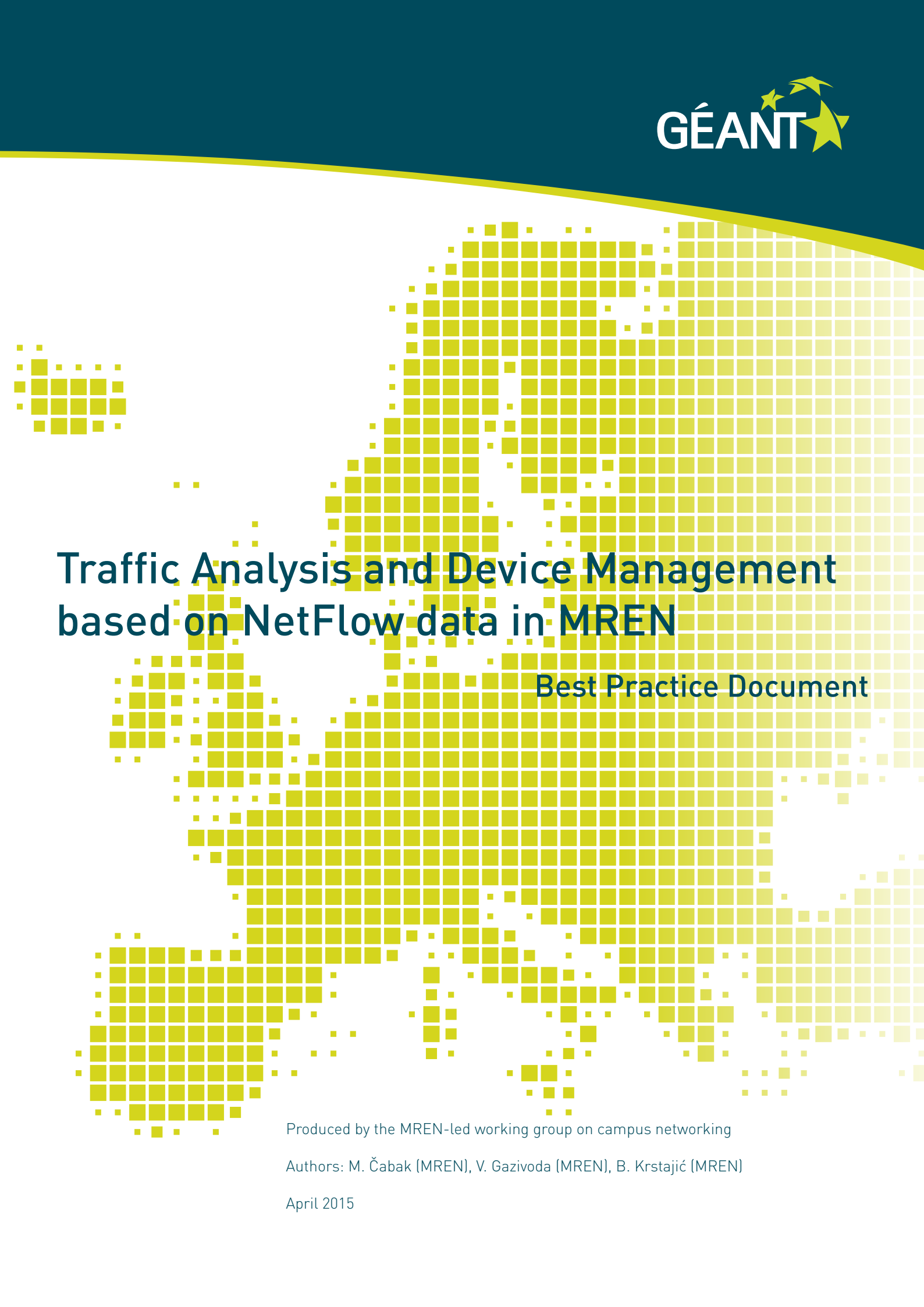


Traffic Analysis and Device Management based on NetFlow data in MREN

Best Practice Document

Produced by the MREN-led working group on campus networking

Authors: M. Čabak (MREN), V. Gazivoda (MREN), B. Krstajić (MREN)

April 2015

Document No:	
Version/ date:	April 2015
Source language:	Montenegrin
Original title:	*"Analiza saobraćaja i upravljanje uređajima na osnovu netflow podataka u AMUCG"*
Original version/ date:	Version 1 / 28 April 2014
Contact:	Milan Čabak, milan@ac.me; Vladimir Gazivoda, vladg@ac.me; Božo Krstajić, bozok@ac.me

MREN/Centar Informacionog Sistema is responsible for the contents of this document. The document was developed by the MREN-led working group on campus networking with the purpose of implementing joint activities on the development and dissemination of documents encompassing technical guidelines and recommendations for network services in higher education and research institutions in Montenegro.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 605243, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3plus)'.

Table of Contents

Table of Figures

Executive Summary

This document describes the monitoring and management methods for computer networks. The quality and availability of computer network services depends on the performance of the monitoring system, as well as the control system.

This document presents the *flow* collector for assembling and analysing data on generated network traffic data obtained from network device exporters. A solution for network traffic analysis will be presented and used for implementing a network devices management system based on the qualitative analysis of network traffic.

Some of the basic techniques for computer network management will also be analysed. The proposed solution sends warnings and automatic actions for changing the configuration of network devices, based on data obtained from qualitative network traffic analysis.

1 Introduction

The complexity of today's computer networks is reflected in the variety of network types, media, technologies, services, and a large number of users. Network administrators require automated tools that will collect information about network elements and successfully manage this complex system.

The need for tools which perform qualitative analysis of network traffic is becoming more prominent in order to have a better insight into the traffic passing through network links. In today's networks we cannot rely only on the information about the quantity of traffic, i.e. availability of links. Information about who or what uses the network resources, and when and how the resources are used, is also necessary. Network monitoring is a complex and challenging task of great importance for network administrators.

Monitoring represents collection and analysis of data about network traffic. Specific mechanisms, which will manage network devices and the system in general, can be activated based on the collected data.

2 Network Traffic Analysis

In modern computer networks, it is of great importance to possess the tools for network traffic analysis. A qualitative analysis of network traffic can be obtained from collection of data contained in *flow* exporters, as well as answers to questions about who or what, when and how uses the network resources. This knowledge is important for network administrators in terms of making the right decisions that can contribute to the whole organisation. There are different methods, and this document will provide a description of a flow-oriented software solution that uses distributed techniques for data collection. Unlike hardware monitoring devices, *flow*-oriented software techniques have advantages in terms of lower investment, easier installation and achieving results in a short period of time.

2.1 Flow technologies

When selecting technologies, one should take into account which versions of *flow* technologies are supported by the network devices in a computer network. Implementation of *flow*-oriented technology requires devices which support *flow* technology in their specifications. Various manufacturers are developing different versions of *flow* technologies that are mainly variations of the same technology. Therefore, our *NetFlow* technology is supported mostly by Cisco devices [1], *sFlow* by Allied Telesis, and others such as *JFlow* by Juniper, *Netstream* by Huawei and *IPFIX* by Nortel network devices.

Selection of the flow technology depends on the types of devices in the network and technologies. These technologies are quite different, but one should take these small differences into account because different software used for collection of this kind of data support one or more different flow technologies. Flow technologies from different manufacturers are not compatible with each other, although efforts are being made to standardise them.

2.2 NetFlow Network Analysis

Each IP packet forwarded by a network device (i.e. an exporter), is examined in order to find the set of IP data. These data are IP packet identifiers and determine whether a packet is unique or part of a set of packets transmitted over a network. The IP packet header consists of a set of 5-7 attributes. IP address

attributes used in NetFlow technology are: source IP address, destination IP address, source port, destination port, type of service (ToS field), and network device interface.

Flow information is very useful in understanding the network behaviour. The source address enables an understanding of who creates traffic; the destination address provides information to whom the traffic is intended; the port indicates which application performs the communication; the type of service determines the priority of the traffic, while grouped packets and data show the amount of generated traffic. The flow also includes the following additional information: the time used for understanding communication duration and packet computing, next hop and the subnet mask of source and destination IP addresses prefix calculations.

3 FNF Exporter Configuration

An ‘exporter’ is a network device with active NetFlow service. An exporter monitors packets passing through a device (i.e interfaces which are monitored) and creates a *flow* of these packages. Collected information is exported to the NetFlow collector in the form of *flow* records.

Flexible NetFlow (FNF), unlike Traditional NetFlow (TNF), has several additional fields, which allows the organisation to oversee more specific information, so that the total amount of information that is exported is reduced, enabling improved scalability and aggregation.

3.1 FNF Records Configuration

Flexible NetFlow (FNF) requires explicit configuration of *flow* records with all necessary and optional fields. The following configuration examples refer to the Cisco ASR 1002 network devices [2].

Basic and additional FNF fields on a network device are defined as follows:

flow record [record name]
    description [record description]
    match [field type] [field value]
    collect [field type] [field value]

Example 1 FNF record configuration with basic and additional fields:

flow record Record-FNF
    description Flexible NetFlow
    match ipv4 tos
    match ipv4 protocol
    match ipv4 source address
    match ipv4 destination address
    match transport source-port
    match transport destination-port
    match interface input
    match flow direction
    match application name
    collect routing source as

collect routing destination as
collect routing next-hop address ipv4
collect ipv4 dscp
collect ipv4 id
collect ipv4 source prefix
collect ipv4 source mask
collect ipv4 destination mask
collect transport tcp flags
collect interface output
collect counter bytes
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last

3.2 Defining the FNF Exporter

NetFlow data that is temporarily kept on the network device is analysed in detail after exporting to the collector. Defining the FNF exporter is only required when exporting data to an external collector. The FNF exporter is configured on a network device as follows:

flow exporter [exporter name]
description [exporter description]
destination [NetFlow collector IP address]
source [interface name]
transport [UDP or TCP] [port number]
export-protocol [protocol name]

Example 2 FNF exporter configuration on a network device:

flow exporter Export-FNF
description FNF v9
destination 10.0.1.100
source GigabitEthernet0/0/2
transport udp 9996
export-protocol netflow-v9
option interface-table
option application-table

3.3 Creating the *flow* monitor

The flow monitor includes the previously defined *flow* record and one or more defined exporters used for the collection and analysis of acquired data. The FNF monitor is configured on a network device as follows:

flow monitor [*flow* monitor name]
description [*flow* monitor description]
record [defined record name]
exporter [defined exporter name]
cache timeout active 60 [export time for "*long-lived*" records]

Example 3 FNF monitor configuration on a network device:

flow monitor Monitor-FNF
    description FNF Traffic Analysis
    record Record-FNF
    exporter Export-FNF
    cache timeout active 60

3.4 *Flow* monitor activation and deactivation

The last step of the configuration is assigning the *flow* monitor to interfaces on the network device which activates traffic monitoring and export of data to the defined external collector. *Flow* monitor activation on a network device is performed as follows:

interface [interface name]
    ip flow monitor [*flow* monitor name] *input*
    ip flow monitor [*flow* monitor name] *output*

Example 4 *Flow* monitor activation:

interface GigabitEthernet0/0/2
    ip flow monitor Monitor-FNF *input*
    ip flow monitor Monitor-FNF *output*

The following commands have to be applied for stopping the export of data and traffic monitoring:

Example 5 *Flow* monitor deactivation:

interface GigabitEthernet0/0/2
    no ip flow monitor Monitor-FNF *input*
    no ip flow monitor Monitor-FNF *output*

3.5 Displaying and removing the FNF record from the exporter

The exporter can display and delete the FNF data directly, as well as remove the collected statistics. Various filters for displaying the target information can be used for displaying data on the exporter.

Example 6 Displaying FNF records on the exporter:

show flow monitor Monitor-FNF *cache*

Example 7 Removing FNF data from the exporter:

clear flow exporter Export-FNF *statistics*
clear flow monitor Monitor-FNF *cache*

4 FNF Collector Installation and Configuration

The *Manageengine NetFlow Analyzer* collector collects and analyses *flow* data acquired from the network device exporter. Advantages of using the *NetFlow Analyzer* software include a scalable architecture supporting a large number of devices, possible centralised and decentralised solutions and a greater number of supported *flow* technologies [3]. The software does not require complex hardware collectors, it can be operated on both *Windows* and *Linux* 32-*bit* and 64-*bit* operating systems and offers a free traffic analysis on up to two interfaces on one or two different devices. It is a commercial solution, meaning that a licence is required for monitoring more than two interfaces (the licence pricing depends on the number of interfaces). The collector can be trialled for up to 30 days without limitations; after the trial period ends the software can be used for monitoring up to two interfaces [4].

The first step in collector implementation is choosing the hardware – the host for the installation of the operating system and the collector. Testing has shown that the *NetFlow Analyzer* operates equally well on both *Windows* and *Linux* platforms. The choice of host depends on the number of monitored interfaces. A greater number of network devices and interfaces require a better hardware configuration. The installation on *Windows* and *Linux* operating systems is equally simple. Following the installation of the operating system and the collector, the flow monitor has to be configured and activated on the exporter from which the data for collector processing is acquired. The configuration is explained in detail in section 3 of this document.

The second step consists of parameter adjustments on the collector. The adjustments include entering the network architecture i.e. the subnetworks (IP Groups) for a simpler review of the collected data.

By using the *NetFlow Analyzer* collector it is possible to gain information on the volume of network traffic, applications, the source and destination addresses participating in the conversation, as well as details of the used protocols, ports and other network traffic characteristics for a given time period, Picture 1 and Picture 2. Apart from this information, the established connections for a chosen host, traffic amount for individual IP Groups, interfaces, percentile representation of used protocols and applications can also be reviewed.

In addition to statistical data, various alerts can be created during the parameter implementation phase which will be described in detail in section 5.

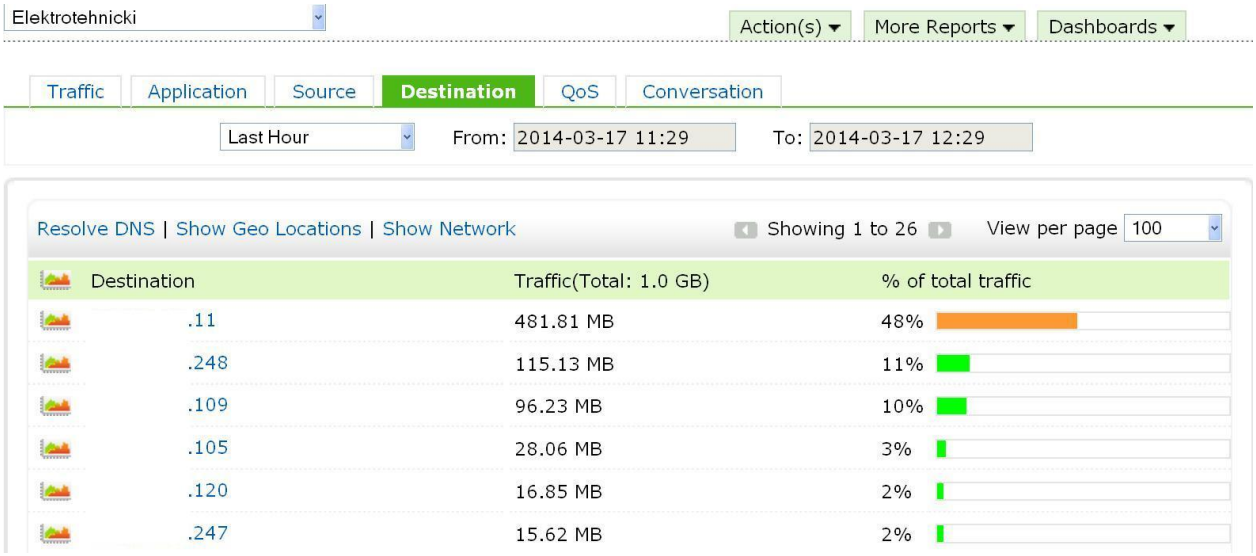


Figure 4.1: Reviewing network traffic

Displaying the selected IP Group from Picture 1 we can notice a host which has consumed 46% of total traffic or 481.81 MB of data. Selecting the IP address displays the list of hosts communicating with the network host and the used communication protocol.

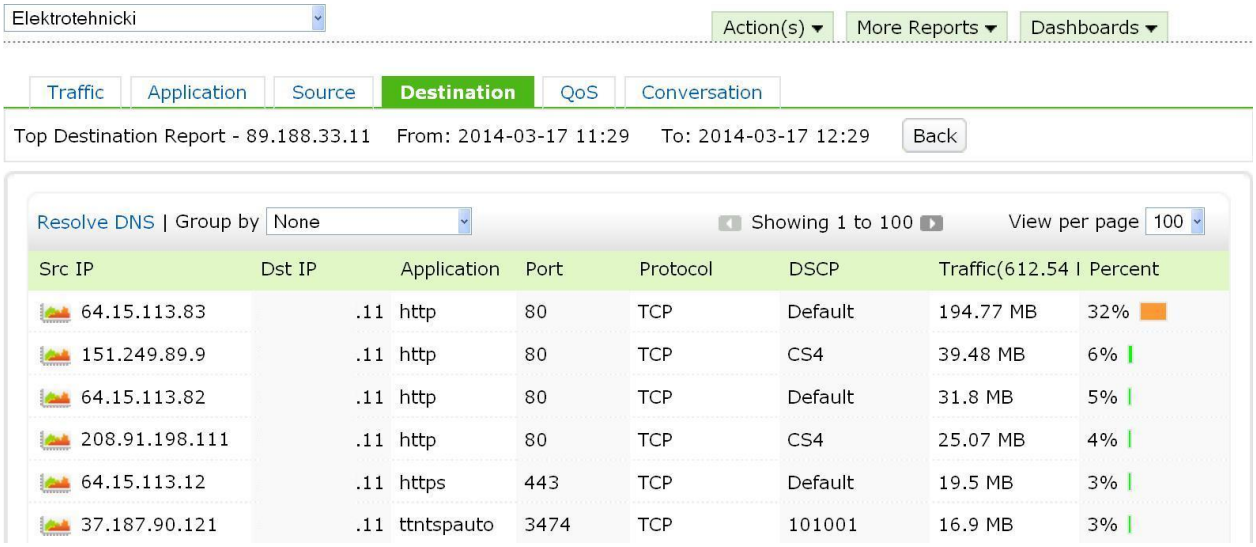


Figure 4.2: Reviewing established connections for the selected host

Selecting the host from Picture 1 and inspecting Picture 2, we can conclude that the host communicates with different hosts on the Internet mostly using standard ports 80 and 443. This example clearly demonstrates how easy it is to identify details of network resource used in a short time period.

Computer network resources are limited and wasted resources can cause a severe performance degradation. Adequate monitoring can shorten the time necessary for the network administrator to discover sources of potential problems. It is possible to determine host IP address, application type, protocol,

consumed traffic in a given time period. Flow technologies enable a quick and simple processing of all notable information and can supply said information using adaptable-format reports, whether monitoring the network in real-time or performing network resource utilisation analysis for different times of day, week or month.

NetFlow Analyzer enables the creation of IP Groups as well as device groups if multiple devices are used for exporting *flow* information, alert profiles for alerting the administrator on notable specific network events, various reports that can be created in specific times of day, week or month, profiles for displaying various information and user accounts [5]. The Device Group Management lets you group tens or hundreds of devices depending on the size of the network and the number of devices exporting data. When monitoring interfaces from different devices as one unit, it is possible to create an Interface Group. Creating Interface Groups is useful for monitoring specific interfaces from multiple devices.

Creating IP Groups lets you display separate monitored subnetworks. IP Groups can be assigned to a specific administrator in charge of a single or multiple subnetworks, while providing the lead administrator with access to all IP Groups and the view of the entire network on a global level.

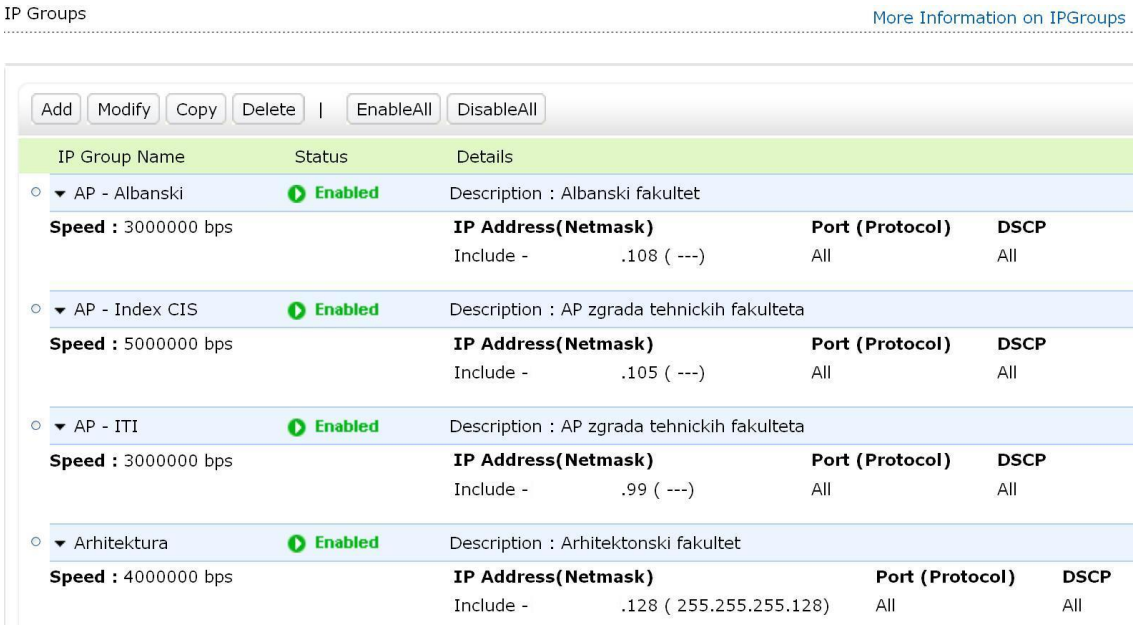


Figure 4.3: Created IP Groups

Subnetworks (IP Groups) whose traffic is monitored from selected interfaces are shown in Figure 4.3. Selecting an IP Group displays the list of IP Group hosts with total generated traffic for each host. Detailed statistics can be accessed by selecting the IP address of a particular host which displays the type of generated traffic, ports and applications used in the communication, as in Figure 4.2.

NetFlow Analyzer provides the network administrators with various functions for achieving significant advantages and saves time during the discovery of network anomalies and so called "*bottlenecks*" critical for network operations.

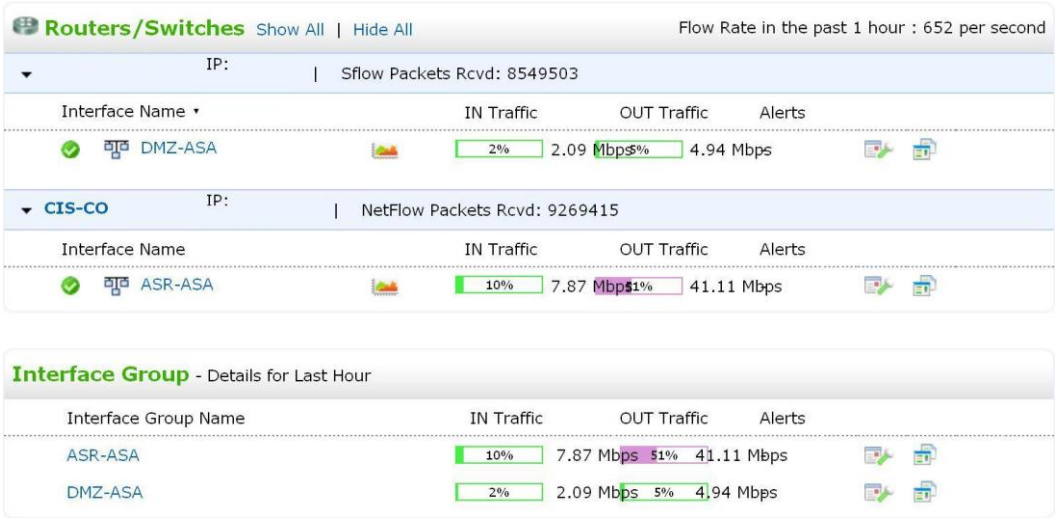


Figure 4.4: Interfaces on devices performing data export

Interface overview for devices performing *flow* data export is shown on Figure 4.4. This example shows data export on two interfaces. Interface traffic can be monitored in any given moment, as well as the number of flow packets received from the collector. Selecting the interface displays detailed statistics.

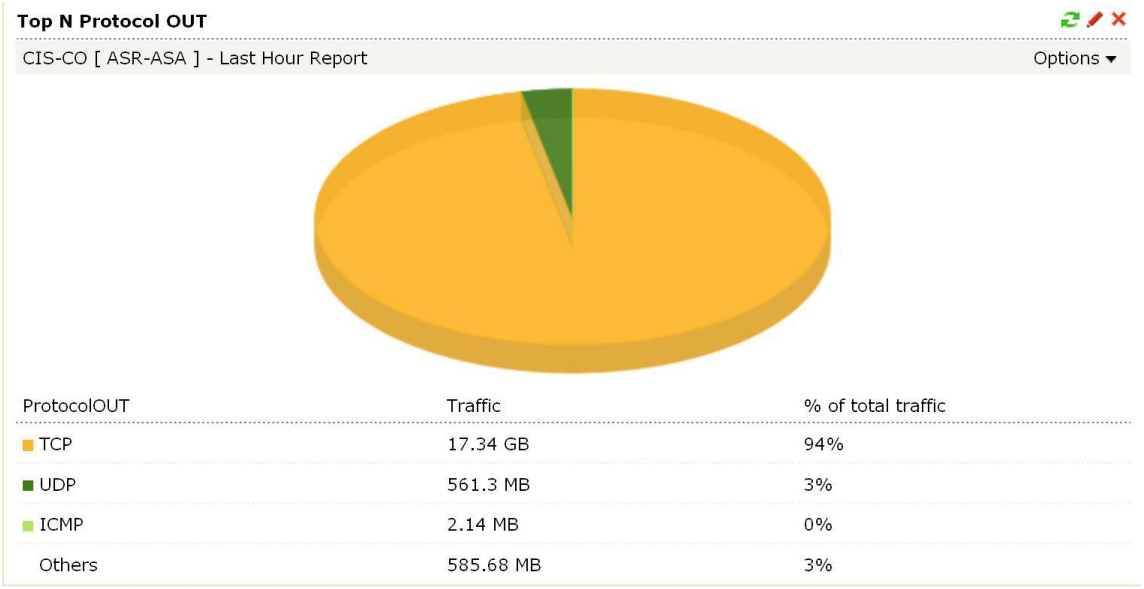
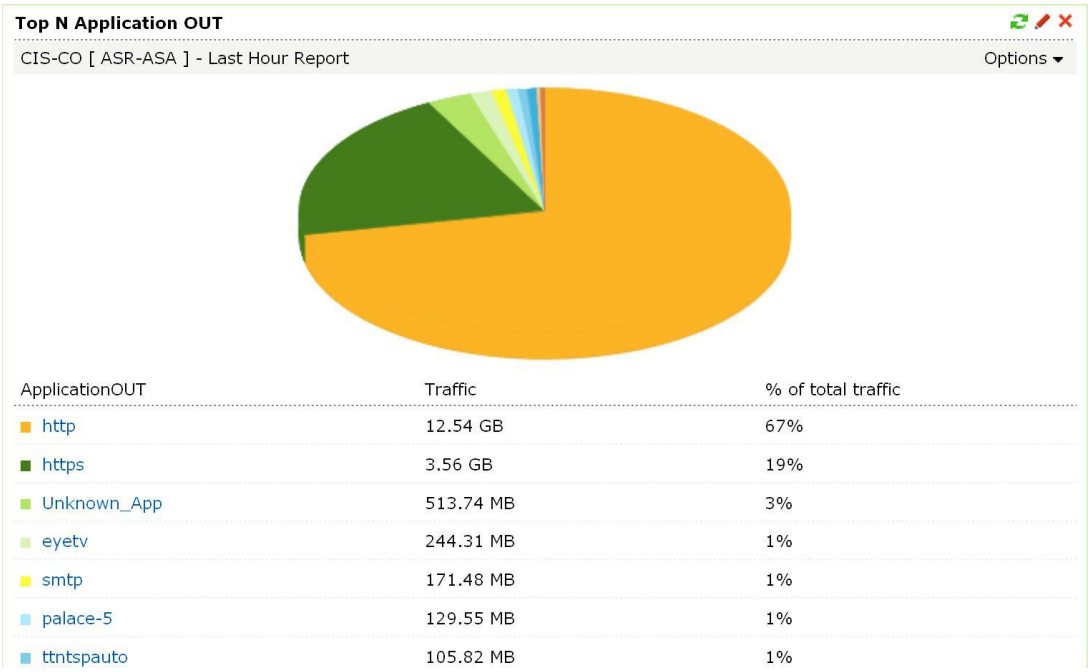


Figure 4.5: Most frequently used protocols

Overview of most frequently used network protocols for a selected time period is shown in Figure 4.5. This example shows that the TCP protocol is used in 94% of established communications, while the UDP protocol is used for only 3% of total generated traffic.



ApplicationOUT	Traffic	% of total traffic
http	12.54 GB	67%
https	3.56 GB	19%
Unknown_App	513.74 MB	3%
eyetv	244.31 MB	1%
smtp	171.48 MB	1%
palace-5	129.55 MB	1%
ttntspauto	105.82 MB	1%

Figure 4.6: Most frequently used applications

Usage of specific applications in a given time period of one hour is shown in Figure 4.6. HTTP applications are responsible for 67% or 12.54 GB of total generated traffic while applications using the HTTPS protocol generated 19% or 3.56 GB of data.

DestinationOUT	Traffic	% of total traffic
.2	2.95 GB	16%
.132	1.03 GB	6%
.108	1.01 GB	5%
.11	675.68 MB	4%
.130	604.04 MB	3%
.109	491.19 MB	3%

Figure 4.7: Top network host conversations

Hosts responsible for generating the most network traffic in one hour are shown in Figure 4.7.

The administrator can select data to be monitored by creating appropriate profiles and filters, as in Figure 4.8.

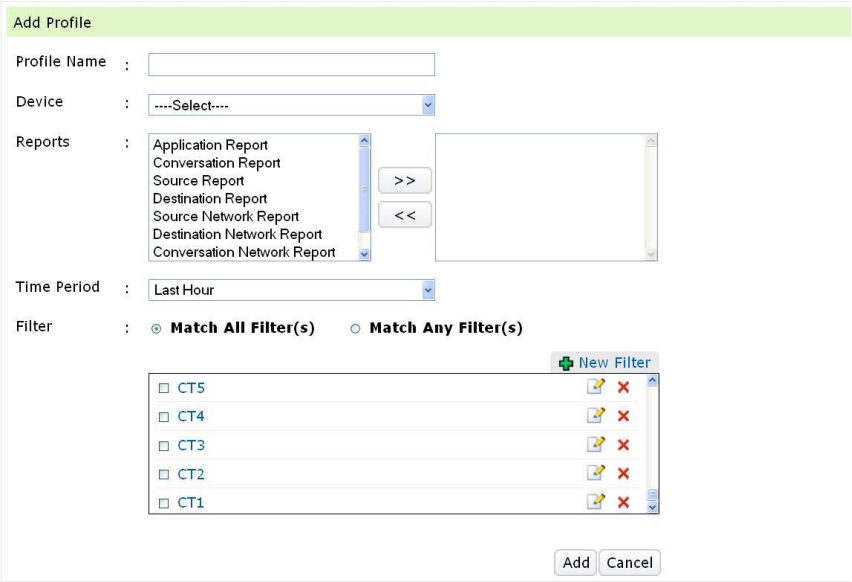


Figure 4.8: Creating profiles for data display

Creating special profiles enables monitoring of only significant information [6]. It is possible to select the device for data monitoring, interface, report type, time period for data monitoring and create appropriate filters. Filters enable the administrator to display data in more detail, extract and display data from a large information pool of and data significant to the monitoring process. He can choose to display only certain applications, source and destination addresses used by those applications, and based on the protocols used in the configuration, Figure 4.9.

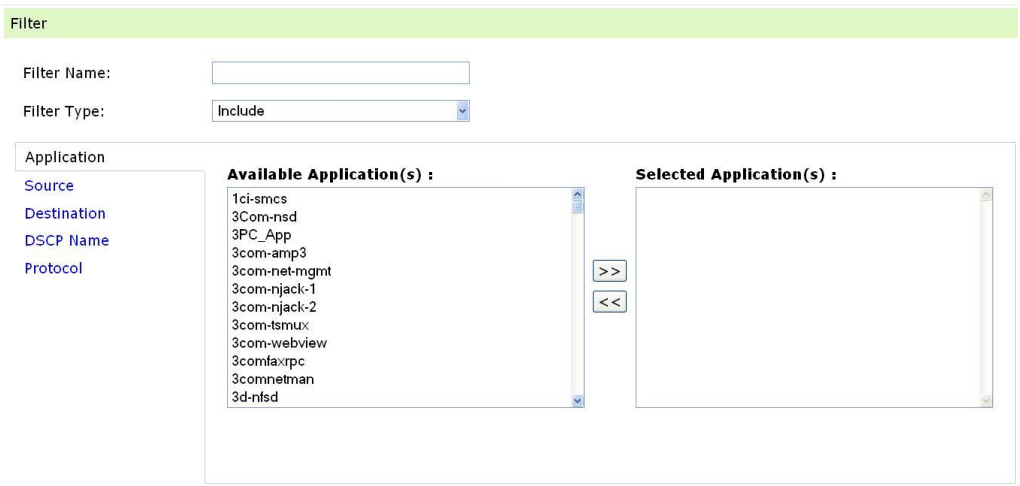


Figure 4.9: Data display using filters

By using the *flow* technology and collectors such as the *NetFlow Analyzer*, it is possible to achieve total control (*monitoring*) of the entire computer network.

Besides the *flow* technology, it is possible to implement other features of network devices for limiting interface, host and protocol flows and achieving control over the network [7][8]. Detailed information on this method and its implementation within the scope of the MREN is presented in section 5.

5 Example of Network Device Management in AMUCG MREN

The following example show the Network Device Management used by Akademske mreze Univerziteta Crne Gore (AMUCG) MREN. The goal of such Management is limiting the flow of IP Group/host which violate the Terms of Use of the Academic network of the University of Montenegro [9].

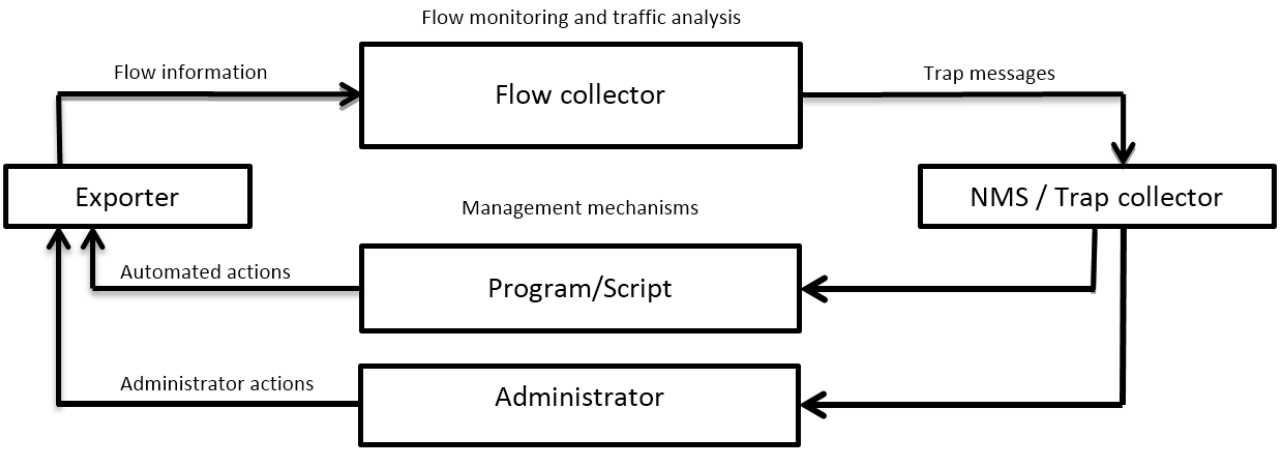


Figure 5.1: Schematic overview of the Network Device Management process

The schematic overview shown in Figure 5.1 is an illustration of the Network Device Management process intended for better understanding the elements, procedures and steps involved in the management process, as demonstrated in the following example.

5.1 Management process

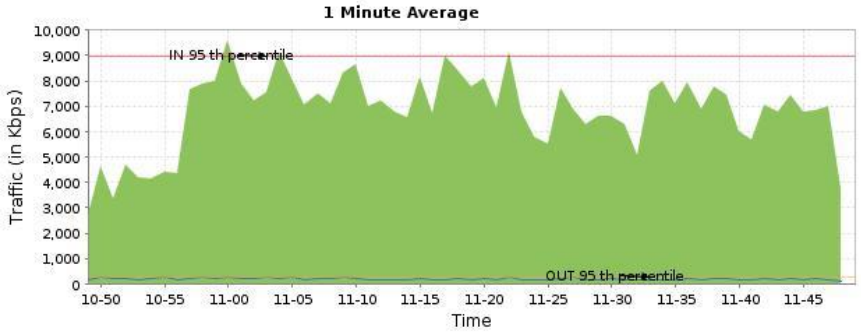
Elements of the displayed system, involved in the process, are: the flow collector, NMS/trap collector, management mechanisms and the exporter.

5.1.1 Flow collector

Flow information from the exporter (router) are forwarded to the flow collector which collects, analyses and displays data. An alert is created at the collector which sends trap messages, based on exceeding configured parameters, to the trap collector [10].

Flow monitoring and traffic analysis on the flow collector

Information flow display for the selected IP Group is shown in Figure 5.2. Based on collected device information, an alert is created on the *flow* collector which will activate the defined action upon the fulfilment of configured parameters.



Category	Total	Max	Min	Avg	Standard Deviation	95th Percentile
IN	3.06 GB	9.55 Mbps	2.68 Mbps	6.69 Mbps	1.48 Mbps	8.95 Mbps
OUT	87.2 MB	275.21 Kbps	92.23 Kbps	190.61 Kbps	38.48 Kbps	258.89 Kbps

Figure 5.2: Information flow display for the selected IP Group

Creating the warning action for exceeding the threshold value

Warning actions for exceeding the threshold value can be created based on multiple parameters previously described in this document. Parameters most frequently selected during the creation of warnings for exceeding the threshold value will be used in the example.

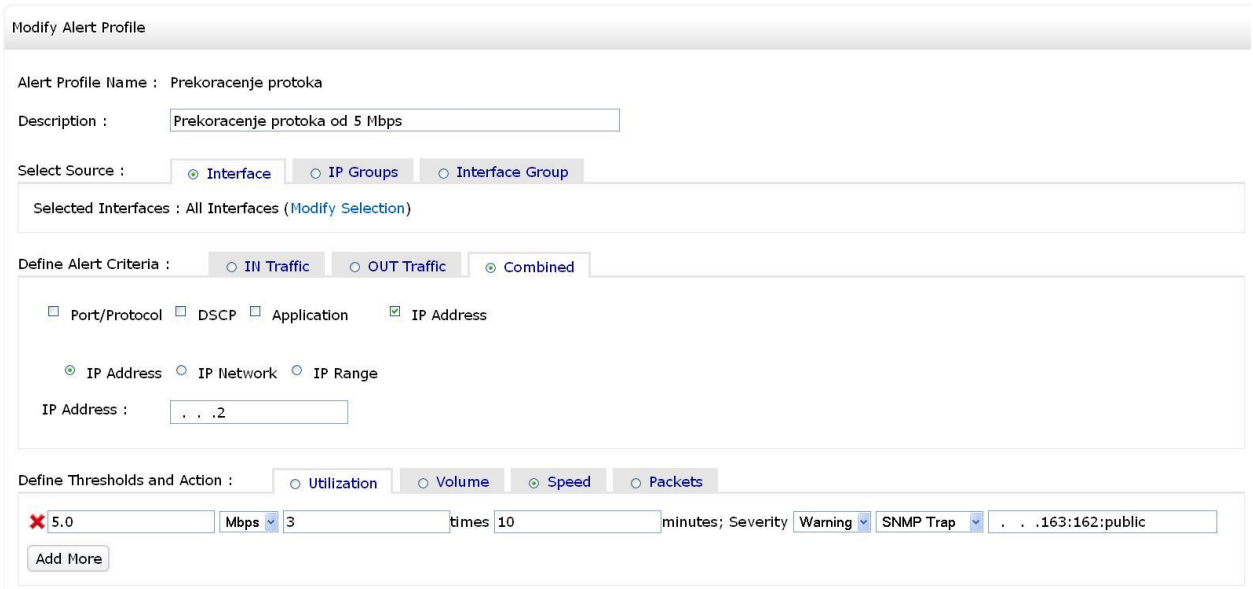


Figure 5.3: Creating the warning action for exceeding the threshold value

Default parameters for forwarding *trap* messages to the *trap* collector are shown on Figure 5.3. Exceeding the threshold flow value of 5 Mbps for the given IP Group will activate the alert. Alerts must be registered at least three times in a time span of 10 minutes to avoid possible false-positive warnings. The selected action sends the trap message to the trap collector at the defined location listening on port 162, as shown in Figure 5.3.

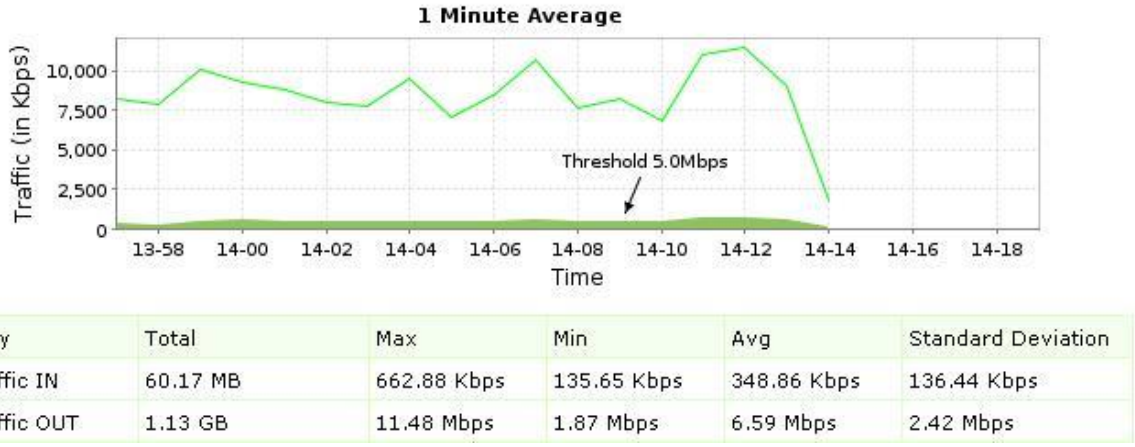


Figure 5.4: Alert for exceeding the threshold value

The alert for exceeding default parameters detailed in the example is shown in Figure 5.4. Exceeding the flow of 5 Mbps, as selected in the example, is registered three times in a time span of 10 minutes and has caused the alert creation and sending of *trap* messages to the *trap* collector.

5.1.2 NMS/Trap collector

The trap collector receives the trap message forwarded from the flow collector and initiates the scheduled actions based on the analysis of the received trap message [10]. Actions can be partially automated (sending e-mail messages to the administrator) and/or fully automated (executing the predefined program and changing device configuration).

Receiving trap messages on the trap collector

The *trap* collector interface with received *trap* messages is shown in Figure 5.5.

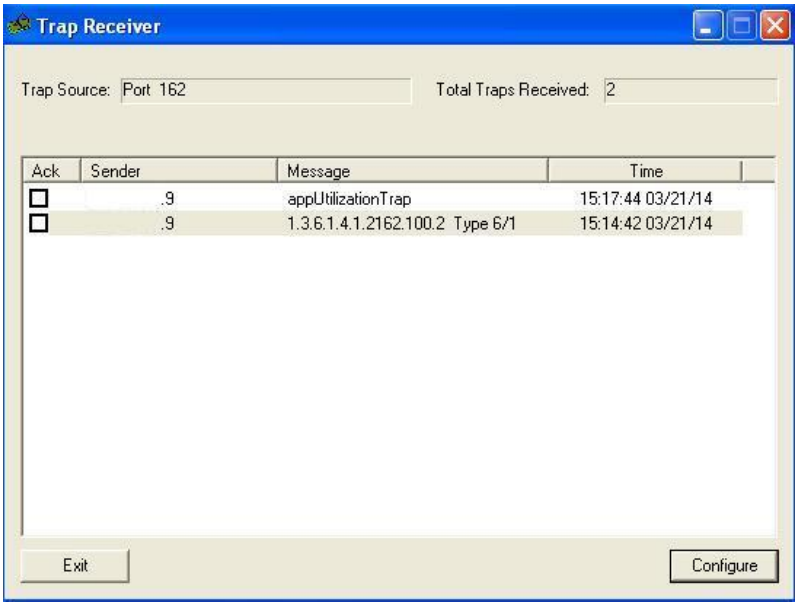


Figure 5.5: *Trap* collector interface with received *trap* messages

The interface is simple and contains only basic fields and information: the port used by the *trap* collector to listen for *trap* messages, total number of received *trap* messages and the *trap* messages containing information on the sender, message type and time of reception.

OID identifiers contained in trap messages

The *trap* collector translates and analyses OID information received in *trap* messages based on the *flow* collector's MIB database data, as shown in Figure 5.6.

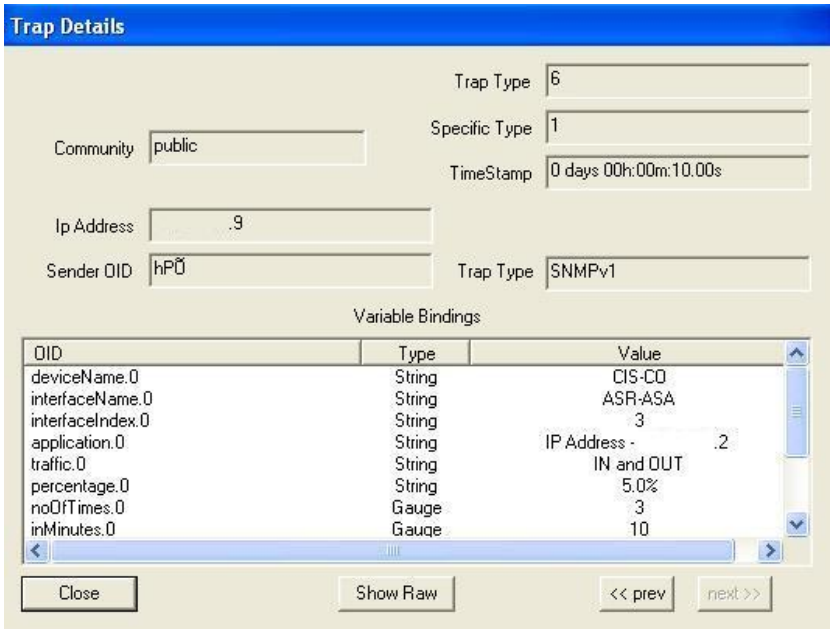


Figure 5.6: Received *trap* message

Trap message identifiers contain values from predefined alerts, see Figure 5.2.

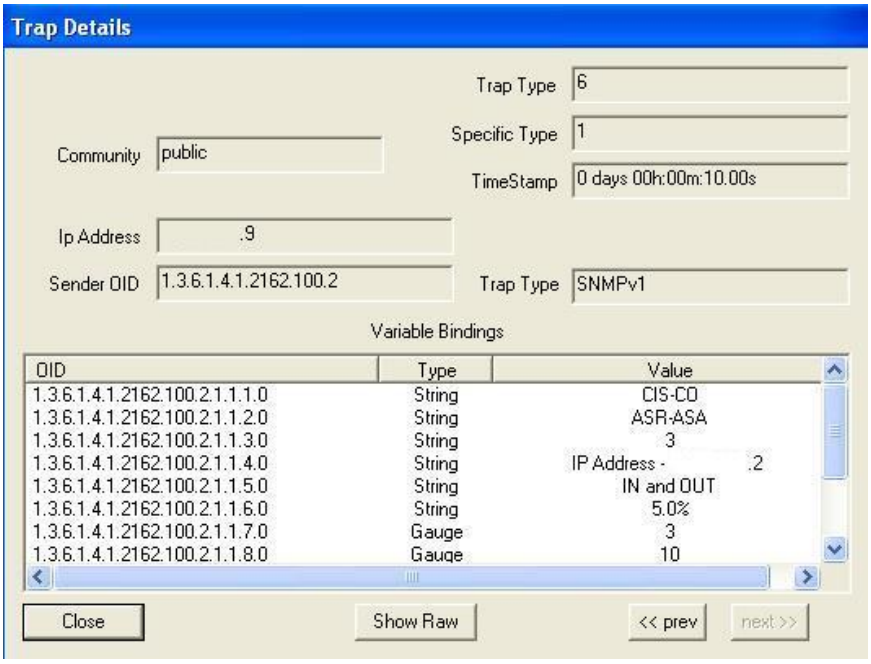


Figure 5.7: Trap message with OID values

Figure 5.7 shows the *trap* message used in the example, but with OID values in their original form, not substituted with values from the MIB database. OID values displayed in this way are difficult to decipher and link to a corresponding OID identifier value.

5.1.3 Management mechanisms

It may be necessary to apply a partially or fully-automated action to the *trap* message received on the *trap* collector.

Defining trap collector actions for the selected trap message

Based on the received *trap* message on the *trap* collector, an automated action may be defined which specifies the program code execution or a partially-automated action which notifies the administrator of the specific alert. The partially-automated action requires more time between the moment an *e-mail* message is sent and the action which involves the administrator making changes to the device configuration.

It is necessary to configure the value of the OID identifier from the MIB database for which the predefined action will be executed. The "*Watch*" field must be set to "*Varbind Value*" for the defined action to be executed based on the selected *trap* message identifier. Field "*Equals*" must contain the OID identifier, value and the equals sign in the format "<OID identifier>:<value>:[<condition>]", as shown in Figure 5.8.

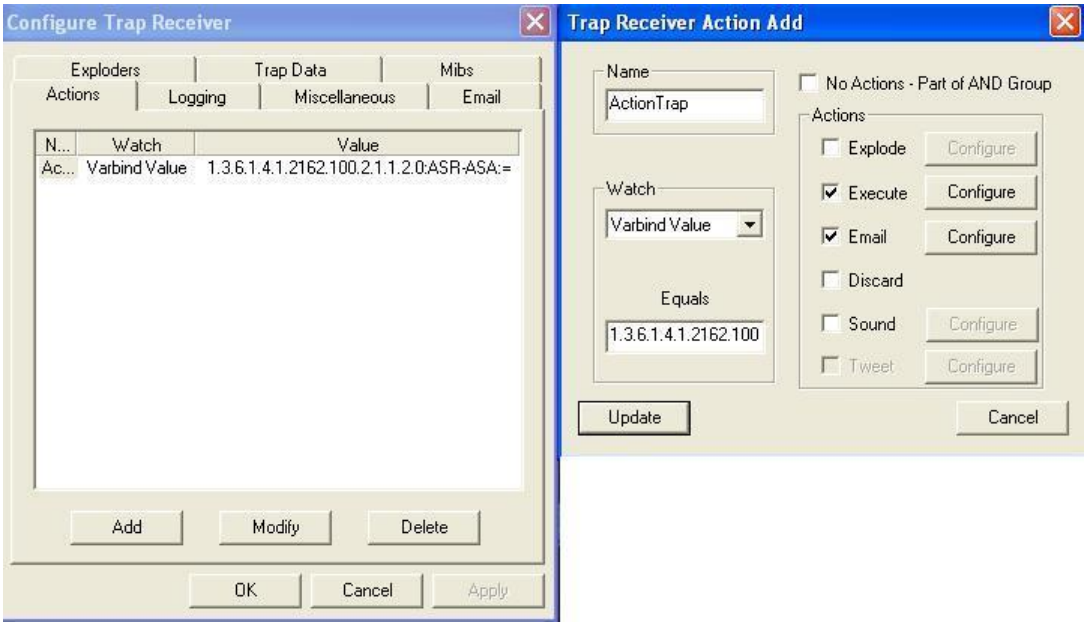


Figure 5.8: Alert definition for the received *trap* message

Automated action program code

Trap collectors can receive trap messages and execute scripts or external programs, but it is not possible to forward a set of CLI commands from the trap collector to the device. To forward such commands to the device, a specific script or program must be created which will perform the authentication to the device using SSH communication, to enter a specific command set and terminate the connection.

To code programs for the automated process of responding to alerts generated by parameters of the *flow* collector requires significant programming expertise in a variety of programming languages. To code scripts in response to received alerts requires a basic understanding of scripting languages, e.g. *Visual Basic Script*.

The PUTTY software is used for establishing communication between the Manager application and devices used in the example. PUTTY is a freeware Telnet and SSH client software. This software provides a quick and easy way for establishing communication with the device and executing proper CLI commands. It also supports the SSH protocol for secure authentication and device communication. The problem arises during the automation of the entire process. A set of commands, to be executed on the device, must be set without the intervention of the administrator. *Visual Basic Script* (VBS) program language will be used in the example to circumvent this problem.

The Visual Basic script (Program code 1) executed by the trap collector must open the PUTTY software, establish communication with the device, successfully create a session using proper access data and execute commands on the device itself, after which it must terminate the communication.

The Visual Basic script contains a few commands. A specific object is created at the start of the script ("WScript.Shell"). Then the session is started using the WshShell.Run command and entering the path to the PUTTY software between quotation marks, and the username and password for accessing the device via the SSH connection [12]. After creating the session, the WScript.Sleep 5000 command is used to input a specific pause of 5000 ms or 5 seconds, until the session is established with the device and the first command can be executed. The WshShell.AppActivate command sets the name of the specific windows which is displayed in the program's title line. WshShell.SendKeys is used for sending CLI commands to the device after successfully establishing the session. The commands can be verified by entering the name of specific keys between curly brackets that have to pressed on the keyboard, e.g. {ENTER}, {ESC}, {DELETE}, or leave an empty space between quotation marks “ ” if the SPACE key is to be used for the command.

Program code 1 is an automated program applied for the received *trap* message with the purpose of limiting the bandwidth of the IP Group using defined ACL policies on the device [15][16].

```
set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run " C:\ putty.exe 192.168.1.1 -l user -pw password"
WScript.Sleep 5000
WshShell.AppActivate "192.168.1.1 - PuTTY"
WshShell.SendKeys "enable{ENTER}"
WshShell.SendKeys "password{ENTER}"
WshShell.SendKeys "configure terminal{ENTER}"
WshShell.SendKeys "int GigabitEthernet 0/3{ENTER}"
WshShell.SendKeys "access-list rate_limit_3000 extended permit ip x.x.x.2 any{ENTER}"
WshShell.SendKeys "access-list rate_limit_3000 extended permit ip any x.x.x.2{ENTER}"
WshShell.SendKeys "exit{ENTER}"
WshShell.SendKeys "exit{ENTER}"
```

Program code 1. Limiting the IP Group bandwidth on the router.

After executing Program code 1, The IP Group will be limited to the specified flow so that further inadequate flow usage will not impact other users of the Academic Network.

With a little skill and understanding of programming basics and IOS CLI commands, it is possible to automate many actions to be executed without the involvement of the administrator, based on the warning messages acquired by the *flow* information analysis on network devices and with the help of the automated Management of Network Devices system introduced in this document.

5.1.4 Exporter

Finally, whether or not the automated actions are implemented or the administrator executes actions on the exporter (i.e. router), the configuration changes because a new entry is added to the ACL list [13][14].

Figure 5.9: Exporter (router) ACL list

The IP Group from the example is added to the ACL list by executing the Program code 1 and the Group's bandwidth is automatically limited, see Figure 5.9.

By applying the suggested example, the flow of the IP Group is limited but the service is still available for use. A rational use of Academic Network resources is made possible for all users by implementing the preceding actions in compliance with the Terms of Use of the Academic network of the University of Montenegro.

6 Application of proposed solution for management in MREN

Members of the Academic Network of the University of Montenegro include: organisational units of the University of Montenegro, scientific, research and educational institutions of Montenegro, libraries, student residences and other non-profit institutions in the service of the academic community in Montenegro.

Based on the Terms of Use of the Academic network of the University of Montenegro, any activity that causes unscheduled or unjustified burden to MREN resources or other networks resources, as well as increased involvement of staff for maintenance of these resources within MREN or using the same, shall be considered as activities that threaten the functionality of MREN and represent unauthorised use of resources.

The current flow of academic network to external links is maximally utilised. Any unauthorised use of resources may cause difficulty in using MREN services. For the aforementioned reasons, it is necessary to carry out continuous monitoring and, with prior warning, to limit the flow to MREN member networks and hosts that do not comply with the Terms of Use of the Academic network of the University of Montenegro.

Flow limitation exemplified in the previous chapter applies in most cases during working hours when the academic network flow is under maximum load. To avoid degradation of network services and possible congestions, the flow is limited for networks and/or hosts that are found to violate the Terms of Use of the Academic network of the University of Montenegro. The application of this rule is stricter during working hours, while outside working hours some of the limitations cease to apply, enabling the networks and/or hosts to use the academic network flow without limitations. The Academic network of the University of Montenegro is responsible for providing Internet services to all its members, however, the principle of flow limitation differs from Internet service providers who carry out limitations on the level of the user. With this method of limitation, each user from his Internet provider receives a guaranteed flow, which in most cases can be used without limitations, depending on the type of the selected service package. In the academic network, this kind of limitation is not possible, for the simple reason, it is not possible to divide the available flow to the user level (host).

Duration, which will be active within working hours, can be limited, whereas access can be limited only for certain services and/or servers. The flow can be limited in one or both directions by activating the set of commands listed in the program codes presented in this document. In addition to the limitations during working hours, it is possible, based on the insight into the amount of traffic generated for a particular month, to set a flow limitation for the network or host until the end of the calendar month in which an excessive amount of network traffic to external links is generated. For traffic generated within the academic network

one should have more understanding and such traffic in most cases is not subject to limitations. However, if it is determined to carry out actions that may jeopardise the academic network, traffic is blocked on the device's firewall.

With implementation of the described system with data obtained by qualitative analysis of network traffic, it is possible to realise both simple and complex management of a computer network by changing the configuration of network devices.

7 Conclusion

Implementing the monitoring system described arose from the need of the administrators of the University of Montenegro academic network for the qualitative analysis of network traffic. Traditional SNMP monitoring has the biggest share in network traffic flow analysis [17][18]. The main drawback of SNMP monitoring is the limited amount of information that can be collected on the generated flow with the aim of solving problems in the network.

During the implementation of the monitoring system, obstacles have been resolved in the form of technology compatibility with available network devices and tested software solutions. After the analysis and testing of several flow technologies, data export with the Cisco ASR 1002 device was selected as the best solution for the University of Montenegro academic network.

Several flow collectors have been tested during work and *NetFlow Analyzer* was chosen, not only because of its functionality, but also for the fact that it was the only tested collector which adequately managed to analyse and display the information obtained by the selected device. *NetFlow Analyzer* is a great solution for data flow analysis with a large number of additional functions.

In the second part of the document, a simple automatic control system was proposed and implemented that, based on warnings received by collector for network traffic analysis; it triggers automated actions for managing network devices and the network as a whole. The recommended system consists of a set of automated procedures combined with direct actions of the administrator.

Examples of automated procedures show that simple and complex computer network management can be realised by changing the configuration of network devices. Some of these examples may be encountered in daily computer network operations and represent routine procedures utilised by the administrator (dynamic flow limiting of the interface/host, port management, etc.).

Finally, the suggested Network Management system, using the *flow* monitoring system, has found its application in the University of Montenegro Academic network and can be used as a basis for developing a more complex system. The system is currently being tested and the application of automated procedures is monitored by the administrators. The continuously changing computer network, as a system, necessitates the implementation of changes, modifications and innovations of program codes forming the base of the management system while making the role of the administrator an irreplaceable one. The application of this and similar computer network management systems can improve the quality of services.

The future of computer networks is unthinkable without the tools for the qualitative analysis of network traffic and their implementation in the management process. The expansion of the computer networks, network applications and number of users creates an increasing need for the best use of resources. The bandwidth of computer networks and other resources are permanently increasing, but are by no means unlimited or free. Each unplanned resource use can create problems in everyday operation of the network and connected services, as well as lead to economic losses. By improving the tools for automated network device management we contribute to the development of smart networks which will reduce or even, eliminate the need for administrator's intervention. These systems, based on the data acquired through qualitative network traffic analysis, can predict and remove a large number of potential incidents in the computer network that would otherwise hinder its functionality.

References

[1] Cisco Systems, "*Introduction to Cisco IOS NetFlow*", Technical Overview, October 2007

[2] Cisco Systems, "*Application Monitoring Using NetFlow Deployment Guide* ", Cisco Systems, Inc., August 2012

[3] Manageengine, "*Enterprise Network Traffic Informatics*", CIO's Hand Guide

[4] ManageEngine, *"University Campus Network Monitoring using NetFlow Analyzer – A case study*", ZOHO Corp., USA, 2010

[5] ManageEngine, "*Bandwidth Monitoring & Traffic Analysis – User Guide*", ZOHO Corp, USA, 2010

[6] ManageEngine, "*Healthcare IT Risk Mitigation – A Network-Centric Approach*", White Paper, NetFlow Analyzer

[7] Čabak Milan, Božo Krstajić, "*Primjer automatskog upravljanja mrežnim uređajima u AMUCG*", XVII Naučno-stručni skup Informacione Tehnologije 2012, Montenegro, Žabljak, February 2012

[8] Čabak Milan, Božo Krstajić, "*Primjer monitoringa i upravljanja računarskom mrežom primjenom FLOW tehnologija*", 19. Telecommunication forum TELFOR 2011, Serbia, Belgrade, November 2011

[9] Centar informacionog sistema UCG, "*Pravila o korišćenju Akademske mreže Univerziteta Crne Gore*"

[10] TrapReceiver.com , "*Trap Receiver User Manual*"

[11] Cisco Systems, "*Understanding Simple Network Management Protocol (SNMP) Traps*", Cisco Systems, Inc., October 2006

[12] Cisco Systems, "*PIX/ASA 7.x: SSH/Telnet on the Inside and Outside Interface Configuration Example*", Cisco Systems, Inc., West Tasman Drive, San Jose, USA, October 2008

[13] Cisco Systems, "*PIX/ASA 7.x and Later: Bandwidth Management (Rate Limit) Using QoS Policies*", Cisco Systems, Inc., West Tasman Drive, San Jose, USA, September 2008

[14] Cisco Systems, "*Configuring QoS",* ASDM User Guide, Cisco Systems, Inc., Chapter 26, September 2006

[15] Cisco Systems, "*Configuring IP Access Lists*", Cisco Systems, Inc., December 2007

[16] Nancy Navato, "*Easy Steps to Cisco Extended Access List*", GSEC Practical Assignment Version 1.2e, SANS Institute, 2001

[17] Cisco Systems, "*Simple Network Management Protocol*", Internetworking Technologies Handbook, Cisco Systems, Inc., Chapter 56, September 2003

[18] Asante Networks, Inc., "*Simple Network Management Protocol – Introduction to SNMP*", April 2005

Glossary

ACL	Access Control List
AMUCG	Akademska Mreža Univerziteta Crne Gore (Academic Network of the University of Montenegro)
CLI	Command Line Interface
FNF	Flexible NetFlow
IP	Internet Protocol
MIB	Management Information Base
MREN	Montenegrin Research and Education Network
NMS	Network Monitoring System
OID	Object Identifier
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TNF	Traditional NetFlow
ToS	Type of Service
UCG	Univerzitet Crne Gore (University of Montenegro)
UDP	User Datagram Protocol
VBS	Visual Basic Script
WAN	Wide Area Network

Complete BPDs are available at http://services.geant.net/cbp/Pages/Home.aspx

campus-bp-announcements@terena.org