Anton Machkasov

Implementation and Testing of QoS Policing

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

April 12, 2016

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages	Anton Machkasov Implementation and Testing of QoS Policing 64 pages + 24 appendices
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Development
Instructor(s)	**Oleksii Fomenko, Experienced Software Developer** **Jarkko Vuori, Principal Lecturer**

This project aimed at creating a software solution capable of translating messages from an IP Operating System to a hardware configuration and creating a testing environment and a set of tests for a router in order to highlight the results of a Quality of Service (QoS), Policing feature work. This thesis covers the implementation and integration of the translation system and a design, the implementation and integration of tests for the QoS Policing feature and processing of the test cases' results.

The feature described in this thesis was developed as a part of a software layer between the IP Operating System and the hardware and was integrated into the solution installed on the new Router 6000 series. The specification of the QoS Policing feature provided a compelling reason for its implementation and integration. One of the most commonly used technologies as a traffic rate control was implemented together with a set of functional tests.

A separate set of functional tests for calculating the error ratio of the policer with various burst sizes configured was implemented along with the set of tests for corner cases of the QoS Policing configuration. All gathered data were processed in order to explore the regression between the main parameters of a policer and traffic such as burst size, rate, running traffic rate and packet size.

Keywords	QoS, Policing, implementation, testing, verification, IXIA, Router

Contents

List of Abbreviations

LAN	Local Area Network
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
MAC address	Media Access Control Address
IP	Internet Protocol
IPv4 address	Internet Protocol Version 4 Address
IPv6 address	Internet Protocol Version 4 Address
OS	Operating System
PVC	Permanent Virtual Circuit
XC	Cross Connect
SI	Service Instance
LAG	Link Aggregation Group
SNMP	Simple Network Management Protocol
MPLS	Multiprotocol Label Switching
OSI	Open Systems Interconnection
L2	Layer 2 of OSI Model
L3	Layer 3 of OSI Model
QoS	Quality Of Service
QoSP	Quality Of Service, Policing
SEK	Swedish Krona
LAA	License Assisted Access
LTE	Long Term Evolution
MIT	Massachusetts Institute Of Technology
ARPA	Advanced Research Projects Agency
MPLS EXP	Multiprotocol Label Switching experimental bits
GUI	Graphical User Interface

1 Introduction

The Quality of Service (QoS), Policing (QoSP) is a core feature of any modern router developed by companies such as Ericsson, Cisco, Juniper, Huawei and others. Its functionality concentrates on providing certain level of bandwidth to the customer with certain accuracy. This thesis concentrates on the implementing of the Quality of Service, Policing feature and a set of automated tests for it for the new Router 6000 series. There are two types of the test cases that will be implemented:

1. Test cases that check the functionality of the feature
2. Test cases aimed at the improving of the check precision of the policing feature and how accurate it is in conjunction with the hardware specifications.

The first type of testing cases aimed at testing the combinations of the feature with different types of configuration units. The second type of test cases is performed on all the possible combinations of the feature's parameters in order to explore the dependencies between those parameters along with the feature's accuracy. The received results are investigated with the regression analysis and embedded to the first type of the test cases. This is done in order to control that the quality of the provided service is on the same level for each update of the third party software. The automation of the tests is a term describing the fact that the test case was developed and written in a form of a script or another kind of a software. It is implied that the running of a test case does not require any usual activity from a developer and done automatically. All the results should be collected and reported by a test case and processed by an automation tool. On the basis of the above it can be concluded that the automation tests serve as continuous control of the software quality, the stability of the final version of the software and for minimizing the human hours spent on testing. The ultimate goal of the thesis is an implementation of the QoSP feature with high quality and stability and the integration of the test cases checking and controlling the accuracy and the functionality into a testing loop.

2 Company Overview

Established in Sweden, Ericsson honourably holds a name of the world's leading provider of communications technology. For almost 140 years, Ericsson has been creating equipment, software and services to enable transformation through mobility. Ericsson is dedicated to provide a whole spectrum of services and technologies related to the networking field such as television and media management, managed services (including designing, planning and building a network to manage day-to-day operations), communications services, networks, support solutions (software solutions for operations and business support systems), communication services, mobile broadband and others. The company has business connections worldwide and as a result, Ericsson can report net sales of more than SEK 200 billion and operating income of more than SEK 6 billion. [1.]

Ericsson is a big international company oriented to take leading positions in all segments of mobile and network areas. It makes significant investment into developing new technologies and products and that fact is reflected in more than 37000 patents owned by Ericsson and more than 25000 Research and Development employees. More than 40% of the world internet traffic goes through the Ericsson networks and it constantly grows. [2.]

Humanity is entering a new era of information technology. According to a prediction of Ericsson, over 50 billion devices will be connected to the internet of things by 2020. [3.] Such large number of devices connected to the Internet will cause a significant growth of traffic. To make it feasible the society needs to have a platform that is properly designed and implemented with the latest technologies. This, in conjunction with a major need of network for rapidly developing technologies, puts Ericsson at the innovation front of providing a platform to provide further development of innovation processes.

Constant development and core values such as respect, professionalism and perseverance allow Ericsson to grow in the European markets and constantly upgrade provided services and line of products. During Mobile World Congress 2015 Ericsson presented new solutions and products which were: Media Delivery Networks, Digital Telco Transformation, Expert Analytics 15.0, Hyperscale Cloud, Connected Maritime Cloud, Connected Traffic Cloud, App Experience Optimization, Networks Software 15B, LTE LAA,

Ericsson Router 6000 Series and Ericsson Radio System. [4.] All of these solutions together provide an opportunity to build networks of a new generation and a reliable platform developing the Network Society.

One of the products presented in the Mobile World Congress is an Ericsson Router 6000. It is a critical component of the Ericsson Radio system and closely integrated with Ericsson Radio and Microwave in order to provide high quality service for a mobile backhaul. This series runs IP Operating System (IPOS). Router 6000 provides high-density 10G interfaces and supports VPN services over IP/MPLS networks and extensive quality of service features. [5.]

3 Technical background

Internet has changed the computer and communication world like nothing before. In 1962, a scientist from MIT and ARPA named J.C.R. Licklider proposed an idea called a “galactic network” of computers that could talk with each other. Already in 1965 an another MIT scientist developed a way of sending information from one computer to another that he called “packet switching” and in 1969 the first message was delivered from one node to another. By the end of 1969, just four computers were connected to the ARPAnet, but the number and the types of devices connected to the Internet (ARPAnet) constantly grew and the number of services based on it increased significantly and finally the term “Internet of Things” (IoT) was invented. The IoT is the concept that describes a future where everyday physical objects will be connected to the network and will be able to communicate with other objects of this environment. [6.] The amount of traffic is directly proportional to correlate with the number of connected devices and finally engineers faced the situation where they had to manage traffic volumes effectively. To solve that problem, devices called routers and a set of routing protocols has been developed.

Routers solved the problem of effective message delivering between nodes from different networks, but with the increase of media traffic and number of customers, the internet providers were faced with a problem that the capacity of the router’s links was not enough to allow data access for all the customers without any restrictions. To solve that problem, the term “Quality of Service” has been introduced. QoS is a collection of technologies which allows applications to request and receive a predictable level of service in terms

of data throughput capacity (bandwidth), latency variations (jitter), and delay. The policing is a part of QoS and aimed at providing the guaranteed bandwidth for a customer and restricting the amount of traffic they consume per unit of time, which allows to distribute link capacity between users and manage traffic streams more effectively. [7.]

3.1 Quality of Service

Before defining QoS as an entity it is necessary to understand the semantic load of it. The term Quality of Service consists of two main entities: quality and service. The quality can be related to many different properties in networking, but engineers generally use quality as a description of the process that allows service provider to deliver data in a reliable way with defined restrictions or within the way that is better than normal. This process includes the ability to define the most optimal use of network resources (such as highest efficiency of the circuit bandwidth or the shortest distance between two endpoints). [8.] The term Service introduces certain uncertainty: it can have several meanings depending on how company of business is structured. People usually use the term service to describe something offered to the end-user or customer of any network: for example, end-to-end communications or client-server application. The term service can cover a wide range of offerings, from e-mail to video, from web browsing to chat rooms. In multiprotocol networks service often has other definitions. In a Novel NetWare network every single service advertisement protocol considered as a separate service. In other situations, services can be defined based on different protocol suites, such as SNA, DECnet, AppleTalk etcetera. [8.] Another granularity that can be applied to service is different types of traffic going through the particular node and different types of interfaces in which QoS can be configured. Taking into consideration the above mentioned possible definitions of terms “Quality” and “Service”, it is possible to specify a definition for Quality of Service as a capability to provide better service to selected network traffic over various technologies and all types of interfaces on the node. [9.] QoS allows providing better service to certain flows. It can be reached by either raising the priority of a certain flow or reducing the priority of other flows.

3.1.1 QoS Architecture

Standard top-level QoS architecture defines three fundamental segments for QoS implementation. The first piece of QoS architecture is identification and marking techniques for coordinating QoS from end to end between network elements. The second part is QoS within the single network element (queuing, shaping, traffic-shaping tools and rate limiting). The last part consists of QoS policy, management and accounting function to control and administer end-to-end traffic through the network. [9.] All the three segments of the QoS architecture shown in figure 1 below.

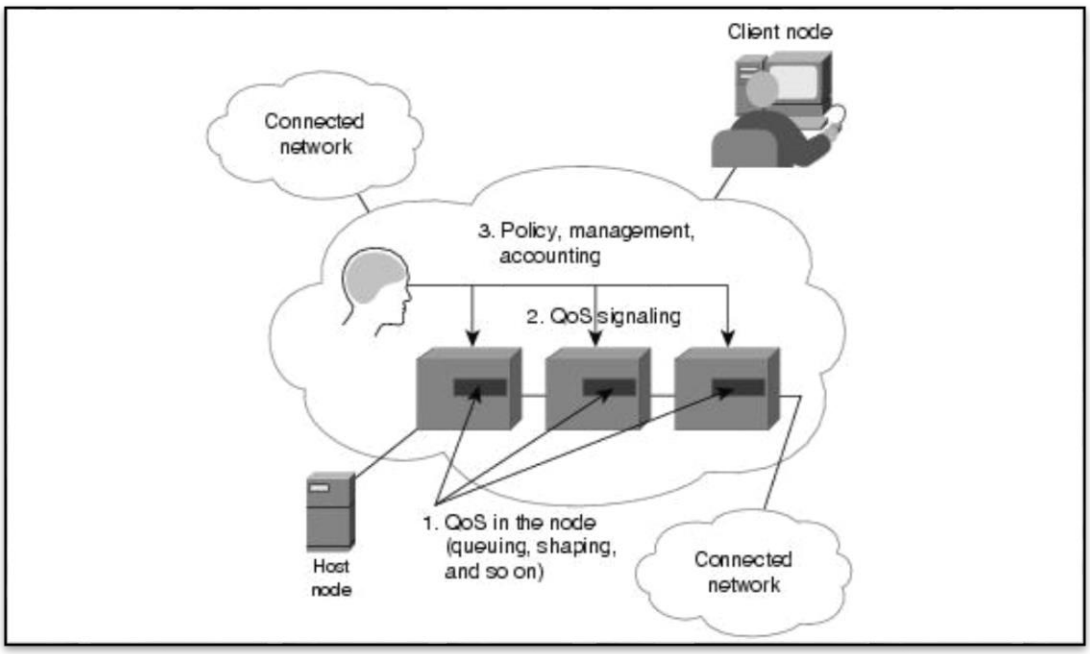

Figure 1. A basic QoS Implementation. Reprinted from Quality of Service Networking (2012) [9]

3.1.2 QoS Identification and marking

The first part of the QoS architecture is the QoS Identification and marking. It can be achieved through the classification and marking of a packet that belongs to a particular traffic flow. After a packet has been classified it can be marked or not. In case it is not

marked, the classification defined as a per-hop based process. The per-hop based classification relates only to the device that it is on, without passing mark on the packet to the next device. In this case just internal services of the single router are able to see the mark of the packet defined by the QoS service. Common use cases of per-hop classification are priority queueing (PQ) and custom queueing (CQ). In the situation when the applied to the packet mark visible on the next router, the IP precedence bits installed. Common methods for identifying flows are: access control lists (ACLs), a policy-based routing, a committed access rate (CAR), and a network-based application recognition (NBAR). [9; 10.]

In general, the classifying and the marking of the traffic consists of two steps:

- A packet must be identified and after that classified into the specific group.
- A packet must be marked on the trust boundaries.

The packet can be classified based on various criteria called traffic descriptors, which can include:

- type of application
- source and destination IP addresses
- ingress interface
- class of Service value in an Ethernet header
- Type of Service value in an IP header (this part of IP header called DSCP or IP Precedence)
- MPLS EXP in a MPLS header. [11.]

A class of service, a type of application and MPLS EXP traffic descriptors are out of the scope of this project, and for that reason they will not be discussed in this chapter. Source and destination IP addresses are parts of the IP header of the packet. Its positions are shown in figure 2 below. [12.]

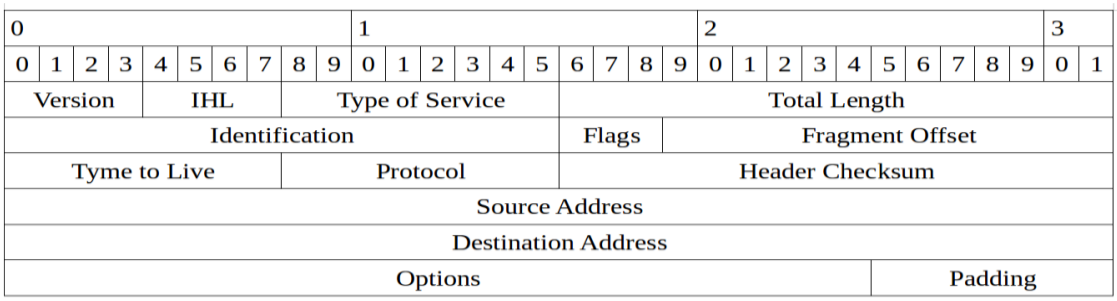

Figure 2. Internet Protocol Datagram Header. Reprinted from RFC791 (1981) [12]

Figure 2 shows that the size of the source and the destination IP addresses is 32 bits each (4 octets). Besides the IP addresses there is a number of other parameters that should be taken into account, i.e. the protocol and the identification fields.

The type of service (ToS) represented by 8 bits in the IP header of a packet and used for providing abstract parameters of the chosen quality of service. Based on its value the actual service parameters will be selected while transmitting a packet through a network. If network offers service precedence (part of ToS field in the packet IP header), which defines that high precedence traffic flow as more important than other traffic, it means in a common case that the node accepts just traffic with the precedence higher than a certain value at the time of a high load. As shown in figure 3 the traffic precedence takes the first three bits of the ToS field.

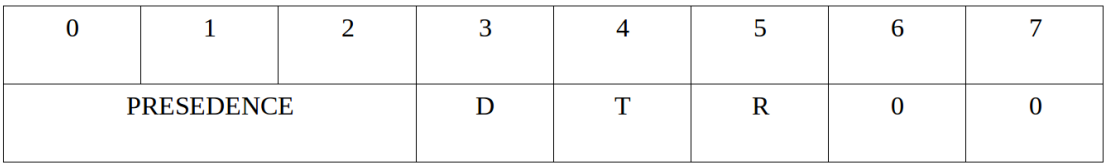

Figure 3. Type of Service field in IP header (Old version). Reprinted from RFC791 (1981) [12]

The rest of the bits represent such desired parameters of the service as delay (D), throughput (T) and reliability (R), the bits 6 and 7 are reserved for future needs. [12.] This type of usage of the ToS bits was defined in RFC 791 in 1981, but the modern redefinition

of the ToS field is a six-bit Differentiated Service Code Point (DSCP) field and two bits Explicit Congestion Notification (ECN). [12; 13; 14.]

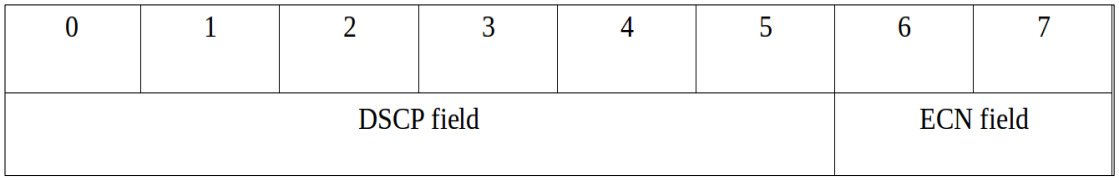

Figure 4. Type of Service field in IP header (Modern version). Reprinted from New Terminology and Clarifications for Diffserv (2002) [14]

The first three bits of DSCP describe the Class Selector of the packet, which is backwards compatible with the IP precedence described above. The last three bits of DSCP assigned to the drop precedence of the packet. Commonly used DSCP values are shown in table 1 below.

Table 1. DSCP and Precedence values. Data gathered from Quality of Service Configuration Guide, Release 4.0(4)SV1(1) [15, 6-1]

DSCP Value	Meaning	Drop probability	Equivalent IP Precedence Value
000 000	Best Effort	N/A	000 - Routine
001 010	AF11	Low	001 - Priority
001 100	AF12	Medium	001 - Priority
001 110	AF13	High	001 - Priority
010 010	AF21	Low	010 - Immediate
010 100	AF22	Medium	010 - Immediate
010 110	AF23	High	010 - Immediate

011 010	AF31	Low	011 - Flash
011 100	AF32	Medium	011 - Flash
011 110	AF33	High	011 - Flash
100 010	AF41	Low	100 - Flash Override
100 100	AF42	Medium	100 - Flash Override
100 110	AF43	High	100 - Flash Override
101 110	High Priority Expedited Forwarding (EF)	N/A	101 - Critical
001 000	CS1		1
010 000	CS2		2
011 000	CS3		3
100 000	CS4		4
101 000	CS5		5
110 000	CS6		6
111 000	CS7		7

As shown in table 1, there are several groups of the DSCP values that can be determined. The first group is AF, which stands for Assured Forwarding. The AF is a state for a class name and encrypts the class selector and the drop precedence of the packet in numbers following by AF abbreviation. Table 1 shows clearly that there are four classes with three different drop precedence values for each of them. Two additional classes are Best Effort or Default class and EF class are also included into the list of possible values of the DSCP field.

There are several differences between the AF classes and two additional classes, which reflected in not having the drop precedence and describing two corner cases of different types of traffic. In case of the default class the packet proceeded within the default actions and in case the EF class the packet proceeds through the bandwidth reserved for the latency sensitive real-time and interactive data.

3.1.3 QoS Within a Single Network Element

The second part of the QoS architecture is the QoS within a single network element. It includes congestion management, queue management, link efficiency, and shaping/policing tools. Congestion management is a tool that allows controlling the traffic flow congestion by specifying the priorities of the packets sent out an interface. The congestion management mechanism encompasses the creation of queues, assignment of the packets to those queues based on the classification of the packet, and scheduling of the packets in a queue for transmission. Queue management is an important part of QoS architecture solution because queues on devices are not of infinite size and can fill and overflow. When a queue is full any new coming packet cannot get in and will be dropped. This situation is called an end-drop. The problem with this scenario is that the router cannot determine that the packet is of a high priority and cannot prevent dropping. To solve this problem Weighted Random Early Detect (WRED) was introduced. WRED provides two mechanisms to support proper packet dropping without a significant effect on the high-priority packets. These mechanisms make sure that there is enough space for high priority packets in a queue, which means that the queue is not completely filled by the low-priority packets. Also these mechanisms provide criteria that allows to ensure that the device tries to drop the low-priority packets from the queue before dropping the high priority ones. This approach allows to save capacity for the high-priority traffic. The shaping is a technology that allows limiting the bandwidth of the particular flow in order to prevent the overflow problem. The policing is a similar technology to shaping but it differs in one important way: traffic that exceeds the specified rate will be dropped, but in shaping it will be buffered. [16; 17; 18.]

3.1.4 QoS policy, management, and accounting functions

The third part of the QoS architecture consists of three technologies: QoS Management, End-to-End QoS levels and Classification-Identifying flows.

The QoS management is a set of methods aimed at setting and evaluating of the QoS policies and goals. A common technique involves the following steps:

- In order to determine the traffic characteristics of the network, fill it with the devices of RMON probes type.
- After the traffic characteristics has been received and an application has been addressed for higher requirements of QoS deploy the QoS techniques.
- Analyze the data received from testing in response of the targeted applications to determine whether the QoS goals have been achieved.

The End-To-End QoS levels term refers to the capability of the network to provide a service needed by the specific network traffic from end to end. The main difference of services here is related to the strictness of the QoS, which is described by how hard the service can be bound by specific bandwidth, delay, jitter, and loss characteristics. There are three basic levels of End-to-End QoS service:

- best-effort service
- differentiated service
- guaranteed service. [19.]

The best-effort service is a communication service that does not provide any feature that will allow recovering lost or corrupted packets during that transmission phase. “Best-effort traffic” in the current global networks uses end-to-end transport protocols such as TCP, UDP, etcetera, with minimal or without requirements of the network to the resource allocation. [20.] Although the simple best-effort traffic has several problems, for example it cannot guarantee that all the packets will be delivered, it also has several useful characteristics like minimal technical requirements on the infrastructure and minimal requirements in terms of economic infrastructure. Besides the useful characteristics, there are several services that are tolerant to packets loss and accuracy, but sensitive to timeliness. For-example in real-time audio/video stream the loss of a small percentage of the packets is not noticeable for the end-user, but time delays caused by recovering of cor-

rupted packets or resending lost packets definitely decrease user satisfaction in the service. Best-service can be described as a FIFO (first in, first out) queue, which has no differentiation between flows. [20; 16; 21,23-25.]

The differentiated service also called as a soft QoS provides an ability to differentiate the traffic flows. The differentiated services architecture based on several principles for traffic flows maintenance. It is possible to summarize them in the following list of actions:

- At the point of entering to the network the traffic should be classified.
- After the classification the traffic should be assigned to the different behaviour aggregates.
- A behaviour aggregate identified by a single DS point (DS point is a differentiated service code point described above).
- Inside the network packets forwarded according to the per-hop behaviour associated with the specific DS code point. [22; 23.]

The hard QoS or guaranteed service refers to an absolute reservation of the bandwidth for the specific traffic flow. It is based on using of resource reservation protocol (RSVP) which allows dynamic reservation of the network resources like a bandwidth and a latency for end-to-end communication. [24.]

3.2 Router Overview and Market Place

The Internet allows our society to store infinite amount of data and content. Exchanging the information increased significantly during last years with the growth of mobile devices connected to the global network. According to Ericsson's prediction the number of mobile devices connected to the IoT would reach 6.1 billion and the overall number of connected devices would reach 50 billion by the end of 2020. The increased number of devices connected to the IoT along with new mobile technologies such as 5G has led to the situation where increasing the router's capacity has become one of the most important aims in order to meet the expectations of end-users about the quality of the Internet connection and the available speed of data transmission.

Under these circumstances, Ericsson presented the new Router 6000 series, which is a key element in Ericsson's next generation portfolio. Ericsson launched the Router 6000 series in order to enable transformation to LTE Advanced, 5G and M2M. It also satisfies

needs for the high-performance backhaul and the metro access networks required by new technologies. As one of the core elements of Ericsson's next generation portfolio the Router 6000 series uses the uniform IP network operating system (IPOS) running on the extensive suite of platforms, covering cell-site and edge routers, mobile core and data centres. Moreover, this series tightly integrated with Ericsson Network Manager (ENM) and other Ericsson's products and meets the requirements of the next generation technologies (LTE Advanced, 5G). [8.] The Router 6000 series supports a multi-standard, multi-band, multi-layer architecture and can be integrated into the new Ericsson radio system. This router introduces the first access router with 100GE interfaces in a single rack unit in industry. Among other features of the router, there is the QoS Policing which allows restricting the traffic for certain users and provides an ability to maintain the bandwidth with high effectiveness. [5; 6.]

3.3 Operating System

An operating system (OS) is a software that acts as an intermediary between a user and a computer system hardware. The main role of any OS is providing an environment in which user can execute applications in a convenient and efficient manner. [25.] Operating systems significantly vary in the way they accomplish their tasks. Different purposes of the hardware where they are installed lead to different architectural solutions. For example, OS for mainframe will be designed mainly for optimal utilization of hardware resources, OS for personal computer will support enormous number of different software written by developers from different companies and should provide a good level of abstraction from hardware and so on. This chapter focused on OS for routers and their architectures.

3.3.1 Cisco Operating System

Cisco Systems has its own version of an operating system for routers, which called an IOS. Originally, the IOS was designed to be a small embedded system. With the development of Internet technologies and growth of routed networks the number of Internet protocols grew significantly and the demand for new functionality of the routers was satisfied by Cisco by adding new features to its operating system. [25.] The IOS was designed to stay lean. That means that it should stay with the hardware constraints of their

original platforms. The first family of the routers had limited resources such as memory and CPU (central process unit) bandwidth for a packet switching. This fact was the result of the simplified architecture compared to the usual Linux based operating system's design. Simplified architecture means that many safeguards like inter-thread memory protection mechanisms are missing from the IOS because of the CPU bandwidth that they consume and the memory overhead. As a result of these pays off the IOS design emphasizes speed at the expense of extra fault protection. [25.]

As shown in figure 5 there are six main components in Cisco IOS. It contains processes, packet buffers, a fast switching software, a kernel, device drivers and a hardware.

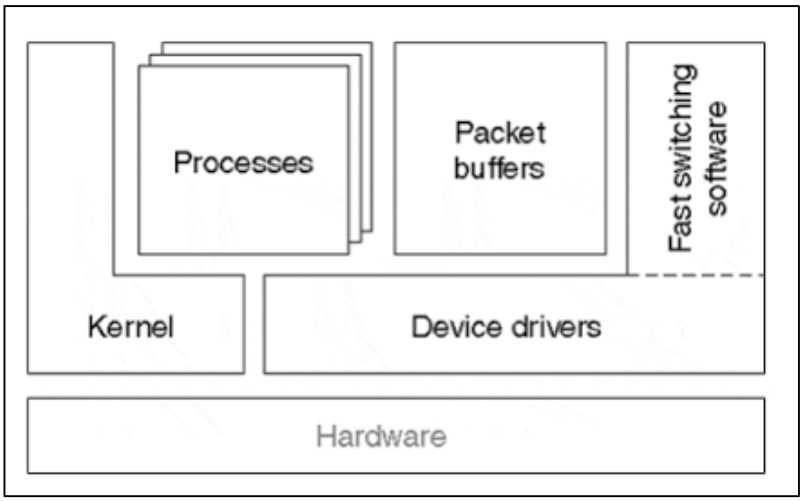

Figure 5. Cisco IOS general architecture. Reprinted from Inside Cisco IOS Software Architecture (2000) [26, 10]

The processes are presented by the separate threads and joined data that work on the different tasks, such as system maintenance, switching of the packets, controlling and configuring routing protocols, configuration of restrictions for traffic flows, logging and others. [25; 27.] The kernel is responsible for memory management and process scheduling and can be described as a software that provides hardware resource management to the processes. The role of the kernel in the IOS is mostly the same as in all Linux-based operating systems: it provides the necessary level of the abstraction in order to hide a low-level hardware management details from a system or a software. [25; 28.] The packet buffers used to store switched packets and implemented as global memory buffers. The device driver functions as an interface between the IOS kernel, the processes, the hardware and the fast switching software and aims at the controlling the network interface, the hardware and peripherals. The fast switching software is a very

foundation of the router. The operation of switching of a packet consists of four basic steps:

- Receiving a packet on an interface.
- Extracting the packet's destination address and comparing it with the list of known addresses.
- If a match is found, the router forwards packet to the appropriate interface, otherwise the packet is dropped.

As is easily visible, the steps are straightforward, but the main challenge in this process is not its complexity. The main challenge in this case is related to the fact that all presented steps are data-intensive operations and it means that all these four steps should be accomplished quickly. The main problem with the data-intensive operation is that increasing the CPU speed is not enough as it depends on bus performance and data memory speed. [25.] One of the possible solutions is to use specific switch chip that is optimized for such kind of operations. With the using of the separate switch chip the configuration from the IPOS command line interface (CLI) downloads to the chip and most of the packets avoid re-sending to the CPU in order to find a destination address. Instead of that it performs looking up of the destinations address on the switch chip that optimized just for that operation and able to provide switching of several hundred thousands of packets per second. [25.]

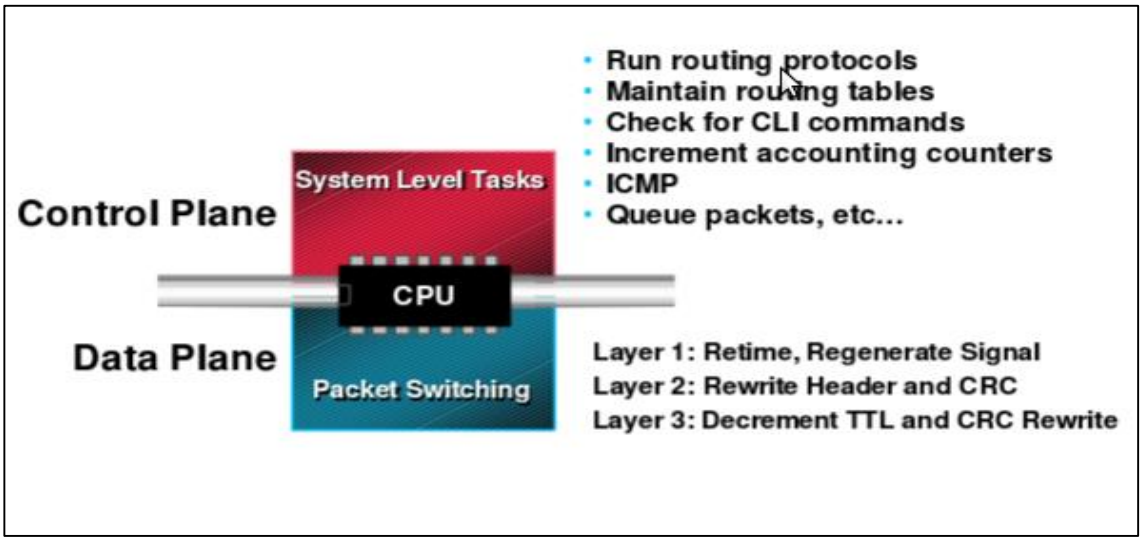

Figure 6. Cisco router control and data planes schema. Reprinted from Cisco Router Architecture (1988) [30, 7]

As shown in figure 6, router functionality in Cisco routers is split into two main areas: control plane and data plane. In the control plane are system level tasks performed and in the data plane the packet switching takes place. In other words, the control plane aims at learning what the router will do with the packet and the data plane actually moves the packet based on what was decided in the control plane. As a conclusion to the Cisco IOS architecture description it is possible to say that the main data flow (here data produced by the CLI input) is going from CLI through the processes (processes can be responsible for QoS configuration, simple network management protocol (SNMP) services and so on) to the hardware driver and finally uploaded to the hardware itself.

3.3.2 Juniper Operating System

Juniper is another vendor of routers able to present its own OS for its products called Junos OS. This company has three main principles of their OS:

- one operating system
- one software release
- one modular software architecture.

The implementation of these principles leads to the modularity of the software that allows the running of each module in its own protected memory space, which protects one module from influence of failures from another one. Another benefit of these principles is a single operating system for all the products, which allows having the same CLI commands and the user interface and decrease the learning curve for the engineers at the moment of integrating Juniper products to the network. The single release principle leads to including each new feature to all the products at the same time. [30.] The architecture of Junos OS from the abstract point of view is represented by two main components:

- the routing engine
- the packet forwarding engine. [31]

A precise schema of the OS processes in Junos OS are presented in figure 7.

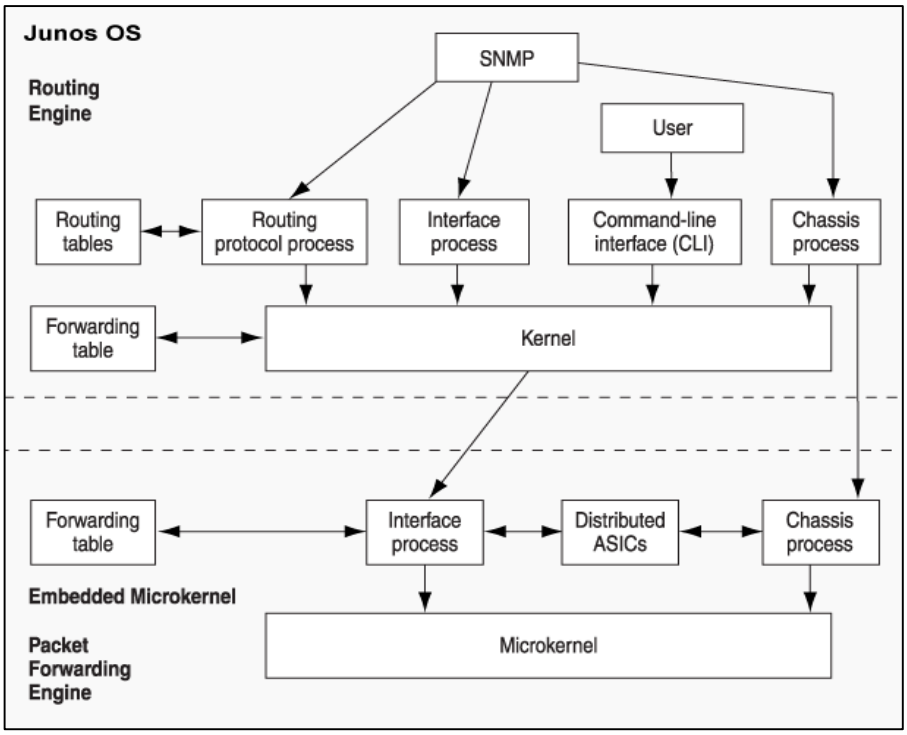

Figure 7. Junos OS architecture. Reprinted from Junos OS Architecture Overview (2012) [31]

As shown in figure 7, the main data transfer channel between the routing engine and the packet forwarding engine is going from the kernel to the interface process which in turn exchange the data between the forwarding table, the distributed ASICs (application-specific integrated circuits) and the microkernel. The packet forwarding engine serves for Layer 2 and Layer 3 packet switching, route lookups and packet forwarding between the input (ingress) and the output (egress) interfaces.

The routing engine was created for performing the system management tasks and controlling of the routing updates. Inside the protected memory of the routing engine there are routing protocol software processes running which are its main components. These process are running on a general-purpose computer platform. In general, the routing engine besides the routing protocols also handles the processes that control the routers' interfaces, the system management and the user access to the router. The main features of the routing engine are:

- Packet processing for routing protocol. This feature allows to all the packets related to the routing protocol service to direct them to the Routing Engine and not to consume Forwarding Engine resources.

- Software modularity. Different software functions are running within the separate processes. This solution does not allow to the failure in one function to affect other functions dramatically.
- Storage and change management. This function performs maintaining of the configuration files, the OS images, and the microcode within the three storages (two secondary and one active storages).
- Management interfaces. The management interfaces are responsible for configuring and controlling the system through the CLI, the craft interface and the SNMP. [31.]

3.3.3 Ericsson Operating System

Ericsson presented its common IP operating system (IPOS) across the IP products in 2013. [32.] Presented IP products included an SSR 8010 router, mini-link SP 415 and SP 420. During this presentation, Ericsson stated that the main aim for the IPOS development unit was to run their entire portfolio based on one IP operating system in order to be able to provide end-to-end solution for IP networks. Running an IPOS on IP Metro and Backhaul IP/MPLS routers was a part of this strategy. After that Ericsson IPOS was mentioned on Mobile World Congress 2015 where Router family 6000 was presented. These routers were also running the IPOS. There is not enough publicly available information about the IPOS's architecture and design, but from the different sources, it is clear there are several main principles applied to the creation of this product:

- Enabling accelerated feature delivery. [33.]
- High resistance and modularity. [40.]

Based on the information about CISCO and Junos OS mentioned above it is possible to say that the IPOS has a unified user interface and a common architecture for all the supported platforms which allows the company to enable the accelerate feature delivery to all the platforms once it was planned and implemented. The high resistance and modularity shows another part of architecture. As was mentioned in the description of IOS and Junos the term modularity applied to the OS architecture implies that the OS is not

a monolithic part of software, but consists of several processes. Each of them is responsible for the separate part of configuration (for example in cisco there are separate processes responsible for SNMP, QoS, port and so on). [26; 29.]

3.3.4 Router Operating System Environment and Obstacles

Based on the information provided in the previous parts it is possible to summarize that all the routers' operating systems have common parts in their architectures:

- modularity (separate processes for different features)
- separation of OS into two main parts: the control plane and the data plane (in terms of IOS).

This work is concentrated on the CLI configuration data flow and its path to the switch chip. The simplified and high-abstract schema of data flow is presented in figure 8. Each element in figure 8 have already been mentioned and described before except the pre-hardware level. Its functionality consists of checking the dependencies between the objects and re-ordering of the messages that sent from all the processes in concurrent mode and set up configuration of the chip.

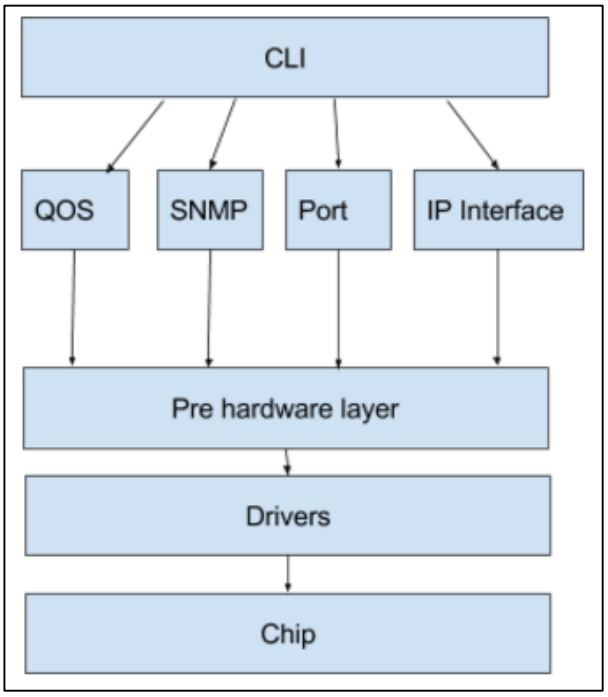

Figure 8. An abstract data flow for the applied CLI commands.

Figure 8 presents objects such as QOS, Port and IP Interface. The first use case is described in order to illustrate the dependencies problem. In this case, a user configures the policer for the IP interface. Three objects are involved in this action: the port, the IP interface and the QoS policer. At first, the Port object should be created, and then the IP interface should be bound to the port and just after that, the QoS policer should be applied to the IP interface. In the situation where all the messages for the creation of these objects come from different processes the race condition occurs. The race condition is a situation where multiple processes access the same data. It occurs when sequence of the commands received by a process is a crucial factor, but because of the nature of the concurrent systems, it is impossible to predict the sequence of the commands. Developers have to invent ways of the dependencies control between the objects in the process-receiver. One of the solutions is to implement a system that controls the dependencies between the objects and re-arranges the messages. Another possible solution is to implement a synchronization mechanism between the processes of the operating system.

4 Environment

4.1 Ixia network

Ixia Company provides application performance and security flexible solutions allowing to validate, optimize and secure business's networks. [35.] In addition, Ixia produces solutions for the functional, conformance and performance testing of the networks and as a part of it of the network devices. [36.] The main requirement for the verification of a high-capacity router is an ability to route the traffic with 100GE speed per rack unit. That can be tested with the device stressing with the maximum supported traffic flow speed per port and checking that the router is able to provide a high quality of service for the different combinations of the traffic flow parameters within the different load applied to all the types of the ports and the circuits. The Ixia test solutions aimed at the verification of the variety of cases such as the network expansion, device insertion, pre-deployment testing and new service implementation. The most valuable part of Ixia's functionality is providing of the testing environment for accomplishing conformance, functional and performance testing of the node. [37.]

Ixia chassis provides 1GE and 10GE interfaces with resources ownership at a per-port level. Each port in Ixia has separate CPU, which allows an independent modulation of the traffic streams on the separate base for each. [36.] High variations of the generated stream parameters allow creating of the automated tests for all kinds of use cases. Support of L2 and L3 protocols emulation allows imitating an integration scenario of the node into the real network with all kinds of service traffic flows enabled. [38.]

Ixia provides clear defined TCL (tool command language) command library which contains an application program interface (API) bringing a full access to the hardware platform. Creating and executing of various programs can be performed through sending configuration within the TCL commands. The TCL scripts allow automating the testing process, which decrease the development cost compared to the development process with manual testing, as a main tool for checking software quality. The data flow of running script process is shown in figure 9.

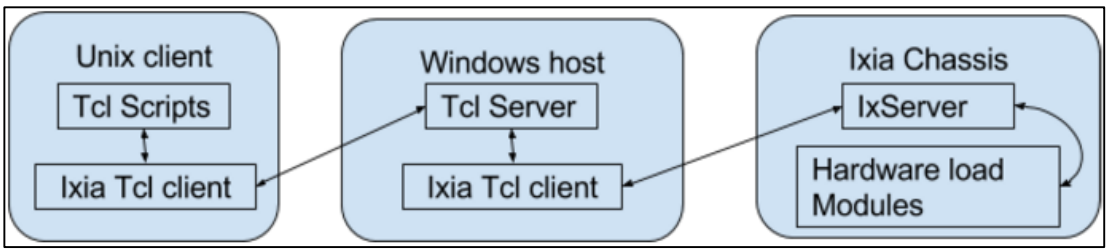

Figure 9. Architecture of the automation testing environment

To set up a proper environment for automated testing it is necessary to have three main software components: a UNIX machine with TCL client installed, a windows server with TCL server and a TCL client installed and an IxServer itself, which downloads a configuration to the actual hardware. [38; 39.] The script will run Ixia TCL client that will establish connection to the TCL server. The TCL server will receive commands sent by the script and re-send it to the Ixia TCL client installed on the “Windows host”. After that, the Ixia TCL client will send commands to the IxServer which will download the desired configuration to the hardware. Besides applying the desired configuration to the Ixia chassis it is possible to manage the state of the chassis’ elements in a real-time mode with such

commands as enable/disable protocols, reserve ports, start/stop transmit traffic and start/stop capture packets on the Ixia side. These commands allow changing dynamically the state of each port and stream separately and provide a possibility to create tests that check several functions of the node at the same time and control the reaction of the node on the peak load for the specific circuit or port. [38; 39; 40.]

4.2 Router Hardware

Router is a device that forwards packets from one subnet to another based on the state of routing and the address resolution protocol (ARP) tables. The target router contains 16 1Ge or 10Ge physical ports and one serial console port. All ports are industry standard interfaces with precise synchronization features. [41.] In addition, the router has a USB port and four additional Ethernet interfaces that are not included into the current project's scope. All of the ports mentioned can be found in figure 10 showing the appearance of the router.

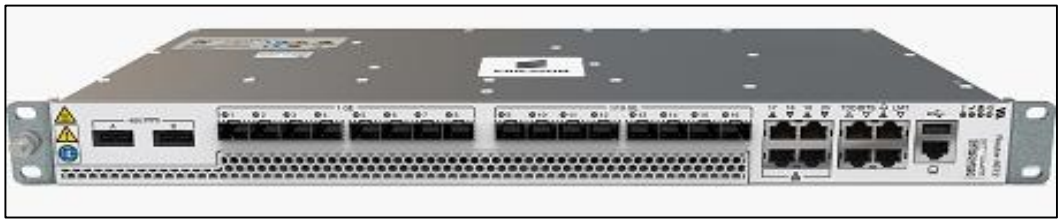

Figure 10. Router 6672 appearance. Copied from Ericsson Router 6672 [41]

For testing purposes, the three ports will be reserved and connected to the corresponding Ixia ports as shown in figure 11 below.

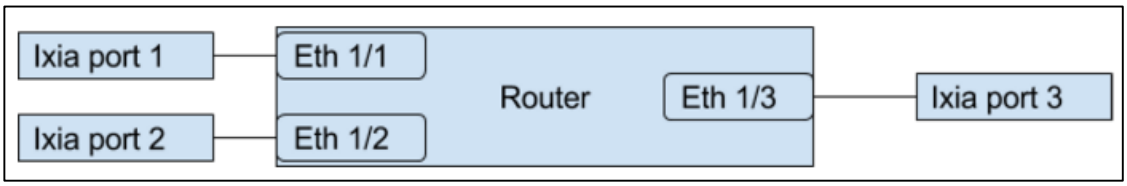

Figure 11. Router hardware schema with Ixia ports.

As shown in figure 11, the router has three Ethernet interfaces available for testing: "Eth 1/1", "Eth 1/2", "Eth 1/3". All the three interfaces are placed on the same line-card, which can be determined from the first digit of the port name. For 80% of cases it is enough to

have two interfaces, but for testing QoS with the policer applied to the link aggregation group (LAG) it is necessary to have three ports. LAG will consume two ports and “Eth 1/3" is an output port for the traffic and is configured as usual Ethernet port.

5 Development Process

5.1 Scrum

The changing requirements in developing a software made leaders of the software development market look for new principles of a software development process that allow satisfying customer and market needs. Increased software complexity, tighter schedules and a need for fast adaptation to the market brought about the new principles of development process called “Agile Manifesto”. Agile manifesto aimed at emphasizing the importance of individuals and their ideas prior to already defined processes and already used tools, changing priorities in the development process so that creating effective software is more important than creating comprehensive documentation. Two more principles of agile manifesto relate to the fact that the cooperation with the customer about the changing of requirements should be more important than the statements in a contract and related to the importance of being ready to adapt to changes and to not restrict the development process with the plan. [42.] After agile manifesto was published several methodologies, which were subclasses of Agile, were developed. On a par with the subclasses of the agile methodology several supporting development process techniques were created. Scrum and continuous integration are the main techniques used in this project.

Scrum is an approach related to the agile family methodologies for developing an innovative software. Scrum is a framework aimed at managing and organizing a work process. In Scrum the product owner has a set of requirements and as a result a vision of the whole product that he wants to create. Through the activity called grooming its vision is broken down into the small pieces in order to receive a set of features. Features gathered into prioritized list called the product backlog. [43.]

The whole working process is divided into sprints. A sprint is a short-time interval chosen in such a way that will allow accomplishing defined tasks by the end of the defined time

period. The sprint's length usually stays between one and four weeks. The sprint starts with sprint planning, continues with the development activities, and ends with the sprint retrospective and review. [43.] A sprint planning is a process that contains picking up tasks (user stories) from a product backlog. The optimal size of the task should be determined from the length of the sprint: it should allow team to accomplish the task (design, implementation and testing) by the end of the sprint. A retrospective in agile is a definition for a meeting that is placed at the end of the sprint and aimed at the improving of the development in a team through the reviewing and analysing of the sprint results. A sprint review is another meeting held at the end of the sprint. Usually it takes the form of accomplished in current sprint feature demos. As a definition of done for the feature most commonly used following list of checks:

1. Feature implemented.
2. Unit tests implemented and coverage for the new code is higher than 90%.
3. Integration/functional tests implemented.
4. Integration/functional tests included into the testing loops.

In conclusion, it can be said that the agile methodologies are belong to the iterative development process class. Each sprint team or developer should deliver fully functional and tested code and as a result each iteration goes through all the steps of development process (planning, design, implementation, testing) and increments functionality of the desired product. [42; 43.] As a result of applying the scrum technique, the following list of actions was created and followed:

- Splitting the project scope into the small tasks (user stories).
- Splitting the time period into two-week sprints.
- Closing of the user story was tightly bound to the list of definition-of-done mentioned above as the most commonly used set of requirements.
- Demos at the end of each sprint should be performed.

5.2 Continuous integration

A continues integration (CI) is a software development technique concentrated on a performing set of actions such as compilation, running automated tests and inspections, running unit tests, deploying software and receiving feedback for each commit or a set

of commits. The main idea of the CI is that each of the commits during the process of merging to a repository should successfully pass the whole integration cycle mentioned above. [44.]

As shown in figure 12, the CI consists of the CI server, the git repository or the version control system in general, the feedback tool and the developer as an actor. The control version system is a core part of the CI system. One of the most popular systems is Git. Git is an open source distributed version control system. [45.] Git provides such functionalities as storing the whole history of changes, creating branches and merge. Git stores all the changes in its internal structures. The key for each change is calculated from the commit's size and the content and represented as SHA1 hash. [46.] Storing of the whole history of changes and providing straightforward access to the version of the repository at any moment in its history allows easy switching between the software states which is important for searching problems that were delivered to the repository and controlling the developing process. Another functionality that Git provides is the ability for a group of people work on the same files. Merge is a process of combining several commits or several branches together and allows moving changes from one working branch to another. The feedback mechanism could be represented by an e-mail or review system.

Build term in CI refers to a process of putting a source code together and verifying that the result works as a unit. A CI use case starts with committing of a source code to the remote repository of control version system. Each commit or a group of commits should trigger a CI cycle. Meanwhile the CI server constantly polling repository for changes. Once it determines that some changes occurred it retrieves the latest copy and then runs the build script, which integrates a software, performs unit and automation testing. In some scenarios also added checks for the new warnings and checks for not adding new lines that were not covered with the unit tests. In addition, different sanity checks like check for a commit message format and check for a code style can be added. After the CI server performed all the checks it sends feedback to the developer through the e-mail system or through the putting a review mark in a code review tool (for example Gerrit). The interaction workflow between git and the CI server is shown in figure 12 below.

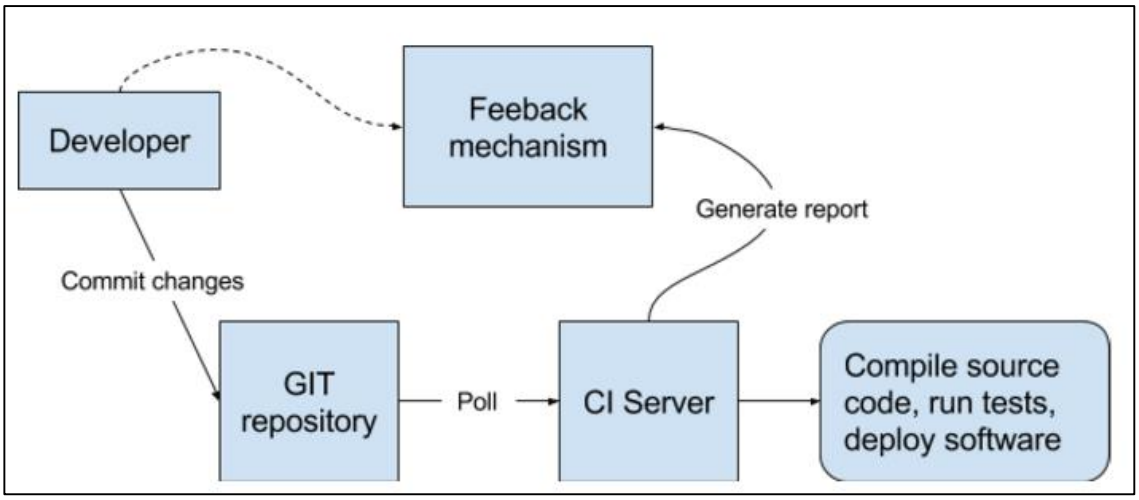

Figure 12. CI system workflow. Reprinted from Paul M. Duvall (2007) [44.]

In the current project, the CI was used in conjunction with the code review system called Gerrit. The system with the code review has different workflow shown in figure 13. The sequence of actions shown in figure 13 is the same as described above except that there is a new acting person in the cycle called reviewer and new system called “Code review”. In this case the developer pushes his changes not directly to the repository but to the separate branch in the code review system. After that, the developer adds a person that he wants to see his change into the list of reviewers. The reviewer goes through the code and if commit quality is acceptable according to his opinion, he gives “+2” to it and Gerrit merges it to the Git repository. After that, the normal CI cycle performs on this commit.

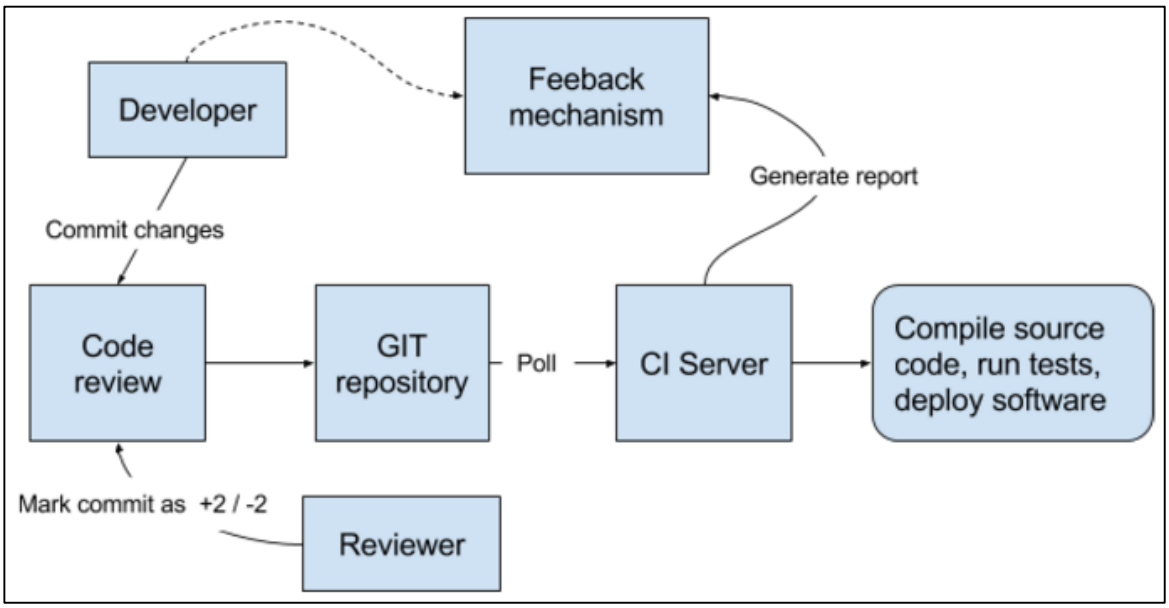

Figure 13. Schema of CI with integrated code review system.

As an enhancement for the usual CI workflow, it was suggested to run the CI cycle on a separate branch that contains the latest version of repository and, merged with it, the developer's change. This solution allows preventing bugs in the repository itself and significantly decreasing the number of bugs appeared in the main branch that stops developing and make all the teams concentrate on the problem fixing in the repository.

The continuous integration together with the code review system is a powerful tool allowing delivery of changes to the repository faster and with higher quality. The CI allows automating routine processes and testing a software more frequently which leads to better quality of the product in terms of its stability and number of bugs in it and perform sanity checks on a particular commit which allows highlighting minor problems in the code style and quality.

6 Scope of the project

This project concentrated on implementing and testing of the Quality of Software, Policing feature for the router Ericsson 6672. All the implemented functionality should be covered with the automated functional tests embedded into the CI loop. The set of tests is aimed at the defining the policing behaviour within the different combinations of the parameters. The results of the tests need to be analysed in order to provide easily interpreted information for a customer and to improve the quality of the rest of the test cases.

6.1 Circuits

The IP operating system is a circuit-centric operating system. It means that each entity that is a part of a routing process has its own circuit. According to the project requirements, policing should be supported on the following circuits: cross-connect (XC), IP interface, PVC (permanent virtual circuit) interface, LAG (link aggregation group), port and bridge. The cross-connect is a layer two of the OSI model (L2) circuit that performs switching of incoming packet based on MAC address. As shown in figure 14 below the cross-connect instance (XC) connects two the service-instances (SI1 and SI2). Each service instance bound to the corresponding port (Port eth 1/1 and Port eth 1/2). Service instance can be bound to the port under the PVC circuit (PVC 1 and PVC 2) as well.

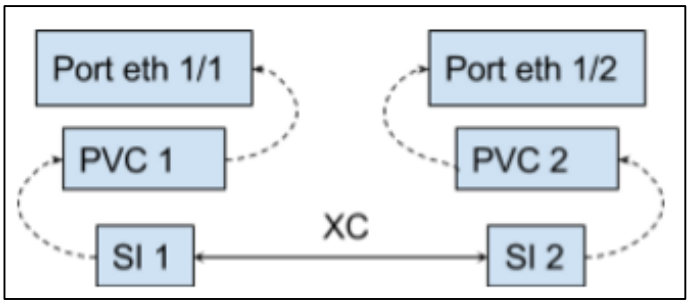

Figure 14. Circuit dependencies for the cross-connect configuration.

Traffic comes to Port eth 1/1 and based on the packet's VLAN (virtual LAN) tag it is forwarded to PVC1, after that based on the destination MAC address it is forwarded to the SI2 and sending out of the node. An IP interface is an interface containing the IPv4 and/or IPv6 addresses and as the SI1 could be bound to the port directly or to the PVC circuit applied to the same port. If the IP interface bound to the port directly it will forward the untagged traffic in other case if the PVC circuit involved, the IP interface will forward traffic just with the specified VLAN tag. A bridge is a circuit used to connect several VLANs. Each of the circuits described above can be applied to the port or LAG. Port is a physical interface represented by 1Ge or 10 Ge Ethernet slots.

LAG is a link aggregation group, which is aimed at combining of the several ports on the layer two of the OSI model, despite the fact that it consists of several ports it is still interpreted as a single circuit from the user's point of view. All of the mentioned circuits have different set of messages created by the CLI towards the back-end processes and different number of objects required to store the related to them information in the memory and this fact requires to implement the proper dependency handling between the target object (policer) and the objects that it dependent on.

6.2 Policing Process

The policing process consists of two steps: marking and applying of a specified action. In this project, the policer will support two actions: "drop" and "mark red". Configuring of the "drop" action will enforce the router to drop all the traffic that will go above the allowed rate. Configuring of the "mark red" action will enforce the router just to re-colour packet as red and drop it just in case of the situation with traffic congestion. All the red packets will be dropped prior to the yellow one.

The process of the packets marking consists of two different phases. Figure 15 represents the situation when the packet being received by the Meter during the first phase. The Meter gets the size of the packet, calculates the result and sends the packet with the calculated result to the Marker. During the second phase the Marker colours (re-colours in case colour-aware mode was chosen) the IP packet. The colouring of the IP packet is a process of changing the differentiated services field (DS field) of the packet. [47; 48.]

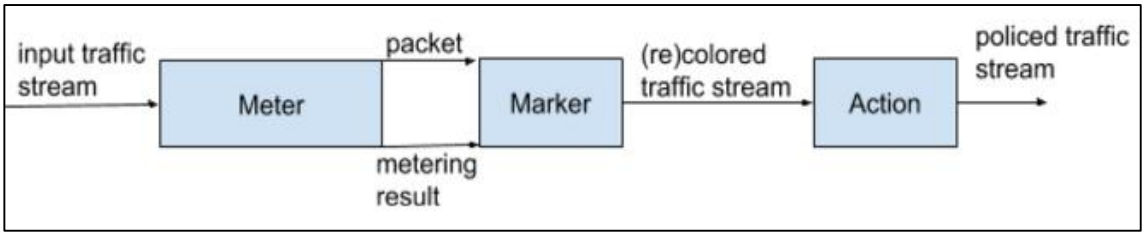

Figure 15. The marking process schema.

6.3 Single-Rate Two-Colour Policer

The single-rate two-colour policer (SR2CM) provides an applying of implicit or specified by user action on the traffic that does not conform to the specified limits. The policer can operate in two modes: colour-blind mode and colour-aware mode.

After configuring the single-rate two-colour policer and binding it to the input (ingress) interface, the policer meters the traffic flow according to the limits defined by the following parameters:

- rate limit (bandwidth limit) - CIR
- burst-size limit - CBS. [49; 50]

The rate limit can be defined in bits per second. It defines the average number of bits per second permitted for the packets received at the interface and refers in documents to the committed information rate (CIR). The burst-size limit specifies the size of the burst data allowed on the traffic flow and refers to the committed burst size (CBS). The single-rate two-colour policer performed within the single bucket algorithm shown in figure 16 below.

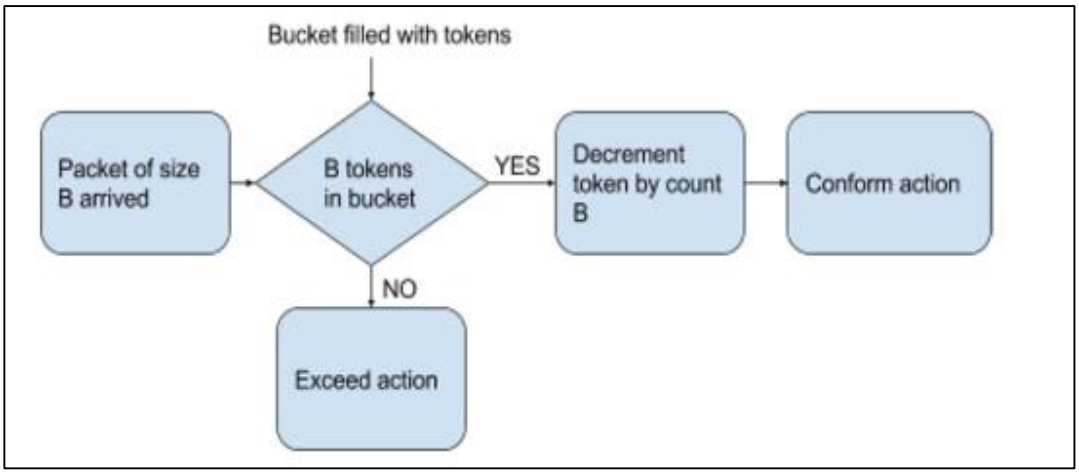

Figure 16. The metering algorithm for the single-rate two-colour policer.

After the CIR and the CBS were defined and policer was applied to the circuit the rate of the ingress traffic flow starts to be metered according to the provided algorithm. When the packet arrives to the ingress circuit within the rate higher than the CIR its size compared with the number of the tokens in the bucket. If the number of the tokens is less than the size of the packet (B), the exceed action is applied. If the number of the tokens in the bucket is greater than B, then the packet is marked as green and number of the tokens decreased by B.

The default configuration of the policer specifies exceed action as marking of the packet as red and drop it during the next phase of the packet processing. Another option that can be configured for the exceed action is to mark red and not drop the packet. For the conform action, just the default case is defined and in case the conform action is applied, the packet is marked as green and should not be dropped.

The flow described above is related to the colour-blind mode of the policer. In the colour-aware mode, the policer will take into account the colour of the incoming packet (the colour that was applied to the packet in a previous process or assigned by the previous node). If the packet is pre-coloured as green, there are no changes in the algorithm, as the “darkest” colour will be applied to the packet. If the packet is pre-coloured as yellow, then the packet will be dropped as in this type of the policer there is no yellow bucket (will be described below) and the nearest darkest colour for the packet is red. If the packet is pre-coloured as red, then no metering is performed on it and the packet automatically receives the colour red and will be dropped during the following phase of the packet processing. [49.]

6.4 Single-Rate Three-Colour Policer

The single-rate three-colour policer (SR3CM) supports two modes: colour-aware and colour-blind. The configuration of the single-rate three-colour policer command uses three parameters:

- rate limit – CIR
- burst size – CBS
- excess burst size – EBS.

The CIR measured in kilo-bits per time interval which is usually equal to the one second. [51.] The size taken into account by the meter excludes the link specific headers. The CBS and the EBS are measured in bytes and in order to enable the two buckets algorithm both of them should be configured with the value greater than zero. In general, it is recommended that the size of the CBS and the EBS should be greater than the maximum possible size of the packet (MTU) in the traffic flow. [18.] The single-rate three-colour algorithm differs from the single-rate two-colour algorithm in the number of supported colours and the number of the buckets participating in storing of the tokens. In the single-rate three-colour policer both buckets share the same CIR and they both have the same source of the tokens, which affects the bucket's speed and order of filling. The EBS bucket starts to fill just when there is no space for the new tokens in the CBS bucket. The EBS is responsible for configuring the yellow bucket size. The packets that came to the excess bucket are marked as yellow, which means that they will be dropped in case of traffic congestion happens under the certain circuit. The algorithm of the single-rate three-colour policer is shown in figure 17.

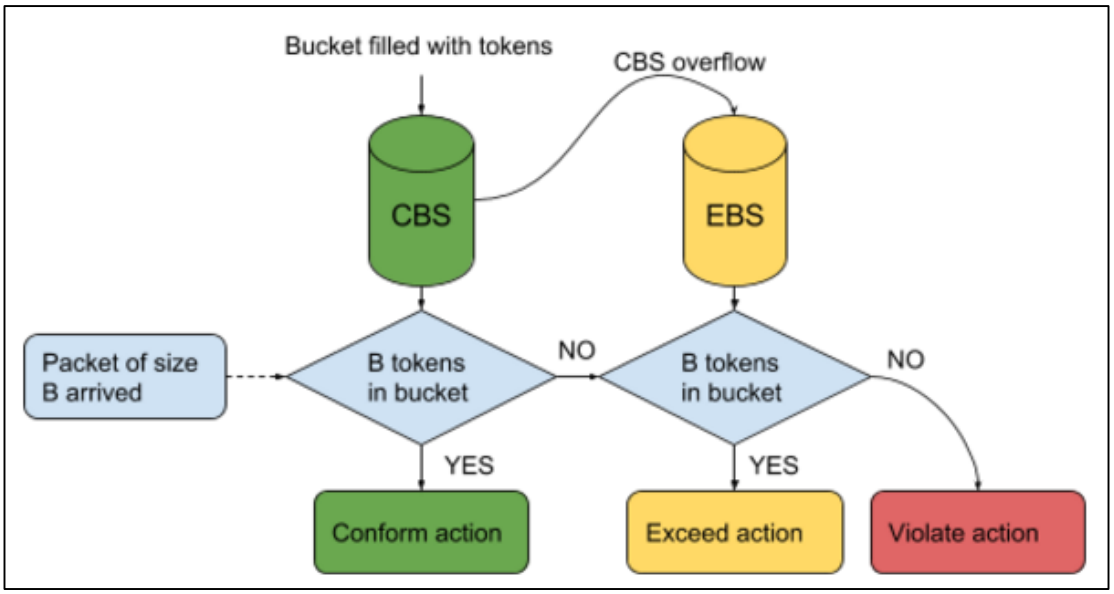

Figure 17. The metering algorithm for the single-rate three-colour policer. Data gathered from RFC2697 (1999) [51]

When the packet arrives to the ingress circuit it goes to the meter stage. During the meter stage the size of the packet compared with the number of available tokens in the green bucket (CBS) and if there are enough tokens the packet coloured as red and the number of tokens decreased on the packet size B. If there are not enough tokens in the green bucket the next step occurs. The next step is a comparison between the size of the packet and the number of tokens in the yellow bucket (EBS). If there are enough tokens in the EBS then the packet will be coloured as yellow and the number of yellow tokens will be decreased, otherwise the packet will be coloured as red and the violating action will be applied.

6.5 Two-Rate Three-Colour Policer

The two-rate three-colour policing (TR3CM) is a technology that was developed and patented (patent No. US 8385206 B2) by Thyagarajan Nandagopal and Thomas Y Woo in Alcatel Lucent company. [52.] The description of the technology in this patent is referred to the RFC 2698. [47.] The Cisco company implemented this technology in “Cisco IOS Release 12.2 (27) SBB” at 2008. [53.]

The drawback of the single-rate three-colour policer is that the service provider should be careful in assigning of the CIR value, because in the situation without congestion poorly chosen values can lead to the offering of less bandwidth to the customer than is

actually possible in current time. The reason for it is in the fact that not all the customers use all their bandwidth simultaneously and the situation of underutilizing the network happens quite often. In this case, the single-rate can transform the other customer's not used bandwidth to the short time period bursts which does not lead to an optimal way of using the network resources. The two-rate three-colour marking algorithm was created to solve this problem and provide better utilizing of the common network resources.

The two-rate three-colour policer supports the same modes as the single-rate three-colour policer (colour-aware and colour-blind), and for its configuration four parameters are required:

- committed information rate -CIR
- excess information rate - EIR
- committed burst size - CBS
- excess burst size – EBS.

The CIR and the CBS have the same meaning as in the single-rate algorithm. The excess information rate is an additional parameter compared to the single-rate policer and it defines the maximum rate of the traffic sending to the customer. The traffic burst that exceed CIR but does not exceed EIR is allowed in network but will be marked for more aggressive discarding. Marking depends on the transport technology and in this project under marking the DSCP bits change assumed. The excess burst size parameter is also presented in the single-rate algorithm, but in the two-rate policer it has a different meaning. The excess burst size or Be is the maximum size of the yellow bucket that could be allowed to support the EIR.

The algorithm of the single-rate three-colour policer shown below in figure 18. As can be seen from the figure 17 the sequence of the packet's size (B) comparing with the buckets' sizes is the same, but the source of the tokens for the buckets is different. In fact, there are two independent sources of the bucket tokens: the EIR and the CIR.

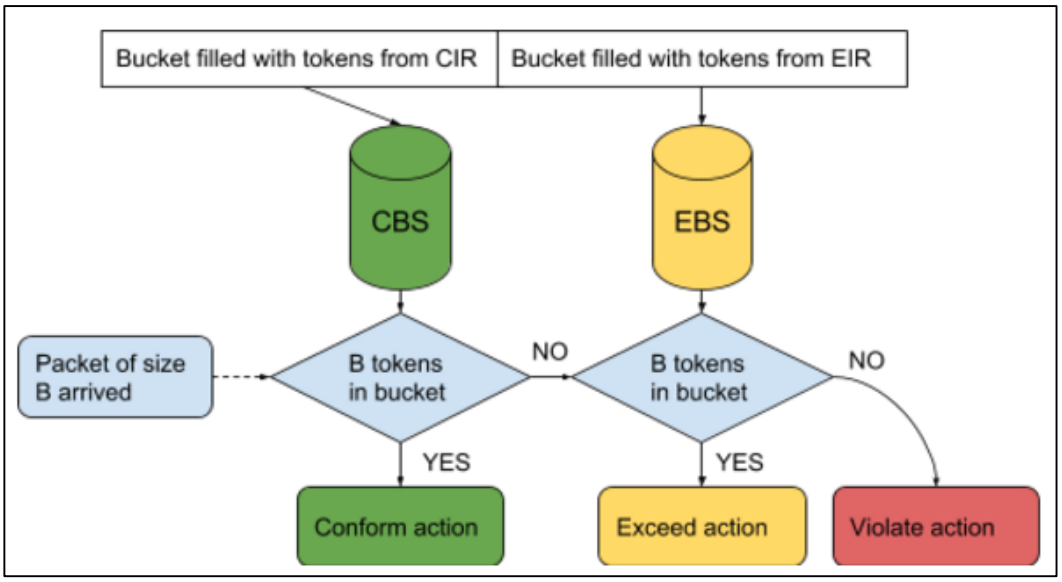

Figure 18. The metering algorithm for the two-rate three-colour policer. Data gathered from RFC4115 (2005) [54]

The first version of the TR3CM algorithm was published in RFC 2698 and it was described above. [47.] The second version of this algorithm was published in RFC 4115. [54.] The previous version of the TR3CM is still used by many routers and is a basic version of the TR3CM technology. The TR3CM described in the RFC 4115 is different from the one discussed above. The first difference is that in RFC 4115 it was suggested to link CIR and PIR together with the following equation:

$$T = \frac{CBS}{CIR} = \frac{EBS}{EIR}, \qquad (1)$$

where T is an expected time of the burst in sec.

A second difference is in handling of the inbound traffic. The incoming red packet will not be tested by the meter and its colour will not be changed. The last difference is that the traffic with defined CIR should be coloured with green mark without passing the additional conformance tests. That part of algorithm fixes a problem of dropping the packet in the first step of the algorithm described in the previous section, although check in the second and the third step should pass. In the current project, the policer algorithm with specification from RFC 4115 is used.

7 Project Implementation

As was partly described above, the scope of the project aimed at creating the part of the software responsible for proceeding messages from the IPOS processes and downloading the result of its proceeding to the hardware. The IPOS is an operating system consisting from several processes that send messages to the developed process. Several problems appear with such kind of architectures. The first problem is related to the sequence of messages that cannot be controlled and potentially can cause misconfiguration of the target object. The second problem related to the intersection of several message flows leading to the possibility of breaking the dependencies between the dependent objects. In this project, the policer in order to affect the traffic flow should be applied on any of the circuits described above. This condition creates dependency between the circuit, physical port and policer. Figure 19 shows a common schema of objects that should be created to enable the policer's functionality.

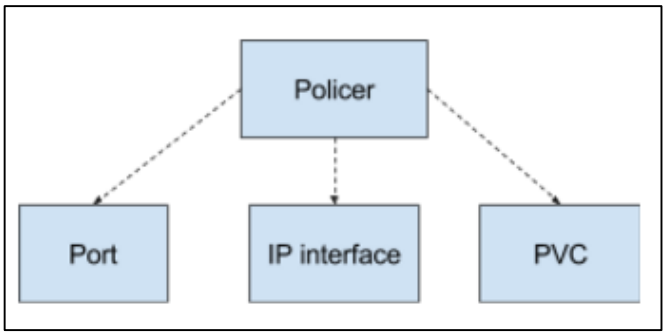

Figure 19. Dependencies between objects in the process of creating and applying the policer to the PVC circuit under the IP interface.

Figure 19 shows that before the policer object switches to the fully functional or created state it requires the Port, the IP interface and the PVC objects to be created, but because of the system's architecture specialty the message for the policer creation can arrive before the message for PVC object creation. The solution for this problem can be implemented based on the observer pattern. [55.]

The observer pattern is a software design pattern aimed at defining of a one-to-many dependency between several objects so that when the main object changes its state all other objects dependent on it will be notified. The notification process usually implemented by calling functions of the dependent objects. There are two key objects in classic observer pattern: subject and observer. A subject may has several observers. All the

observers should be notified on any occasion of subject state change. In response each observer can request the subject to synchronize its state with the subject's state. Figure 20 shows the flow of actions of the observer pattern described above.

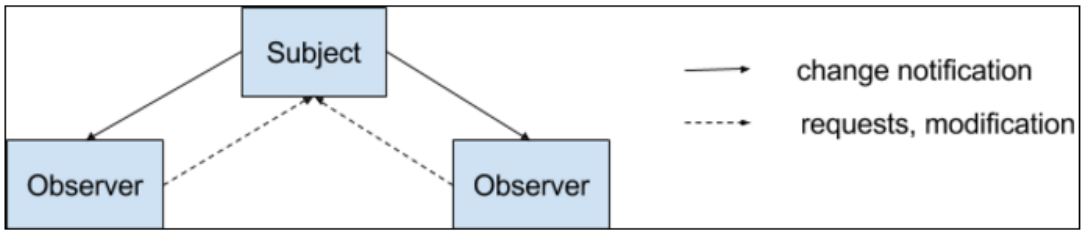

Figure 20. The observer pattern notification flow. Reprinted from Gamma E., Helm R., Johnson R., Vlissides J. (2000) [55.]

In order to adapt the observer pattern for the current project the dependency schema shown in figure 21 was implemented. This figure represents the scenario where the policer bound to the PVC interface under the IP interface. In this case, the Policer object and the IP interface objects are observers for the PVC object and the PVC object is observer for the Port object.

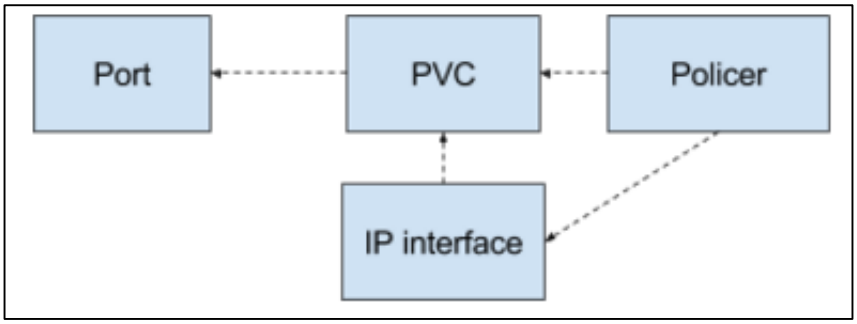

Figure 21. Dependencies between the objects for the scenario in which policer applied to the PVC circuit under the IP interface.

The PVC object provides the information to the policer about its readiness and availability of resources and creation of necessary structures on the hardware. The IP interface object provides information about the type of the configured IP addresses (IPv4 and IPv6) and its changes. The Port object provides information about the physical state of the link and the current state of the interface.

8 Testing

8.1 Testing implementation

The aim of the testing process is to define a set of test cases for the feature or its implemented part to be tested. There are two fundamental test methods: specification-based and code-based testing. The specification-based testing (originally called “functional testing”) follows the approach where a software is treated as a function that maps a set of input parameters to certain values in output parameters. This approach can also be called black box testing. The main idea of the black box testing is that the tester does not know the implementation of the feature and the functionality of the software represented just by its input and output parameters. The specification-based approach allows creating test cases that are independent from the implementation and in case of changing the implementation of the test cases still should pass. Another benefit of the specification-based testing is that the test case development could be performed in parallel with the feature development, which is extremely useful for cross-functional team, which is a common development unit in a scrum. Code-based testing is another approach. In contradiction to specification-based testing it supposes that the tester knows the implementation details of the feature and allows performing testing for each line of code in the functional tests. [56.]

The implementation of the tests is in the scope of this project and consist of three main parts presented below.

1. Testing of the combinations of the circuits and the traffic types with colour-blind policer mode and the default action defined as dropping of the red packets.
2. Testing of the policer accuracy for the different levels of the burst size.
3. Testing algorithms with configured colour-aware mode and the violate action as mark red.

8.1.1 Circuits

In each software testing process there are more than one condition and more than one rule and its results. In order to maintain a situation where a number of action combinations are taken with a different set of conditions a decision table can be built. The decision table consists of two main areas, which are a condition part and an action part. The condition part is represented by a certain number of lines placed in the top part of the table. The action part is represented by several lines placed under the condition part. In this table, each column represents one certain rule. In this project, the decision table was created for controlling tested combinations between the policer type, the circuit type and the traffic types. Table 2 was created in the form of the decision table and is presented below. This table presents the combinations of actions and results for the different types of circuits, all types of policer algorithms and all traffic types.

Table 2. Decision table for policer testing.

C1: policer SR2CM	X			X			X			X			X			X			X	
C2: policer SR3CM		X			X			X			X			X			X			X
C3: policer TR3CM			X			X			X			X			X			X		
C4: Circuit XC	X	X	X	X	X	X	X	X												
C5: Circuit IP: IPv4									X	X	X	X								
C6: Circuit IP: IPv6													X	X	X	X				
C7: Circuit Bridge																	X	X	X	X
C8: Interf.: Port	X	X			X	X			X	X	X	X	X	X	X	X	X	X		
C9: Interf.: LAG			X	X			X	X											X	X
C10: PVC (VLAN tag 100)					X	X	X	X	X	X			X	X	X		X	X		
C11: Untagged	X	X	X	X							X	X				X			X	X
C12: Traffic type L2	X	X	X	X	X	X	X	X	X		X						X	X	X	X
C13:	X	X	X	X	X	X	X	X	X		X		X				X	X	X	X

Traffic type L3 IPv4																			
C14: Traffic type L3 IPv6	X	X	X	X	X	X	X	X		X		X		X	X	X	X	X	X	X
C15: Traffic tag 100		X		X		X		X	X	X			X		X		X		X	
C16: Traffic tag: untagged	X		X		X		X				X	X		X		X		X		X
A1: Limit the rate	X		X			X		X	X		X				X	X	X			X
A2: Not limit the rate		X		X	X		X							X				X	X	
A3: Traffic should not forward										X		X	X							

Table 2 presents the set of decisions applied to the packets based on the circuit types and the traffic types. The specified algorithm of the policer is not crucial here as any checks of how properly an algorithm works is out of the scope of this table. The fluctuation of the policer's algorithms, represented in the first three lines of the table, were applied in order to facilitate the identification of the problem with an algorithm if it will appear after the driver update. With this solution, it will be clearly seen from the logs that just test cases with certain algorithm failed and the time for problem detection will significantly decrease.

Different types of the circuits needed to be tested because of different algorithms of getting required information for the policer. Also, table 2 shows that there are three general types of the traffic included in the test cases: the layer 2, the IPv4 and the IPv6 traffics. Each traffic should be tested for each circuit type in order to check that the policer will police all the traffic that goes through the specific circuit. In this project, the L2 traffic defined as a traffic flow forwarded based on the source and destination media access control (MAC) addresses. For clearer distinguishing of the L2 traffic from the IP traffic the IP header could be deleted from the packet blueprint in the traffic generator. The IP traffic defined as a traffic forwarded based on the forwarding tables filled by the routing protocols and the address resolution protocol (ARP) requests. The IPv4 and IPv6 traffics were distinguished from the IP traffic as according to the command line interface user can

configure both IPv4 and IPv6 address to the same interface and all the combinations of it should be tested.

Each of the traffic types and the circuit can be applied under the PVC circuit. In table 2, just PVC tag 100 is mentioned, but in order to get better coverage it can be randomized in the test cases. For the testing quality in this project, the value of VLAN tag is not crucial, and because of that just two types of the VLAN tags were used: tagged and untagged traffic. Based on the test conditions there are three types of actions that can be applied. If the policer is applied to the traffic - it should be rate limited, if traffic goes through the node but does not satisfy the requirements of the policer it should not be rate limited. If the configuration does not assume a condition on the node that allows forwarding the traffic through it, then it is expected that the number of received packets should be equal to zero on the Ixia traffic generator side.

8.1.2 Traffic Analysis

The Ixia traffic generator has several restrictions on the traffic flow analysis. It allows the tester to receive data about the sending and the receiving traffic flow rate in the form of the average rate, but it does not allow receiving proper data about the maximum, the minimum and the current rates. These restrictions lead to inventing of the way of the policer testing through the catching and analysing the set of packets. Figure 22 below shows the graph representation of the incoming traffic flow rate. This graph illustrates that the standard granularity for the representing real-time traffic rate is 1 second, which does not fit the requirements for the testing of the policer's accuracy within the high-intensive traffic flow. The graph on the top called "Tx Test Packets" represents the rate of packets transmission and the graph on the bottom represents the rate of the packets that was received. The horizontal axis defines the time in seconds and the vertical axis shows the rate of the traffic in packets per second.

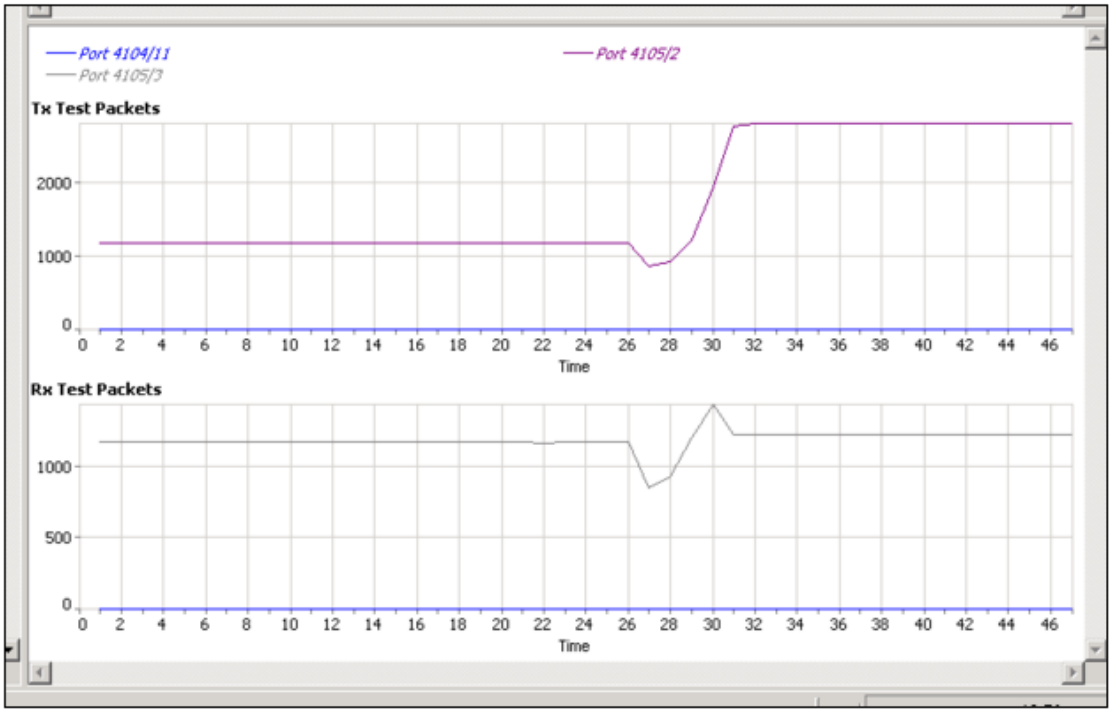

Figure 22. Policed traffic with configured single-rate two-colour marking algorithm.

Figure 22 shows that the traffic burst can be seen as a peak from the second 28 to the second 32. After the second 32, the bucket was empty and the policer started to drop the traffic, which resulted at traffic restriction on the level of 1100 packets per second.

The representation of the traffic rate on a real-time graph is useful for determining the fact that the policer was applied and the burst of some (impossible to calculate burst size from this information) size was configured, but it is not enough for the algorithm testing with the precise check of the burst size and the types of used buckets. As a solution for the problem of insufficient information that can be received from Ixia, the process of capturing of all the packets and its analysing was implemented. Ixia allows packet capturing with all the data including timestamps.

Capturing of the packets has several restrictions: starting and stopping of the capturing takes more than 2 seconds each and each command on the transmission rate modification takes extra time. Because of that, there is a certain number of packets in the captured list that is not needed for the analysis and should be filtered out. As shown in figure 23, it is possible to receive the information related to the packet's time arriving with higher precision than 1 sec interval.

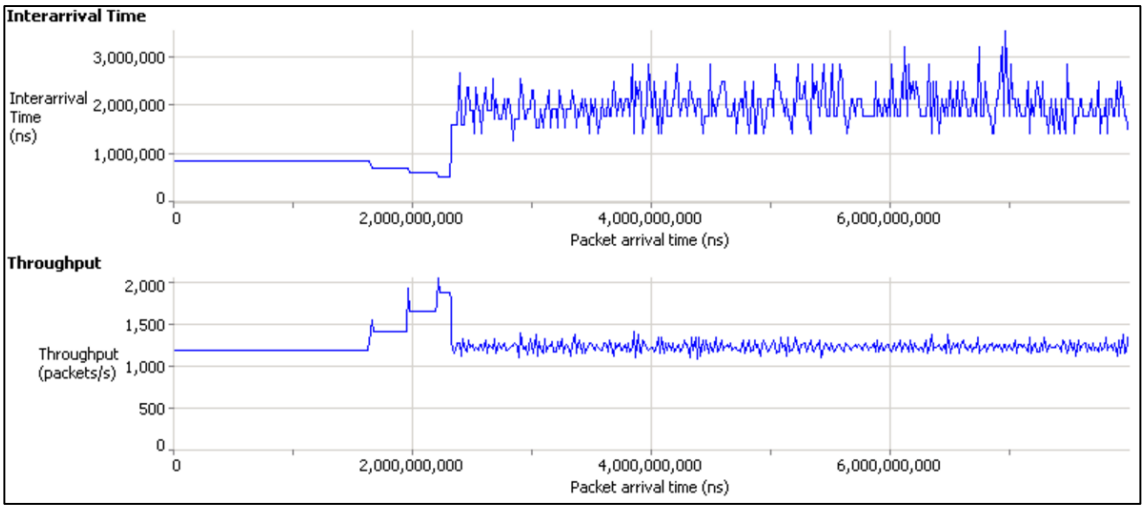

Figure 23. Captured packets analysis in Ixia traffic generator GUI.

In order to check the burst accuracy, it is necessary to run the analysis of the captured packets. The first graph represents an inter-arrival time for each packet in nanoseconds. The inter-arrival time is a calculated difference between the timestamps of the two neighbouring packets. The second graph shows the throughput of the packets per second. The throughput here is the rate represented in packets per second but with the higher precision than in figure 22. Unlike figure 22, which shows the real-time graph within the 1-second granularity, figure 23 presents the rate within the nanoseconds precision, which allows determining the exact size of the burst in packets.

There are three main areas that can be distinguished on the graphs in figure 23. The first part of the graph shows the period where traffic was generated with the rate below the configured CIR. It allows the policer to fill the bucket with the tokens. The second period shows decreasing of the inter-arrival time between the packets and represents the time interval where the tokens from the bucket were consumed by the extra traffic. This period is called a burst period. The last period represents the situation where the traffic rate allowed to go through the node is equal to the CIR on average. This period is characterized by the average rate equal to the CIR and visible fluctuations of the traffic rate around the CIR level caused by saving of a certain number tokens in the bucket and then immediately spending them on the short bursts.

Ixia software can save all the captured packets to the file, but it is not able to send all the captured information through the CLI, which prevents receiving all the necessary information about the captured packets through the script. The solution for the burst size and its accuracy checking was implemented through the getting packets one by one and then

analysing of the received information. The problem with this solution is that each request for the packet's timestamp is sent to the server and after that, the server extracts the necessary data and sends it back to the script. This causes n operations to be performed on each run, which significantly slows down the script. Another restriction of the Ixia software is that it cannot form sets of packets containing more than 1000 packets and with the fact that average number of captured packets is more than 6000 it requires splitting the list of the packets into the pieces within the size of 1000 items in each and analyse it separately. Figure 24 shows a simplified graph of the packets' inter-arrival time with policer applied to the ingress traffic flow.

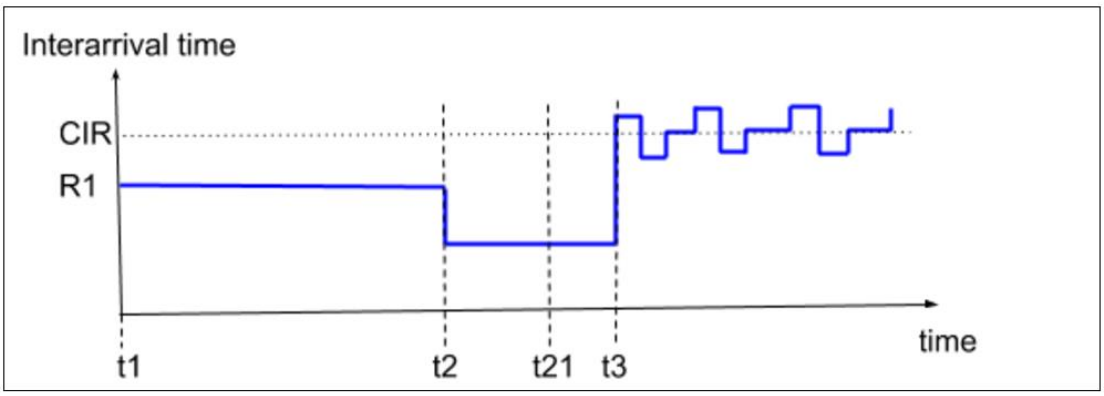

Figure 24. Inter-arrival time of the captured packets.

Calculating of the policer's precision based on the comparison between the expected and the actual number of the packets in the burst period. On the graph above, the burst period is an interval between points t2 and t3. Because of the restriction of the Ixia software described above the operation of reading the packet time is very expensive in terms of execution time and for the algorithm that firstly fetches data from the server and then performs an analysis on it the complexity will be O (n). The O (n) complexity for 6000 packets takes more than five minutes which is two expensive in terms of hardware usage because the router and the Ixia ports will be locked during that time and are not available for the other developers. It is seen from the graph that just two points needed in order to determine the burst size. The first part of the graph (from t1 to t2) has constant rate R1 that represents the period of filling the bucket with the tokens. Because of constancy of the R1, it is possible to split it into equal intervals and request the data for the packets that are borders of these segments. Comparing of the inter-arrival time of the packet at the beginning of the interval and at the end of the interval allows determining the segment where the burst starts. In order to avoid the situation where the length of the segment is

too big and the whole burst interval is placed in a single segment, the size of the interval can be represented as half of the expected burst size.

The expected burst size can be calculated from the CIR, the EBS and the traffic stream rate according to the following equation:

$$ExpectedBurstSize = \frac{EBS}{1 - \frac{CIR}{Ra}}\ , \tag{2}$$

where ExpectedBurstSize is the number of the packets in the burst interval including green packets which consumed the tokens from the bucket and the packets went through the buckets as belonging to the CIR. The Ra is the rate of the traffic flow when it was adjusted to the higher level than the CIR in order to enable the burst traffic. The ratio between the CIR and Ra represents the ratio between the packets that went through the policer and did not consume the tokens and the packets that decreased the number of the tokens from the bucket. After the expected size of the burst was calculated, the next action is to find the segment of the data where point t2 is presented. As can be seen from the graph inside the determined segment, there are just two levels of the inter-arrival time, which allows to interpret the segment as a sorted array. In order to find the point where the traffic starts to consume tokens from the bucket the binary search can be applied.

The binary search allows to decrease the cost of the operation from O (Nb), where Nb is the size of the segment to the O (log (Nb)). Searching of point t3 cannot be optimized with the usage of the binary search because of the fluctuating nature of the graph's third area, which starts from point t3. Although it is impossible to use the same optimization as described above, it is possible to decrease the number of calls to Ixia with the information about point t2. The expected burst size was calculated and in order to find t3 point it is possible to define point t21, which will be calculated using the equation below:

$$t21 = t2 + x * ExpectedBurstSize, \tag{3}$$

where x is a part of the burst size that can be skipped. These calculations allow decreasing the complexity of the burst's right border search from the O (Nb), to
O ((1- x) *Nb), where x is less than 1.

The described approach to the traffic analysis allows determining the actual borders of the burst period with the precision of one packet and calculating the actual size and the relative error for the applied policer. The relative error is calculated with the equation below:

$$Error = \frac{ExpectedBurstSize - ActualBurstSize}{ActualBurstSize}. \quad (4)$$

8.1.3 Policer's Counters Testing

The policer counter is an entity that stores the information about the packets and the bytes went through the policer that can be retrieved from the CLI command. The policer counter provides information about the following marks of the packets:

- green
- yellow
- marked red
- dropped red.

The method of analysing the captured data is useful for checking the policer's accuracy. Nevertheless, it is not the optimal one as it uses the shared resources like the router and the Ixia for a significant period. A more optimal approach exists for checking quickly which policer's algorithm was applied and what sizes of buckets were configured. This approach relies on the state of the counters for the specific circuit. The drawback for the quick check of the configured policer is that it is less accurate and its main aim is to check the type of the policer and how it reacts to different types of the traffic, but not its accuracy. Besides that, this type of check is the only way to check the policer if it is configured not with the standard action, but with the action "mark red". In this case, the red packets are not dropped and the whole approach for the traffic analysis described in the previous chapter meaningless. Table 3 below presents the decision table for the testing of the policer's parameters such as colour-blind mode, colour-aware mode, drop and mark red actions. The set of the test cases represented in table 3 covers all the branches of the single-rate two-colour policer algorithm described above. All of the action checks guarantee that the applied policer has just green bucket (the yellow counter always equal to zero) and in the colour-aware mode all the pre-coloured with yellow colour packets are

treated as red packets. The checks for “mark red” and “drop red” actions guarantee that the traffic was restricted as was expected and the specified action was applied.

Table 3. Decision table for policer’s mode with SR2CM algorithm.

C1: Mode: colour-aware	X	X	X	X	X	X	-	-	-	-	-	-
C2: Mode: colour-blind	-	-	-	-	-	-	X	X	X	X	X	X
C3: Action: drop	X	X	X	-	-	-	X	X	X	-	-	-
C4: Action: mark red	-	-	-	X	X	X	-	-	-	X	X	X
C5: SR2CM	X	X	X	X	X	X	X	X	X	X	X	X
C8: Traffic: green	X	-	-	X	-	-	X	-	-	X	-	-
C9: Traffic: yellow	-	X	-	-	X	-	-	X	-	-	X	-
C10: Traffic: red	-	-	X	-	-	X	-	-	X	-	-	X
A1: Counter: green	X	0	0	X	0	0	X	X	X	X	X	X
A2: Counter: yellow	0	0	0	0	0	0	0	0	0	0	0	0
A3: Counter: mark red	0	0	0	X	X	X	0	0	0	X	X	X
A4: Counter: drop red	X	X	X	0	0	0	X	X	X	0	0	0

Table 4 below is the decision table presenting the set of conditions and requirements for the three-color algorithms. It has one main difference from table 3: the yellow colour counter is presented in all the combinations except the case where the policer is configured in colour-aware mode and traffic stream pre-coloured as red. The difference between the SR3CM and the TR3CM determined from the ratio between red and yellow counters.

Table 4. Decision table for all policer’s modes with SR3CM and TR3CM algorithms.

C1: Mode: colour-aware	X	X	X	X	X	X	-	-	-	-	-	-

C2: Mode: colour-blind	-	-	-	-	-	-	X	X	X	X	X	X
C3: Action: drop	X	X	X	-	-	-	X	X	X	-	-	-
C4: Action: mark red	-	-	-	X	X	X	-	-	-	X	X	X
C6: SR3CM/TR3CM	X	X	X	X	X	X	X	X	X	X	X	X
C8: Traffic: green	X	-	-	X	-	-	X	-	-	X	-	-
C9: Traffic: yellow	-	X	-	-	X	-	-	X	-	-	X	-
C10: Traffic: red	-	-	X	-	-	X	-	-	X	-	-	X
A1: Counter: green	X	0	0	X	0	0	X	X	X	X	X	X
A2: Counter: yellow	X	X	0	X	X	0	X	X	X	X	X	X
A3: Counter: mark red	0	0	0	X	X	X	0	0	0	X	X	X
A4: Counter: drop red	X	X	X	0	0	0	X	X	X	0	0	0

The calculations for determining the algorithm is presented below. Two kind of checks can be implemented for the counters:

1. Check if counter zero or not.
2. Check the exact value of the counters with expected high error level.

The first approach is not precise enough and not flexible to the extra packets that could be forwarded through the node and caught by the Ixia traffic generator. The extra packets can come from such service protocols as ARP, OSPF (Open Shortest Path First) and others. The second approach allows defining the algorithm and the size of the bucket and flexible enough as it is possible to include some gaps for the expected number of packets in order to not fail the test case if any of the service packets will be captured. In general, the policer works with the following schema: if running rate is twice higher than the CIR, each second packet will be dropped. That dependency allows counting the expected rate of the red packets after the bucket became empty.

Figure 25 presents the main areas of the graph that should be considered in the algorithm of checking the counters.

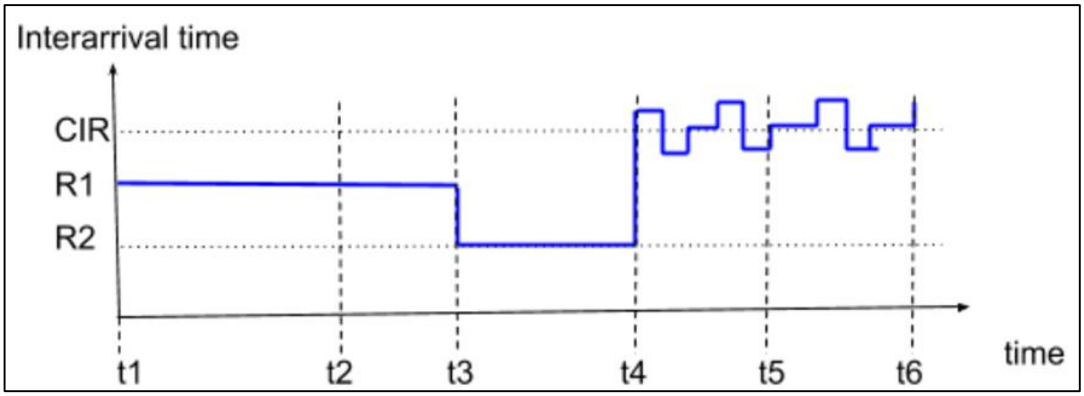

Figure 25. The inter-arrival time of the packets with distinguished areas for the counters check.

The first interval [t1, t2] shows the time of the traffic run with the rate below the CIR. The number of Kbits arrived to Ixia port is equal to $R1 * t$ Kbits/s. The second interval [t2, t3] represents the time of reaction on the command of increasing of the rate sent to the Ixia. This interval is one of the points that decrease the accuracy of the measurement, as it is impossible to determine precisely the time of reaction on the command. It should be measured from the time stamps before command and after it. After the t3 point the running rate will not be changed and will be equal to the R2. The third interval [t3, t4] represents the burst interval, but because of the test restrictions, it is impossible to determine the precise size of the burst interval, although it is possible to calculate the approximate size and time of the burst interval. The burst interval contains two kinds of packets:

1. Packets that marked as a part of the CIR.
2. Packets that consume tokens from the bucket.

The second type of the packets has a certain size equal to the EBS. From the equation above the expected burst size can be calculated and the approximate time of the burst interval can be calculated as:

$$BurstTime = \frac{ExpectedBurstSize}{R2}. \quad (5)$$

The rest of the time on interval [t4, t6] the rate of the traffic is restricted on the level of the CIR rate. There are just two periods that are known exactly by developer: [t1, t2], [t3, t5]. The period between points t4 and t5 can be calculated as the difference between intervals [t3, t5] and the BurstTime from the equation 5. The last period [t5, t6] represents the time of reaction on the command of stopping the traffic generation. This period cannot be calculated and it should be measured from the several runs of the special test case aimed at the testing and gathering measurements for the “Stop traffic generation command”. The result of the information provided above can be presented by the following equation:

$$ExpectedBytes = \mathrm{R1} * \mathrm{t13} + \mathrm{ExpectedBurstSize} + \mathrm{CIR} * (\mathrm{t36} - \mathrm{BurstTime}), \quad (6)$$

where t13 is a time of the period [t1; t3], t36 is a time of the period [t3; t6]. The only period of marking packets as red is [t4; t6]. Based on it, the expected size of the counters for single-rate two-colour marking policer can be calculated from the equations below:

$$ExpectedRedBytes = (R2 - CIR) * t46, \quad (7)$$

$$ExpectedYellowBytes = 0, \quad (8)$$

$$ExpectedGreenBytes = ExpectedBytes - ExpectedRedPackets. \quad (9)$$

For the single-rate three-color marking algorithm, the equations will be different:

$$\mathrm{BurstTime} = \frac{(\mathrm{EBS} + \mathrm{CBS}) * \mathrm{R2}}{1 - \frac{\mathrm{CIR}}{R2}}, \quad (10)$$

$$ExpectedBytes = \mathrm{R1} * \mathrm{t13} + \mathrm{ExpectedBurstSize} + \mathrm{CIR} * (\mathrm{t36} - \mathrm{BurstTime}), \quad (11)$$

$$ExpectedGreenBytes = \mathrm{ExpectedByts} - \mathrm{ExpectedRedBytes} - \mathrm{EBS}, \quad (12)$$

$$ExpectedYellowBytes = EBS, \quad (13)$$

$$ExpectedRedBytes = (R2 - CIR) * t46. \quad (14)$$

The two-rate three-colour marking algorithm has several differences from single-rate three-colour algorithm:

1. The result rate of the policer is a sum of CIR and EIR (in case without traffic congestions).
2. In order to get full burst the bucket should be filled with the rate smaller than CIR.
3. The number of yellow packets after t4 calculated according to the equation (15).

$$\text{YellowPacketsAfterBurst} = \text{EIR} * \frac{\text{t6} - \text{t4}}{\text{PacketSize}}, \quad (15)$$

In order to increase the accuracy of calculating the expected packets the fact of running script from a remote workstation and the fact of not immediate commands performing actions to the Ixia should be considered. The problematic areas are explicitly shown in figure 26 below.

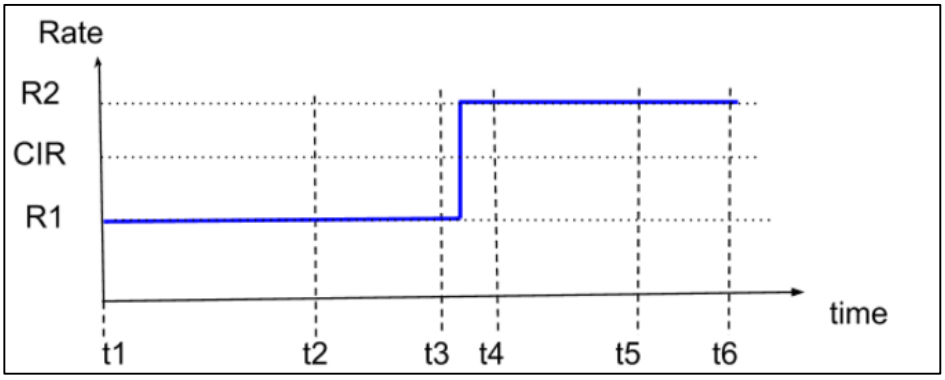

Figure 26. Traffic flow with the time of reacting on the script's commands.

Figure 26 presents the situation where the traffic started with the rate R1 was running during [t1, t3], after that the load on the link was modified and traffic was running with the rate R2 during [t4, t6]. There are three difficult analysis areas. The first one is a period [t1, t2] which represents the traffic that went through the node but the script did not receive an answer from Ixia that a traffic generation process started. The second area is a period [t3, t4], which represents the result of the command that modifies the load on the link. The main problem here is that after the script called a certain function it took some time before Ixia received information about it. Then undefined time was spent on the actual command execution on the server side and as a final step, it took one more undefined period for Ixia to send the response back to the script, so it can continue the execution. The last uncertain period is [t5, t6]. This period represents the time that is needed to stop the traffic.

As a first part of determining the specified problematic areas above, the situation will simplified and the interval [t4, t5] will be excluded, which allows receiving the first equation. The simplified situation is presented in figure 27.

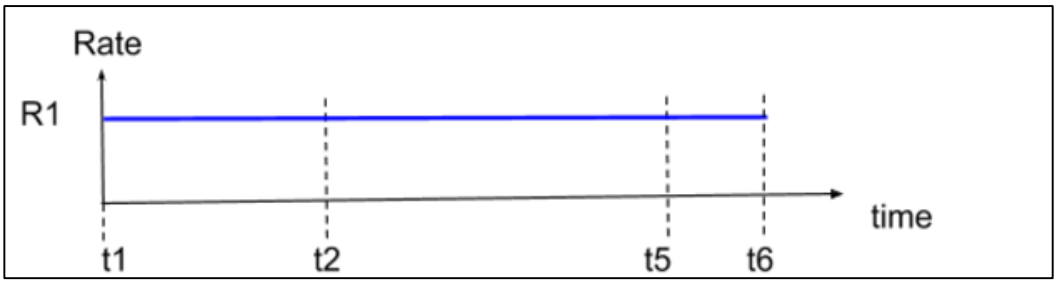

Figure 27. Single rate traffic.

The interval between t2 and t5 in figure 27 is a time during which the traffic runs intentionally. After the traffic run, there is a number of packets received by Ixia from the router and the time period [t1, t6]. R1 is a rate in packets and the number of captured packets is PktCapturedSingleRate. As a result, it is possible to create an equation for the sum of periods t12 and t56:

$$\mathrm{t12} + \mathrm{t56} = \frac{\mathrm{PktCapturedSingleRate} - \mathrm{R1} * \mathrm{t25}}{R1}, \tag{16}$$

After several tests, it was defined that the sum of t12 and t56 is the same in all the cases within the different rates and equal to 2.4 seconds.

After the sum of periods t12 and t56 was received it is possible to return to the more complicated case shown in figure 26. At first, the period t34 should be determine. As was mentioned above, this period related to the adjusting of the traffic rate command. The difference between these commands from the start/stop traffic commands is that it is an atomic command on the server side, but it is still necessary to calculate the time of delivering function call to the server and the time of response. For this purpose, the separate test case was created which measured the time of the whole procedure of changing the traffic rate. In all the case's runs it took 0.8 second. As this operation is an atomic on the server, it can be assumed that it was immediate and was perform at the end of the t34 period. This assumption decreases the accuracy of measurement, but still allows satisfying the requirements. Based on the above, the second equation for figure 26 is created:

$$(t12 + t34) * R1 + t56 * R2 = PktCapturedTwoRates - t23 * R1 - t45 * R2. \tag{17}$$

The final set of equations for this case is:

$$\begin{cases} t12 + t56 = \dfrac{PktCapturedSingleRate_s - R1 * t25_s}{R1} \\ (t12 + t34) * R1 + t56 * R2 = PktCapturedTwoRates - t23 * R1 - t45 * R2 \end{cases}, \quad (18)$$

where all the variables with the subscript “s” belong to the test case performed with the single rate configuration and the variables without the subscript belong to the test case performed with the two rate configuration.

Based on the created set of equations (18) the result of t12 and t56 can be calculated with following equations:

$$\begin{cases} t56 = \dfrac{PktCapturedTwoRates + (t25_s - t23) * R1 - R2 * t45 - PktCapturedSingleRate_s}{R2 - R1} \\ t12 = \dfrac{\mathrm{PktCapturedSingleRate} - \mathrm{R1} * \mathrm{t25}_s}{R1} - \mathrm{t56} \end{cases}, \quad (19)$$

With known values of the periods t12 and t56, it is possible to determine the value of all the counters for the policer. These values allow distinguishing the policer’s algorithm and its parameters like the size of the buckets, committed rate and excess rate.

8.1.4 Stress testing and key performance indicators testing

The key performance indicator (KPI) testing is a set of tests that checks the border values of the software parameters. These parameters include the highest and the lowest values of the CIR, EIR, CBS, and EBS. All the possible combinations of mentioned parameters allow checking that no side effects of the specific chip implementation were highlighted.

The stress testing is aimed at creating of an intensive flow of data between the CLI and the policer object. The stress testing should perform active switching of the policer object configuration and the entire object, which it depends on. There are seven basic schemas of the dependencies between the objects of different types in figure 26. In order to test the flexibility of the system it is necessary to check that all kinds of updates will be proceeded properly.

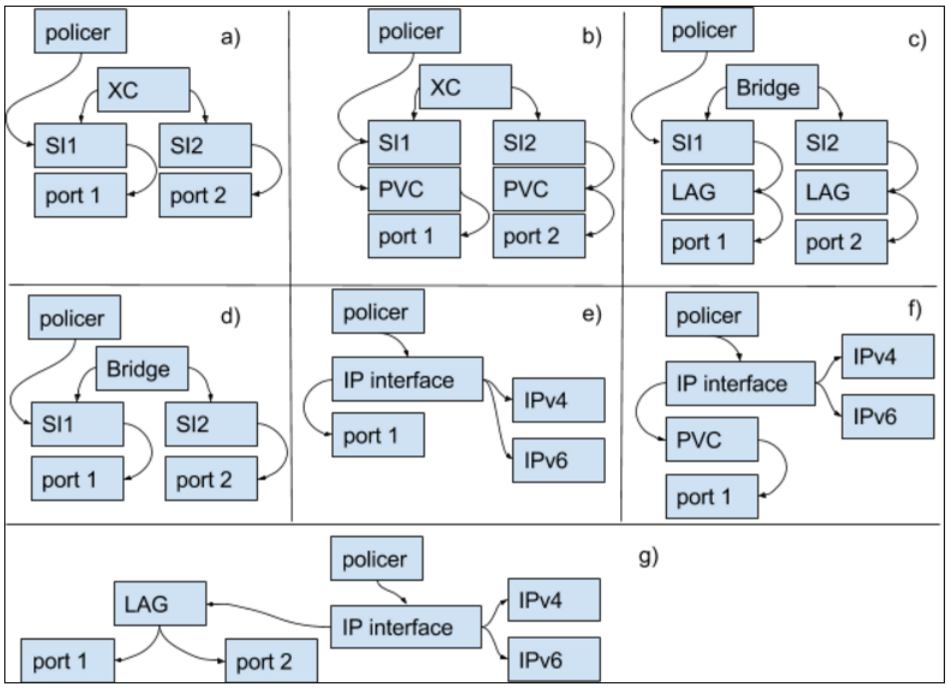

Figure 27. Possible combinations of objects with configured policer.

In order to test the flexibility of the system it is necessary to check that all kinds of updates will be proceeded properly. To create a significant flow of messages from the CLI it is necessary to flap the configuration several times without pauses between the commands. In case ‘a’ in figure 26 there are three possible points that will cause the recreation message flow for the policer: policer recreation, cross-connect (XC) recreation and the service instance (SI1) recreation. Part ‘b’ has the same critical points with one additional point related to the PVC. Creation of the PVC causes the initialization of an additional object and should be tested separately. Case ‘c’ shows the dependency between the bridge and the LAG objects. The policer testing with LAG requires separate hardware configuration as a LAG feature is a combination of ports and non-standard states of LAG in conjunction with the policer should be tested. These states presented with LAG and policer applied to it, but without ports added to it, testing of influence of the LAG with two and three ports on the policer’s result. Area ‘d’ shows the combination of the bridge and the service instance. Cases ‘e’ and ‘f’ represent the combination of the policer with the IP interface object. The case with the configured PVC should be tested separately because of extra dependency. There are two additional dependencies from IP interface to IPv4 and IPv6 IP addresses. These dependencies require additional test cases in order to ensure that adding, deleting of IP address proceeded properly and just configured type of traffic will be policed. Also fast re-configuring of the IP addresses cause intensive message flow, which should be tested as well.

8.2 Testing results

The Router 6000 supports link speed up to 10 Gbit/s. Within the different rates of the traffic and the different combinations of the parameters, the accuracy of performing the traffic restrictions by a policer is different. In order to find the influence of different policer's parameters on its accuracy the traffic testing with the check of the actual burst size described in the previous chapter was performed. The results of this test can be found in appendix 1.

The first scenario was implemented in order to find out the dependency between the committed rate, the burst size and the policer's accuracy. Another parameter that can affect the policer's accuracy is the packet size. It was introduced as a constant value on the lowest possible level in order to increase the test quality for the low rate and the low burst cases. The size of the packet equals to 64 bytes and is the same for all the tests. Figure 28 presents the graph of the error ratio in percentage based on the different rates and burst configured to the policer.

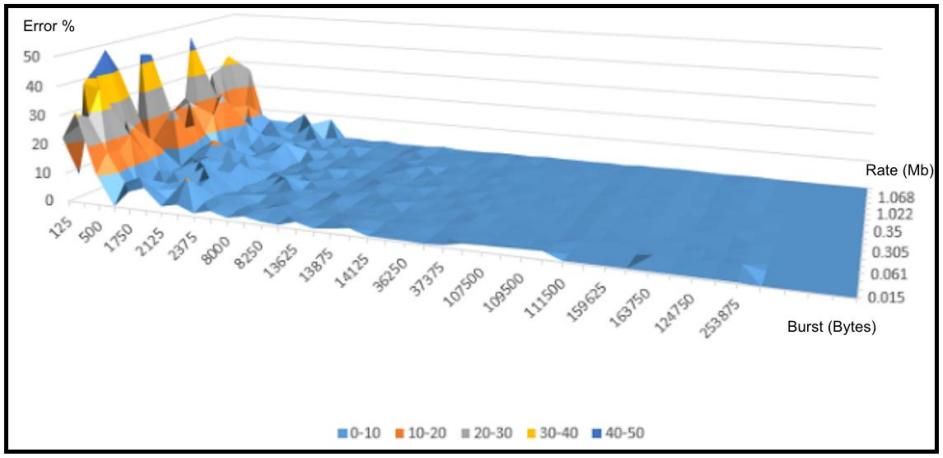

Figure 28. Error ratio by the rate and the burst of the policer.

The aim of the test was to check the actual burst size and compare it with the expected value. It is clearly seen that within the small size of the CBS the error ratio increases exponentially, but with the increasing of the CBS size it decreases despite the value of the rate. Based on figure 28, it is possible to say that the regression equation could be close to the logarithmic function. After excluding the rate from the expression because of its low impact on the result, it is possible to calculate the coefficients by the following equation:

$$Error = a + b * \ln(CBS). \quad (20)$$

To simplify the calculation it is necessary to substitute z = ln (CBS) and the final equation transforms to the linear form:

$$Error = a + b * z, where\ z = ln\ (CBS). \quad (21)$$

After that, it is possible to apply the least-square technique for the linear equation. The equations for the coefficients 'a' and 'b' in respect to the original data will be as follow:

$$\begin{cases} a = \dfrac{N * \sum_{i=1}^{N} \ln(x_i) * y_i \ - \sum_{i=1}^{N} \ln(x_i) \ * \sum_{i=1}^{N} y_i}{N * \sum_{i=1}^{N} \ln^2(x_i) \ - (\sum_{i=1}^{N} \ln(x_i)\ )^2} \\ b = \dfrac{\sum_{i=1}^{N} y_i \ - a * \sum_{i=1}^{N} x_i}{N} \end{cases}, \quad (22)$$

where y = Error, x = CBS, N is a number of experiments for the single rate. After calculation a = 20.0186 and b = -1.696.

To check the quality of the regression function it is necessary to calculate the coefficient of determination. The coefficient of determination or R^2 is a number between 0 and 1, or between 0 and 100 if it represented in percent. It indicates how well experimental data fit to the regression function or statistical model. The coefficient of determination can be calculated with the following equations:

$$\begin{cases} R^2 = 1 - \frac{SS_{reg}}{SS_{tot}} \\ SS_{reg} = \sum_i (f_i - \overline{y})^2 \\ SS_{tot} = \sum_i (y_i - \overline{y})^2, \end{cases} \quad (23)$$

where f_i is a calculated from regression function value of the error, y_i is an experimental value of the error and $\overline{y}$ is a mean value of all the experimental values of the error. According to the equation (23) the R^2 for the logarithmic regression function is equal to 29%, which shows that the chosen type of the regression function is not correct and the statistical model is not reliable.

Another function whose graph looks similar to the experimental data is a hyperbolic function:

$$Error = a + \frac{b}{CBS}, \quad (24)$$

For this function the coefficients 'a' and 'b' can be calculated with the following equations:

$$\begin{cases} a = \dfrac{N * \sum_{i=1}^{N} \ln\left(\frac{1}{x_i}\right) * y_i \; - \sum_{i=1}^{N} \ln\left(\frac{1}{x_i}\right) * \sum_{i=1}^{N} y_i}{N * \sum_{i=1}^{N} \ln^2\left(\frac{1}{x_i}\right) \; - \left(\sum_{i=1}^{N} \ln\left(\frac{1}{x_i}\right)\right)^2} \\ b = \dfrac{\sum_{i=1}^{N} y_i \; - a * \sum_{i=1}^{N} \frac{1}{x_i}}{N} \end{cases}, \quad (25)$$

After the calculations a = 2194.458 and b = 2.303. Compared to the previous regression function the hyperbolic function gives better R^2 value equal to 39%. However, the R^2 rate still shows that the current hypothesis is not good enough. As the next step, data were analysed one more time and it was discovered that the bursts values that were chosen have a nature of not uniform distribution. As a solution for it, it was decided to perform more tests for the single rate, but with the uniform distribution of the burst levels. The data of the tests is presented in appendix 2. After performing the additional test run, the more uniform distribution was received of the experimental points on the graph which allowed me to receive more reliable regression function with R^2 = 98%. The final function is presented in equation (26):

$$Error = \frac{6976.34}{CBS}, \quad (26)$$

R^2 check is not a perfect check as in some cases it does not provide reliable results, so in order to have a better check for the regression function quality it is necessary to check residuals and analyse its set. The residual is defined as difference between the experimental and predicted values. Figure 29 presents the graph of residuals by the burst.

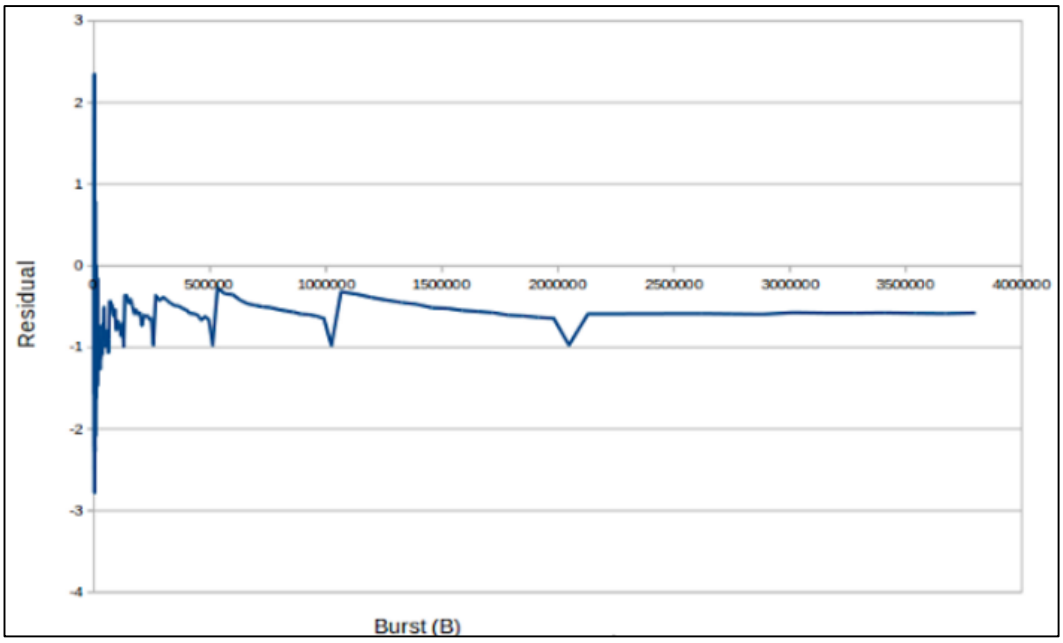

Figure 29. Residuals of the hyperbolic regression function.

It is possible to say that the nature of the residuals is independent from the error value, which also confirms that the calculated regression function has a high level of reliability. Figure 30 presents the received experimentally set of data and the graph of the regression function.

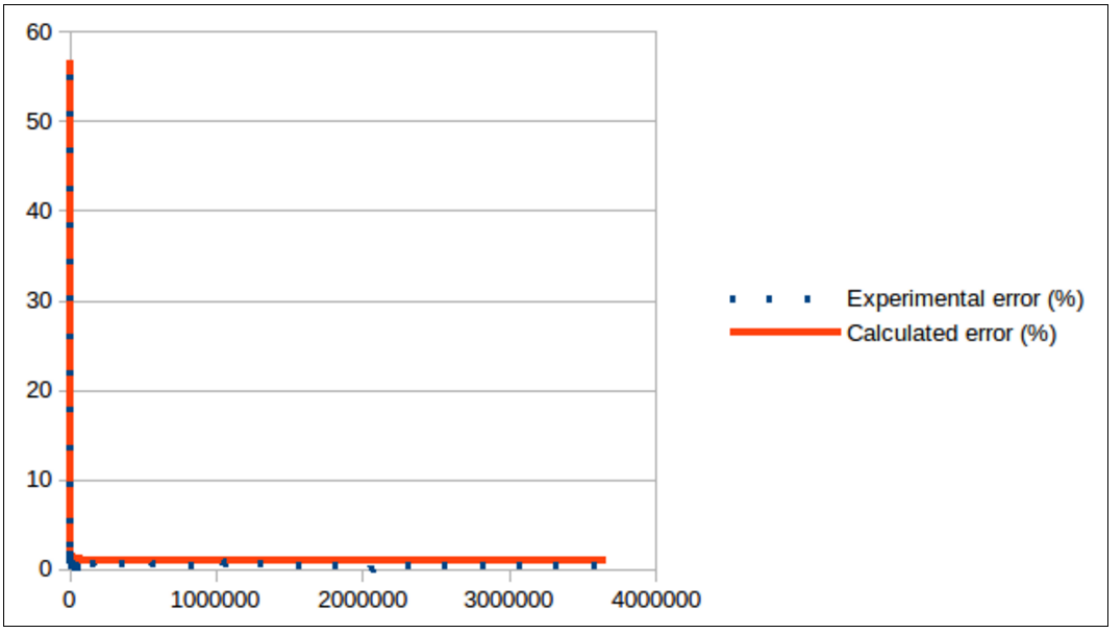

Figure 30. Regression function and experimental data of the policer's error.

Because of the nature of the error check in the tests it is necessary to transform the regression function to the minimal possible Big-O function for the experimental data, so that the final function for the error check has the following equation:

$$Error = 2.3 + \frac{6976.34}{CBS}, \tag{27}$$

where the number 2.3 that raises the regression function graph up was chosen based on the graph of residuals by the burst and allows to ensure that the maximal expected error will be above the maximal value of the error plus maximal level of the residual. That statement can be presented with the following equations:

$$\begin{cases} Error = C + \frac{6976.34}{CBS} \\ C = \max(r_i) \\ r_i = ErrorExp_i - \frac{6976.34}{CBS_i} \end{cases}, \tag{28}$$

where r_i is a residual of the burst with index i and ErrorExp is an experimental value of the error for the specific CBS.

The rest of the algorithms have the same dependencies except for the case of colour-aware mode for the two-rate three-colour policer and the yellow traffic running. In this case, the final rate depends on the CPU clock rate and the rate of transferring tokens to the yellow bucket (EBS). The main problem with these parameters is that they are hidden from the user and the calculations of the final output rate can be difficult. To improve the situation, I will find the regression function that allows to calculate the output rate based on the policer's parameters and the rate of the running traffic flow. The main benefit of this approach is that the output rate can be calculated from the parameters that totally under control of the user. The results of the tests presented in appendix 3, which represents the output rate for the different rates, excess-rates, bursts and the excess-bursts parameters of the policer. The main hypothesis for the regression function is that it is a multiple linear regression. The final regression function is presented by equation (29).

$$R_{out} = -2 * 10^{-6} * CIR + 5.9 * 10^{-9} * CBS + 10^{-3} * EIR + 2.12 * 10^{-7} * EBS + 1.4 * 10^{-3}. \tag{29}$$

It was calculated with the least square method. The residual sum of squares (rss) for this function equals to 8.65 and the R^2 is equal to 99.992954%. The quality of the regression function is on a high level and the residuals are very small, but it is clearly seen the coefficients for the parameters CBS, CIR and EBS are significantly lower than for the rest of them. There are two ways to check if the CBS variable really affects the final rate:

1. Exclude CBS from the data set and perform a new regression analysis.
2. Perform correlation analysis between all the parameters on the original data set.

The first approach allows checking the result with already created equations which will be faster. After the exclusion of the burst from the data set, the following equation was received:

$$R_{out} = -2.09 * 10^{-6} * CIR + 10^{-3} * EIR + 2.11 * 10^{-7} * EBS + 6.4 * 10^{-3} . \quad (30)$$

Residual Sum of Squares: rss = 8.65.
Coefficient of Determination: $R^2 = 9.9929 \cdot 10^{-1}$.
It is clearly seen that the coefficient of determination did not change and the residual sum of squares stayed the same and these facts allow concluding that the burst parameter does not affect the final rate. The next parameter that needed to be checked within the same approach is the EBS, which is an excess-burst in the equation (30). After excluding of the excess-burst from the latest data set the following data were received function was received:

$$R_{out} = -4.75 * 10^{-6} * CIR + 10^{-3} * EIR + 0.13 . \quad (31)$$

Residual Sum of Squares: rss = 13.75.
Coefficient of Determination: $R^2 = 9.9887 \cdot 10^{-1}$.

As can be seen from the comparison of the two latest results, that after the excluding of the excess-burst from the data set the R^2 decreased and the rss increased which shows that the excess-burst parameter has an influence on the output rate of the two-rate three-colour algorithm.

The last parameter that should be checked is the CIR. After excluding the CIR from the equation (30) the final regression equation converted to the equation (32).

$$R_{out} = 7.9 * 10^{-9} * CBS + 10^{-3} * EIR + 2.15 * 10^{-7} * EBS - 1.91 * 10^{-2} . \quad (32)$$

Residual Sum of Squares: rss = 8.704199264.

Coefficient of Determination: $R^2 = 9.992912888 \cdot 10^{-1}$.

As can be seen from the latest results the rss increased and the R^2 value decreased which can be interpreted as the statement that the CIR parameter also affects the final rate of the single-rate three-colour policer. Based on the experiments on the combination of the parameters for TR3CM policer the final regression function for TR3CM policer with yellow traffic running is presented by equation (28).

9 Discussion

9.1 Results

The main result of this project is a stable feature that was integrated into the software solution for the Router 6000 series. As a part of the integration process three sets of test cases were implemented.

The first set checks the feature behaviour in different scenarios. The scenarios covered in this case include testing of the feature applied on top of all kinds of circuits and its combinations. Different kinds of traffic types were running during these test cases and the reaction of the feature on each of them was checked and documented. All the results were compared with the RFC standards. As a result of these checks and comparisons, the behaviour of the feature was described in the documentation with references to the RFC standards which is very important for such heavily standardized industry as “Networks and telecommunications”.

The second set of test cases was implemented in order to check the software behaviour in case of fast changing of the circuits’ states and its configuration flapping. These tests allow checking that in all the possible situations of configuration change, which can call the message storm between the processes of the operating system, the feature will be configured properly and that those changes will not leave the router in an undefined state.

The third set of tests was implemented in order to check the behaviour of the feature that depends on the hardware specifications and limitations. These tests include checks of the policer’s accuracy. The accuracy was explicitly checked for the burst parameter. This set also included test cases that aimed at precise checking of the counters for the policer and decreasing time for the policer’s type checks.

The fourth set of test cases was implemented in order to define the feature behaviour in some specific cases, like TR3CM policer applied to the traffic pre-coloured as yellow. These test cases allowed transferring the calculation of the final result from the parameters not controlled by the customer to the parameters that can be installed by the user.

As a result, it can be stated that during this project the QoS Policing feature was developed with high quality, delivered to the latest stable version of the whole project and that all the test cases were integrated into the continuous loop testing cycle. It can be concluded that the feature development is finished and approved for the current release of software.

9.2 Future Development

Although the results of the tests, the application's behaviour and the quality of the test cases' implementation were accepted, there is a still a window for further development. The precision of the counters' expected values can be improved. Currently the gap between the expected and the actual values of the counters is 10%. In order to improve this situation more tests should be developed. These tests should verify the actual behaviour and the reaction of the Ixia traffic generator and the network environment. In the current implementation, the expected counters' values are calculated only if the bucket is full before the traffic started and if there is just one command that modifies the traffic load applied. The counter's check does not work with the commands start/stop capturing if they were executed between the start/stop traffic commands. The reason for these restrictions is in instability in terms of execution time of the capturing commands. The function for the counters check should be implemented in a more generic way, so that it will accept any sequence of commands to the traffic generator and provide an accurate estimation of the expected values.

For the functionality related to the circuits more test cases should be implemented. These test cases should be able to work with more than three active ports and have more complex configurations with more than two streams at the same time. In addition, the time of reaction between the CLI configuration and the actual applying of the entered commands should be defined.

9.3 Implementation Issues

During the various stages of this project, there were several issues that should be solved. The first problem is related to the Ixia software quality. The traffic generator is not able to provide access to the real-time data through the script, which leads to inventing of the

workarounds during the test cases' developing phase. The second problem is that Ixia cannot send the data of the packet set through the network to the script, which slows down the development process and the script time execution.

The usage of the internal framework for the testing leads to dealing with not very detailed documentation and increased time spent on learning of the basics. The speed of the test cases execution and the project's build time also slowed down the development process along with the lack of hardware resources.

10 Conclusion

This project aimed at the development of the QoS Policing feature and implementation of the verification test cases. The feature was developed according to the RFC standards. Test cases that verified feature behaviour within different sets of routing objects (circuits) configured were implemented. Test cases causing a message storm between the IPOS process were implemented and the feature behaviour in these cases was checked. Additional test cases were implemented in order to highlight the feature behaviour in corner cases and improve the quality of the test cases.

All the test cases were integrated to the continuous integration loop, which allowed constant monitoring of the state of the feature and performing all the necessary checks for each code delivery to the master branch or latest stable version branch. All the data gathered from the test cases was analysed and with the statistical methods several regression functions were created and verified, which allowed transferring the dependencies from the hidden customer values to the parameters that he can actually control, which is a significant improvement of the documentation.

The goals of the project were fully achieved, although there is a still a possibility to improve the test cases. Based on the test results, it was confirmed that all the necessary RFC standards were supported in all the cases and the developed software had a predictable behaviour in all corner cases.

References

1. Ericsson. Ericsson reports fourth quarter and full year results 2014 [online]. Stockholm; January 2015.
   URL: http://www.ericsson.com/news/1889730
   Accessed 26 April 2016.

2. Ericsson External Communications, Region Middle East. Ericsson Mobility Report: Mobile Data Traffic to Grow 16 Times by 2021 in Middle East and North East Africa [online]. Press Release; November 2015.
   URL: http://www.ericsson.com/ae/news/mobile-data-traffic_146370883_c
   Accessed 26 April 2016.

3. Hunn Nick. Ericsson slashes cellular IoT device forecast by 20 billion [online]. Wireless connectivity; February 2016.
   URL: http://www.nickhunn.com/
   ericsson-slashes-cellular-iot-device-forecast-by-20-billion/
   Accessed 26 April 2016.

4. Ericsson. Ericsson at mobile world congress [online]. Barcelona, Spain; March 2015.
   URL: http://www.ericsson.com/mwc2015/#launches
   Accessed 26 April 2016.

5. Ericsson. Ericsson Router 6000 Series [online].
   URL: http://www.ericsson.com/ourportfolio/products/
   router-6000-series?nav=productcategory004|fgb_101_0463
   Accessed 26 April 2016.

6. Cisco Systems, Inc. Internet of Things (IoT) [online]. Cisco Press.
   URL: http://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html
   Accessed 26 April 2016

7. Microsoft. “What is QoS?” [online]. March 2003.
   URL: https://technet.microsoft.com/en-us/library/cc757120(v=ws.10).aspx
   Accessed 26 April 2016.

8. Paul Ferguson, Geoff Huston. Delivering QoS on the internet and in Corporate Networks. Boston, MA: Pearson Education Inc.; January 1998.

9. Cisco Systems, Inc. Quality of Service Networking [online]. Indianapolis, Cisco Press; 1999
   URL: http://docwiki.cisco.com/wiki/Quality_of_Service_Networking
   Accessed 26 April 2016.

10. Cisco Systems, Inc. Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 [online]. Cisco Press: January 2014.
    URL: http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/
    configuration/guide/fqos_c/qcfconmg.html#wp1000872
    Accessed 26 April 2016.

11. Aaron Balchunas. QoS Classification and Marking v1.32 [online].
    URL: http://www.routeralley.com/guides/qos_classification.pdf
    Accessed 26 April 2016.

12. Information Sciences Institute University of Southern California. Internet Protocol [online]. California, Information Sciences Institute University of Southern California: September 1981.
URL: https://tools.ietf.org/html/rfc791
Accessed 26 April 2016.

13. Nichols K., Blake S., Baker F., Black D. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers [online]. The Internet Society: December 1998.
URL: https://tools.ietf.org/html/rfc2474
Accessed 26 April 2016.

14. Grossman D. New Terminology and Clarifications for Diffserv [online]. The Internet Society: April 2002.
URL: https://tools.ietf.org/html/rfc3260
Accessed 26 April 2016.

15. Cisco Systems, Inc. Cisco Nexus 1000V Quality of Service Configuration Guide, Release 4.0(4)SV1(1) [online]. San Jose, Cisco Press: November 2009
URL:http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus1000/sw/4_0/qos/configuration/guide/nexus1000v_qos.pdf
Accessed 26 April 2016.

16. Cisco Systems, Inc. Quality of Service Networking [online]. Cisco Press: October 2012.
URL: http://docwiki.cisco.com/wiki/Quality_of_Service_Networking
Accessed: April 26 2016.

17. Cisco Systems, Inc. Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 [online]. Cisco Press: January 2014.
URL:http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfconmg.html#wp1000872
Accessed: April 26 2016.

18. Cisco Systems, Inc. Cisco 10000 Series Router Quality of Service Configuration Guide [online]. Cisco Press: November 2013.
URL: http://www.cisco.com/c/en/us/td/docs/routers/10000/10008/configuration/guides/qos/qoscf/10qpolce.html#wp1041352
Accessed: April 26 2016.

19. Cisco Systems, Inc. DiffServ -- The Scalable End-to-End QoS Model [online]. Cisco Press: August 2005.
URL: http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html
Accessed April 26 2016.

20. Floyd S., Allman M. Comments on the Usefulness of Simple Best-Effort Traffic [online]. Network Working Group: July 2008.
URL: https://tools.ietf.org/html/rfc5290
Accessed April 26 2016.

21. Vittorio Ghini. QoS-Adaptive Middleware Services Thechnical Report UBLCS-2002-05 [online]. Bologna (Italty), University of Bologna: May 2002.

URL: http://www.informatica.unibo.it/it/ricerca/technical-report/2002/pdfs/2002-05.ps.gz
Accessed April 26 2016.

22. Cisco Systems, Inc. Quality of Service - The Differentiated Services Model [online]. Cisco Press: September 2008.
URL: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/differentiated-services/product_data_sheet0900aecd8031b36d.html
Accessed April 26 2016.

23. Blake S., Black D., Carlson M., Davies E., Wang Z., Weiss W. An Architecture for Differentiated Services [online]. Network Working Group: December 1998.
URL: https://tools.ietf.org/html/rfc2475
Accessed April 26 2016

24. Cisco Systems, Inc. Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2 [online]. Cisco Press: January 2014.
URL: http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfintro.html
Accessed April 26 2016.

25. Silberschatz A., Baer Galvin p., Gagne G. Operating system concepts, 9th Edition. Wiley: October 2012.

26. Vijay Bollapragada, Curtis Murphy, Russ White. Inside Cisco IOS Software Architecture. Indianapolis, Cisco Press: 2000.

27. Cisco Systems, Inc. Monitoring System Processes and Logs [online]. Cisco Press: October 2003.
URL: http://www.cisco.com/en/US/docs/storage/san_switches/mds9000/sw/rel_1_x/1_2_1a/san-os/configuration/guide/SysMontr.html
Accessed April 26 2016.

28. Tutorialspoint. Operating System - LINUX [online]. Tutorialspoint.com: 2016.
URL: http://www.tutorialspoint.com/operating_system/pdf/os_linux.pdf
Accessed April 26 2016.

29. Cisco Systems, Inc. Cisco Router Architecture [online]. USA, Cisco Systems, Inc.: 1988.
URL: http://www.cisco.com/networkers/nw99_pres/601.pdf
Accessed April 26 2016.

30. Juniper Networks, Inc. Junos OS Overview [online]. November 2012.
URL: http://www.juniper.net/documentation/en_US/junos12.3/topics/concept/junos-software-introduction.html
Accessed April 26 2016.

31. Juniper Networks, Inc. juniper OS Architecture Overview [online]. November 2012.
URL:http://www.juniper.net/documentation/en_US/junos12.3/topics/concept/junos-software-architecture.html
Accessed April 26 2016.

32. Ericsson. Ericsson demonstrates common operating system across IP Products at MPLS & Ethernet World Congress [online]. March 2013.
URL: http://www.ericsson.com/news/
130318-common-os-across-ip-products_244129229_c
Accessed April 26 2016.

33. Ericsson. Ericsson Router 6000 Series [online].
URL: http://www.ericsson.com/fi/ourportfolio/products/
router-6000-series?nav=productcategory004|fgb_101_0463
Accessed April 26 2016.

34. Ericsson. Sercomtel modernizes broadband user management with an unprecedented solution in Brazil [online]. October 2014.
URL: http://www.ericsson.com/lc/news/2014-10-16-sercomtel-en_254740125_c
Accessed April 26 2016.

35. Ixia. Testing, visibility, and security solutions to strengthen applications across physical and virtual networks [online]. Ixia: 2016.
URL: https://www.ixiacom.com/company
Accessed April 26 2016.

36. Ixia. XM2 Portable Chassis [online]. Clabasas, SA, Ixia: July 2015
URL: https://www.ixiacom.com/sites/default/files/resources/datasheet/
ch_optixia_xm2.pdf
Accessed April 26 2016.

37. Ixia. Network Test Solutions [online].
URL: http://www.ixiacom.com/solutions/network-test-solutions
Accessed January 14 2016.

38. Ixia. Routing and Switching [online].
URL: http://www.ixiacom.com/routing-switching
Accessed January 22 2016.

39. Ixia. IxOS Tcl Development Guide Release 5.50. Calabasas SA, Ixia: July 2009.

40. Ixia. Tcl Automation Environment [online].
URL: https://www.ixiacom.com/sites/default/files/resources/datasheet/tcl_api.pdf
Accessed April 26 2016.

41. Ericsson. Ericsson Router 6672 [online].
URL: http://www.ericsson.com/ourportfolio/products/router-6672?nav=
productcategory004|fgb_101_0463|fgb_101_0515
Accessed April 26 2016.

42. Robert C. Martin. Agile Software Development: Principles, Patterns, and Practices. USA, Prentice Hall: 2002.

43. Kenneth S. Rubin. “Essential Scrum. A Practical Guide To The Most Popular Agile Process”. Boston MA, Pearson Education Inc.: July 2012.

44. Paul M. Duvall. Continuous Integration Improving Software Quality and Reducing Risk. Boston MA, Pearson Education Inc.: June 2007.

45. Tommi Virtanen. Git for Computer Scientists [online].
URL: http://eagain.net/articles/git-for-computer-scientists/
Accessed 26 April 2016.

46. John Wigley. Git from the bottom up [online]. December 2009.
URL: http://ftp.newartisans.com/pub/git.from.bottom.up.pdf
Accessed April 26 2016.

47. Americas Headquarters Cisco Systems, Inc. (2013). Cisco 10000 Series Router Quality of Service Configuration Guide [online].
URL:http://www.cisco.com/c/en/us/td/docs/routers/10000/10008/configuration/guides/qos/qoscf.pdf
Accessed 21 March 2015.

48. [62] Nandagopal, T. (2013). Single and dual rate three color marker systems, United States Patent US8385206, [online].
URL:https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US8385206.pdf.
Accessed 20 March 2015

49. Juniper Networks Inc. Basic Single-Rate Two-Color Policers [online]. February 2012.
URL: http://www.juniper.net/documentation/en_US/junos12.3/topics/topic-map/policer-single-rate-two-color.html#jd0e156
Accessed April 26 2016.

50. Juniper Networks Inc. Basic Single-Rate Two-Color Policers [online]. April 2014.
URL: http://www.juniper.net/documentation/en_US/junos15.1/topics/topic-map/policer-single-rate-two-color.html#jd0e166
Accessed April 26 2016.

51. Heinane J., Guerin R. A Single Rate Three Color Marker [online]. Network Working Group: September 1999.
URL: https://tools.ietf.org/html/rfc2697
Accessed April 26 2016.

52. Aboul-Magd, O., Rabie, S. (2005). A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic [online]. The Internet Society: December 1998.
URL:https://tools.ietf.org/html/rfc2474
Accessed 23 March 2015.

53. Hakyong, K., Changmo, Y., Woo-Young, J. (2003). Simulation Study on the Effect of the trTCM Parameters [online]. 03, 1482-1488.
URL:http://ieeexplore.ieee.org.ezproxy.metropolia.fi/stamp/stamp.jsp?tp=&arnumber=1191653.
Accessed 23 March 2015.

54. Aboul-Magd O., Rabie S. A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic [online]. Network Working Group: July 2005.
URL: https://tools.ietf.org/html/rfc4115
Accessed 26 April 2016.

55. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns Elements of Reusable Object-Oriented Software. USA, Pearson Education Corporate Sales Division: November 2000.

56. Jorgensen Paul C. Software testing A Craftsman's approach Fourth Edition. Northwest, Washington, D.C., CRC Press Taylor & Francis Group: October 2013.

Appendix 1. Error rate by the different burst sizes and the commit rates.

Error (Abs. value)	Commit rate (Mb/s)	Burst (B)
0.005349626452	1.052	37375
0.005068666457	1.052	37875
0.003168494917	1.052	107500
0.003911072408	1.052	108500
0.004248539945	1.052	109500
0.002824994325	1.052	110500
0.00396747104	1.052	111500
0.005541006656	1.052	157625
0.005674052963	1.052	159625
0.005098188207	1.052	161750
0.004991302978	1.052	163750
0.003318491581	1.052	247750
0.002806455683	1.052	124750
0.002825568368	1.052	251875
0.003597321706	1.052	253875
0.000530079653	1.052	256000
0.01715805784	1.068	125
0.2933184479	1.068	250
0.05736050737	1.068	500
0.1462876284	1.068	625
0.04498627652	1.068	1750
0.05719924886	1.068	2000
0.0164781223	1.068	2125
0.001040108454	1.068	2250
0.0293543983	1.068	2375
0.007313895605	1.068	7750
0.009197133177	1.068	8000
0.00890811965	1.068	8125
0.01082121376	1.068	8250
0.0067233129	1.068	8375
0.005915391963	1.068	13625
0.01541555063	1.068	13750
0.003199374163	1.068	13875
0.003920614012	1.068	14000
0.002335588718	1.068	14125
0.001748464563	1.068	35750
0.00559968038	1.068	36250
0.0009658631493	1.068	36750
0.0007066723091	1.068	37375
0.001070684537	1.068	37875
0.003088766382	1.068	107500
0.003699257458	1.068	108500
0.004585575049	1.068	109500
0.003286407791	1.068	110500
0.003836923445	1.068	111500
0.006164906722	1.068	157625
0.004734086169	1.068	159625
0.005829229805	1.068	161750
0.006257322054	1.068	163750
0.004065505291	1.068	247750
0.00299900042	1.068	124750
0.004097049142	1.068	251875

0.003884743852	1.068	253875
0.0005187586323	1.068	256000
0.1767733171	4.943	125
0.3348879409	4.943	250
0.0880788535	4.943	500
0.004482688414	4.943	625
0.07472145804	4.943	1750
0.03941242703	4.943	2000
0.01909985596	4.943	2125
0.06968147095	4.943	2250
0.006157484671	4.943	2375
0.02285301161	4.943	7750
0.01255880849	4.943	8000
0.01034401707	4.943	8125
0.004620838658	4.943	8250
0.006825220599	4.943	8375
0.005302171439	4.943	13625
0.00009348421591	4.943	13750
0.001297254831	4.943	13875
0.002328570176	4.943	14000
0.00242834358	4.943	14125
0.003683295866	4.943	35750
0.006324383748	4.943	36250
0.003684927251	4.943	36750
0.007812500715	4.943	37375
0.003080027165	4.943	37875
0.004744160651	4.943	107500
0.003303349553	4.943	108500
0.003541040298	4.943	109500
0.003033960567	4.943	110500
0.003861406719	4.943	111500
0.005957718912	4.943	157625
0.005454028916	4.943	159625
0.005255380003	4.943	161750
0.005456713561	4.943	163750
0.004231819642	4.943	247750
0.001528502962	4.943	124750
0.004043095973	4.943	251875
0.003676107893	4.943	253875
0.0003363401604	4.943	256000
0.3076216439	5.004	125
0.1845802632	5.004	250
0.268402421	5.004	500
0.02537568261	5.004	625
0.01022204637	5.004	1750
0.05467991553	5.004	2000
0.09616431928	5.004	2125
0.0108967275	5.004	2250
0.08292492744	5.004	2375
0.009342110589	5.004	7750
0.0109951162	5.004	8000
0.004958183316	5.004	8125
0.0112980022	5.004	8250
0.00330414917	5.004	8375
0.004744533569	5.004	13625

0.003725469491	5.004	13750
0.009685083293	5.004	13875
0.00291808026	5.004	14000
0.002284816906	5.004	14125
0.002655642262	5.004	35750
0.005509740891	5.004	36250
0.00458594449	5.004	36750
0.002415151766	5.004	37375
0.006359829888	5.004	37875
0.00395974581	5.004	107500
0.004179947409	5.004	108500
0.003093186572	5.004	109500
0.004766812798	5.004	110500
0.002018958697	5.004	111500
0.006583203336	5.004	157625
0.004764020951	5.004	159625
0.006338949746	5.004	161750
0.005850210382	5.004	163750
0.003644248939	5.004	247750
0.002008036094	5.004	124750
0.003743679503	5.004	251875
0.003765123665	5.004	253875
0.0004327679756	5.004	256000
0.353479619	5.065	125
0.2693184669	5.065	250
0.05321542035	5.065	500
0.1109258873	5.065	625
0.000227100631	5.065	1750
0.05797280349	5.065	2000
0.06856582262	5.065	2125
0.0101484077	5.065	2250
0.01487086849	5.065	2375
0.01791902797	5.065	7750
0.001375632199	5.065	8000
0.0007322322849	5.065	8125
0.008597049697	5.065	8250
0.01800285453	5.065	8375
0.005994272525	5.065	13625
0.001895221364	5.065	13750
0.005422255715	5.065	13875
0.01406675544	5.065	14000
0.003771124484	5.065	14125
0.005459735542	5.065	35750
0.006593912749	5.065	36250
0.004862153823	5.065	36750
0.003580964626	5.065	37375
0.005021913232	5.065	37875
0.004204911503	5.065	107500
0.004222114066	5.065	108500
0.003406174967	5.065	109500
0.003620550309	5.065	110500
0.004319152755	5.065	111500
0.005084085135	5.065	157625
0.005501092348	5.065	159625
0.004863557433	5.065	161750

0.00537336499	5.065	163750
0.00333883366	5.065	247750
0.003714789652	5.065	124750
0.003728432876	5.065	251875
0.004100188259	5.065	253875
0.0001871394122	5.065	256000
0.3934014738	13.793	125
0.005349626452	1.052	37375
0.005068666457	1.052	37875
0.003168494917	1.052	107500
0.003911072408	1.052	108500
0.004248539945	1.052	109500
0.002824994325	1.052	110500
0.00396747104	1.052	111500
0.005541006656	1.052	157625
0.005674052963	1.052	159625
0.005098188207	1.052	161750
0.004991302978	1.052	163750
0.003318491581	1.052	247750
0.002806455683	1.052	124750
0.002825568368	1.052	251875
0.003597321706	1.052	253875
0.000530079653	1.052	256000
0.01715805784	1.068	125
0.2933184479	1.068	250
0.05736050737	1.068	500
0.1462876284	1.068	625
0.04498627652	1.068	1750
0.05719924886	1.068	2000
0.0164781223	1.068	2125
0.001040108454	1.068	2250
0.0293543983	1.068	2375
0.007313895605	1.068	7750
0.009197133177	1.068	8000
0.00890811965	1.068	8125
0.01082121376	1.068	8250
0.0067233129	1.068	8375
0.005915391963	1.068	13625
0.01541555063	1.068	13750
0.003199374163	1.068	13875
0.003920614012	1.068	14000
0.002335588718	1.068	14125
0.001748464563	1.068	35750
0.00559968038	1.068	36250
0.0009658631493	1.068	36750
0.0007066723091	1.068	37375
0.001070684537	1.068	37875
0.003088766382	1.068	107500
0.003699257458	1.068	108500
0.004585575049	1.068	109500
0.003286407791	1.068	110500
0.003836923445	1.068	111500
0.006164906722	1.068	157625
0.004734086169	1.068	159625
0.005829229805	1.068	161750

0.006257322054	1.068	163750
0.004065505291	1.068	247750
0.00299900042	1.068	124750
0.004097049142	1.068	251875
0.003884743852	1.068	253875
0.0005187586323	1.068	256000
0.1767733171	4.943	125
0.3348879409	4.943	250
0.0880788535	4.943	500
0.004482688414	4.943	625
0.07472145804	4.943	1750
0.03941242703	4.943	2000
0.01909985596	4.943	2125
0.06968147095	4.943	2250
0.006157484671	4.943	2375
0.02285301161	4.943	7750
0.01255880849	4.943	8000
0.01034401707	4.943	8125
0.004620838658	4.943	8250
0.006825220599	4.943	8375
0.005302171439	4.943	13625
0.00009348421591	4.943	13750
0.001297254831	4.943	13875
0.002328570176	4.943	14000
0.00242834358	4.943	14125
0.003683295866	4.943	35750
0.006324383748	4.943	36250
0.003684927251	4.943	36750
0.007812500715	4.943	37375
0.003080027165	4.943	37875
0.004744160651	4.943	107500
0.003303349553	4.943	108500
0.003541040298	4.943	109500
0.003033960567	4.943	110500
0.003861406719	4.943	111500
0.005957718912	4.943	157625
0.005454028916	4.943	159625
0.005255380003	4.943	161750
0.005456713561	4.943	163750
0.004231819642	4.943	247750
0.001528502962	4.943	124750
0.004043095973	4.943	251875
0.003676107893	4.943	253875
0.0003363401604	4.943	256000
0.3076216439	5.004	125
0.1845802632	5.004	250
0.268402421	5.004	500
0.02537568261	5.004	625
0.01022204637	5.004	1750
0.05467991553	5.004	2000
0.09616431928	5.004	2125
0.0108967275	5.004	2250
0.08292492744	5.004	2375
0.009342110589	5.004	7750
0.0109951162	5.004	8000

0.004958183316	5.004	8125
0.0112980022	5.004	8250
0.00330414917	5.004	8375
0.004744533569	5.004	13625
0.003725469491	5.004	13750
0.009685083293	5.004	13875
0.00291808026	5.004	14000
0.002284816906	5.004	14125
0.002655642262	5.004	35750
0.005509740891	5.004	36250
0.00458594449	5.004	36750
0.002415151766	5.004	37375
0.006359829888	5.004	37875
0.00395974581	5.004	107500
0.004179947409	5.004	108500
0.003093186572	5.004	109500
0.004766812798	5.004	110500
0.002018958697	5.004	111500
0.006583203336	5.004	157625
0.004764020951	5.004	159625
0.006338949746	5.004	161750
0.005850210382	5.004	163750
0.003644248939	5.004	247750
0.002008036094	5.004	124750
0.003743679503	5.004	251875
0.003765123665	5.004	253875
0.0004327679756	5.004	256000
0.353479619	5.065	125
0.2693184669	5.065	250
0.05321542035	5.065	500
0.1109258873	5.065	625
0.000227100631	5.065	1750
0.05797280349	5.065	2000
0.06856582262	5.065	2125
0.0101484077	5.065	2250
0.01487086849	5.065	2375
0.01791902797	5.065	7750
0.001375632199	5.065	8000
0.0007322322849	5.065	8125
0.008597049697	5.065	8250
0.01800285453	5.065	8375
0.005994272525	5.065	13625
0.001895221364	5.065	13750
0.005422255715	5.065	13875
0.01406675544	5.065	14000
0.003771124484	5.065	14125
0.005459735542	5.065	35750
0.006593912749	5.065	36250
0.004862153823	5.065	36750
0.003580964626	5.065	37375
0.005021913232	5.065	37875
0.004204911503	5.065	107500
0.004222114066	5.065	108500
0.003406174967	5.065	109500
0.003620550309	5.065	110500

0.004319152755	5.065	111500
0.005084085135	5.065	157625
0.005501092348	5.065	159625
0.004863557433	5.065	161750
0.00537336499	5.065	163750
0.00333883366	5.065	247750
0.003714789652	5.065	124750
0.003728432876	5.065	251875
0.004100188259	5.065	253875
0.0001871394122	5.065	256000
0.3934014738	13.793	125
0.005349626452	1.052	37375
0.005068666457	1.052	37875
0.003168494917	1.052	107500
0.003911072408	1.052	108500
0.004248539945	1.052	109500
0.002824994325	1.052	110500
0.00396747104	1.052	111500
0.005541006656	1.052	157625
0.005674052963	1.052	159625
0.005098188207	1.052	161750
0.004991302978	1.052	163750
0.003318491581	1.052	247750
0.002806455683	1.052	124750
0.002825568368	1.052	251875
0.003597321706	1.052	253875
0.000530079653	1.052	256000
0.01715805784	1.068	125
0.2933184479	1.068	250
0.05736050737	1.068	500
0.1462876284	1.068	625
0.04498627652	1.068	1750
0.05719924886	1.068	2000
0.0164781223	1.068	2125
0.001040108454	1.068	2250
0.0293543983	1.068	2375
0.007313895605	1.068	7750
0.009197133177	1.068	8000
0.00890811965	1.068	8125
0.01082121376	1.068	8250
0.0067233129	1.068	8375
0.005915391963	1.068	13625
0.01541555063	1.068	13750
0.003199374163	1.068	13875
0.003920614012	1.068	14000
0.002335588718	1.068	14125
0.001748464563	1.068	35750
0.00559968038	1.068	36250
0.0009658631493	1.068	36750
0.0007066723091	1.068	37375
0.001070684537	1.068	37875
0.003088766382	1.068	107500
0.003699257458	1.068	108500
0.004585575049	1.068	109500
0.003286407791	1.068	110500

0.003836923445	1.068	111500
0.006164906722	1.068	157625
0.004734086169	1.068	159625
0.005829229805	1.068	161750
0.006257322054	1.068	163750
0.004065505291	1.068	247750
0.00299900042	1.068	124750
0.004097049142	1.068	251875
0.003884743852	1.068	253875
0.0005187586323	1.068	256000
0.1767733171	4.943	125
0.3348879409	4.943	250
0.0880788535	4.943	500
0.004482688414	4.943	625
0.07472145804	4.943	1750
0.03941242703	4.943	2000
0.01909985596	4.943	2125
0.06968147095	4.943	2250
0.006157484671	4.943	2375
0.02285301161	4.943	7750
0.01255880849	4.943	8000
0.01034401707	4.943	8125
0.004620838658	4.943	8250
0.006825220599	4.943	8375
0.005302171439	4.943	13625
0.00009348421591	4.943	13750
0.001297254831	4.943	13875
0.002328570176	4.943	14000
0.00242834358	4.943	14125
0.003683295866	4.943	35750
0.006324383748	4.943	36250
0.003684927251	4.943	36750
0.007812500715	4.943	37375
0.003080027165	4.943	37875
0.004744160651	4.943	107500
0.003303349553	4.943	108500
0.003541040298	4.943	109500
0.003033960567	4.943	110500
0.003861406719	4.943	111500
0.005957718912	4.943	157625
0.005454028916	4.943	159625
0.005255380003	4.943	161750
0.005456713561	4.943	163750
0.004231819642	4.943	247750
0.001528502962	4.943	124750
0.004043095973	4.943	251875
0.003676107893	4.943	253875
0.0003363401604	4.943	256000
0.3076216439	5.004	125
0.1845802632	5.004	250
0.268402421	5.004	500
0.02537568261	5.004	625
0.01022204637	5.004	1750
0.05467991553	5.004	2000
0.09616431928	5.004	2125

0.0108967275	5.004	2250
0.08292492744	5.004	2375
0.009342110589	5.004	7750
0.0109951162	5.004	8000
0.004958183316	5.004	8125
0.0112980022	5.004	8250
0.00330414917	5.004	8375
0.004744533569	5.004	13625
0.003725469491	5.004	13750
0.009685083293	5.004	13875
0.00291808026	5.004	14000
0.002284816906	5.004	14125
0.002655642262	5.004	35750
0.005509740891	5.004	36250
0.00458594449	5.004	36750
0.002415151766	5.004	37375
0.006359829888	5.004	37875
0.00395974581	5.004	107500
0.004179947409	5.004	108500
0.003093186572	5.004	109500
0.004766812798	5.004	110500
0.002018958697	5.004	111500
0.006583203336	5.004	157625
0.004764020951	5.004	159625
0.006338949746	5.004	161750
0.005850210382	5.004	163750
0.003644248939	5.004	247750
0.002008036094	5.004	124750
0.003743679503	5.004	251875
0.003765123665	5.004	253875
0.0004327679756	5.004	256000
0.353479619	5.065	125
0.2693184669	5.065	250
0.05321542035	5.065	500
0.1109258873	5.065	625
0.000227100631	5.065	1750
0.05797280349	5.065	2000
0.06856582262	5.065	2125
0.0101484077	5.065	2250
0.01487086849	5.065	2375
0.01791902797	5.065	7750
0.001375632199	5.065	8000
0.0007322322849	5.065	8125
0.008597049697	5.065	8250
0.01800285453	5.065	8375
0.005994272525	5.065	13625
0.001895221364	5.065	13750
0.005422255715	5.065	13875
0.01406675544	5.065	14000
0.003771124484	5.065	14125
0.005459735542	5.065	35750
0.006593912749	5.065	36250
0.004862153823	5.065	36750
0.003580964626	5.065	37375
0.005021913232	5.065	37875

0.004204911503	5.065	107500
0.004222114066	5.065	108500
0.003406174967	5.065	109500
0.003620550309	5.065	110500
0.004319152755	5.065	111500
0.005084085135	5.065	157625
0.005501092348	5.065	159625
0.004863557433	5.065	161750
0.00537336499	5.065	163750
0.00333883366	5.065	247750
0.003714789652	5.065	124750
0.003728432876	5.065	251875
0.004100188259	5.065	253875
0.0001871394122	5.065	256000
0.3934014738	13.793	125
0.005349626452	1.052	37375
0.005068666457	1.052	37875
0.003168494917	1.052	107500
0.003911072408	1.052	108500
0.004248539945	1.052	109500
0.002824994325	1.052	110500
0.00396747104	1.052	111500
0.005541006656	1.052	157625
0.005674052963	1.052	159625
0.005098188207	1.052	161750
0.004991302978	1.052	163750
0.003318491581	1.052	247750
0.002806455683	1.052	124750
0.002825568368	1.052	251875
0.003597321706	1.052	253875
0.000530079653	1.052	256000
0.01715805784	1.068	125
0.2933184479	1.068	250
0.05736050737	1.068	500
0.1462876284	1.068	625
0.04498627652	1.068	1750
0.05719924886	1.068	2000
0.0164781223	1.068	2125
0.001040108454	1.068	2250
0.0293543983	1.068	2375
0.007313895605	1.068	7750
0.009197133177	1.068	8000
0.00890811965	1.068	8125
0.01082121376	1.068	8250
0.0067233129	1.068	8375
0.005915391963	1.068	13625
0.01541555063	1.068	13750
0.003199374163	1.068	13875
0.003920614012	1.068	14000
0.002335588718	1.068	14125
0.001748464563	1.068	35750
0.00559968038	1.068	36250
0.0009658631493	1.068	36750
0.0007066723091	1.068	37375
0.001070684537	1.068	37875

0.003088766382	1.068	107500
0.003699257458	1.068	108500
0.004585575049	1.068	109500
0.003286407791	1.068	110500
0.003836923445	1.068	111500
0.006164906722	1.068	157625
0.004734086169	1.068	159625
0.005829229805	1.068	161750
0.006257322054	1.068	163750
0.004065505291	1.068	247750
0.00299900042	1.068	124750
0.004097049142	1.068	251875
0.003884743852	1.068	253875
0.0005187586323	1.068	256000
0.1767733171	4.943	125
0.3348879409	4.943	250
0.0880788535	4.943	500
0.004482688414	4.943	625
0.07472145804	4.943	1750
0.03941242703	4.943	2000
0.01909985596	4.943	2125
0.06968147095	4.943	2250
0.006157484671	4.943	2375
0.02285301161	4.943	7750
0.01255880849	4.943	8000
0.01034401707	4.943	8125
0.004620838658	4.943	8250
0.006825220599	4.943	8375
0.005302171439	4.943	13625
0.00009348421591	4.943	13750
0.001297254831	4.943	13875
0.002328570176	4.943	14000
0.00242834358	4.943	14125
0.003683295866	4.943	35750
0.006324383748	4.943	36250
0.003684927251	4.943	36750
0.007812500715	4.943	37375
0.003080027165	4.943	37875
0.004744160651	4.943	107500
0.003303349553	4.943	108500
0.003541040298	4.943	109500
0.003033960567	4.943	110500
0.003861406719	4.943	111500
0.005957718912	4.943	157625
0.005454028916	4.943	159625
0.005255380003	4.943	161750
0.005456713561	4.943	163750
0.004231819642	4.943	247750
0.001528502962	4.943	124750
0.004043095973	4.943	251875
0.003676107893	4.943	253875
0.0003363401604	4.943	256000
0.3076216439	5.004	125
0.1845802632	5.004	250
0.268402421	5.004	500

0.02537568261	5.004	625
0.01022204637	5.004	1750
0.05467991553	5.004	2000
0.09616431928	5.004	2125
0.0108967275	5.004	2250
0.08292492744	5.004	2375
0.009342110589	5.004	7750
0.0109951162	5.004	8000
0.004958183316	5.004	8125
0.0112980022	5.004	8250
0.00330414917	5.004	8375
0.004744533569	5.004	13625
0.003725469491	5.004	13750
0.009685083293	5.004	13875
0.00291808026	5.004	14000
0.002284816906	5.004	14125
0.002655642262	5.004	35750
0.005509740891	5.004	36250
0.00458594449	5.004	36750
0.002415151766	5.004	37375
0.006359829888	5.004	37875
0.00395974581	5.004	107500
0.004179947409	5.004	108500
0.003093186572	5.004	109500
0.004766812798	5.004	110500
0.002018958697	5.004	111500
0.006583203336	5.004	157625
0.004764020951	5.004	159625
0.006338949746	5.004	161750
0.005850210382	5.004	163750
0.003644248939	5.004	247750
0.002008036094	5.004	124750
0.003743679503	5.004	251875
0.003765123665	5.004	253875
0.0004327679756	5.004	256000
0.353479619	5.065	125
0.2693184669	5.065	250
0.05321542035	5.065	500
0.1109258873	5.065	625
0.000227100631	5.065	1750
0.05797280349	5.065	2000
0.06856582262	5.065	2125
0.0101484077	5.065	2250
0.01487086849	5.065	2375
0.01791902797	5.065	7750
0.001375632199	5.065	8000
0.0007322322849	5.065	8125
0.008597049697	5.065	8250
0.01800285453	5.065	8375
0.005994272525	5.065	13625
0.001895221364	5.065	13750
0.005422255715	5.065	13875
0.01406675544	5.065	14000
0.003771124484	5.065	14125
0.005459735542	5.065	35750

0.006593912749	5.065	36250
0.004862153823	5.065	36750
0.003580964626	5.065	37375
0.005021913232	5.065	37875
0.004204911503	5.065	107500
0.004222114066	5.065	108500
0.003406174967	5.065	109500
0.003620550309	5.065	110500
0.004319152755	5.065	111500
0.005084085135	5.065	157625
0.005501092348	5.065	159625
0.004863557433	5.065	161750
0.00537336499	5.065	163750
0.00333883366	5.065	247750
0.003714789652	5.065	124750
0.003728432876	5.065	251875
0.004100188259	5.065	253875
0.0001871394122	5.065	256000
0.3934014738	13.793	125
0.005349626452	1.052	37375
0.005068666457	1.052	37875
0.003168494917	1.052	107500
0.003911072408	1.052	108500
0.004248539945	1.052	109500
0.002824994325	1.052	110500
0.00396747104	1.052	111500
0.005541006656	1.052	157625
0.005674052963	1.052	159625
0.005098188207	1.052	161750
0.004991302978	1.052	163750
0.003318491581	1.052	247750
0.002806455683	1.052	124750
0.002825568368	1.052	251875
0.003597321706	1.052	253875
0.000530079653	1.052	256000
0.01715805784	1.068	125
0.2933184479	1.068	250
0.05736050737	1.068	500
0.1462876284	1.068	625
0.04498627652	1.068	1750
0.05719924886	1.068	2000
0.0164781223	1.068	2125
0.001040108454	1.068	2250
0.0293543983	1.068	2375
0.007313895605	1.068	7750
0.009197133177	1.068	8000
0.00890811965	1.068	8125
0.01082121376	1.068	8250
0.0067233129	1.068	8375
0.005915391963	1.068	13625
0.01541555063	1.068	13750
0.003199374163	1.068	13875
0.003920614012	1.068	14000
0.002335588718	1.068	14125
0.001748464563	1.068	35750

0.00559968038	1.068	36250
0.0009658631493	1.068	36750
0.0007066723091	1.068	37375
0.001070684537	1.068	37875
0.003088766382	1.068	107500
0.003699257458	1.068	108500
0.004585575049	1.068	109500
0.003286407791	1.068	110500
0.003836923445	1.068	111500
0.006164906722	1.068	157625
0.004734086169	1.068	159625
0.005829229805	1.068	161750

Appendix 2. Error rate for uniform distribution of the burst sizes' values

Error, %	**Burst (B)**
55.24139693	125
11.57686786	750
8.420511648	1375
2.262567352	2000
1.011673585	2500
3.058236836	3000
0.7887807445	3500
0.4178498081	4125
3.271689507	4625
3.145376383	5125
0.1567444395	5625
0.7108450278	6125
2.008959141	6625
0.3582977968	7125
0.607041749	7625
1.845553038	8250
0.5972456431	8750
0.4518121667	9250
1.18951635	9750
0.7810968657	10250
0.5427478658	10875
0.359853006	11375
0.5259398723	11875
0.1467075687	12375
1.128912957	12875
0.4951031091	13375
0.03291582377	13875
0.3030172939	14375
1.139984026	14875
0.8736432364	15375
1.275514949	16000
0.4362304925	16625
0.5297867542	17625
0.213430819	18625
0.6220323078	19625
0.1662835958	20625
0.5749821663	21750
0.1353857367	22750
0.172747835	23750
0.3570308397	24750
0.004459104159	25750
0.2715746344	26875
0.1222802641	27875
0.1178153092	28875
0.1014140932	29875
0.1518214416	30875
0.1873337837	32000
0.2678400391	33250
0.1155727734	35250

0.4659819323	37375
0.293893262	39375
0.4167938377	41375
0.6519478682	43500
0.2676107581	45500
0.2142604055	47500
0.149885905	49625
0.1540823816	51625
0.3294545278	53750
0.2413276964	55750
0.2963922834	57750
0.2461493937	59875
0.04927581529	61875
0.2091558037	64000
0.666707334	66500
0.663617039	70625
0.5991960738	74750
0.6031054607	78750
0.4816483354	82875
0.4633982778	87000
0.5355548782	91125
0.287025947	95125
0.2965035065	99250
0.3884123766	103375
0.293230634	107500
0.322131939	111500
0.2032924662	115625
0.2211428661	119750
0.3526664657	123875
0.06478833431	128000
0.6874305169	133000
0.680454254	141250
0.5915678364	149500
0.6319914738	157625
0.5504164286	165875
0.4506856467	174000
0.4979203906	182250
0.4427598717	190375
0.4534415448	198625
0.304429648	206750
0.4257312185	215000
0.4093347363	223125
0.4180218839	231375
0.3698543976	239500
0.3795234646	247750
0.0559135804	256000
0.6592772993	266125
0.5993833397	282500
0.6381290833	299000
0.597861848	315375
0.5580861535	331750

0.5327006	348125
0.5265373836	364500
0.496528928	380875
0.4776965795	397250
0.4331227876	413625
0.428784103	430000
0.4131111817	446375
0.3502350684	462750
0.3928381007	479125
0.3536162277	495500
0.03965767743	512000
0.7367809208	532375
0.670153137	565125
0.6590788188	598000
0.5888311969	630750
0.5442454088	663500
0.5247525752	696250
0.5063635571	729000
0.497556696	761750
0.473053763	794500
0.4554106539	827375
0.4397288871	860125
0.4153165572	892875
0.4096760474	925625
0.3906774444	958375
0.3635276838	991125
0.03028101398	1024000
0.6881374764	1064875
0.6594638916	1130375
0.6175993671	1196000
0.5835513489	1261500
0.5551835191	1327000
0.5322445648	1392625
0.4910516217	1458125
0.483331027	1523625
0.4564995551	1589125
0.4434601353	1654750
0.4282143693	1720250
0.3992696283	1785750
0.3887717098	1851375
0.3715609942	1916875
0.3598118269	1982375
0.03151658543	2048000
0.7128917744	2129875
0.630318792	2260875
0.3018506338	2392000
0.3770701902	2523125
0.4887963028	2654125
0.4624968926	2785250
0.426667468	2916250
0.47110270771	3047375

0.47782799311	3178375
0.4253033574	3309500

Appendix 3. Output rate of the two-rate three-colour marking policer.

CIR (Kb/s)	CBS (B)	EIR (Kb/s)	EBS (B)	Output Rate (Mb)
1000	125000	1000	125000	1.068311273
1000	125000	2000	125000	2.106461091
1000	125000	3000	125000	3.083170909
1000	125000	5000	125000	5.096913455
1000	125000	7000	125000	7.111028364
1000	125000	10000	125000	10.10185309
1000	125000	20000	125000	20.11098764
1000	125000	30000	125000	30.12049455
1000	125000	1000	125000	1.099589818
1000	250000	1000	125000	1.099589818
1000	625000	1000	125000	1.099589818
1000	875000	1000	125000	1.099589818
1000	1250000	1000	125000	1.099589818
1000	2500000	1000	125000	1.099589818
1000	3750000	1000	125000	1.099589818
1000	125000	1000	125000	1.099589818
1000	125000	1000	250000	1.101451636
1000	125000	1000	625000	1.102196364
1000	125000	1000	875000	1.101824
1000	125000	1000	1250000	1.101824
1000	125000	1000	2500000	1.102568727
1000	125000	1000	3750000	1.101824
5000	125000	1000	125000	1.099217455
5000	125000	2000	125000	2.106461091
5000	125000	3000	125000	3.082798545
5000	125000	5000	125000	5.097285818
5000	125000	7000	125000	7.111400727
5000	125000	10000	125000	10.10148073
5000	125000	20000	125000	20.11136
5000	125000	30000	125000	30.12086691
5000	125000	1000	125000	1.099589818
5000	250000	1000	125000	1.099589818
5000	625000	1000	125000	1.099589818
5000	875000	1000	125000	1.099589818
5000	1250000	1000	125000	1.099589818
5000	2500000	1000	125000	1.099589818
5000	3750000	1000	125000	1.095121455
5000	125000	1000	125000	1.099217455
5000	125000	1000	250000	1.101451636
5000	125000	1000	625000	1.102196364
5000	125000	1000	875000	1.101824
5000	125000	1000	1250000	1.101824
5000	125000	1000	2500000	1.102941091
5000	125000	1000	3750000	1.102196364
6000	125000	1000	125000	1.099217455
6000	125000	2000	125000	2.106461091
6000	125000	3000	125000	3.082798545
6000	125000	5000	125000	5.097285818
6000	125000	7000	125000	7.111028364
6000	125000	10000	125000	10.10148073
6000	125000	20000	125000	20.11098764
6000	125000	30000	125000	30.12012218

6000	125000	1000	125000	1.099589818
6000	250000	1000	125000	1.099589818
6000	625000	1000	125000	1.099589818
6000	875000	1000	125000	1.099589818
6000	1250000	1000	125000	1.099589818
6000	2500000	1000	125000	1.099217455
6000	3750000	1000	125000	1.099589818
6000	125000	1000	125000	1.099589818
6000	125000	1000	250000	1.102196364
6000	125000	1000	625000	1.101451636
6000	125000	1000	875000	1.101079273
6000	125000	1000	1250000	1.102196364
6000	125000	1000	2500000	1.101451636
6000	125000	1000	3750000	1.751226182
7000	125000	1000	125000	1.099589818
7000	125000	2000	125000	2.106461091
7000	125000	3000	125000	3.083170909
7000	125000	5000	125000	5.097285818
7000	125000	7000	125000	7.111400727
7000	125000	10000	125000	10.10185309
7000	125000	20000	125000	20.11136
7000	125000	30000	125000	30.12012218
7000	125000	1000	125000	1.099589818
7000	250000	1000	125000	1.099589818
7000	625000	1000	125000	1.099589818
7000	875000	1000	125000	1.099589818
7000	1250000	1000	125000	1.099589818
7000	2500000	1000	125000	1.099589818
7000	3750000	1000	125000	1.099589818
7000	125000	1000	125000	1.099589818
7000	125000	1000	250000	1.101824
7000	125000	1000	625000	1.101451636
7000	125000	1000	875000	1.101824
7000	125000	1000	1250000	1.102196364
7000	125000	1000	2500000	1.101824
7000	125000	1000	3750000	1.102196364
10000	125000	1000	125000	1.099589818
10000	125000	2000	125000	2.106461091
10000	125000	3000	125000	3.083170909
10000	125000	5000	125000	5.097285818
10000	125000	7000	125000	7.111028364
10000	125000	10000	125000	10.10185309
10000	125000	20000	125000	20.11136
10000	125000	30000	125000	30.12012218
10000	125000	1000	125000	1.099589818
10000	250000	1000	125000	1.099589818
10000	625000	1000	125000	1.099589818
10000	875000	1000	125000	1.099589818
10000	1250000	1000	125000	1.099589818
10000	2500000	1000	125000	1.099589818
10000	3750000	1000	125000	1.099589818
10000	125000	1000	125000	1.099589818
10000	125000	1000	250000	1.101079273
10000	125000	1000	625000	1.101451636
10000	125000	1000	875000	1.101824

10000	125000	1000	1250000	1.101824
10000	125000	1000	2500000	1.100334545
10000	125000	1000	3750000	1.100706909
20000	125000	1000	125000	1.099589818
20000	125000	2000	125000	2.106461091
20000	125000	3000	125000	3.082798545
20000	125000	5000	125000	5.096913455
20000	125000	7000	125000	7.111028364
20000	125000	10000	125000	10.10185309
20000	125000	20000	125000	20.11136
20000	125000	30000	125000	30.12086691
20000	125000	1000	125000	1.099217455
20000	250000	1000	125000	1.099589818
20000	625000	1000	125000	1.099589818
20000	875000	1000	125000	1.099589818
20000	1250000	1000	125000	1.099217455
20000	2500000	1000	125000	1.099589818
20000	3750000	1000	125000	1.099589818
20000	125000	1000	125000	1.099589818
20000	125000	1000	250000	1.101079273
20000	125000	1000	625000	1.101079273
20000	125000	1000	875000	1.102196364
20000	125000	1000	1250000	1.102196364
20000	125000	1000	2500000	1.101451636
20000	125000	1000	3750000	1.102196364
30000	125000	1000	125000	1.099589818
30000	125000	2000	125000	2.106461091
30000	125000	3000	125000	3.083170909
30000	125000	5000	125000	5.096913455
30000	125000	7000	125000	7.110656
30000	125000	10000	125000	10.10222545
30000	125000	20000	125000	20.11136
30000	125000	30000	125000	30.12049455
30000	125000	1000	125000	1.099589818
30000	250000	1000	125000	1.099589818
30000	625000	1000	125000	1.099217455
30000	875000	1000	125000	1.099217455
30000	1250000	1000	125000	1.095121455
30000	2500000	1000	125000	1.099589818
30000	3750000	1000	125000	1.099217455
30000	125000	1000	125000	1.099589818
30000	125000	1000	250000	1.101824
30000	125000	1000	625000	1.101079273
30000	125000	1000	875000	1.102568727
30000	125000	1000	1250000	1.102196364
30000	125000	1000	2500000	1.101824
30000	125000	1000	3750000	1.102196364
1000	125000	1000	125000	1.099217455
5000	125000	1000	125000	1.099589818
6000	125000	1000	125000	1.099589818
7000	125000	1000	125000	1.099589818
10000	125000	1000	125000	1.099589818
20000	125000	1000	125000	1.099589818
30000	125000	1000	125000	1.099589818
1000	125000	1000	125000	1.099589818

1000	250000	1000	125000	1.094376727
1000	625000	1000	125000	1.099589818
1000	875000	1000	125000	1.099217455
1000	1250000	1000	125000	1.099589818
1000	2500000	1000	125000	1.094004364
1000	3750000	1000	125000	1.099217455
1000	125000	1000	125000	1.099589818
1000	125000	1000	250000	1.101451636
1000	125000	1000	625000	1.101824
1000	125000	1000	875000	1.101824
1000	125000	1000	1250000	1.101079273
1000	125000	1000	2500000	1.102568727
1000	125000	1000	3750000	1.101451636
1000	125000	2000	125000	2.106461091
5000	125000	2000	125000	2.106461091
6000	125000	2000	125000	2.106461091
7000	125000	2000	125000	2.106461091
10000	125000	2000	125000	2.106461091
20000	125000	2000	125000	2.106461091
30000	125000	2000	125000	2.106461091
1000	125000	2000	125000	2.106461091
1000	250000	2000	125000	2.106461091
1000	625000	2000	125000	2.106461091
1000	875000	2000	125000	2.106461091
1000	1250000	2000	125000	2.106461091
1000	2500000	2000	125000	2.106461091
1000	3750000	2000	125000	2.106461091
1000	125000	2000	125000	2.106461091
1000	125000	2000	250000	2.192849455
1000	125000	2000	625000	2.208861091
1000	125000	2000	875000	2.193221818
1000	125000	2000	1250000	2.209233455
1000	125000	2000	2500000	2.206999273
1000	125000	2000	3750000	2.19136
1000	125000	3000	125000	3.082798545
5000	125000	3000	125000	3.083170909
6000	125000	3000	125000	3.083170909
7000	125000	3000	125000	3.083170909
10000	125000	3000	125000	3.083170909
20000	125000	3000	125000	3.083170909
30000	125000	3000	125000	3.083170909
1000	125000	3000	125000	3.082798545
1000	250000	3000	125000	3.082798545
1000	625000	3000	125000	3.082798545
1000	875000	3000	125000	3.082798545
1000	1250000	3000	125000	3.083170909
1000	2500000	3000	125000	3.083170909
1000	3750000	3000	125000	3.083170909
1000	125000	3000	125000	3.083170909
1000	125000	3000	250000	3.174027636
1000	125000	3000	625000	3.281640727
1000	125000	3000	875000	3.280896
1000	125000	3000	1250000	3.283130182
1000	125000	3000	2500000	3.281640727
1000	125000	3000	3750000	3.279778909

1000	125000	5000	125000	5.096913455
5000	125000	5000	125000	5.096913455
6000	125000	5000	125000	5.097285818
7000	125000	5000	125000	5.096913455
10000	125000	5000	125000	5.097285818
20000	125000	5000	125000	5.096913455
30000	125000	5000	125000	5.097285818
1000	125000	5000	125000	5.097285818
1000	250000	5000	125000	5.096913455
1000	625000	5000	125000	5.097285818
1000	875000	5000	125000	5.096913455
1000	1250000	5000	125000	5.096913455
1000	2500000	5000	125000	5.096913455
1000	3750000	5000	125000	5.097285818
1000	125000	5000	125000	5.096913455
1000	125000	5000	250000	5.187770182
1000	125000	5000	625000	5.459223273
1000	125000	5000	875000	5.637585455
1000	125000	5000	1250000	5.493480727
1000	125000	5000	2500000	5.489757091
1000	125000	5000	3750000	5.496087273
1000	125000	7000	125000	7.111400727
5000	125000	7000	125000	7.111028364
6000	125000	7000	125000	7.111028364
7000	125000	7000	125000	7.111400727
10000	125000	7000	125000	7.111028364
20000	125000	7000	125000	7.111028364
30000	125000	7000	125000	7.111400727
1000	125000	7000	125000	7.111028364
1000	250000	7000	125000	7.111028364
1000	625000	7000	125000	7.111400727
1000	875000	7000	125000	7.111400727
1000	1250000	7000	125000	7.111028364
1000	2500000	7000	125000	7.111400727
1000	3750000	7000	125000	7.111400727
1000	125000	7000	125000	7.111028364
1000	125000	7000	250000	7.201885091
1000	125000	7000	625000	7.472965818
1000	125000	7000	875000	7.652072727
1000	125000	7000	1250000	7.701597091
1000	125000	7000	2500000	7.672180364
1000	125000	7000	3750000	7.706437818
1000	125000	10000	125000	10.10185309
5000	125000	10000	125000	10.10185309
6000	125000	10000	125000	10.10148073
7000	125000	10000	125000	10.10185309
10000	125000	10000	125000	10.10185309
20000	125000	10000	125000	10.10148073
30000	125000	10000	125000	10.10185309
1000	125000	10000	125000	10.10185309
1000	250000	10000	125000	10.10185309
1000	625000	10000	125000	10.10185309
1000	875000	10000	125000	10.10185309
1000	1250000	10000	125000	10.10185309
1000	2500000	10000	125000	10.10185309

1000	3750000	10000	125000	10.10185309
1000	125000	10000	125000	10.10185309
1000	125000	10000	250000	10.19270982
1000	125000	10000	625000	10.46379055
1000	125000	10000	875000	10.64252509
1000	125000	10000	1250000	10.91658473
1000	125000	10000	2500000	11.08005236
1000	125000	10000	3750000	10.99850473
1000	125000	20000	125000	20.11098764
5000	125000	20000	125000	20.11098764
6000	125000	20000	125000	20.11098764
7000	125000	20000	125000	20.11136
10000	125000	20000	125000	20.11136
20000	125000	20000	125000	20.11098764
30000	125000	20000	125000	20.11098764
1000	125000	20000	125000	20.11136
1000	250000	20000	125000	20.11136
1000	625000	20000	125000	20.11098764
1000	875000	20000	125000	20.11136
1000	1250000	20000	125000	20.11098764
1000	2500000	20000	125000	20.11061527
1000	3750000	20000	125000	20.11136
1000	125000	20000	125000	20.11136
1000	125000	20000	250000	20.20221673
1000	125000	20000	625000	20.47329745
1000	125000	20000	875000	20.652032
1000	125000	20000	1250000	20.92609164
1000	125000	20000	2500000	21.83168
1000	125000	20000	3750000	21.98434909
1000	125000	30000	125000	30.12049455
5000	125000	30000	125000	30.12049455
6000	125000	30000	125000	30.12049455
7000	125000	30000	125000	30.12049455
10000	125000	30000	125000	30.12049455
20000	125000	30000	125000	30.12049455
30000	125000	30000	125000	30.12049455
1000	125000	30000	125000	30.12086691
1000	250000	30000	125000	30.12086691
1000	625000	30000	125000	30.12086691
1000	875000	30000	125000	30.12012218
1000	1250000	30000	125000	30.12086691
1000	2500000	30000	125000	30.12049455
1000	3750000	30000	125000	30.12086691
1000	125000	30000	125000	30.12049455
1000	125000	30000	250000	30.21135127
1000	125000	30000	625000	30.482432
1000	125000	30000	875000	30.66116655
1000	125000	30000	1250000	30.93559855
1000	125000	30000	2500000	31.84081455
1000	125000	30000	3750000	32.74640291
1000	125000	1000	125000	1.068311273