ANALYSIS AND CONFORMITY ADAPTATION PROPOSAL OF MEASURING DEVICE BASICMETER WITH IPFIX AND PSAMP STANDARDS[1]

Miroslav POTOCKÝ, František JAKAB
Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics,
Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel.: + 421 55 602 2553,
E-mail: Miroslav.Potocky@cnl.tuke.sk, Jakab@elfa.sk

SUMMARY

The present document aims to explore the possibilities of introduction of the IPFIX (IP Flow Information eXport) and PSAMP (Packet Sampling) standards at implementation of a non-intrusive device for monitoring the QoS parameters in high speed networks. Since no implementation of such device, fully conform with both standards, currently exists, this paper provides the basic points for its development. It analyzes in detail all requirements related to documents released by the task groups that deal with the proposals of these standards. Furthermore, it suggests possible solutions of problems related to their implementation into the measuring device itself, as well as possible modifications of these standards in future versions, with aim to simplify the development of further IPFIX and PSAMP based applications. The requirements of these standards are analyzed step-by-step, in the same way as they are found in the documents of their respective task groups.

The majority of solutions presented in the paper is, or will be, included in the measuring device which is being developed in the Computer Network Laboratory of the Department of Computers and Informatics, at the Technical University, Kosice. At the same time, the majority of important aspects of these standards is assessed in connection to this tool, and conclusions related to their implementation and further development are drawn.

Keywords: *IPFIX, PSAMP, conformity, implementation, QoS, computer network*

1. INTRODUCTION

There are many IP flows export systems used to various tasks related to analysis and measurement of network traffic attributes. However, these systems are mutually incompatible and usually it is impossible to use more of them in one environment in a cooperative way. Therefore, a need to develop a standard has emerged, by which the network devices would export information about flows in a format that would be understood by all external systems, e.g. statistical tools, accounting systems, QoS meters, and network management tools.

The emerging standard for the IP flows export is called IPFIX (IP Flow Information Export) and is being developed by an IETF (Internet Engineering Task Force) working group. The documents describing the new standard in detail that have been created within this group, are the subject of the present analysis.

Beside the analysis of the requirements for the proposed IPFIX standard the PSAMP task group is also important for implementation of the mentioned systems in high speed networks (1, 10, 100 Gbps). This group analyzes and unifies use of the network traffic sampling and filtering. Documents of this group describe in detail the requirements and techniques for selection of relevant network traffic data subset needed to acquire sufficient knowledge about network traffic, while keeping the load of the measuring point within reason.

No implementation of tool for monitoring the network traffic parameters, or other activity related to the IP flow export, which would conform with the mentioned proposed standards, exists yet. Therefore it is quite important to evaluate implementation possibilities of individual requirements at development of network monitoring tools. Next chapters provide an overview of these possibilities, and also proposals and ways out of the problems that can appear.

2. TERMINOLOGY

2.1. IP flow

A flow is defined as a subset of IP packets going through a network point during certain time interval. All packets belonging to certain flow have some common features. A packet is said to belong to the flow if it fulfils completely all features defined in the flow.

In the IPFIX a flow is defined as follows: A flow is a set of IP packets, or encapsulated IP packets, going through the network observation point during certain time interval. All packets belonging to certain flow have a set of common properties. Each property is defined as a result of applying a function on values:

1. One or more fields of the current packet header, e.g. target IP address, or field in the encapsulating packet header, e.g. end points of IP-v-IP tunnel or fields of transport header (target port number), or fields of application header.
2. One or more properties of the packet iself, e.g. packet length.

[1] *This work is partially supported by the Slovak Science Grant Agency (VEGA No 1/2175/05 "Evaluation of operational parameters in broadband communication infrastructures: research of supporting platforms").*

3. One or more properties resulting from the packet processing, e.g. next jump address etc.

A packet is defined as belonging to flow if completely meets all defined flow properties. Each of the items (1., 2., 3.) is called a flow key.

This definition includes flows containing all packets monitored in network, down to one-packet flows.

2.2. Observation Point

Observation point is a network point where the IP packets can be monitored. Examples include line with connected meter, shared medium such as Ethernet based LAN, router port, or set of router interfaces (physical or logical). An observation point can also be a set of other observation points.

2.3. Measuring Process

Measuring process generates records about flows. The process input are packet headers monitored in observation point and packet processing in monitoring point. The measuring process consists of a set of functions which contains packet headers capturing, timestamping, sampling, sorting, and flow records management.

Flow records management includes recording, treating of existing records, calculations of flow statistics, derivation of further flow properties, flow expiration signalling, relaying of the flow records to export process, and flow records elimination.

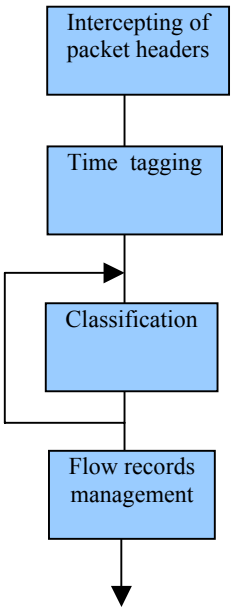

Fig. 1 Functions of measuring process

Sampling / filtering function and classification can be applied several times with different parameters. Figure 1 shows how these functions are applied in measuring process. The sampling / filtering is not represented on the scheme because it can be carried our before or after any of other functions.

2.4. Flow record

The flow record contains information about specific flows which have been observed in the observation point. The record includes also measured and characteristic properties of the flow.

2.5. Export process

Export process relays the flow records to one or more Collecting processes. The flow records are generated by one or more measuring processes.

2.6. Collecting process

Collecting process receives the flow records from one or more export processes. Collecting process can store the received flows or treat them further or send them to be processed by other application.

2.7. Observation domain

A set of observation points and their corresponding measuring processes is called observation domain. To identify exported packets observation domain presents unique ID to collecting process. One or more observation domains can cooperate with the same export process.

2.8. Template

A template is an ordered n-element sequence used for complete identification of structure and semantics of certain information which is to be transferred from IPFIX export process to collector. Each template is uniquely identified.

3. IPFIX ANALYSIS

3.1. Distinguishing packets into flows

Packets are mapped to flows by evaluating their attributes. The packets with similar parameters are assigned to the same flow. A packet with one or more different attributes is assigned to different flow.

In the following sections the properties, that measuring process has to know to recognize in order to map incoming packets to flows, are mentioned. Which of them are used in the mapping decision depends on the measuring process configuration. Essentially, it does not need to be always the whole subset of properties, but only its part. For particular use it is possible to evaluate also properties not mentioned in this paper.

Interfaces

The measuring process has to be able to distinguish flows according to input interface where the packet was observed or according to output

interface where the packet leaves the measuring point.

IP header fields

The measuring process has to be able to separate flows according to the following IP header fields:

- source IP address
- target IP address
- protocol (TCP, UDP, ICMP, ...)

To compare with the source or target address, beside the complete match, a mechanism for comparison by the IP address prefix has to be at disposal.

The measuring process should also enable to distinguish packets to flows according to the IP protocol version, if the measuring point is located in a device supporting more than one protocol version.

Fields of transport protocol header

The measuring process has to be able to recognize flows according to the port numbers of the TCP or UDP transport header. At the same time, when dealing with a transport protocol of the SCTP type [6], the recognizing according to ports has to be supported.

For identification it has to be possible to use the source or target port, or combination of both.

MPLS label

If the measuring point is located in a device supporting MPLS (Multi Protocol Label Switching) [7] then the measuring process has to be able to distribute packets into export flows also by this rule.

DiffServ Code Point

If the measuring point is located in a device supporting DiffServ (Differentiated Services) [8], then the measuring process has to be able to distribute packets into export flows by this rule, too.

3.2. Reliability

The measuring process has to be reliable, or if it is not, this fact has to be clearly indicated.

Whether the measuring process is reliable is determined by that, whether every packet observed in measuring point is processed according to the measuring point configuration. So, if for instance in case of overload some packets can not be processed or some other incidence hinders normal function of the measuring process, this fact has to be detected and clearly signalled in some proper way.

3.3. Sampling / filtering

Sampling means systematic or random selection of a subset of elements (a sample, in our case, of packets), from the original set (parent population, in this case it is the whole network traffic observed in measuring point). Sampling is non-deterministic selection.

Filtering is deterministic selection of elements from the parent population, which fulfil certain, previously given parameters. Filtering is deterministic selection. Usually, the purpose of sampling / filtering is to find the parameters of the parent population by evaluating only the packets from the sample. Sampling or filtering techniques can be applied either on observed packets, which are subsequently mapped to flows, or on the proper flows generated by measuring process.

Sampling / filtering methods differ in their strategies (e.g. systematic or random mask, mask / match, hash filtering ...). In the case of sampling it can be also event that triggers selection of a packet. Thus, the packet selection can be influenced by entrance timing (time based sampling), order within overall network traffic (count based sampling), or directly by packet content (content based sampling).

The measuring process can support sampling / filtering. If those mechanisms are supported and utilized, their configuration has to be clearly defined, as it directly determines the selection method and its parameters.

If change of sampling or filtering configuration occurs during operation of the measuring point, this change has to be indicated to all collecting processes, which register flows from that measuring process. The change of configuration includes the following:

- addition of sampling or filtering function to the measuring process
- removal of sampling or filtering function from the measuring process
- change of sampling or filtering method
- change of sampling or filtering parameter(s)

In case of any change of sampling / filtering configuration it is necessary that all flows measured by the previous configuration be terminated correctly and exported according to the previous configuration. The flows generated by the previous and new sampling / filtering configurations must not be mixed.

3.4. Overload behaviour

In case of overload of the measuring process (e.g. due to insufficient system resources) this can change its behaviour in order to handle the problem. Possible solutions include:

- reduction of amount of measured flows. This can be achieved by setting the flow attributes to more “coarse” values, so that one flow intercepts more flows. This prevents generating of large amount of similar flows and thus saves the system resources.
- starting the sampling / filtering process before processing packet into the flow. If this is

already happening, the system resources can be saved by changing the sampling / filtering parameters.
- stopping the measurement
- changing the process priority.

The behaviour of the measuring process at overload is not limited by the abovementioned choices. Generally, however, when using some mechanism to deal with insufficient system resources, this mechanism has to be clearly defined in configuration.

For some flows such change can influence the saved data, for instance when change of the sampling frequency or classifier criteria occur. When applying the mechanism for handling overload, these flows have to be closed and exported outside the flows generated after the change. These flows must not be mixed. The collecting process has to be able to distinguish flows generated before and after the change of the measuring process behaviour. This requirement does not concern the flows unaffected by the measuring process conditions change.

3.5. Timestamping

The measuring process must be able to generate the timestamps for the first and last observed flow packets in the measuring point. The timestamp precision has to be at least that of *sysUpTime* [9], i.e. one tenth of second.

3.6. Time synchronization

It has to be possible to synchronize the timestamp generated by measuring process with the UTC (Coordinated Universal Time).

It is important, that the possibility of synchronization of the timestaps of one measuring process with the UTC implies the possibility to mutually synchronize the timestamps generated by various measuring processes.

It has to be noted that the abovementioned does not implies generating the timestamps by the measuring process in the UTC. They can be generated in local time of the measuring process and when exporting, this will be provided with another, UTC timestamps, which will become a reference for timestamps synchronization according to the time shift with respect to the UTC.

3.7. Flow export

The measuring process has to be able to detect flow expiration. A flow is considered expired if during certain time no packet belonging to this flow is observed. The measuring process can support flow expiration even before this time is over, for instance by detecting FIN or RST bits in TCP connection. The flow expiration detection procedure has to be clearly defined.

3.8. Multicast flows

For multicast flows, which contain packets replicated to several output interfaces, the measuring process has to be able to keep separated flows for every output interface. For example, a measuring process should be able to capture incoming multicast packet which has been replicated to 4 different output interfaces in four different flows, which differ in their output interface.

3.9. Packet fragmentation

In the case of the packet fragmentation it can happen, depending on classification scheme, that only packet with zero offset of one fragmented packet will contain sufficient information to classify this packet. (This packet should be the first one generated by device doing fragmentation, but does not have to be the first one spotted in the measuring point.) That's why the measuring process can keep information about mapping of fragments which do not have enough information, so that after obtaining sufficient number of fragments this packet could be classified and assigned to flow.

3.10. Port copy packet ignoring

The measuring process should be able to ignore the packets generated by port copy function located in the device containing that measuring point.

3.11. Information model of data export

Information model of data export is a list of attributes which will be included in exported data (together with semantics of those attributes).

The following list presents attributes that the measuring system should be able to export. However, it does not mean that every flow must contain all here presented attributes. On the other hand, it has to be possible to configure the export process in a way that allows transferring of all required attributes to the colleting process (processes) for each exported flow.

The measuring process can further provide a possibility to export attributes which are not listed here. Then the flow can include the standard attributes, as well as some others, covering for example future technologies.

If the measuring process has to fulfil the IPFIX requirements, it has to be able to export all mandatory fields, even though in some cases only a small part of these fields is necessary.

The export process has to be able to export the following attributes (mandatory fields):

1. *IP version*
   This requirement applies only to measuring points supporting more than one IP version.

2. *source IP address*

3. *target IP address*

4. *type of IP protocol (TCP, UDP, ICMP, …)*

5. *number of source port*
   Only for UDP or TCP protocols.

6. *number of target port*
   Only for UDP or TCP protocols.

7. *packet count*
   If the packet is fragmented, each fragment is considered as an individual packet.

8. *byte count*
   Sum of all lengths of IP packets in bytes. The whole packet length includes the IP header and IP payload.

9. *type of service octet (in case of IPv4)*
   octet traffic class (in case of IPv6)
   According to [8] these octets contain DiffServ CodePoint, which is 6 bits long.

10. *flow label (in case of IPv6)*

11. *the highest MPLS label FEC (forwarding equivalence class [7])*

12. *timestamp of the first flow packet*

13. *timestamp of the last flow packet*

14. *sampling / filtering configuration*

15. *unique identifier of the observation (measuring) point*

16. *unique identifier of the export process*

The export process should be able to export the following attributes:

17. *ICMP type and code (in case of ICMP protocol)*

18. *input interface*
    Does not apply for devices acting as probes.

19. *output interface*
    Does not apply for devices acting as probes.

20. *multicast replicate factor*
    Number of packets coming out from device generated after obtaining one multicast packet. This is a dynamic variable of multicast flows that changes with time. For unicast flows it is a constant 1. Its export value is a factor in the point of exporting the flow.

Export process can be able to export the following attributes:

21. *TTL (Time To Live, in case of IPv4)*
    Hop limit (in case of IPv6)

22. *IP flags*

23. *TCP flags*

24. *count of packets discarded in the measuring point*
    If the packet is fragmented, each fragment is considered as an individual packet.

25. *fragmented packets count*
    Counter of all packets with fragmentation flag.

26. *next hop IP address*

27. *number of source BGP AS (Autonomous System)*

28. *number of target BGP AS (Autonomous System)*

29. *number of next hop BGP AS (Autonomous System)*

3.12. Data model of flow record

Data model describes how the information is represented in the flow record.

The data model has to be extensible by future attribute addition. Even though the set of attributes in the flow record is fixed, the data model has to provide a possibility to extend the record through configuration.

The data model used for flow information export has to be flexible with respect to the flow attributes included in the flow record. The flexible format should provide a possibility to define records independently on type and number of attributes.

The data model should be independent on transport protocol used.

3.13. Data transfer

Requirements for the data transfer include reliability, overload handling capability, and security issues. To satisfy these requirements the export process can use security mechanisms of the device in which it acts and/or the ones provided by the transport net. For instance it can utilize existing authentication and encrypting mechanisms and/or use physical protection of separated network to transfer the flow information.

3.14. Handling of (line) overload

For data transfer a protocol able to handle line overload has to be used.

3.15. Reliability of data transfer

Any loss of flow records during the data transfer from the export to collecting process has to be indicated in the collecting process. This indication has to enable to show the number of lost flow records. The possible reasons for loss of flow record include (but are not limited to):

1. *measuring process limits (insufficient memory, computing capacity,...)*

2. *limits of export process (insufficient memory, computing capacity,...)*

3. *transfer problems*
   Packets with information from export to collecting process are discarded during transfer (line failure, protocol failure, ...)

4. *limits of collecting process*
   The process can be jammed and unable to manage incoming flow records.

5. *Operation and maintenance limits*
   The collecting process can be disabled due to maintenance or administrative reasons.

If an unreliable transport protocol is being used, the reliability can be provided by higher layers. In such case only overall reliability loss needs to be indicated.

The exporting – collecting processes data transfer has to be open for reliability modifications including at least:

- retransmission of lost flow record
- connection break or failure detection
- confirmation of receiving the record by collecting process

This extensibility can be utilized to provide additional reliability. Again, the extended protocol has to satisfy the requirements for the original one, among other, it still has to be able to handle overload.

3.16. Security

The exporter - collector data transfer confidentiality has to be guaranteed. Also their integrity and authenticity has to be conserved.

These security requirements stem from the potential security threats more widely summarized at the end of the paper.

3.17. Data export modes

Generally, there are two options to decide about data transfer:

- push mode
- pull mode

In the push mode the decision about sending the records is made by the export process, without outside trigger activity.

In the pull mode the record sending is triggered by outside request from the collecting process.

The export process has to support the push mode export. It can support the pull mode export.

3.18. Regular export interval

The export process should be able to export the measured traffic regularly according to the given time interval.

3.19. Message about specific events

The export process can have the ability to send a message to the collector when a specific event occurs. Such event can be, for example, arrival of the first packet of a new flow, or flow termination after its time interval expiration.

3.20. Anonymization

The export process can be able to anonymize the source and target IP addresses in the flow data before exporting. It can support anonymization of the port numbers and other fields, too.

Originally, anonymization is not an application requirement, it is derived from general requirements for processing of the measured network traffic.

For certain applications the anonymization is not applicable. Examples are accounting, traffic engineering. In spite of that, for the sake of user privacy protection it should be applied whenever possible. In most cases it is sufficient to apply the anonymization in the collecting process, immediately after reception of export information. This provides sufficient security provided that the export confidentiality is maintained.

When anonymized data are exported, it has to be indicated to all collecting processes receiving data from this exporting process, so that the anonymized and unanonymized data can be distinguished.

3.21. Configuration

If configuration is being made remotely, its security, including its confidentiality, integrity and authenticity, should be guaranteed.

3.22. Configuration of measuring process

The measuring process has to provide option to configure the traffic measurement. The following parameters should be configurable:

- observation point specification (e.g. interface, interface list)
- specification of flows to be measured
- time intervals of flows
- sampling / filtering methods and parameters (if supported)
- overload handling (if supported)

3.23. Configuration of export process

The export process has to provide options to configure the data export. The following parameters should be configurable:

- export data format
  Specification of the export data format has to include selection of the export attributes for each flow.
- collecting processes which are exported to flow
- export interval
  This option is applicable only when the exporter supports exporting in regular time interval.
- messages to be sent to collecting processes
  This option is applicable only if the collecting process supports messaging.
- flows anonymization
  This option is applicable only if the exporting process supports flow anonymization.

3.24. Openness

IPFIX conform implementations should be open for future technologies. This includes extensibility of configuration of measuring and exporting process.

Openness is also required in connection to expandability of data model.

3.25. Scalability

The data collection from hundreds exporting processes has to be supported. The collecting process has to be able to identify several hundreds exporting processes by their identifiers.

3.26. Larger number of collecting processes

The export process should be able to export the flow information to more than one collecting process. If the exporting process is able to export to more collecting processes simultaneously, it has to guarantee safe identification of flow records to prevent duplication and problems with double computing.

3.27. Security risks

An IPFIX conform implementation of measuring tool has to be able to transport data through public Internet. Therefore it can not be excluded that an attacker intercepts and changes packets or adds new ones.

This section describes security requirements for IPFIX implementations. As for other requirements, also the security ones differ between applications. The motivation to change the collected accounting data or IDS (Intrusion Detection System) is usually higher than to change the traffic profiling data.

The following potential security problems were identified in connection with exposing the IP flow information, flow record creating, and DoS (Denial of Service) attacks.

3.27.1. Flow data revealing

Content of the data exchanged within IPFIX implementation should be secret while transferred between concerned processes. Observation of flow records gives attacker detail information about active network flows, communication points, and traffic samples. This information can be used not only to observe the user's behaviour but also to plan future attacks. Therefore the security requirements include secrecy of transferred data. This can be achieved by encryption.

Similarly, privacy of users, either senders or receivers, has to be maintained. In many countries the right to save the personal information (including network traffic profile) is limited by law or by regulations.

Together with encryption, some privacy part can be protected by anonymization too. For many flows such anonymized data are usable equally as the original ones.

3.27.2. Creation of new data flows

If the flow records are used as a base for accounting and/or security applications, strong motivation to create new IPFIX flow records arises.

Especially, it is necessary to monitor situations where the flow measurement is a basis of a security application.

3.27.3. Denial of Service (DoS) attacks

Routers or other IPFIX protocol devices using the record sending, can become targets of the DoS attacks. However, such attacks are not caused by the IPFIX implementation and therefore they can not be solved on this level.

4. PSAMP ANALYSIS FOR IPFIX

In the PSAMP draft, which is being developed together with the IPFIX standard, a complete packet sample export protocol is presented. To implement a conform tool with support of packet sampling and filtering the following requests are made:

4.1. Requirements of general selection process

- omnipresence
  The selector has to be so simple that is can be implemented everywhere at the highest transfer speed.
- applicability
  The set of selectors has to be sufficiently large to support the large group of existing and newly created measuring applications and protocols.
- expandability
  The implementation has to be prepared to be expanded by selectors currently yet undefined.

- flexibility
  Implementation has to support packet selection from various network protocols, or from encapsulation.
- robust selection
  packet selection has to be so robust as to resist the attempts to create artificial packet flow, from which the selection would be disproportional.
- causality
  The selection rule for every packet has to depend on other packets incoming only very weakly or not at all.
- encrypted packets
  The selectors interpreting the packet header fields have to be configurable to ignore encrypted packets.

4.2. Selectors

The PSAMP categorizes two selector types:

- filtering
  Filtering selects packet deterministically according to packet content, packet handling, and functions of both. The examples are:
  - Field-match filtering.
  - Hash-based selection
- sampling
  Sampling is a non-filtering selector. That means the packet selection can not be derived from the packet content only. The sampling operations can be divided to two subtypes:
  - Content-independent Sampling
    Does not use the packet content to obtain sampling result. Examples include periodic and uniform pseudorandom sampling dependent on random number, independent of the packet content.
  - Content-dependent Sampling
    Uses the packet content to obtain sampling result. An example is pseudorandom probability sampling depending on the packet content.

4.3. PSAMP defined selectors

PSAMP selection process has to support at least one of the following selectors:

4.3.1. Systematic time based sampling

Packet selection is an instance of a periodic function separated by spacing. All packets incoming within certain time interval are selected.

4.3.2. Systematic number based sampling

Similar to the previous one. The difference is that the selection is defined by the packet order not by time. Packet selection is made periodically after certain number of packets.

4.3.3. Uniform probability sampling

Packets are selected independently, with fixed probability, *p.*

4.3.4. Non-uniform probability sampling

Packets are selected independently with probability *p* depending on the packet content.

4.3.5. Probability n-z-N sampling

From each *N* consecutive packets *n* is randomly selected.

4.3.6. Match filtering

Filtering schemes are based on the IPFIX flow definition. In this method the packet is selected if specific packet field is equal to a pre-defined value. The possible fields with which this value can be compared are shown in [2].

The packet is selected if Field = Value. Masks and value intervals are admissible where [2] allows it.

AND operations are possible by chaining of filters, creating thus a composite selection operation. In this case the order of filtering is implicitly defined.

OR operations are not supported by simple models.

4.3.7. Hash filtering

Hash filtering utilizes one or more hash functions. Hash function is applied on a subset of packet content. The packet is selected if the function result lies within a certain interval. The stronger hash function, the better approximation of uniform probability sampling.

4.3.8. Router state filtering

This selector chooses packets using following conditions, which can be combined by logical operators AND, OR, and NOT.

- input packet interface
- output packet interface
- violation of ACL by packet
- RPF failure (Reverse Path Forwarding) for packet
- RSVP failure (Resource Reservation)
- no route for packet
- source BGP AS
- target BGP AS

Other conditions usually depend on the router architecture.

5. IMPLEMENTATION ANALYSIS OF SOME IPFIX AND PSAMP REQUIREMENTS

5.1. Implementation of packet classifier to flows

According to the IPFIX, packet classification by several IP header fields and some fields of packet transport header is obligatory.

One of the options (except locating the measuring point in MPLS or DiffServ environment) is the method described in [13].

$$id = \sum_{i=0}^{n} p_i \, mod \, b$$

where *id* is packet flow affiliation identifier, *n* is the number of considered fields, p_i is *i*-th field, and *b* is the number of considered flows or maximum number of flow stack.

This method is implemented in the measuring device BasicMeter, namely in its classifier. The method's disadvantage is the large number of collisions if the function input obtains incorrect values. Then, it is highly probable that the packet ends up in completely different flow. This is caused by the weak hash characteristic of this function.

Better and more universal usability is provided by frequently used strong hash functions MD5 or SHA. The collision probability of these functions is very low and when using more packet header fields, it is almost zero.

Other options:

- unidirectional vocabulary functions
- checksums
- compress algorithms

5.2. Sampling and filtering

Sampling and filtering belong to standard mechanisms to lower the demands for the system resources at monitoring the high speed networks. Every packet going through a measuring point has to be open down to several layers. It is clear that the time demands for monitoring each packet grows with the network speed.

In the current implementations of monitoring applications (in the OS Linux environment) the basis is usually the library *libcap*, which provides low level (but still user-space) capturing of packet and passing on its content, together with attached headers, to user application. This library uses so called BPF (Berkeley Packet Filter), which is a standard part of any larger distribution.

For the library *libcap*, i.e. for programmer, the BPF provides a complete implementation of the field map filtering.

The meaning of the hash filtering is, above all, its approximation of the uniform probability sampling. Since not every device has a random number generator, in this selector it is possible to use the result of the strong hash function of given packet part as a random numbers source. The stronger the hash function, the more it approximates the mentioned sampling.

In the case of the router state filtering the implementation is quite complicated. It would be necessary to simulate the packet route through whole routing subsystem and the result would be a set of operations done on the monitored packet. Therefore this filtering is usually a part of commercial implementations, such as NetFlow and others.

Sampling using the abovementioned sampling algorithms is done mostly by simple functions, where as input serves time, or order of incoming packet, or probability of sampling given either as fixed or as random function. The output of these functions will be a binary value for every packet, indicating whether the packet shall be selected from the population or not.

5.3. Time synchronization

The time synchronization is one of the most important requirements for the IPFIX protocol implementation, namely for evaluation of the QoS parameters in computer networks. All time characteristics (especially those for multipoint measurements) depend on precise synchronization.

Unfortunately, the most of hardware clocks have low precision. This is simply because the time controlling frequency is never the same. Error of 0.001% leads to abour 1 second deviation in one day. That is why, for precise measurements, the PPM (Part Per Million), i.e. 0.0001% (= 1E-6), is used.

Since the average values of the QoS parameters, such as OWD and RTT (One Week Delay, Round Trip Time) are given in orders of milliseconds to tenths of millisecond, synchronization precision in order of microseconds is necessary.

To find the actual precision of the NTP synchronization the command *ntp –c rl* can be used. With more powerful workstations and good connection it is possible, with 1-2 severs, to achieve precision of about tens of microseconds.

Since the precision in PPM depends on many physical quantities (heat, magnetic fields, etc.) it is advisable to perform the synchronization immediately before and after the measurement.

5.4. Information model

Information model of the IPFIX protocol is quite large. Therefore for implementing the flow export so called templates were introduced. These templates determine which fields of the information model will be included into the export of every exported flow.

The most suitable for description of these templates seems to be the XML language, because it is robust, scalable, and easily recognizable. An example of such template, exporting number of transferred bytes and packets in flow, is:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<template>
    <bytes_32 />
    <pkts_32 />
</template>
```

5.5. Data model

Data model of transfer is critical for performance of the IPFIX protocol implementation. Transfer of disproportionally large amount of data can slow the exporter and other parts of measuring architecture will have to adapt to its speed, if they do not have implemented a powerful buffer. Therefore, it is necessary to choose for data model a form which satisfies requirements of expandability, flexibility and independence on the transfer protocol.

Here, too, the optimal seems to be implementation of the XML, which is sufficiently expandable, flexible, and transfer protocol independent. Moreover, for suitable choice of transferred XML data format it is undemanding with respect to data volume.

5.6. Reliability

Transfer reliability or indication of export-collector data loss is, when using the TCP, simple. It is sufficient to rely on the sequential packet numbers.

When using the unreliable UDP protocol, it is necessary to implement the data loss indication at higher levels. An option is additional numbering of flows in such order as they leave the exporting protocol. On the collecting side the set of flows that arrived correctly would be periodically checked, and a loss of some number or its excessively delayed arrival would be indicated to user.

Since the IPFIX implementation requires openness for introducing further mechanisms for assuring the security, it is possible to extend the XML data transfer model by control information, which would guarantee the reliability at the application level.

5.7. Configuration

It has to be possible to set up the configuration parameters of the measuring, export, and collecting processes beforehand. This condition can be satisfied simply by creating a configuration file. Implementation details related to the format of this file are again solved by the XML language, which provides sufficient freedom for modification of parameters and their values.

6. ANALYSIS OF DEFICIENCIES OF THE IPFIX PROTOCOL

One of the greatest defects encountered when implementing the new standard into the measuring device BasicMeter is absence of support for unambiguous identification of the packets themselves. This key element of multipoint measurements of the QoS parameters, which enables to trace single packet on route through the network and a number of measuring points, was in the IPFIX architecture proposal indicated too late, and even to this date its support is not great. The problem of the unique packet identification has the greatest challenge in the concept of aggregation. The flow export allows viewing the network traffic as a set of flows but information about particular packet is lost. Only at the end of 2005 the internet draft was released [15], introducing the concept of "Flow per packet" within the IPFIX. This concept allows existence of one-packet flows, by which it effectively excludes the aggregation principle, which until now prevented successful introduction of the unique packet identification and multipoint measurements for evaluation of time characteristics of QoS in computer networks.

7. CONCLUSION

The realized analysis enabled to obtain valuable knowledge for implementation of the measuring device BasicMeter in the Computer Networks Laboratory. Much of it will be soon included into the project for development of a measuring device for the QoS parameters evaluation in computer networks.

During the analysis also many deficiencies, present in the current implementation of the measuring device, have been found. They were caused by incorrect interpretation of the currently emerging standard. In future, the communication of the device developers with the IPFIX and PSAMP working groups, as well as with associated persons will have to be improved.

REFERENCES

[1] Quittek, J., Zseby, T., Claise, B., Zander, S., Carle, G. and Norseth, K.C.: Requirements for IP Flow Information Export, RFC3917, Network Working Group, October 2004

[2] Quittek, et al.: Information Model for IP Flow Information Export, Internet draft, Network Working Group, September 2005

[3] Zseby, T., Molina, M., Duffield, N., Niccolini, S. and Raspall, F.: Sampling and Filtering Techniques for IP Packet Selection, Internet Draft, PSAMP Working Group, July 2005.

[4] Sadasivan, et al.: Architecture for IP Flow Information Export, Internet draft, IP Flow Information Export WG, August 2005

[5] Leinen s.: Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX), RFC 3955, Network Working Group, October 2004

[6] Stewart, et al.: Stream Control Transmission Protocol, Network Working Group, RFC2960, October 2000

[7] Rosen, et al.: Multiprotocol Label Switching Architecture, Network Working Group, RFC3031, January 2001

[8] Nichols, et. al.: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Network Working Group, RFC2474, December 1998
[9] Presuhn, et al.: Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), RFC3418, December 2002
[10] Hronský, M., Jakab, F., Potocký, M., Jakab, R. and Giertl, J.: Sampling Algorithms for Nonintrusive Measurement in Network Oriented Educational Systems, In: 4th International Conference on Emerging e-learning Technologies and Applications (ICETA 2005), Košice, Slovak Republic, 13 - 14 September 2005, elfa, s.r.o., 2005, pp. 165-171, ISBN 80-8086-016-6
[11] Jakab, F.: Methods of Management Optimization and Evaluation in Computer Networks: Measurement and Evaluation of Operational Parameters, PhD thesis, Technical University of Košice, Košice, Slovak Republic, 2005, 133 pp, (in Slovak)
[12] Jakab, F., Jakab, R., Kaščák, M. and Giertl, J.: Improving efficiency and manageability in IPFIX network monitoring platform, In: International Network Conference 2006 (INC2006), Plymouth, United Kingdom, 2006 (in printing)
[13] Kvačkaj, P., Baroňák, I.: Connection Admission Control (CAC) in ATM Systems. In: International Conference – Applied Electronics. Pilsen, 8-9 September, 2004, ISBN 80-7043-274-8, pp. 134-138
[14] André, M.: Meranie a vyhodnocovanie prevádzkových parametrov v pocítacových sietach, Diploma thesis, Košice 2006
[15] Boschi, M.: Use of IPFIX for Export of Per-Packet Information, Internet-Draft, June 2005

BIOGRAPHY

František Jakab was born in 1959. He received the MSc.degree in Systemotechnic engineering from the St. Petersburg Electrotechnical Institut (Rusia) in 1984 and the PhD. degree in 2005. He is employed as an assistant professor at the Dept. of Computers and Informatics, Technical university of Kosice. He is a head of the Computer Engineering Group and Computer Networks Laboratory (www.cnl.tuke.sk). His research interests include projecting of computer network, modeling, simulation and network management, new form of multimedia-based communication, QoS, tele-learning systems, intelligent tutoring systems and virtual universities. He has been a coordinator of several large international e-learning oriented projects supported by EC. He is a coordinator of the Cisco Networking Academy Program for the Slovak Republic and head of the Application Section of the Communication Technology Forum Association in Slovak Republic (www.ctf.sk).

Miroslav Potocký was born on 1982 in Trebišov. He is student of Information systems and technologies (in 2006 he will graduate MSc.) at the Department of computer and informatics of the Faculty of Electrical Engineering and Informatics at Technical University of Košice. He is a member of Computer Network Laboratory (www.cnl.tuke.sk) assigned to QoS@Lab project (qos.cnl.tuke.sk). Area of his research interests include developing software for evaluting QoS parameters of computer networks and implementation of sampling algorithms in high-speed computer networks.