IPFIX Mediation Framework of the SLAmeter Tool

Rastislav Kudla, Peter Feciľak, and Adrián Pekár

Department of Computers and Informatics
Technical University of Košice, Letná 9, 042 00 Košice, Slovakia

Abstract. Presented paper deals with the IP Flow Information Export (IPFIX) Mediation Problem. The problem was elaborated by IPFIX working group in RFC 5982 and RFC 6183. The aim of this work is to design and implement an IPFIX Mediation Framework of the SLAmeter monitoring tool based on these documents. The analytical part is devoted to the IPFIX protocol and the IPFIX Mediation Problem. Monitoring applications SLAmeter and BasicMeter are analysed as well. The core part of the paper is represented by the design and description of implementation workflow of IPFIX Mediation Framework. Accuracy of solutions is confirmed by experimental verification at the end of the paper. The main contribution of this work is to extend the possibilities of monitoring application SLAmeter.

Key words: IPFIX, Mediation, Mediator, Collector, Exporter, SLAmeter, BasicMeter, MONICA

1 Introduction

The trend of recent years in computer networking are converged networks which combine data, voice and video transmission into one common infrastructure. However, with increasing growth of opportunities raises demand for quality. In order to compare or even increase quality of networks and service, it is necessary to have the ability to measure network parameters.

SLAmeter is measuring architecture being developed by the MONICA research group in Computer Networks Laboratory at the Technical University of Košice. It is a tool for passive flow-based measurement and subsequent network traffic analysis with the purpose of determing the grade of network quality [1].

Flow-based measurement is a popular method for various network monitoring usages. The sharing of flow-based information for monitoring applications having different requirements raises some open issues in terms of measurement system scalability and capacity, flow-based measurement flexibility, and export reliability. *IP Flow Information Export (IPFIX) Mediation* may help resolve these issues [2].

2 Motivation

To fulfill application requirements with limited system resources, the IPFIX architecture needs to introduce an intermediate entity between Exporters and Collectors. From a data manipulation point of view, this intermediate entity may provide the aggregation, correlation, filtering, and modification of Flow Records to save measurement system resources and to perform preprocessing tasks for the Collector. From a protocol conversion point of view, this intermediate entity may provide conversion into IPFIX, or conversion of IPFIX transport protocols (e.g., from unreliable, connectionless UDP to the Stream Control Transmission Protocol (SCTP)) to improve the export reliability [3].

3 Goals

The aim of this work is to analyze, design, implement and test an framework for intermediate entity (Mediator) between exporter and collector in the IPFIX protocol. It also integrates the solution with the existing architecture of the SLAmeter monitoring tool. Framework put a great emphasis on its modularity in order to make it easy and convenient to implement new mediation modules and thereby increase monitoring possibilities of SLAmeter.

4 Analysis

4.1 Terminology and Definitions

The IPFIX-specific terminology used in this paper is defined in [4]. Let us define the IPFIX Mediation-specific terms here.

- **Record stream** – a stream of records carrying flow-based information. The records are encoded as IPFIX Data Records.
- **IPFIX Mediation** – the manipulation and conversion of a record stream for subsequent export using the IPFIX protocol.
- **Intermediate Process** – takes a record stream as its input from Collecting Processes, or other Intermediate Processes performs some transformations on this stream, based upon the content of each record and passes the transformed record stream as its output to Exporting Processes, or other Intermediate Processes, in order to perform IPFIX Mediation.
- **IPFIX Mediator** is an IPFIX Device that provides IPFIX Mediation by receiving a record stream from IPFIX Exporter or other Mediator, hosting one or more Intermediate Processes to transform that stream, and exporting the transformed record stream to IPFIX Collector via IPFIX Messages. [2]

Fig. 1 depicts one of the possible IPFIX Exporter – Mediator – Collector architecture examples.

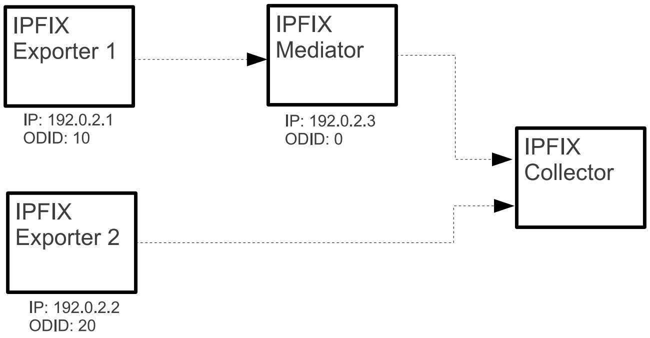

Fig. 1. One of the possible IPFIX Exporter – Mediator – Collector architecture examples

4.2 Problem statement

Network administrators generally face the problems of measurement system scalability, Flow-based measurement flexibility, and export reliability, even if some techniques, such as Packet Sampling, Filtering, Data Records aggregation, and export replication, have already been developed.

The problems consist of adjusting some parameters of metering devices to resources of the measurement system while fulfilling appropriate conditions: data accuracy, Flow granularity, and export reliability. These conditions depend on two factors – **measurement system capacity** and **application requirements**.

Due to resource limitations of the measurement system, it is important to use traffic data reduction techniques as early as possible, e.g., at the Exporter. However, this implementation is made difficult by the heterogeneous environment of exporting devices. On the other hand, keeping data accuracy and Flow granularity to meet the requirements of different monitoring applications requires a scalable and flexible collecting infrastructure.

This implies that a new Mediation function is required in typical Exporter-Collector architectures. [2]

4.3 Mediation Applicability Examples

RFC 5982 [2] provides several examples of IPFIX Mediator applications. Let us mention a few of then:

- Data record anonymization,
- Interoperability between legacy protocols and IPFIX,
- Adjusting flow granularity,
- Flow-Based sampling and selection,
- Distributed collecting infrastructure,
- Time composition,
- Spatial composition.

5 Solution and Results

5.1 Application design

IPFIX Mediation Problem Framework and its architecture is analysed in RFC 6183 [3]. Based on the IPFIX Mediation reference model as an extension of the IPFIX reference model presented in *Architecture for IP Flow Information Export* – RFC 5470 [5] was designed application *Mediator v1.0* architecture. Fig. 2 covers the various possible scenarios that can exist in the *Mediator v1.0* monitoring system. The functional components within each entity are indicated within brackets []. *Mediator v1.0* receives IPFIX Flow Records from other IPFIX Mediators and Exporters via UDP transport protocol. Processed IPFIX Flow records are exported to Collector or other IPFIX Mediator via UDP.

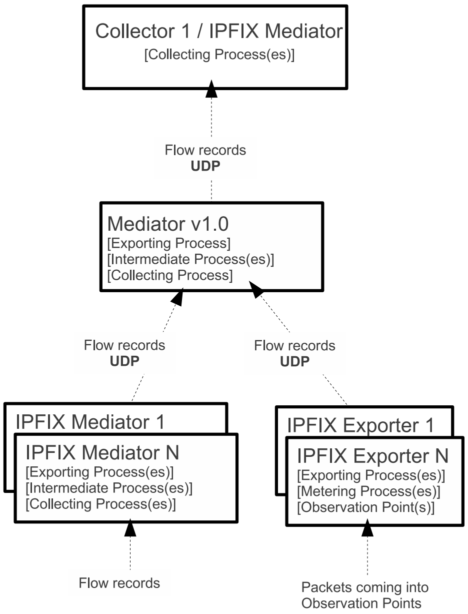

Fig. 2. Mediator v1.0 reference model overview

Basic *Mediator v1.0* component model is shown in the Fig. 3. The Mediator may contain one or more Intermediate Processes hierarchically located between one Collecting and one Exporting Process.

Intermediate Processes are key functional blocks for IPFIX Mediation. Data flow between them is managed in the following ways:

- **Parallel processing -** record stream is processed by Intermediate Processes simultaneously. In this setup, each Intermediate Process receives a copy of the entire record stream as its input.

- **Serial processing -** In order to ensure flexible manipulation of a record stream, the Intermediate Processes are connected serially. In that case, an output record stream from one Intermediate Process forms an input for a succeeding Intermediate Process.

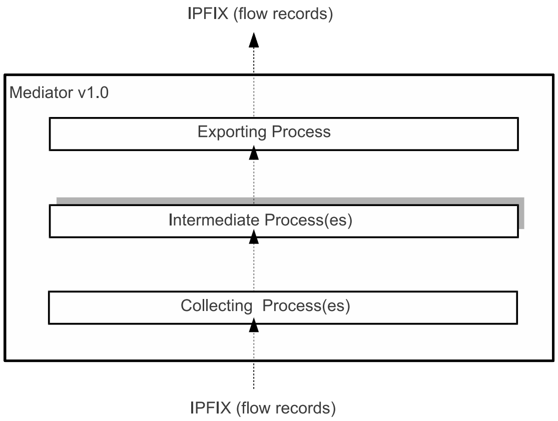

Fig. 3. Basic Mediator v1.0 component model

5.2 Implementation of Mediator v1.0

Main class The role of the main class is to process command line arguments and then start all threads needed to run the application. These threads are following: UDP Server, UDP Processor, UDP Exporter and each Intermediate Process is single thread as well. In case of error is Mediator shut down correctly so it realeases memory and stops all running threads. Same situation occurs after pressing `Ctrl-c`.

Collecting Process Logical structure of this process involves two stages, where each stage is one thread. Let us briefly describe the individual stages of the Collecting Process.

First stage is shown in Fig. 4. A core class of this stage is *UDPServer* which receives byte stream of IPFIX messages sent from Exporter via UDP protocol. This stream is wrapped into the *PacketObject* and written to fast, blocking FIFO queue called *PacketCache*. Application design allows for future extension with another transport protocols, such as TCP and SCTP.

Scheme of second stage depicts Fig. 5. The UDP Processor thread works in cycle – it reads packet objects from cache and sends them to IPFIX Parser. IPFIX Packets are split up there into IPFIX message header, IPFIX template

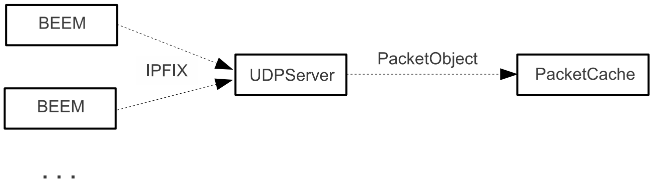

Fig. 4. First stage of Collecting Process of Mediator v1.0

sets, data sets and options template sets. These entities are processed and transformed to flow records. IPFIX Parser sents each flow record with additional information to class representing dispatcher of flow records.

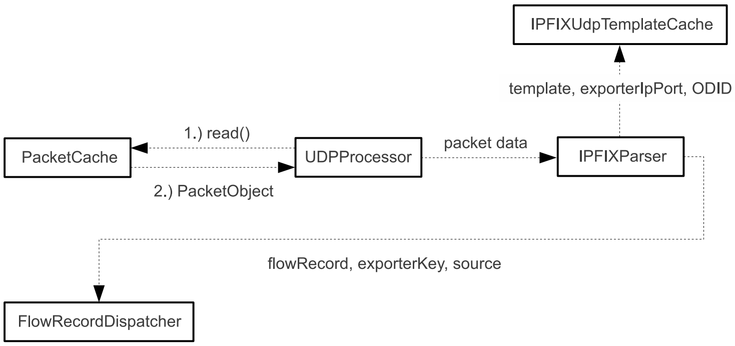

Fig. 5. Second stage of Collecting Process of Mediator v1.0

6 Acknowledgment

This work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

7 Conclusion

References

1. Výskumná skupina MONICA: *SLAmeter* [online], 2013. <`http://wiki.cnl.sk/Monica/SLAmeter`>.
2. KOBAYASHI, A. – CLAISE, B. et al.: *IP Flow Information Export (IPFIX) Mediation: Problem Statement.* RFC 5982. 2010

3. KOBAYASHI, A. et al.: *IP Flow Information Export (IPFIX) Mediation: Framework.* RFC 6183. 2011
4. CLAISE, B. et al.: *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information.* RFC 5101. 2008
5. SADASIVAN, G. et al.: *Architecture for IP Flow Information Export* RFC 5470. 2009