user desires to collect, *where* the data should be collected, *when* the data should be collected and *where to send* the data. The MArC uses this information to create filters that the MPs understand. The filters that the MP uses are a combination of all the user supplied rules, combined in such a manner that all requests are met in a best effort style. The MArC keeps track of the MPs and their capabilities, thus it knows how many filters a MP can handle before it runs into performance problems. The MArC also monitors the performance of the MArN and reject user rules that could cause performance problems within the MArN. If a MP is to obtain a filter list that would push it into a region of potential performance problems, the MArC will alter the filters in order to minimize the number of filters. By doing this the load on the MP is kept at a reasonable level, but this approach requires the consumers to do some filtering of their own. Hence, it is up to the user to supply the desired Consumer with a filter. The filters within a MP are arranged in such a manner that no packet is reported twice by the MP.

Let's give a simple example, we have one MP and two consumers C1 and C2. Initially we have two rules (using BPF syntax):

R1 {*tcp host A.a*} which sends its data to C1.
R2 {*ip net A*} which targets C2

Here two approaches are possible; the first during low load would have the following filters sent to the MP:

F1 {*tcp host A.a*}→ M1
F2 {*ip net A*}→ C2

Here M1 is a multicast address that C1 and C2 listens to. If the load on the MP approaches a high level then only one filter would be sent to the MP

F1 {*ip net A*}→ M1

In this case the C1 consumer would need to perform filtering in order to select the TCP segments of host A.a. By default a consumer should always filter the measurement data that it receives, ensuring that it passes a correct stream to the analysis/storage entity.

4 Privacy & Security Issues

A MP will see all the traffic passing on a link that it is tapping, which can be viewed as a intrusion of privacy. Furthermore, since the majority of the network protocols used today were not designed with security in mind, user credentials might pass on the link and be clearly visible to the MP. This can be an intrusion of privacy and should require special care on behalf of the measurement system and its users. If the data collected from the system is only intended for internal use, it might be enough that all users and the network-owner have agreed to that their traffic can be monitored to allow for measurements. However, if the data is to be shared with researchers in other organizations, the data should be deprivatized. Deprivatization [7] can be done on various levels, from the removal of parts in the application data to the removal of all network data. We believe that the system should minimize the alternation of the captured data and leave the anonymization to the consumers. If the MP would anonymize the data, e.g. through scrambling of addresses [8], some consumers such as intrusion detection systems or charging systems might not be able to operate anymore. However, if the system does deprivatization by default, this should be done in the MPs. If address scrambling is utilized, this causes problems when the user specifies the measurement rules. If the unscrambled address was used, the user will obtain scrambled addresses matching his requirement and then it is possible to reverse-engineer the scrambling system. If the scrambled address was used, the user would need to know how to create that scrambled address. Probably, the first method should be chosen. In that case, the only person that is capable of reverse-engineering the packet trace is the user requesting the trace, since he knows both scrambled and unscrambled address. Now, if the packet trace is stolen, the thief cannot match packets to individual hosts/users unless he has access to a descrambler and the scrambling key.

Privacy issues will probably have to be addressed by specialized consumers. For instance, we have two consumers, a intrusion detection system (IDS) and a link utilization estimator (LUE). The IDS needs undistorted information. The LUE could on the other hand use deprivatized data, but since the MP will not send two copies of the same packet there is a problem. It is probable that a network owner would like to have control of the information that leaves his network, so it would be easier for the network owner to supply an export consumer that deprivatizes the data according to his own policies, which might not meet the particular desires of the user. For our own measurements, the agreement we made with the system owner was the following: The MPs are only allowed to capture headers, not user payload. Furthermore, the data leaving a consumer may only be in statistical form, or deprivatized in such a manner that it is impossible to reverse-engineer the data to obtain information that allows you to identify a particular individual.

From a security point of view, all components in the system should be protected from unauthorized access. The simplest way to do this is to have the system operating on a separate network, with no connection to any other networks. This would however be expensive and unpractical in measurements distributed over a wide area. The solution to this it to utilize Super Measurement Areas (SMAr), see Figure 8. SMAr's are used to connect to MAr's at different locations using existing infrastructure. A SMAr can be seen as a MAr at a higher level, the MAr's MP becomes SMArFilters (specialized consumers that attach to the MArN), the MArs consumers are called SMArConsumers. Between the SMArFilters and SMArConsumers TCP is used to provide reliable communication. The MPs and the MArN need to be protected from unauthorized access, both physical and logically. Physical protection of the MAr subsystems is the first requirement in giving logical protection; the consumers and the MArC need to be protected from intrusions via their connection to the users.

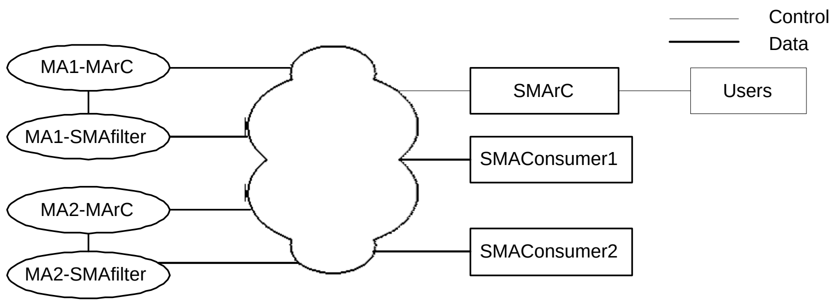


Fig. 8. Example of a SMAr.

5 Examples of Use

As of writing two MAr have been implemented and used. One is available online via http://inga.its.bth.se/projects/dpmi and is mainly used in a controlled environment. The second MAr consisted of two measurement points each monitoring a gigabit link on a campus network. In both cases only one physical consumer was used, but it was sufficient to handle up to eight logical consumers. Examples of consumers are: estimation of traffic distribution (at link, network, transport and application level); link utilization; packet inter arrival time; communication identification; and bottleneck identification [9]. At the time of writing we are preparing a third MAr to be deployed in an ISP network, where it will initially be used for bottleneck identification. In Figure 9 we visualize the result from a analyzer that identifies bottlenecks. It uses two consumers to estimate the link bit rate over a given time intervall, these are then transferred to a database which is accessed by the visualizer that estimates the bottleneck.

In Figure 10 the MArC (prototype) interface for adding a rule is shown. In this implementation all tasks are done manually, the goal was to develop the MP not the MArC. The following filtering options are availible, the MASK fields are used to mask the packet value.

- CI: Physical interface identifier.
- VLAN_TCI: VLAN number and priority.
- ETH_TYPE: Ethernet type.
- ETH_SRC/DST: Ethernet source/destination address.
- IP_PROTO: IP payload type.
- IP_SRC/DST: IP source/destination address.

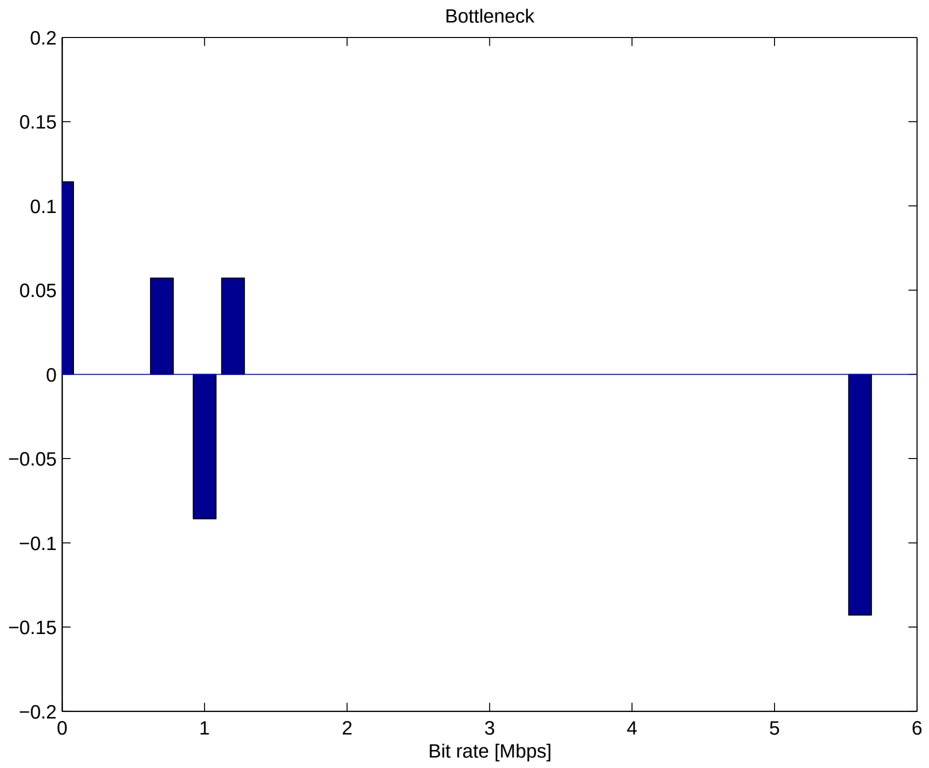
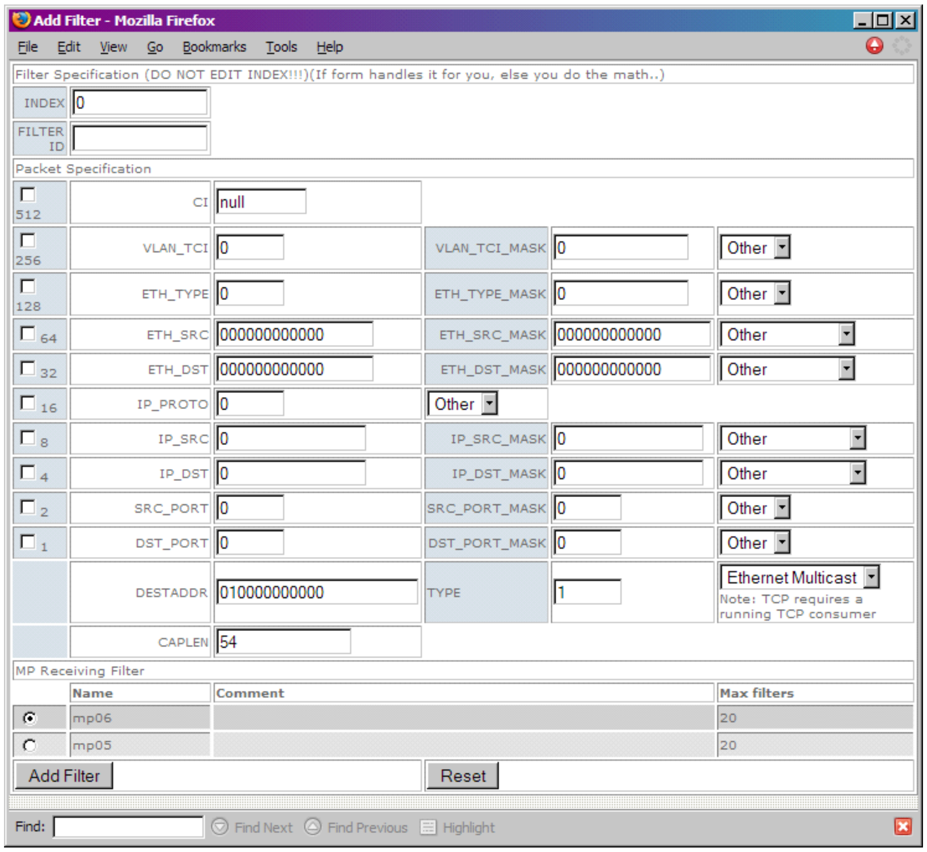


Fig. 9. Example of a consumer: Visualization of a bottleneck through bitrate histogram difference plots (c.f. [9]).

- SRC/DST_PORT: Transport protocol source/destination port numbers (if applicable).
- DESTADDR: What Ethernet address should receive the measurement data?
- TYPE: Which type of transport should the MP use? Ethernet, UDP or TCP.
- CAPLEN: How much of each captured frame should we store?

FilterID is a number that specifies in which order the MP should check its filters, starting with number zero. Index will indicate which fields that are used in the rule specification. For instance if we wish to collect all packets caught on a specific CI the index would be 512, and the CI field would hold a string identifying the CI. If we would like to capture IP packets caught on a specific CI, index would be 640, ETH_TYPE=2048 and CI a string specifying the interface.



Filter Specification (DO NOT EDIT INDEX!!!)(If form handles it for you, else you do the math..)

INDEX	0
FILTER ID	

Packet Specification

☐ 512	CI	null			
☐ 256	VLAN_TCI	0	VLAN_TCI_MASK	0	Other
☐ 128	ETH_TYPE	0	ETH_TYPE_MASK	0	Other
☐ 64	ETH_SRC	000000000000	ETH_SRC_MASK	000000000000	Other
☐ 32	ETH_DST	000000000000	ETH_DST_MASK	000000000000	Other
☐ 16	IP_PROTO	0	Other		
☐ 8	IP_SRC	0	IP_SRC_MASK	0	Other
☐ 4	IP_DST	0	IP_DST_MASK	0	Other
☐ 2	SRC_PORT	0	SRC_PORT_MASK	0	Other
☐ 1	DST_PORT	0	DST_PORT_MASK	0	Other
	DESTADDR	010000000000	TYPE	1	Ethernet Multicast Note: TCP requires a running TCP consumer
	CAPLEN	54			

MP Receiving Filter

	Name	Comment	Max filters
●	mp06		20
○	mp05		20

Fig. 10. User interface for adding rules.

6 Ongoing and Future Work

Initial experiences with the system are encouraging, and development of consumers is currently ongoing. The experience of the demo has indicated that the MP's software needs to be changed in such a manner that the MPs periodically flush their measurement buffers, in order to prevent consumers from waiting long times. We are considering a modification of the system so that the MArC supplies the consumers automatically with the information that they need with regards to filters and multicast addresses.

To handle the increased link speeds, new devices with better timestamping accuracy are needed. Even if we can obtain this accuracy, a single device will probably run into problems when measuring such a link. Hence another task would be to investigate how to distribute the measurement task of a link onto several MPs. Compression of frame data is also considered to be implemented, this would could enable us to do full frame capturing without requiring a MArN that is more powerful that the observed link. We also need to evaluate the performance of a MArN.

The infrastructure is being considered as a part of the EuroNGI WP.JRA.4.3 [10] Measurement tool. This tool will support traffic generation, measurement, analysis and visualization.

7 Conclusions

In this paper we have presented a distributed passive measurement infrastructure, which has separate components for packet capturing, control and analysis. We discussed how the system deals with multiple users and their request for data. Since the infrastructure is passive we addressed the security and privacy issues associated with this. Furthermore, we gave examples of current usage and future work.

References

1. CAIDA: CoralReef. (2005) http://www.caida.org/tools/measurement/coralreef (Verfied in January 2005).
2. Sprint: IPMON (2005) http://ipmon.sprint.com (Verified in January 2005).
3. AT&T: Gigascope (2005) http://www.research.att.com/info/Projects/Gigascope (Verified in January 2005).
4. IETF: PSAMP Workgroup. (2005) http://www.ietf.org/html.charters/psamp-charter.html (Verfied in January 2005).
5. IETF: IPFIX Workgroup. (2005) http://www.ietf.org/html.charters/ipfix-charter.html (Verfied in January 2005).
6. IETF: A Framework for Packet Selection and Reporting. (2005) http://www.ietf.org/internet-drafts/draft-ietf-psamp-framework-10.txt (Verfied in January 2005).

7. Pang, R., Paxson, V.: A high-level programming environment for packet trace anonymization and transformation. In: SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, ACM Press (2003) 339–351
8. Xu, J., Fan, J., Ammar, M., Moon, S.B.: On the design and performance of prefix-preserving ip traffic trace anonymization. In: IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, ACM Press (2001) 263–266
9. Fiedler, M., Tutschku, K., Carlsson, P., Nilsson, A.A.: Identification of performance degradation in ip networks using throughput statistic. In: Proceedings of the 18th nternational Teletraffic Congress (ITC-18), ELSEVIER (2003) 399–408
10. EuroNGI: Homepage (2005) http://www.eurongi.org (Verified in January 2005).
11. TCPDUMP Public Repository: Homepage. (2005) http://www.tcpdump.org (Verified in January 2005).
12. Endace Measurement Systems: Homepage. (2005) http://www.endace.com (Verified in January 2005).