Measuring Platform Architecture Based on the IPFIX Standard

Alžbeta Kleinová, Anton Baláž, Jana Trelová, Norbert Ádám

Department of Computers and Informatics, Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
E-mail: Alzbeta.Kleinova@tuke.sk, Anton.Balaz@tuke.sk, Jana.Trelova@tuke.sk, Norbert.Adam@tuke.sk

Abstract: In the present network monitoring is very important particularly by the network protection and security. It is necessary to know a lot about accessible monitoring tools and know principles, protocols, recommendations and standards, on which monitoring tools are based. Measuring tool BasicMeter was also based on this knowledge.

Keywords: IPFIX, NetFlow, flow record, template, passive measurement

1 Introduction

Measuring tool BasicMeter is proposed within the Computer Network Laboratory on the Department of Computers and Informatics of Faculty Electrotechnic and Informatics Technical University of Košice. Proposal is concentrated on the tool's evolution, which would realize passive methods of the measurement. Measured results are possible to use by network parameter's evaluation for example from network security point of view. Measuring tool was based on the IPFIX standard foundation, from which comes out and is realized on the specification of the NetFlow version 9 protocol. One of aims is gradually introduced this measuring tool into real traffic and eliminate its shortages.

Because whole philosophy of measuring tool BasicMeter is based on passive (non-intrusive) measurements, in the article is described principle, advantages and disadvantages of this kind of measurement. This article is also devoted to the standard IPFIX, on which measuring tool is based. Further in this article are analyzed individual components of the IPFIX standard architecture in detail. The article is devoted to proposal alone and concentrates also on the analysis of measuring platform architecture how it is proposed within the research on Department of Computers and Informatics. Whole research is based on Cisco technologies.

2 Passive Measurement

Passive (non-intrusive) measurement which measuring the network characteristics and the QoS (Quality of Services) parameters, is one of three measuring methods (active, passive, semi-active) which are specialized in measuring the QoS parameters in the network. The passive measurements are executed only on the ground of the real traffic in the network. Results are so good interpreted and usable in the praxis. The elements of the passive measurement in the network are not loaded with the subsidiary traffic. Any traffic test is sending. This is an advantage of the passive measurements. As there is any subsidiary traffic, so there is not the possibility of the influence of the measurement results [3].

The passive measurements are uncontrolled experiments because they are created any specific traffic test. This is a disadvantage of the passive measurements. The given traffic is not influenced and the controlled data can not be transferred. Therefore it is needed to have the subsidiary control traffic for the transit of the measurement result. When the time characteristics (one-way delay, jitter) are measured, the external synchronization is needed in the measuring points.

3 IPFIX Standard

In the present different standards, recommendation and protocols exists but each of them focuses on another area. For example the IPPM standard defines parameters which are measured within the BasicMeter project on Department of Computers and Informatics as well as individual measurement progresses. The IPFIX standard deals with measuring platform mainly on the lowest layer level as measuring process.

IPFIX (Internet Protocol Flow Information eXport) is a standard, which was developed by the IETF (Internet Engineering Task Force) [10]. The IETF chose for the IPFIX standard Netlow version 9 protocol. IPFIX is a foundation for standardized IP data flow export. This standard will make easier for network administrators to gain the information, which they need to control their network.

Evolution of the IPFIX standard corresponds to the evolution of version 9 for NetFlow protocol. NetFlow as well IPFIX can export information almost of any data type and so it will be simplified to monitor the applications, such as multicasting [10].

IPFIX defines format by which IP flow information can be transferred from exporter to the collector. Applications supporting IPFIX make to represent statistics accepted from any router supporting the IPFIX standard.

IPFIX is a format for data export based on the templates. It is suitable for the network administrators to use the templates because the administrators need not to modify their software to support a new format always when the administrators decide to see their traffic statistics. [2]

Fig. 1 shows the IPFIX standard architecture. Asterisk (*) marks these parts which are not components of the IPFIX standard. The interfaces drawn by the dashed line are defined by the IPFIX standard; the interfaces drawn by the full line are not defined.

Next part of this chapter describes IPFIX standard in detail.

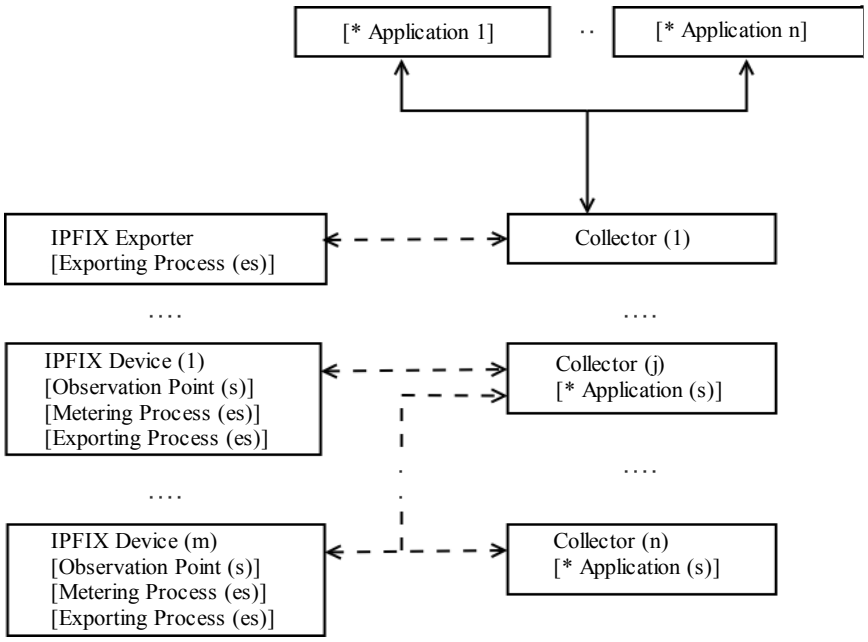

Figure 1
IPFIX architecture

Collector

The collector receives flow records from one or more exporters. It can modify or save received flow record. The collector is the subsystem in the mutual interaction whit one or more IPFIX devices. According to the definition of the IPFIX standard collector is a device, a network element, on which is running the collecting process. The functions of the collector can include [4]:

- identifying, accepting and decoding the export packets from the exporting process and from the observation domain,
- running the IPFIX protocol,

- storing the control information and the flow records receiving from the IPFIX device,
- announcing state and problems to the IPFIX device.

The collector receives the template definition from the exporter before accepting flow records. Flow records can be decoding and locally storing on the devices. In the case, that template definition was not accepted in time of flow record acceptance the collector should hold flow record for the later decoding until the template definition will be accepted.

The collector must not suggest that data FlowSet and attached template ID are exported in the same export packet.

The collector must not suggest that only one template FlowSet is included in the export packet. In the rarely case export packet can contain several template.

The templates exist only given time interval. The template lifetime should be subtracted on the collector on the time foundation where the last template FlowSet was accepted from the exporter. The collector must not try decoding flow records with the expired validity of the template. The collector should hold such a record: <exporter, exporter interface, template ID, template definition, recent acceptance>.

If a new template definition is accepted (e.g. in the case of the exporter restart), the existing definition should be immediately substituted.

Observation Point

The observation point is the position in the network where the IP packets can be observed. It can be a link to which a probe is connected, a shared medium e.g. LAN based on the Ethernet, a simple port of the router or a set of the interfaces on the router.

Metering Process

The metering process generates flow records. Input to the metering process are headers of the packets monitoring in the observation point. The metering process consists of a set of the functions including flow records about their capturing, timestamping, sampling, classifying and maintaining header. [4]

Exporting Process

The exporting process sends flow records to one or more collecting processes. Flow records are generated by one or more metering processes. The exporting process must be capable of providing the following information about each measured flow:

- the version number of the Internet protocol,
- the source IP address,

- the destination IP address,
- the IP protocol type (TCP,UDP,ICMP,...),
- in the case of the protocol type TCP or UDP – the source port,
- in the case of the protocol type TCP or UDP – the destination port,
- the packet counter,
- the byte counter,
- the syllable of the service type (Type of Service – ToS),
- in the case of IP version 6 – flow label,
- in the case of the MultiProtocol Label Switching – the first label,
- timestamp of the first packet,
- timestamp of the last packet,
- unambiguous identifier of the observation point,
- unambiguous identifier of the exporting process.

Collecting Process

The collecting process receives flow records from one or more exporting processes. The exporting process can execute further processing of flow records [1].

The collecting process should receive flow records without connection with template records. If the template records were not received in the time of receiving data records, the collecting process should store data records on the short period of the time interval and decode them after receiving template records. The time interval of the storing data records must be shorter than the template lifetime.

The template lifetime of the collecting process is limited to fixed refresh timeout. The collecting process must be connected with the lifetime of each receiving template via UDP protocol. The templates which are not renewed by the exporting process in the timeout are expired in the collecting process. If the template is not renewed by exporting process before expiring timeout, the collecting process must discard the template and also all current and succeeding data records connecting with this template.

In every time the collecting process should hold the following format for all current template records and template option records[7]: <exporting process, source ID of the observation domain, template ID, template definition, recent acceptance>.

Data network with IP traffic consists of IP flows passing through the network elements. Thus the IPFIX collecting process should be capable of receiving flow information passing through the several network elements in data network. This

required uniformity in the methods representing flow information and the way of the flow communication from the network elements to the collecting point. The IPFIX protocol provides the access to the IP flow information.

Observation Domain

The set of the observation points which is the biggest set of the flow information on the IPFIX device is called an observation domain. The observation domain provides unique ID to the collector for the identification of the export packets generating by the collector.

The observation domain can be connected with the same exporting process. For example the observation domain can be router line-card composed of more interfaces where each interface represents observation point. [6]

4 Proposal of the Measuring Platform Architecture

This chapter gives the proposal of the measuring platform architecture so as it was realized within the laboratory terms on the Department of Computers and Informatics. The measuring tool comes out from the proposal of the IPFIX standard architecture. According to this standard the measuring platform architecture consists of three components. Sometimes the SQL database is drawn to architecture too but the database does not express the concept of the measuring platform according to the IPFIX standard.

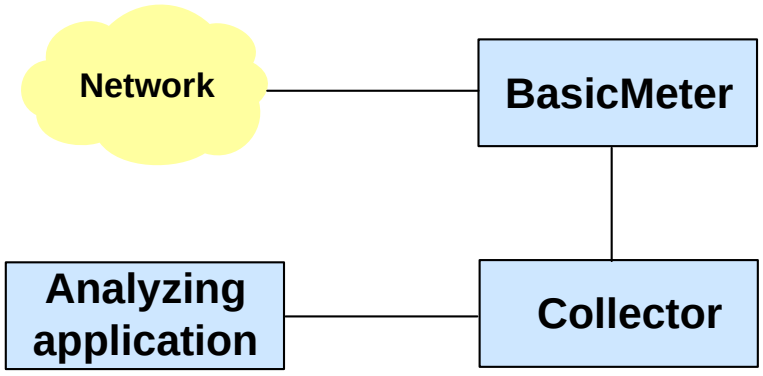

Figure 2
Measuring platform architecture

The measuring platform architecture consists of these components:

- **BasicMeter** is an application serving as a metering process. It is given for the packet capturing. It creates data for the collecting process too.

- **Collector** is the collecting process. It is used on processing of the export packets from the exporting process.
- **Analyzing application** has an access to the exported data. It also creates graphic or statistic analyze on the user demand.

Fig. 2 shows the measuring platform architecture according to the IPFIX standard.

The title measuring platform includes whole measuring platform architecture (Fig. 2), measuring tool architecture and concept of the experiment. Architecture of the measuring tool BasicMeter consists of these parts:

- packet capturing and sampling,
- classification,
- configuration,
- network communication,
- exporting process,
- NetFlow 9 metrics.

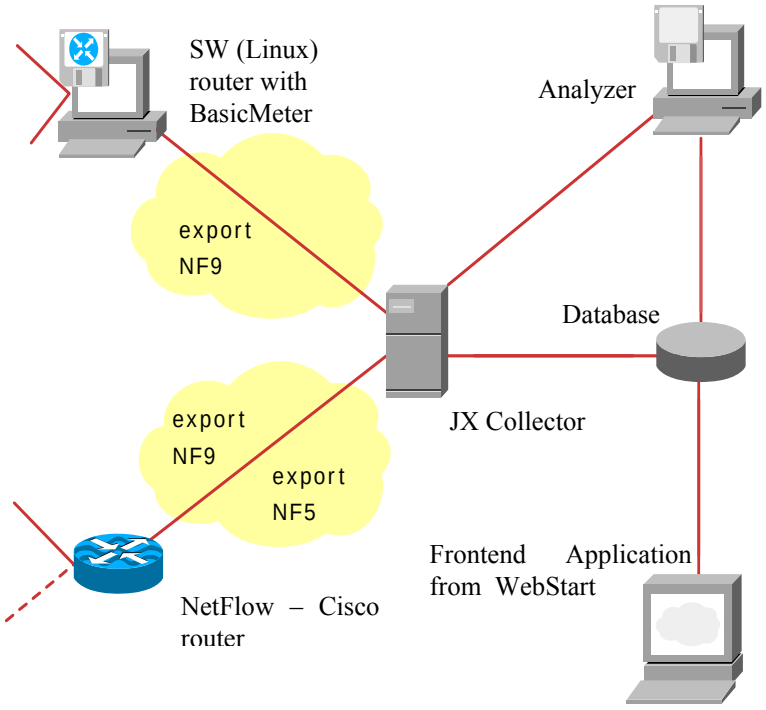

Figure 3
Model of the measuring tool

All these parts are in the evolution phase of the BasicMeter project in the Computer Network Laboratory on the Department of Computers and Informatics. The concept of the experiment (Fig. 3) serving on the simulation of the measuring tool BasicMeter was proposed in this laboratory terms.

4.1 Implementation and Functionality of the Measuring Architecture

Within the meaning of proposal of the measuring platform architecture and on the basis of the research in the Computer Network Laboratory on the Department of Computers and Informatics the following part of the article deals with the implementation and functionality of the particular components of the measuring architecture and describes the reciprocal relations among these components.

BasicMeter Information

An application is designed as a standalone command line program. The C++ language was chosen as the main implementation language because of its wide spread on the Unix OS's and the platform independency. With the respect for the maximum program portability no nonstandard functions of the OS core will be used.

The application is logically divided into two parts:

- packet capturing part,
- exporting part (exporting process).

The packet capturing part uses for this purpose well known libpcap library. This library was chosen because of its wide support on different operating systems (OS). The main function of this library is packet capturing but it is also possible use this library on the work with an easy packet filter included in almost every Unix OS's cores.

The exporting process is an independent part of that program. It processes data captured by the capturing part. This part was designed so it can support NetFlow version 9 with the option on the creating template. The export packets transport is realized by UDP protocol. This proposal is verifying in the laboratory terms within the research on the Department of Computers and Informatics.

JXColl Information

An abbreviation JXColl stands for Java XML Collector. It is software taking care of the flow collection (data flow information, Netflow 5/9) from one or several exporters. Its main purpose is to prepare data for the next evaluation and presentation by the front-end application (analyzer). It is able to process data from the several exporters emitting UDP packets with the flows.

It performs some simple actions with the content of the flows and prepares the data for storing in the archives or sends them directly to the analyzer for immediate output to user. So far it was tested on the free DBMS (Database Management Systems) such as PostgreSql and MySql but with the certain problems. The database design and the low level of the data distribution contributed to the weak performance which resulted in long lasting requests for data. So the analyzer needed some time to produce the outputs in the acceptable time. Therefore the direct connection was implemented between the analyzer and the collector. The collector optimizes queries and sends only data which are needed for the given measuring method.

JXColl is developed as the Java console application; XML stands for an option of recognizing and using the contents of incoming packets using XML description.

Analyzer Information

The main purpose is to provide the interface and the outputs (graphs, statistics, etc.) for a user. It is also developed in Java for the user needs as an application elaborated using Java Web Start technology.

The analyzer can make the databases of the flows available and can connect directly to the collecting process. Currently it supports PostgreSql and MySql databases. Security is verifying by the connection; either the database or the collector requires valid login and password. Analyzer presents the data flow for the given method in a graphical form either as static output (traffic history analyze) or as a pseudo real-time measurement.

The output can be filtered on the foundation of the several criterions (IP address, port number and time interval with the flexible timestamp). In this point the methods of the measurement are used. Currently the analyzer completely utilizes the methods for the bandwidth monitoring and the number of the packets. The work on packet length, port usage, one-way delay, jitter and packet loss is in the progress.

It could trigger an event based on the actions such as notification, log, ... In the principle all other parts of such a subsystem only prepare data for an intelligent application, which is ready to process them and produce the answers or the outputs.

Database Information

The database is a set of data that relate together in some manner. Any kind of database (file, database working with file system, reserved disk partition for storing unformatted data and so on) is supported for storing a set of data - export packets.

Although SQL database is not a part of the IPFIX specification or the NetFlow version 9 protocol it was selected as a database store because of its simple use and

good possibilities of the further manipulation with stored data and for the good possibilities of getting stored data.

In the following part the reciprocal relations among these components are mentioned.

Metering Process

The metering process consists of a set of the functions including packet header capturing, timestamping, sampling, classification and flow records messages. This progress is the same for every packet passing through the given router.

If the time limit expires, the flows are exported. Only these flows are exported which are considered for the expired flows. Together with the flow export must be exported the template too. On the foundation of the exported template the collecting process can decode data which arrives to the collecting process (in other case it throws away the data). [5]

Collecting Process and Collecting into Database

The collector collects data from one or several exporters, preprocesses and archives them in the database (SQL queries on the database).

Collecting Process and Direct Connection

For the pseudo real-time measurements the connection to the database is avoided and it is created the direct connection to the analyzer to which are sent only data necessary for evaluating the given measurement (on the ground of the agreement between the collector and the analyzer).

Analyzer and Database

The analyzer chooses from the database needful data by using SQL queries for the presentation of the measurement results.

Conclusion

This article deals with the proposal of the measuring platform architecture based on the IPFIX standard. Measuring tool BasicMeter was proposed within the measuring platform. This tool is verifying in the Computer Network Laboratory on the Department of Computers and Informatics of Faculty Electrotechnic and Informatics Technical University of Košice.

The aspects of the security concern with the communication among the particular parts of this tool. If it is used the UDP protocol (according to the NetFlow protocol), it should exist the separate measuring circle. It is supposed, that instead of the transport UDP protocol (unreliable, has the smallest transfer requirements) among the particular parts of the tool should be used the TCP protocol (in conjunction with SSL) or the SCTP protocol (according to the IPFIX standard).

The general aim is everything solve so that the final version of the proposed measuring tool was the smallest from the aspect of the processor performance and the network alone. In the future the research should focus on the implementation of the IPFIX protocol as the alternative to the NetFlow protocol.

References

[1] Jakab, F.: Tvorba sieťových prostredí pre televzdelávanie (Metódy optimalizácie hodnotenia a riadenia v počítačových sieťach – meranie a vyhodnocovanie prevádzkových parametrov v počítačových sieťach), Košice, 2004

[2] Kohler, P. - Claise, B.: IPFIX fine-tunes traffic analysis, [online], August 11, 2003, Accessible on the internet: <http://www.nwfusion.com/news/tech/2003/0811techupdate.html>

[3] Koščo, Ľ., Jakab, R., Giertl, J., and F. Jakab: Contribution to Implementation of New Standards of Measurement Architecture: Protocol for Direct Connection Between Capturing and Analyzing Application, Proceedings of the 4th International Conference on Emerging e-learning Technologies and Applications (ICETA 2005), Elfa s.r.o., Košice, Slovakia, September 13-14, 2005, pp. 187-193, ISBN 80-8086-016-6

[4] Norseth, K.: Architecture Model for IP Flow Information Export, Draft-ietf-ipfix-architecture-02.txt, [online], December 2002

[5] P ankaj, G. - Mckeown, N.: Packet Classification on Multiple Fields, Proc. Sigcomm, Computer Communication Review, Harvard University, Vol. 29, No. 4, September 1999, pp 147-160

[6] Quittek, J. - Zseby, T. - Clasise, B. - Zander, S.: Requirements for IP Flow Information Export (IPFIX), RFC 3917, [online], October 2004

[7] Sadasivan, G. et al.: Architecture for IP Flow Information Export, Draft-ietf-ipfix-architecture-07, [online], March 2005

[8] Vokorokos, L. - Jelšina, M.: Počítače: základy technických prostriedkov, Mercury - Smékal, s.r.o., Košice, 2004, ISBN 80-89061-90-7

[9] Vokorokos, L.: Digital Computer Principles, 1.vydanie, Typotex Publish House, Budapest, 2004, ISBN 9639548 09 X

[10] Waltner, Ch.: Cisco's NetFlow Selected as Basis for IETF Standard, IP data flow export technology cornerstone to network management, [online], 2005, Accessible on the internet: <http://newsroom.cisco.com/dlls/innovators/software_standards/idw_052003.html>

Supported by VEGA project No. 1/1064/04