Automatic configuration of QoS parameters in IP RAN

Henrik Hjalmarsson

2009-11-06
Automatic configuration of QoS parameters in IP RAN

Examensarbete utfört i Kommunikations- och Transportsystem vid Tekniska Högskolan vid Linköpings universitet

Henrik Hjalmarsson

Examinator Di Yuan

Norrköping 2009-11-06
Uphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extras-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativt art.

Uphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Henrik Hjalmarsson
Abstract
This thesis discusses methods for automating the configuration process of an IP network. Focus lies on automating the configuration of Quality of Service parameters in an Ericsson IP RAN, a network that poses some constraints to the work, such as containing a diverse set of nodes and users with extremely high demands on service quality compared to other IP networks. It is possible to automate almost any task during IP network configuration, provided that there is a general system for transmitting data to every node in the network. Currently there is no such general system at Ericsson. The system created in this thesis is a prototype that shows how an application could be created and how it will have to interact with the communication part of an existing system to provide functionality for automation.
# Contents

1 Introduction ......................................................................................................................... 1
   1.1 Background .................................................................................................................. 1
   1.2 Purpose ....................................................................................................................... 1
   1.3 Scope .......................................................................................................................... 1
   1.4 Objectives ................................................................................................................... 2
   1.5 Method ......................................................................................................................... 2
   1.6 Outline ......................................................................................................................... 2
   1.7 Ericsson corporate information .................................................................................... 2

2 Quality of Service.................................................................................................................. 3
   2.1 Quality of Service metrics ........................................................................................... 3
   2.2 Quality of Service mechanisms .................................................................................... 5
   2.3 Achieving Quality of Service ....................................................................................... 8

3 IP RAN ................................................................................................................................. 13
   3.1 RNC/BSC site ............................................................................................................ 15
   3.2 RBS site ....................................................................................................................... 15
   3.3 Transport network nodes ............................................................................................ 15

4 Network management ......................................................................................................... 17
   4.1 Standards for network management .......................................................................... 17
   4.2 OSS-RC and AIPCM .................................................................................................. 22

5 Automation of IP network nodes ...................................................................................... 24
   5.1 Drawbacks of manual configuration ........................................................................... 24
   5.2 Purpose of automatic configuration ............................................................................ 24
   5.3 Drawbacks of automatic configuration ...................................................................... 29
   5.4 Requirements for an automatic configuration system .................................................. 29
   5.5 Network view ............................................................................................................ 30
   5.6 Network state ............................................................................................................ 31

6 Automation of QoS parameter configuration ................................................................... 33
   6.1 Application properties ............................................................................................... 33
   6.2 Consistency check ....................................................................................................... 35
   6.3 Use cases .................................................................................................................... 36

7 System design ...................................................................................................................... 39
   7.1 Communication between devices in the RAN ............................................................. 42
   7.2 Storage of network information ................................................................................ 42
   7.3 Design specification ................................................................................................... 43
   7.4 Extensions to AIPCM ................................................................................................ 51

8 Discussion ............................................................................................................................ 53

9 Conclusion ........................................................................................................................... 54

10 Future work ......................................................................................................................... 55

References ............................................................................................................................... 56

Appendix A - Abbreviations ................................................................................................. 58
Automatic configuration of QoS parameters in IP RAN

Henrik Hjalmarsson

Figures
Figure 2-1 - The IPv4 header as of RFC 791 ................................................................. 10
Figure 2-2 - The IPv6 Header ......................................................................................... 11
Figure 2-3 - MAC layer frame ...................................................................................... 12
Figure 2-4 - Ethernet and VLAN headers ..................................................................... 12
Figure 3-1 - IP RAN and mobile backhaul coexistence ............................................. 14
Figure 3-2 - RBS and RNC/BSC sites ......................................................................... 14
Figure 5-1 - Connected networks in an IP RAN ......................................................... 26
Figure 6-1 - Structure of the configuration process .................................................. 38
Figure 7-1 - Simplified structure of the configuration process .................................. 40
Figure 7-2 - Configuration update of a SIU ................................................................. 42
Figure 7-3 - User input, IP addresses ......................................................................... 43
Figure 7-4 - Map DSCP and P-bit values to queues .................................................. 44
Figure 7-5 - Map DSCP to P-bit values ...................................................................... 45
Figure 7-6 - User input, scheduling principles ............................................................. 46
Figure 7-7 - Confirm configuration .............................................................................. 47
Figure 7-8 – All steps of the automation process ....................................................... 48
Figure 7-9 - Processing of DSCP values, detailed ..................................................... 51

Tables
Table 2-1 - The AF PHB classes and levels ................................................................. 9
Table 2-2 - The history of the TOS/DSCP field .......................................................... 10
Table 4-1 - SNMP messages ....................................................................................... 18
Table 4-2 - Netconf messages ..................................................................................... 19
Table 6-1 - Traffic classes, DSCP values and suggested applications ..................... 34
1 Introduction
This introduction gives an overview of how the thesis is structures. Background information as well as limitations are explained here.

1.1 Background
Over the last few years there has been an increased interest in accessing Internet based applications from mobile units in cellular networks. When more users are sending and transmitting bulk data over a voice network there are possible larger gains if the data traffic, together with voice traffic, only has to pass through a network adapted for data traffic. If the Internet protocol, IP, is used as data carrier from the base stations to the base station controller site, BSC, or radio network controller site, RNC, it is possible to send traffic from separate mobile networks, 2G, 3G and LTE, through one single network instead of using separate links for each technology. However, there are problems when real time sensitive traffic and bulk data traffic must share a network that is not built to give any assured service quality to any traffic flow. By using existing methods for how to forward traffic it is possible to give the network the properties necessary for most traffic types and thus make sure that all applications can be used as intended, a process that can be considered as providing Quality of Service, QoS, to a network. All nodes in a network affect Quality of Service and it is therefore important to make sure that Quality of Service rules are applied to the entire network and in the correct way. By automating the process of configuring network nodes it will be possible to set up the network faster without losing control of the process.

1.2 Purpose
The purpose of this master’s thesis is to evaluate whether or not it is possible to configure Quality of Service parameters at a series of nodes in an IP RAN without the need for the user to access each router individually. Since Quality of Service parameters are discussed there is also a study on how it should be able to combine configurations on different nodes, focus is mainly on the automation process of configuration, not on how performance is affected by each possible configuration. The study ends with how to create a tool for automatically configure a series of nodes in an IP RAN.

1.3 Scope
Since different routers are configured using different syntax the application has to focus on a few products, otherwise the thesis would expand too much. Ericsson manufactures a few products themselves that perform routing in the IP radio access network, IP RAN. These products are SmartEdge, Marconi OMS1410, MINI-LINK and the site integration unit, SIU. In their services they also use Summit routers from Extreme Networks. This thesis will not discuss any other products, and the products mentioned above are not the main focus of interest, i.e. their exact properties will not be discussed in great detail. Some of their properties are important for the work but most of them are irrelevant for the outcome of this work. The time limit on the work is a limitation, which makes the development of any non-existing Quality of Service principle difficult to perform. All QoS-parameters have to be supported by the routers and their operating system, which makes the development of such principles very difficult to implement.
1.4 Objectives

The goals set up for this thesis is to create a system that can be used by Ericsson or any of its customers to automatically configure a set of parameters regarding QoS on a set of network nodes and to provide an overview of techniques possible to achieve automation of various levels. The goals can be summarized as follows:

- Gather information of how automatic router configuration can be performed
- Set up rules that nodes must follow to provide consistent QoS rules
- Set up rules for how an application must be set up in order to meet the requirements

1.5 Method

A literature study is carried out to evaluate the possibilities and limitations for automating the configuration process in an IP-based network. The findings of this literature study are a base for how to set up constraints and boundaries for how an application should be developed. Together with the limitations that are posed on the work and the possibilities that exist internally at Ericsson, a prototype is created with the purpose to configure a set of network nodes automatically.

1.6 Outline

In chapter 2 the concept of Quality of Service is explained together with some of the metrics used to calculate Quality of Service and mechanisms that make it possible to affect these metrics. Chapter 3 includes an introduction to an IP RAN; this part also contains information about the different products used in the Ericsson IP RAN solution. Some protocols and procedures for controlling IP nodes and how these can be used for automation are explained in chapter 0. A complete view of automating IP node configuration is explained in chapter 5 and continues with a discussion on how to create a system for automating QoS parameters in an IP RAN in chapter 6, especially how the application would be used is explained here. Chapter 7 explains details of how the application was implemented and the work is discussed and concluded in chapters 8 and 9.

1.7 Ericsson corporate information

Telefonaktiebolaget LM Ericsson or Ericsson AB is one of the largest companies in the telecommunication industry. Their point of interest is towards mobile communications where they are the market leader in several segments. They manufacture products for communication via several technologies such as GSM, WCDMA and LTE, i.e. 2G, 3G and “Beyond 3G”. Ericsson equipment is used in approximately 175 countries worldwide and about 40 percent of all mobile calls are made through Ericsson’s systems [1]. As of December 31 2008 Ericsson had 78,750 employees [2].
2 Quality of Service

“Quality of Service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow.” [3] It can also be seen as “a set of communication characteristics that is required by an application. QoS defines a specific transmission priority, level of route reliability, and security level”. [4] According to Cisco [5], “Quality of Service refers to the capability of a network to provide better service to selected network traffic”, and further, and very important, to make sure that “providing priority for one or more flows does not make other flows fail”. These definitions should hopefully help to give a good view of what Quality of Service is all about, to provide a good enough user experience at all times, which is especially important when congestion increases and packets are lost in the network.

2.1 Quality of Service metrics

The Internet Engineering Task Force, IETF, standardized the Internet Protocol, IP, in the early 1980’s and it offers a best effort node-to-node transport service. The intention with IP was to provide a protocol that made sure that all traffic was delivered error free but without the demands on delivery that exist in a public switched telephone network, PSTN. In PSTN, as well as in other circuit-switched networks, resources are reserved for every connection, which is given a certain guaranteed bandwidth, a method that is very practical for telephone networks. IP however, was developed for data traffic that is bursty by nature and sometimes need to send a lot of data and sometimes no data at all. The only Quality of Service included in IP is its ability to detect bit errors, thus avoiding transmitting corrupt frames.

In order to provide a useful service in an IP network there are some performance metrics that have to be taken account for. These metrics are not possible to change in a controlled manor but are merely a result of the network characteristics. However, some metrics depend heavily on a certain network characteristic and management and thus can be controlled to some extent whereas others are depending on current network traffic. All metrics can be improved with network over dimensioning but this method is very expensive and should therefore be avoided. The most commonly used performance metrics are presented in the following sections.

2.1.1 Delay

In all communication networks it takes a certain amount of time for information to pass from the point of origin to the point of destination. This time is referred to as delay and varies a lot due to several factors of the network. Delay is measured as either round-trip delay or as one-way delay, where round-trip delay is about twice the size of one-way delay for a given path. The reason for having two metrics describing the same thing is that certain applications have certain characteristics, which makes measuring either one-way or two-way a more realistic approach. One-way delay is generally used for real-time applications while round-trip delay is used for non real-time applications. [6]

Delay is introduced in different ways in the entire network and it can be divided in 5 different types of delay; different types that have their specific characteristics that affect the quality of the
network in different ways. These different types of delay are propagation delay, transmission delay, switching delay and scheduling delay.

Propagation delay is the time it takes for a series of binary zeros and ones to travel from one end of a link to the other end. This delay depends on the length of the link and its physical properties; for example does an optical fiber has the ability to transport data faster than a twisted copper wire. [6]

Transmission delay is the time it takes to convert a series of bits into an electromagnetic signal that can be transmitted on the physical media. This delay varies very much between home equipment, i.e. a modem that can transmit 56 Kbps, and high-end ISP equipment, i.e. a backbone router that can transmit several Gbps. Transmission delay is similar to store-and-forward delay, which is the time it takes for a stream of bits to arrive at a node before it is checked for errors. [7]

When a packet arrives at a network router it is subject to two types of delay, switching (or processing) delay and scheduling (or queuing) delay. Switching delay is the time between arrival on a router interface and the enqueuing of the packet in one of the output queues, i.e. the time it takes for a packet to traverse through the switching fabric of the router. Depending on the size of the queue in which the packet placed, and the amount of packets in any other queues that has a higher priority due to scheduling principles of the router, it will take some time before the packet is sent out on the link from the router, which is the scheduling delay. Different scheduling principles will affect packets differently; this means that the scheduling delay can vary very much between packets in a router. [6]

For some applications, such as telephony or interactive services, it is important that delay is kept at a low level since large delays make the conversation hard to follow. With delays lower than 150 ms the user satisfaction is high, but when delay exceeds 400 ms it gets very difficult for users to retain a normal conversation [6].

2.1.2 Jitter

Packets in a data stream do not always travel the same path between their point of origin to their point of destination. When different links and nodes are used the packets in the stream will be exposed to different conditions that affect the time it takes for traveling to the end node. Even if two packets travel the exact same path it is unlikely that the delay for these two packets would be the same since routing delay is varying, thus delaying packets in a rather arbitrary manor. This difference in delay is called jitter and it is a very important Quality of Service metric, especially for real time applications. When jitter is high, i.e. packet inter arrival time is varying considerably, some packets might arrive too late to be played out at the receiver. If jitter is too high, playback at the receiver might have to be delayed in order to keep the application from discarding too many packets. If playback is delayed too much, user perceived quality is decreased according to section 2.1.1. [6]

2.1.3 Packet loss

Packet loss can occur everywhere in a network and for various reasons. The major reasons that prevent packets from reaching their destination are bit errors that cannot be resolved or packet
drops in routers and end nodes. Also, if there is an error in an underlying layer, packets will not be delivered to the end node.

Bit errors are caused by noise, which exists in all physical media; some media however are more resilient to noise thus generating less bit errors than other media. The bit error rate, BER, is extremely low in optical fibers, whereas in wireless networks BER can be a big problem. There are techniques to avoid dropping packets due to single bit errors; some of these are cyclic redundancy checks, CRC, and parity checks. These techniques can detect and in some cases even correct single bit errors. However, if neither CRC nor parity check can resolve the bit error, the packet is discarded.

The third major reason for packet loss is when packets are discarded in a router or an end node due to congestion or that packets arrive too late to be used. When the network experiences congestion, buffers in the network routers overflows and packets need to be dropped. [6]

2.1.4 Throughput

Throughput is a measurement that describes the amount of data possible to transmit on a path in a few various ways. From an end user perspective, throughput is the amount of actual user data sent or received during a specified time frame, expressed in bits per second. This measurement is also known as bulk transport capacity and it describes how much data that can be transmitted over a link long-term. Another metric for throughput is link capacity, which describes the transmission capacity on the data-link layer. This metric is constant for any specific link and depends on the physical media. On a path consisting of several links, the capacity of the link with the lowest capacity is the upper theoretical limit of throughput on the path. Due to noise and packet loss this upper limit of capacity might not be reached. [6]

Throughput for one user using any type of service is also affected by network congestion. During peak-load hours several flows will fill the network, thus decreasing throughput for all flows passing through the congested links. When congestion is present in a network, methods to achieve Quality of Service become more important than during hours with low traffic since several flows compete for scarce resources. When there is little traffic in the network all flows will be delivered with little delay. [6]

2.2 Quality of Service mechanisms

In a best effort network there are several techniques available for giving different traffic flows as much as possible of the bandwidth available. These techniques vary from separating traffic based on its properties, origin or destination to setting up logical links that provides a guaranteed fixed, or minimum, bandwidth, large enough for the application. As these mechanisms are discussed they are considered as the only mechanism present in any network, it is possible to combine several of the mechanisms, how that can be done is not discussed.

2.2.1 Marking

When marking a traffic flow, any of the headers of each packet in the flow can be used. The TCP/IP protocol suite supports two main mechanisms for marking packets; using the DSCP field of the IP header, using the p-bit field of the VLAN tag of the 802.1 layer-2 header. These techniques are described further in sections 2.3.2 - DiffServ. If an end node is able to mark
packets itself, it may do so according to rules that states how a certain packet should be marked, if the end system itself cannot mark a packet the first trusted node will do so according to its rules, for example that all incoming traffic to an interface or port is marked with a specific value.

### 2.2.2 Classification

Classification of packets can be used to separate different traffic flows containing different types of application data from each other in order to provide different service levels suitable for different applications. The classification can be based on various properties such as the protocol used for transmission, the IP address of the destination or any other property that is easy to get from the data by any node in the network. Depending on how classification is done, the Quality of Service received can be affected and must thus be done with much consideration. The most common technique for classifying packets is to use the differentiated services codepoints of the IP header, which is used in DiffServ and described further in section 2.3.2.

### 2.2.3 Reservation

If all entities in a network, through which a flow passes on its way from one user to another, can spare the bandwidth, resources, needed for a certain flow, the users can expect this flow to meet a certain quality level, corresponding to the amount of resources reserved. When resources in a network are reserved the network gets attributes similar to those in the PSTN where a certain bandwidth is reserved during each call. The main advantage with reservation is that when resources are reserved the network can guarantee a level of QoS that is sufficient for the application that requested the resources. This approach has some disadvantages, when resources are to be reserved throughout an entire network the course of action must be synchronized to avoid that the sending node starts to transmit data before all nodes involved have reserved all resources necessary for the transmission.

Integrated services, which is based on bandwidth reservation to give all flows an environment where there is no risk of packet losses due to congestion, uses reservation, and especially the resource reservation protocol, RSVP, to ensure a good Quality of Service. RSVP can also be considered as being used for admission control that is discussed in the next section. Integrated services is explained further in section 2.3.1.

### 2.2.4 Admission control

Admission control is a group of techniques that can be used to prevent new flows from being set up if there is a risk that the new flow may degrade the service for flows already existing in the network. In a network that is provisioned to cope with a certain amount of traffic, i.e. a network that is not over provisioned, there is, during peak hours, a large probability that any new flow may cause routers to drop packets due to congestion. In this case, a system that prevents new traffic flows on congested links will make sure that no other flows are being degraded. The approach between the different admission control methods varies and every network provider must decide which method suites their needs best. [6]

For elastic flows such as transporting bulk data, such as large files, using the transmission control protocol, TCP, where a minor change in available bandwidth does not affect service very much, admission control is not very important. For inelastic traffic, such as VoIP, a minor decrease in
available bandwidth can eliminate the quality of the application entirely, thus creating a need for some form of admission control. [6]

There are two main types of admission control, off-path and on-path control. With off-path admission control there must be a central node present that can calculate the available bandwidth in the network for a certain flow, the approach here can be to either keep track of the current amount of traffic in the network or to have a predefined setup of bandwidth available, e.g. there is a server that is aware of how much bandwidth is available for each link in the network. These approaches require either lot of calculations by nodes that are not taking part of the data exchange or it leaves bandwidth unutilized due to its inflexibility. [6]

The most used part of the latter method, on-path admission control, is the resource reservation protocol, RSVP, which is used in the integrated services model. IntServ is explained further in section 2.3.1.

2.2.5 Policing
Policing is a mechanism that can be implemented in any node in a network to make sure that any incoming flow does not exceed any predefined bit rate. Using a so called token bucket it is possible to calculate whether or not a flow exceeds a certain limit that conforms to the rate at which the bucket is filled with tokens and the amount of tokens in the bucket. The token is continuously filled with tokens at a rate R, which is the maximum average rate of the flow, and can contain B tokens, which is the maximum burst size of the flow. When a packet arrives at the node, the number of bytes in the packet is compared to the number of tokens in the bucket; if there are enough tokens left the packet is forwarded and tokens, one token for every byte, are removed from the bucket. If there are not enough tokens in the bucket, the flow exceeds its limits and is discarded or marked differently than other packets of the flow. What actually happens in practice is a question for the network administrator and not further discussed here. [6]

There are variations of this simple token bucket that are more complex and provide a more fine-grained policing. More information about such methods can be found in [6].

2.2.6 Dimensioning
Dimensioning is a means for providing enough bandwidth to support the different applications that should be served by the network. It is in most cases possible to avoid large delays and high jitter by over dimensioning a network, if there is no risk of congestion in a network all flows will always reach their destination without suffering from any noticeable amount of delays. This approach is however difficult in practice, such a network will be very large, very expensive to build and contain a large amount of redundant nodes. There are several other approaches to dimensioning, but only over dimensioning may solve issues regarding Quality of Service if dimensioning is used as the only method for achieving Quality of Service. If there is almost infinite capacity in the network, other approaches may improve the ability to cope with Quality of Service matters, it will however not solve these issues.

Dimensioning may not be considered as a mechanism for achieving a good Quality of Service the same way as other mechanisms presented. However, it is an important task during network roll-out to ensure that the network is suited for the amount of data expected without being to expensive to deploy.
2.3 Achieving Quality of Service

In order for a network to support Quality of Service there are two major models, standardized by the IETF, that can be used, integrated services and differentiated services. The integrated and differentiated services models are discussed in this section. Focus however, will be on differentiated services since this is the model used in the Ericsson IP RAN reference model [8]. In both models, flows are considered being given their priority by any node that is trusted to do so by the network, the mechanism for how this is done is of minor interest in this thesis. In a cellular network the RBS is responsible for this task.

2.3.1 Integrated services

The integrated services model, IntServ, is a model in which resources are allocated to users when they need them. The sender of data requests a certain level of Quality of Service from the network which reflects the type of data being sent, i.e. what application is being used and what characteristics this certain application has. The request is transmitted to all nodes in the network on the path between the sender and the receiver which in turn allocates the resources needed. If some node cannot reserve enough resources, it has to announce this to the sending host. IntServ was originally defined in RFC1633, which describes the architecture of the model and the basic mechanisms that are necessary to provide Quality of Service throughout an entire network. [9] IntServ was later extended with a protocol that defines how to set up resources in the network; this protocol is called Resource Reservation Protocol, RSVP. When using RSVP, the receiving host sends an RSVP reservation request to the sending host via all nodes on the path that the data packets will traverse. Each node on the path enters the reservation state, which is kept until they receive an RSVP path message from the sending host. If every node on the path is able to provide the requested QoS, data transmission can be started. [10].

The advantage with reserved resources is that all flows are given the appropriate Quality of Service needed for their respective application throughout the entire data transmission session. There is a problem when resources are scarce, at that point new sessions will not be given the appropriate QoS, and thus the service must be degraded. The reason for this is that existing RSVP connections cannot be torn down by any other entity in the network than the receiving or the sending node themselves. The second, and most important, disadvantage is that IntServ is poorly scalable; there has to be a lot of extra data sent in order to keep track of all flows. RSVP does not transmit any data itself but is merely a signaling protocol on top of IP, similar to ICMP and IGMP for example. [10]

2.3.2 Differentiated services

The other major approach to provide QoS in a large network is to use differentiated services, DiffServ. This model derives from the use of the type of service, TOS, field in the IPv4 header. This part of the IP header has been subject to updates and changes since the first introduction of the Internet Protocol in RFC 791[11]. Ever since this introduction there have been a few updates of the IPv4 header, as shown in table Table 2-2, of which the two latest are slightly different from their predecessors. The first of these two versions was introduced together with the concept of differentiated services, and it is in these two versions that there is a differentiated services field in the IP header. The definition of DiffServ can be found in RFC’s 2474 and 2475[12, 13], where the architecture of the DiffServ model and the DS field of the IP header are discussed.
DiffServ takes another approach to providing QoS than IntServ, there are no negotiations between hosts or calculations of network usage prior to setup of new flows. Instead, each host starts transmitting data, which is given a certain priority; packets with higher priority are to be forwarded before packets with lower priority. This gives data flows with high priority less store-and-forward delays than flows with lower priority, and thus less total delay. This approach does not provide any guaranteed Quality of Service but is used to prioritize flows over each other. If there is congestion in the network this will lead to decreased service quality for some flows, such as web browsing, whereas delay and jitter sensitive applications, such as VoIP, do not suffer as much since these packets are not dropped due to the congestion.

DiffServ only uses information that is possible to extract from the IPv4 header that makes it suitable to use in an environment where entities are unaware of properties in remote locations of the network. This mechanism of the architecture also makes DiffServ highly scalable.

### 2.3.2.1 Codepoints and Per-Hop-Behavior

The DS field of the IP header is used to extract the differentiated services codepoint, DSCP which is mapped to a certain Per-Hop-Behavior, PHB. Each PHB is given a specific priority in the outgoing interface of any DiffServ compliant router in the network. To do this mapping the first six bits of the DS field octet are used, the last two bits of the fields are currently unused. All PHB’s may be used whereas every DS field value must correspond to exactly one PHB. In RFC 4594, “Configuration Guidelines for Differentiated Services”, all these PHB’s are presented; some of the PHB’s are discussed further in separate RFC’s [14, 15]. The different PHB’s are “default forwarding”, DF, “assured forwarding”, AF, “expedited forwarding”, EF and “class selector”, CS. These PHB’s are used for different traffic classes and should be treated accordingly. Packets associated to the EF PHB group have a generally high priority whereas packets belonging to the CS and AF classes could vary in priority from extremely high to very low. The AF PHB group has four internal classes with three loss probability levels in each class resulting in a total of twelve differently prioritized levels of traffic. In table Table 2-1 the different general AF PHB classes and levels are displayed, ranging from high priority – low loss probability in the upper left corner to low priority – high loss probability in the lower right corner.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF11</td>
<td>AF12</td>
</tr>
<tr>
<td>AF21</td>
<td>AF22</td>
</tr>
<tr>
<td>AF31</td>
<td>AF32</td>
</tr>
<tr>
<td>AF41</td>
<td>AF42</td>
</tr>
</tbody>
</table>

There is a final PHB in DiffServ, one that is used to provide backward compatibility with older versions of the IP header. When differentiated services was introduced the IP header was changed and to keep compatibility with the older version of the header, the precedence bits where given an own per hop behavior, called class selector, or CS for short. Table Table 2-2 in the next section shows how the type of service field of the IP header has been changed. When RFC 2474 became a full standard the CS PHB was introduced to maintain compatibility.
2.3.2.2 DiffServ in IPv4

The techniques and principles for DiffServ described above are valid for DiffServ together with IPv4. The TOS field was present in the first RFC that discussed IP but it has been subject to changes in later RFC’s. Table 2-2 shows how the eight bits of the TOS field has been changed and in what RFC the change was done. In RFC 791, where the Internet protocol was introduced, the TOS field consisted of a three-bit Precedence field that states the type of traffic being sent and three bits to declare the type of service. This type of service marks the needs for the data sent according to one of the three parameters delay, throughput and reliability. [11]

The precedence field was kept unchanged in the first two updates where the TOS field changed to five and four bits respectively, the MBZ-field in RFC 1349 is the “Must-be-zero” field, hence this bit is always set to zero. [16, 17]

As already discussed, the TOS field was changed to the DSCP field, which is currently used. The latest change in the DS field was done in RFC 3168 that introduces explicit congestion notification, this change has minor impact on DiffServ and QoS, thus it will not be discussed further. [18]

Table 2-2 - The history of the TOS/DSCP field

<table>
<thead>
<tr>
<th>RFC</th>
<th>Bits</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>791</td>
<td>Precedence</td>
<td>TOS</td>
<td>Res</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1122</td>
<td>Precedence</td>
<td>TOS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1349</td>
<td>Precedence</td>
<td>TOS</td>
<td>MBZ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2474</td>
<td>DSCP</td>
<td>CU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3168</td>
<td>DSCP</td>
<td>ECN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How the IPv4 header has been changed since the introduction in RFC 791 is outside the scope of this thesis, Figure 2-1 shows the header as it looked when it was introduced. In theory it would be possible to separate traffic based on its source and destination address as well as the TOS/DSCP field, this approach would require large tables at every node in which the priority for each pair of source and destination address is explicitly stated. This means that this approach is very process demanding and thus very difficult in practice.

Figure 2-1 - The IPv4 header as of RFC 791

2.3.2.3 DiffServ in IPv6

Version six of the Internet protocol, IPv6, is a non-interoperable extension of IPv4 that has some major differences compared to IPv4 apart from the address space, which has little or no impact on Quality of Service and forwarding. The two fields in the header that might affect routing
efficiency is the traffic class and the flow label. The traffic class is intended to be used similar to the DS field of the IPv4 header, thus similar properties are to expect when using DiffServ with IPv6. This requires that traffic classification must be done in a similar fashion as when using DiffServ with IPv4. It does not mean that a certain traffic class must have the same bit representation as in IPv4, only that traffic classes must be designed similar to the traffic classes that already exists when using IPv4. Figure 2-2 shows the IPv6 header as it is specified in RFC 2460, it shows the eight bit traffic class field and the intention is to support the same functionality as the DS field of the IPv4 header. The intended use of the flow label is to give a sequence of packets from a source to its destination a value in the header that can be used by nodes on the path to take a certain action. It is not yet completely decided how the flow label is to be used, especially regarding its use with Quality of Service. [19]

![Figure 2-2 - The IPv6 Header](image)

### 2.3.2.4 DiffServ in layer 2 networks

DiffServ can be used in a layer-2 transport network, such as a RAN, using the possibilities of the Ethernet frames. A standard Ethernet frame itself does not contain any fields to be used for differentiating traffic, but this frame can be extended to provide differentiating capabilities. A standard Ethernet frame is a frame that follows the IEEE 802.1D standard [20].

Virtual local area network is a standard defined in IEEE 802.1Q and it is an extension to bridged local area networks, defined in IEEE 802.1D. A bridged local area network creates a network that is identical to a LAN from the end user perspective even if the end systems are separated topologically, i.e. end systems can communicate with each other using only information on layer two of the OSI model, the MAC layer. A MAC datagram consists of the source and destination addresses, a protocol identifier and data from upper lying protocols, such as an IP packet. The entire datagram is shown in Figure 2-3. Compared to the header in Figure 2-4, which is slightly simplified the standard MAC datagram does not contain the VLAN tag. [21]
As the Ethernet frame itself has no ability to transfer information about any QoS related questions by default, there is a field in the VLAN tag that can be used to differentiate flows from each other. This field contains the priority bits, P-bits, also known as priority code points, PCP. The bits in this field can be used just like the DSCP field of the IP header. The problem is that as there are eight bits in the DSCP field, of which six are used for traffic classification, there are only three priority bits. This disparity means that for each DSCP value there must be one, and only one, P-bit value whereas one P-bit value can be mapped to several DSCP values. [22]

2.3.2.5 Mapping between DSCP and P-bits

In a network with several different routers and switches some nodes will only read layer-2 data while other nodes only read layer-3 data, i.e. some nodes read the DSCP value and some the P-bits to decide on how to process incoming packets. To solve this possible problem there must be a consistent mapping between DSCP and P-bit values. There is no official standard on how to map DSCP to P-bits and vice versa but this has to be done separately for each network, i.e. by each network provider. In the Ericsson IP RAN solution it is described how this mapping should be performed, this figure is found in table B-1 in appendix B.

2.3.2.6 Non-Differentiated Services-compliant routers

All nodes in a network may not be compliant with DiffServ, e.g. they are not able to read the IP header as it is stated in RFC 2474. This node might consider the IP header according to neither RFC 791, in which the TOS field is introduced, nor to RFC 1349, where the TOS field is updated. If such nodes are present in the network they should use the properties of the TOS field, which provides some compatibility to the DS field even if it is not possible to get the exact same behavior. If possible, these nodes should not be the bottleneck in the network, if they are the effect on QoS will be larger than otherwise necessary.
3 IP RAN

The radio access network, RAN, is the part of a mobile telecommunication network that connects the radio base stations, RBS, to its control node. The RBS is either a base transceiver station, BTS, or a Node B, depending on if it is a 2G or 3G radio access network. In a 2G network the control station is the base station controller, BSC, and in case of a 3G network, the radio network controller, RNC. The traditional view of a RAN is that incoming TDM traffic from the users are sent from the RBS to the BSC as TDM traffic, thus keeping the time slots for each user throughout the RAN. This is very efficient in a network where only voice traffic is served; all users will have their determined time slots throughout the entire network to the other end user. As a result of this technology all users will receive a good Quality of Service since resources are allocated to provide a guaranteed bandwidth during the entire session. This approach is identical to the approach used in PSTN where resources are allocated during a phone call. As long as there are no other services than voice calls in the network this approach provides good Quality of Service. However, when other services are introduced in the network, problems might occur, problems that arise when applications that are not based on time slots are used in a network where users are given time slots during which they can communicate. Examples of such applications are almost any application that uses any Internet service; such applications are based on the idea that bandwidth is allocated when needed with the result that the data rate achieved can vary from very low to very high. When web services and traditional voice service are sent through the same network there is a need to use only one type of network, either a circuit switched network suitable for voice traffic or a packet switched network that is suitable for services that occasionally need to transmit large amounts of data. In an IP RAN the approach is to use a packet switched network through which both voice and other services are transmitted. This gives a network with the same properties as a normal IP network for the operator, and the properties of a PSTN for the user. The RNC, BSC, BTS and Node B do not send any traffic using IP and thus these nodes are not a part of the IP RAN.

The IP RAN coexists with the mobile backhaul and it is difficult to draw a line where one ends and the other begins, the mobile backhaul can be seen as lying underneath the IP RAN, even if there may be nodes within the mobile backhaul that use layer-3 information for forwarding of data. However, this network is mainly a level 2 network. Figure 3-1 shows the coexistence of IP RAN and mobile backhaul, as seen here the mobile backhaul lies in between of the different parts of the radio access network.
Automatic configuration of QoS parameters in IP RAN

Henrik Hjalmarsson

Figure 3-1 - IP RAN and mobile backhaul coexistence

Figure 3-2 shows how the RNC/BSC and RBS sites can be configured and connected to each other through the transport network, which is the IP RAN. There are several different ways of setting up both the RNC/BSC site as well as the RBS site, these different approaches vary according to the demands of each specific site and the details of these approaches are not discussed here. In the remainder of the chapter follows a brief description of the most important nodes and sites of the IP RAN.

Figure 3-2 - RBS and RNC/BSC sites

To conclude the discussion about IP RAN it can be considered as a normal IP network, just like the Internet. However, it has a very hierarchical structure ranging from small radio base stations to large aggregation routers. In the next sections follows a brief presentation about the Ericsson products that are included in the IP RAN solution, and the mobile backhaul solution since these...
nodes affect traffic in the IP RAN. As seen in these sections, the environment of the IP RAN is very heterogeneous. Furthermore, each node uses its own management and operating system. However, they all are interoperable and can transport data end to end as long as configured properly. There are more nodes such as time servers and security gateways in the network, these nodes do not affect QoS as the other nodes of the network do, thus they are left out of the discussion.

### 3.1 RNC/BSC site

At the site where the control nodes are placed in the network, the RNC or the BSC, there is a need for a high-speed aggregation node that is able to forward large amounts of data to and from the RNC/BSC. The aggregation router usually consists of two routers instead of one due to resilience questions; it would be devastating for a network provider if a RNC/BSC is unable to connect to the network.

#### 3.1.1 Ericsson SmartEdge

The Ericsson SmartEdge is a high capacity router used as aggregation point of incoming flows at the RNC/BSC site in IP RAN. The SmartEdge has a capacity ranging from 10 to 10 Gbps per IP port and can also support STM-1 on its interfaces adding up to at total of 80 Gbps in the switching fabric. Since the SmartEdge consists of modules that are connected to each other it is possible to adapt the router to the needs of the site. Regarding Quality of Service matters the SmartEdge series offers full compliance with all mechanisms used in the DiffServ model. [23]

### 3.2 RBS site

The RBS can be either a GSM BTS or a UMTS Node B base station that connects users to the network. Multiple users connect to a single RBS, and transmit data, using either TDM or WCDMA depending on the technology used. Theses signals can not be sent immediately over an IP RAN but transformation must be done into IP packets prior to transmission from the RBS site. At the RBS site there is a need to convert the data sent from end users using GSM and WCDMA, in the future also LTE, into IP packets to send to the control nodes, for this purpose the site integration unit, SIU, can be used. The main functionality is to transform and aggregate incoming GSM, WCDMA, LTE and LAN traffic from the RBS into IP packets on one outgoing interface to the RAN. Thus it is connected to the RBS on the one side and the RAN on the other. It supports both E1/T1 for connections to the RBS as well as WAN interfaces from 10 Mbps up to 1000 Mbps on both electrical and optical fiber as physical media for the interface to the RAN. Further, the SIU supports Quality of Service settings according to DiffServ and VLAN, which both are necessary to work in an operational environment. [24]

### 3.3 Transport network nodes

The IP RAN consists of a large network that connects the RBS to the RNC/BSC site. The Ericsson IP RAN solution contains a set of different nodes that can be used for this purpose. There must be a versatile set of nodes in the IP RAN solution since a radio access network must be possible to build in any physical and geographical environment, thus there is support for transport through optical fiber, copper wires and wireless links, for example micro waves.
3.3.1 Marconi OMS 1410

Marconi is originally a British company that was purchased by Ericsson in 2005 [25]. They offer telecommunication equipment for a wide range of purposes. At the moment, only one of their products is a part of the IP RAN solution, the OMS 1410. The OMS 1410 is a node mainly for optical transport using Ethernet, SDH or TDM, and thus it can be used in traditional mobile networks as well as in IP networks. It has a capacity of up to 80 Gigabit Ethernet ports but can be scalable from 20 GE ports to support the needs of each individual site, further it can support a wide range of physical links on each port, ranging from E1 to 10 Gigabit Ethernet. [26]

3.3.2 MINI-LINK

The MINI-LINK is a microwave based node used in environments where it is difficult, or too expensive, to use wireline connections. It consists of a parabolic antenna connected to modules for wireline connections; the wireless part transmits data at frequencies of 6-38 GHz through distances up to 50 km. The MINI-LINK is usually deployed close to the cell site, thus its capacity is lower than the capacity for nodes at RNC/BSC sites. Still the possible throughput is quite high, up to 850 Mbps for each wireless antenna and 1 Gbps per wireline Ethernet connection.

The MINI-LINK connects to the IP RAN at the cell site and to mobile backhaul towards RNC/BSC site, i.e. it resides in the part of mobile backhaul that is sometimes referred to as low RAN or LRAN. [27]
4 Network management

Network management can be described as the activities to be performed in order for a network to fulfill the duties posed on the network based on the services that the network should be able to offer. It can further be described with FCAPS, an acronym that stands for “Fault, Configuration, Accounting, Performance, and Security” [28]. This thesis will not consider the security and accounting matters of network management and the configuration aspects will be discussed deeper than fault and performance issues, which are of some interest.

Issues concerning fault, configuration and performance are quite tightly bound together since any changes made to the configuration task may impose an unforeseen misconfiguration that affects both performance and fault. During configuration of network nodes a logic or programmatic fault might be introduced that makes it possible for configuration errors to occur. This is especially common if non-standardized CLI scripts are used. This section discusses some of the standardized ways of communicating with nodes without manually using a CLI. There are also methods whose framework is standardized but where implementations can vary from one system to another. In this discussion, there is some focus on IP RAN, but the methods are valid for any IP network.

Configuration of network nodes is an important task that must be performed when a network is extended, a new service is introduced or whenever there is a need to change the current configuration due to customer demands. Independent on the reason for a network configuration, the actual configuration will not be done the same; this because of the many commands that exists in the various operating systems. All the existing operating systems combined with the total amount of commands available makes it probable that errors are introduced in any of the nodes. The Internet community has had discussions on how to ease the task of network configuration, which has resulted in several standardized methods, SNMP, NETCONF, CIM and QPIM, which are described further in sections 4.1.1 to 4.1.4.

4.1 Standards for network management

There are several standards that are created either solely for managing network entities or that can be used for this purpose as well, even if it was not created with this specific task in mind. These protocols are presented in the sub sections of this chapter.

4.1.1 SNMP

The first attempt to create a method for network management ended up in the creation of the simple network management protocol, SNMP, which was first described in RFC 1098. The protocol resides at the application level which makes it able to collect data from and send data to nodes of all vendors whose products are SNMP compliant. As the name suggests SNMP is a simple protocol that is only able to send eight different types of messages from the SNMP manager to the SNMP agent of which three are used to request data, one to respond accordingly to a request, one to set a variable at the agent side and two message types to exchange information between SNMP managers. The last message is called the TRAP message and is used to send data from an agent to a manager if some predefined event occurs that calls for administrator intervention, or as information to the administrator. [29, 30]
Since there are only two types of messages that the SNMP manager can use to interact with the agent, either the “SetRequest” message or any of the “GetRequest” messages, interaction is simple without decreasing the type and amount of data that is possible to collect. All SNMP messages are shown in Table 4-1. The structure of SNMP is simple and in order to be able to retrieve the data wanted from a node one must know the data structure that is used within each node. SNMP itself does not contain any information about this data structure thus two separate protocols must be used, the management information base, MIB, and a structure of management information, SMI. SMI is used to set the structure of information, that is: information on how to name objects and to define the type of data, e.g. strings or integers that objects can store. It is also responsible for how encoding should be done. MIB on the other hand contains information about each node and exists at each agent. In order to collect MIB variables the manager site must do this remotely. The MIB is a subset of ASN.1, defined by OSI, and it is the responsibility of SMI to define that subset. [29, 30]

SNMP exists in three different versions, whose functions are similar to each other. Currently SNMP version 3 is a full standard, which means that the previous versions have become obsolete. SNMPv3 is described RFC 3411 – 3418 in which it is stated that SNMPv3 should be based as much as possible on already existing technologies, e.g. previous version of SNMP, and that it should be kept as simple as possible [31].

The main advantage with SNMP is that it is vendor independent which makes it useful in a network with routers from multiple vendors. All routers can be configured without information about the vendor displayed to the operator.

Table 4-1 - SNMP messages

<table>
<thead>
<tr>
<th>Message Type</th>
<th>From</th>
<th>To</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRequest</td>
<td>Manager</td>
<td>Agent</td>
<td>Get the value of a specific parameter</td>
</tr>
<tr>
<td>GetNextRequest</td>
<td>Manager</td>
<td>Agent</td>
<td>Get the value of a parameter whose index is unknown</td>
</tr>
<tr>
<td>GetBulkRequest</td>
<td>Manager</td>
<td>Agent</td>
<td>Get a large amount of data from an agent</td>
</tr>
<tr>
<td>SetRequest</td>
<td>Manager</td>
<td>Agent</td>
<td>Set or change a specific parameter value</td>
</tr>
<tr>
<td>Response</td>
<td>Agent</td>
<td>Manager</td>
<td>Reply to any of the get-messages</td>
</tr>
<tr>
<td>Trap</td>
<td>Agent</td>
<td>Manager</td>
<td>Reply to a predefined property</td>
</tr>
<tr>
<td>InformRequest</td>
<td>Manager</td>
<td>Manager</td>
<td>Get data from an agent under control of another manager</td>
</tr>
<tr>
<td>Report</td>
<td>Manager</td>
<td>Manager</td>
<td>Error reporting between managers</td>
</tr>
</tbody>
</table>

SNMP in itself cannot be used for automatic network configuration; however it can provide an important part of a system for configuration. If all entities in a network support SNMP it is possible to use a small set of messages and still have the ability to set any parameter at a remote network node.

4.1.2 Netconf

Netconf is a protocol that is a proposed standard by the IETF but whose work is still in progress. It is, similar to SNMP, a protocol at the application layer, which means that it is independent of underlying technologies and characteristics, i.e. the OS and physical capabilities of the entity running Netconf. Even though Netconf itself resides on the application layer, it is necessary to
use additional protocols on the same layer, these protocols are remote procedure call, RPC, and a protocol that provides security, for example SSH. When configuring a network node using Netconf, the operator sets the parameter value of interest and sends this parameter value to the node. The parameter and its value is encoded using extensible markup language, XML, and sent to the node using RPC. SSH is used by RPC for providing security to the connection. The input to Netconf is configuration data, which uses any of the built in base operations. [32]

One main advantage of Netconf is that it is able to distinguish configuration data, e.g. routing parameters that can be changed, from state data, e.g. network statistics that are calculated by the node itself. The separation has been made possible to minimize the data sets used and the possibility of errors since all parameters can be read and written.

4.1.2.1 Base operations

The proposed standard suggests a set of operations that all Netconf compliant entities must be able to support. These base operations can be extended with what is called “capabilities” that can be created for almost any task. Capabilities are discussed further in section 4.1.2.2. When not using any capabilities, which are device specific, there are nine different messages that all Netconf compliant nodes must support. The small amount of predefined messages makes Netconf appear similar to SNMP, which only supports eight different types of messages. Nine message types are however enough to retrieve configuration data from a router and change the configuration if needed. The different message types are described in Table 4-2.

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>Collect data from the currently running configuration</td>
</tr>
<tr>
<td>Get-config</td>
<td>Collect data from any configuration at the node</td>
</tr>
<tr>
<td>Edit-config</td>
<td>Edit any configuration at the node</td>
</tr>
<tr>
<td>Copy-config</td>
<td>Replace an entire configuration from another existing complete configuration</td>
</tr>
<tr>
<td>Delete-config</td>
<td>Delete any not running configuration</td>
</tr>
<tr>
<td>Lock</td>
<td>Lock the system from being changed by any other user or system</td>
</tr>
<tr>
<td>Unlock</td>
<td>Unlock a previous locked session</td>
</tr>
<tr>
<td>Close-session</td>
<td>Terminates the session</td>
</tr>
<tr>
<td>Kill-session</td>
<td>Terminates the session, aborts any running processes</td>
</tr>
</tbody>
</table>

As seen in Table 4-2, there are only five messages responsible for configuration data exchange. By using a set of parameters to each message, a lot of configuration changes can be done with
each message type. However, these message types have been, and can still be, extended using capabilities.

4.1.2.2 Capabilities

The base operations presented in the previous section is a subset of all the capabilities a Netconf device may provide, this specific subset is one that all Netconf devices must provide. During the set up process of a Netconf session a list of all capabilities supported by a node is sent to the node requesting the session and vice versa. If any capability is only supported by one of the nodes the other node must not read any message using this capability. Capabilities can be created by anyone thus increasing the ability of the system to perform network administrator tasks since the new capabilities can have any feature desired, as long as they conform to some structural demands specified in the standard. [32]

4.1.3 CIM

The common information model, CIM, can be described as an object-oriented model for defining and naming a system’s elements, their properties and also their relationship to other elements. These elements use a single model for abstraction; hence all elements can be managed independent of vendor and type. This approach is suitable when several operators are managing elements that usually must be managed in different ways. One operator may not be able to understand how to apply some settings to a certain element if the device specific commands are to be used. When using the CIM, all elements are controlled in a way that the operator only has to set the value of a certain parameter without knowing exactly how any changes are actually done in the element. [33]

CIM can be divided in three parts, the core model, the common model and an extension schema. The core model and the common model do not have a very distinct boundary, and they are both parts of the CIM schema. The core model only provides a small set of classes and properties that are applicable on all areas whereas the common model broadens this perspective with classes that offer functionality for various specific areas with the common denominator of being vendor and version independent. These classes can support several attributes, which are presented further in table 5-3. The extension schema represents objects that are specific for a certain element, i.e. a certain OS, network device or application. [33]
Table 5-3 CIM types and usage area

<table>
<thead>
<tr>
<th>Type</th>
<th>Description/Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>A group of Classes used for administration and class naming</td>
</tr>
<tr>
<td>Class</td>
<td>A collection of instances that support the same Properties and Methods</td>
</tr>
<tr>
<td>Property</td>
<td>A value to characterize instances of a Class</td>
</tr>
<tr>
<td>Method</td>
<td>A declaration of a signature (method name, return type and parameters)</td>
</tr>
<tr>
<td>Trigger</td>
<td>A recognition of a state change</td>
</tr>
<tr>
<td>Indication</td>
<td>An object created as a result of a Trigger</td>
</tr>
<tr>
<td>Association</td>
<td>A class that contains two or more References</td>
</tr>
<tr>
<td>Reference</td>
<td>Defines the role each object plays in an Association</td>
</tr>
<tr>
<td>Qualifier</td>
<td>Characterizes Named Elements</td>
</tr>
</tbody>
</table>

In order to use the CIM it is needed to have information about each element and how to control it. This is done by using managed object format, MOF, which describes the classes that each element supports, what methods can be called and what properties to set. Each element uses a specific MOF-file, which preferably is the same for all element of the same type, in which information about the configuration environment of the element is possible to collect, i.e. it does not contain information about how an element is, or can be, configured. When configuring an element the MOF data can be used to discover what attributes are needed for a certain task.

4.1.4 QPIM

The work of the IETF Policy Framework working group has been focused on finding a solution to the problem of converting a high-level network policy into actual low-level configuration. Using the QoS Policy Information Model, QPIM, which is presented in RFC 3644, this functionality can be achieved. The QPIM uses information about the network and combines that information with device specific information to create low-level configurations. The network information that is needed is a business policy, i.e. a policy on how what services to provide, the network topology and the QoS methodology, e.g. DiffServ. What QPIM does is that it provides “a means (...) to specify functionality in a standard way that is independent of the capabilities of different vendors’ devices” [34]. The network topology used is a description about all interfaces on all routers in the network that can be used when settings are applied to the nodes. This description tells the system what interface is used for what kind of traffic. When applying settings all interfaces with the same description will get the same setting. [34]
Since the QPIM is created solely for use with Quality of Service it will most definitely be of good use when it becomes a full standard, currently it has however not yet been adopted as a full standard and there is more work to be done. The strength of QPIM is that it is able to support both IntServ and DiffServ, and because of this any network provider and operator can use it. Whether or not it will require some time for acceptance by operators cannot be answered at the moment, but until it reaches the status of a full Internet standard by the IETF. At the moment it is only a proposed standard and as such it is difficult to draw any conclusion about how it will be implemented when it reaches the status of a full standard, if it ever does. [34, 35]

### 4.2 OSS-RC and AIPCM

Ericsson’s Operations Support System - Radio and Core, OSS-RC, can be used by network providers to configure their networks as well as collecting information from the network. The application is able to set up connections to various nodes and routers and also getting and setting parameter values at all nodes. It supports far more entities than discussed in this thesis, both in wired and wireless networks, especially in TDM networks in which it has been deployed worldwide.

Using OSS-RC as a base to create a tool for automatic configuration of QoS parameters in an IP network, the tool would get instant access to a large amount of features that otherwise would require several weeks of work to create. There is also a risk that a stand-alone application would get lost at Ericsson, thus not taking advantage of the work done.

AIPCM stands for Abis over IP Configuration Management and is a subsystem of OSS-RC; it is used to configure nodes that transmit Abis data. Abis is a collection name for all types of data being sent in a 2G RAN, i.e. it contains signaling traffic, bulk data and voice. The name suggests that only 2G networks are configurable using AIPCM, it is however possible to configure nodes transmitting Iub as well. Iub is a collection name for data sent in a 3G RAN. Regarding Quality of Service there is no difference between 2G and 3G data since IP is used for transmission and latter work in this thesis does not take these differences into consideration. AIPCM is still being developed and currently it does not support all IP parameters in the network.

Configuring a router requires the user input to be parsed by AIPCM into the correct format, what happens is that the new configuration is saved as an object in Java that contains the new settings. The actual communication process between AIPCM and the nodes in the network is done differently depending on what node is configured. When a SmartEdge is configured the process is quite simple, from the Java object a Netconf message is created which is passed to Net OP, a system created to communicate solely with a SmartEdge router. Net OP translates the Netconf message into commands that can be sent to the router. Other nodes are not able to handle this simple process for data exchange, when a SIU is configured the process starts with AIPCM sending a message via SSH that tells the node to send its current configuration back to AIPCM. The SIU creates an XML-file containing its configuration and sends this file via SFTP to a server that is connected to AIPCM. When this is done, AIPCM can collect the XML-file, changing it according to the new configuration, and send it back to the server. The next step is that AIPCM sends a similar message directly to the SIU which tells the SIU to get its new configuration from the same server. The process ends with the SIU getting the new XML-file from the server and applying the new change.
Even if these approaches are very different it is still possible to configure both nodes without knowing exactly what data each specific message contains. From a user point of view the wanted configuration is instantly sent to both nodes.
5 Automation of IP network nodes

The standards that are established, or proposed, do not offer a complete view on how to automate configuration in an IP network. This task is far more complex than to only provide means for collecting or setting some parameter value at a remote network node. The problem when automating the configuration process of IP network nodes is that the new configuration may affect the network in an unintended fashion. This behavior stems from the task of parsing user-friendly high-level intentions into router specific low-level commands, a task that any application for automation must be able to perform without any risk of creating commands that have negative side effects on the network. Automation of node configuration can be focused on various things, automation of some individual parameter at a set of nodes, automation for introduction of a new service or partial automation of the process to make manual configuration easier [36].

5.1 Drawbacks of manual configuration

The reason for automating the configuration process is because of the drawbacks and potential risks when manually configuring nodes in a network. The major drawback is that manual configuration is expensive, due to the complexity of configuration languages and the need for manual labor. The configuration languages available on the market often use a command line interface, CLI, from which the user inserts one command at the time. These languages contain up to several hundred, or even over a thousand, different commands [37]. For an operator to learn the commands of a single OS and a specific version of an OS takes a lot of training and costs large amounts of money. There are also an ever-increasing amount of options and new services applied to every new software version, which creates a need for constant ongoing training of the network operators. As long as there are only a small set of products, and thus a small set of operating systems in the network, it is possible for a group of operators to maintain the network. However, the network will become a lot more difficult to maintain manually as the network grows, difficulties that arise from two main reasons. The first reason is if the network is extended with new nodes similar to those already present in the network, in this case the operators will be able to maintain the new products and reconfigure them if needed. If the network is extended with a new product from a new vendor, that uses a different OS, the operators will not be able to maintain these products unless they are trained accordingly, i.e. the operators must follow the same learning curve as they already did once when they learned their first OS. This problem can also have the impact that a network provider cannot incorporate products from other vendors in their network. [36]

One potential risk that originates from manual configuration is that a minor typo during configuration of a single node can cause an error that affects the entire network. A minor error during configuration can lead to unimagined network behavior, for example that some nodes or hosts are unavailable. Totally, approximately 60% of all network downtime is due to human errors. There is also a possibility that security holes arise that do not affect the network immediately, but that can be used by an intruder. Such errors are difficult for an operator to detect while configuring a node that also makes troubleshooting difficult. [38]

5.2 Purpose of automatic configuration

To avoid the problems with manual configuration, a system for automating the configuration process can be very helpful. The problems that might appear during manual configuration are
however still present during automatic configuration and there is also a larger need for checks since the control of the configuration otherwise is taken away from the operators to some extent. Depending on the focus of the system the need for control varies. The main problem during automation is that there must be a network-wide set of rules that must be obeyed by all routers in order to avoid any errors that might affect network behavior, i.e. there must be consistency between all nodes in the network. To achieve a consistent network there must be a set of rules that checks whether or not a configuration is valid. Such controls are described further in 5.5.1. There are some examples of configurations that do not have to be exactly the same on all routers in a network for the network to provide a decent service, some of which concern Quality of Service and explained further in 6.2.

The advantage when automating any process is that instead of doing time consuming and repetitive tasks manually, a system created the right way is able to do all the involved tasks with no or little human intervention. This behavior makes it possible to get large savings during the configuration process if every operator is able to configure more nodes in the same time, or use less time for the same number of nodes. For a process that concerns a telecommunication network it is possible to have a system that, when created correctly, is able to make any changes in the network at any time without risking any downtime or inconsistencies, thus not risking not to serve any customer needs which in the end leads to customer losses due to poor service delivery. An operator that manually configures a network may not discover some of the errors created due to the similarity of configuration files, especially troubleshooting is difficult since a minor error is not easily found manually. These minor errors can however have a large impact on network behavior, and thus affecting users negatively. For troubleshooting purposes it would be sufficient to use a simple parser that detects differences between configuration files, but this is not to be considered as configuration automation even if it might be useful in some cases. Such a system must also be able to detect which differences that are actual errors and not only differences that exist since all routers have some node specific information, for example the internal IP addresses.

Inconsistency is in itself not a problem that must be avoided at any cost; however there is an increased risk that serious errors are introduced in the network for each pair of nodes where inconsistency prevails, no matter the cause of the inconsistency. The more inconsistencies that exist in the network, the harder it is to calculate with these inconsistencies when nodes are about to be reconfigured, which leads to a more difficult way to overview the consequences of the reconfiguration. It is not sure that inconsistency between two routers, or two subnets leads to any seriously negative effects but with more inconsistent subnets the risk that a service will suffer from decreased service quality increases, even if the error lies between two subnets not directly connected to each other. Figure 5-1 shows a network in which there are three sub networks, A, B and C; these three networks should preferably have the same configuration internally but there might be differences across the networks. This might cause interruptions, or decreased service levels, for traffic between two of the networks whereas all other traffic is forwarded without any problems. Hence there is a need to check for inconsistencies not only within a minor part of a network, i.e. between a pair of routers, but also on a much larger level, i.e. between domains in the network. The optimal solution would of course be to have consistent configuration settings in all three networks. A system for automatic configuration can, and should, be used to detect inconsistencies at any level, throughout an entire AS or only in a minor part of a network, which makes it possible for the network administrator to put less effort on the work to discover any
possible problems due to the new configuration.; when this is calculated automatically the administrator can decide whether to change the suggested configuration or to change an existing configuration at a router that was earlier not considered. [39]

![Diagram of connected networks in an IP RAN](image)

**Figure 5-1 - Connected networks in an IP RAN**

Depending on the level of automation there are different savings to be made, this is also heavily dependent on how each network provider decides to implement services in the network, in a network where a routing protocol is used there is no need for a system that automatically sets up static routing tables. A quite simple system can achieve savings if implemented correctly if the automated process is used often by the network operator. [40]

There is not one true way of automating network configuration, depending on the demands on the system and the properties of the network any of the approaches discussed in sections 5.2.1 through to 5.2.3 can be used. The difference between them can be quite vague and which approach to take when developing a system depends on the way of working for the organization that is going to use the system.

### 5.2.1 Partial automation of manual configuration

Partial automation can be considered as helping the operator perform configuration easier and faster using systems and applications that detect possible configuration errors and set up bounds for possible parameter values. There are techniques developed to aid the operator during the configuration process. These aids can consist of applications that transform high level user input into lower level vendor specific configuration language commands. Such systems are very useful if the operators want to keep the control of all actions being performed during configuration while being able to focus on other parts of the process than providing correct configuration language syntax. The most important part of such a system is its ability to parse user input to router commands, an ability that can be provided in different ways. The easiest way to create a system that parses user input into router readable commands is to create a database that contains
all commands that are allowed for any router. This first approach would be very static and
does not require a lot of manual labor prior to any savings in terms of fewer errors and a faster
configuration process. Once a database is created it will need recurring updates to provide
functionality for new router features. Unless new features are to be left out all new features have
to be entered manually into the database decreasing the usability of the system to some extent. A
system with a static database containing all commands will however be intelligent enough to
support a large majority of the most common commands thus saving a lot of time for the operator
even though not all commands are supported by the system. [38, 41]

The second approach is to have a system that is able to perform parsing of router commands in a
more dynamic way using configuration files from several routers to learn how the configuration
language work and what commands to use and how to use them. This dynamic approach requires
a lot more effort before the system actually works but when it does it will be able to parse any
configuration language thus decreasing the need for network operators to learn the differences
between languages. [41]

The use for a tool that supports operators with the correct syntax is, independent on how that
support is achieved, valuable for operators during their day-to-day work. However, the savings
are not as big when using a system that only assists the operators creating router commands
compared to a system that is more automated and offers more support to the operator.

The main advantage with this approach is that control of the configuration process remains at the
operator instead of being invisible to the operator inside software. Until the users trust the
software completely there is a risk they will not use the system no matt
er how intelligent it
actually is. The main disadvantage is that, since there is only little aid, the gain is not so big due
to the lack of functionality of the system.

5.2.2 Automation for introduction of new services

When a new service is to be introduced in a network there is, most certainly, a need to set some
certain parameter at every node that will support the new service. A new service is in this respect
not a service that the network users use, but instead a service that calls for a certain network
behavior, such as a routing protocol or introduction of a new method like MPLS or VLAN etc.
These services need a new set of parameters, which initially should be set to some default value
determined by the network administrator. To set these service specific parameters automatically
the system needs to parse the high-level rules from the administrator into vendor and OS specific
low-level configuration commands. It is also possible to use any underlying technology, such as
SNMP or Netconf to actually set the pa
rameter. The main advantage using a system that provides
configurations of a new service is that all service specific parameters can be created by the
system, thus there is no need to check for existing parameters and their values. An extension of
this property is that running configurations do not have to be parsed into a vendor independent
format before configurations can be done. [36]

Depending on the topology of the network there might be a need to use different parameters at
different nodes, how this must be done is service-dependent; one service might require identical
settings on all affected nodes whereas other services are still supported if there are different
settings on different nodes. The conclusion is that when creating a system that automates the
configuration of the parameters of a not yet existing service it is important to study that particular service closer and see exactly what type of user input is needed and if any alterations of the input are necessary to fit all affected nodes.

A new service is not often introduced in a network, or a domain of a network, but when it is, a lot of routers are affected. To ensure that the roll out of the new service is done correctly, and efficiently, a system that automatically implements all actions necessary through the push of a button would of course be the preferred solution for anyone in charge of the service roll out. The solution of creating a system that specifically creates the environment for a non-existing service has some drawbacks that make it almost impossible, or at least not cost effective, to implement. This solution is more likely to be created from a third party that is possible to create a system that can be used in several networks, thus getting the same advantages but for less money than if each network provider is to create its own system for deploying a certain service.

5.2.3 Automation to set a specific parameter

A system for automating the process of setting a specific parameter is in some ways quite similar to the task of introducing a new service in a network. There are, however, some details that are different, first there should preferably be a phase that collects the current running configuration from each node to avoid setting a new set of parameters without deleting the current configuration. When parameters not being used are saved in the configuration file these files will eventually build up a large amount of not used data making a manual check of a configuration file difficult when necessary. If an error occurs during an automated process it requires a lot of effort to detect such errors [42].

When changing a parameter at several nodes in a network it is important that not only the routers that are about to be reconfigured are considered, routers that are connected to those must also be taken into account due to the errors that otherwise may be introduced in the network. For this to be possible it is required to get access to the routers adjacent to those being configured and collect the parameter values of the parameter to be configured.

When configuring a set of nodes with respect to one single parameter, or a small set of parameters concerning the same task, e.g. QoS, routing etc, it is important to take the existing configuration into consideration when setting up the new parameter values. Unlike when setting up an environment for a new service this method requires retrieving existing values from both the nodes whose configuration should be updated and also the nodes that are connected to these nodes. This makes it necessary for a system to have the ability to read existing configurations and process the configuration into machine-readable text. Unless this function is at hand the automation process will be lost. Assumed that there is an underlying system for collecting and processing data from routers the next step is to find out what routers that affect each other; in a network there might be some routers that are affected by a configuration at some nodes whereas other nodes are not affected at all. Unless the administrator has full knowledge about these relationships errors might be introduced in the network. It is also possible to check adjacent routers after the operator enters the new configuration into the system; this approach has the advantage that it is most certainly easier for the operator to read a few error messages after applying a new setting than to try to adapt the new configuration based on the current configuration of a large set of routers. [38]
This method is the one that most resembles a system for actual automation of configuration
which can be used for maintenance on an everyday basis in a live commercial network. A system
that can automate every part of router configuration is basically a system that is able to automate
several minor tasks, i.e. to configure a large set of single parameters. There is no reason to
believe that there would be a system that offers all possible functionality at once; instead, any
system for automation would more likely start off with a small set of supported functions and
then be extended over time. In IP networks there is still progress being made and new services
are continuously introduced, therefore it might be a good idea to develop a system that easily
implements the new service, but there is a big disadvantage with this approach, when a service is
new it is difficult to know whether or not that service will be used to a wide enough extent that
calls for a system that automates any setting changes for that service. When the service is first
introduced there is a possibility that the service does not become widely deployed across the
entire network; if this happens a lot of effort has been spent on a system that is never used. In the
case a third party software is used, the automation software might be used to deploy a service that
is not used in the future; this approach does not require the same effort as the previous approach
but it is just as costly. As different parameters require different support from the software and the
network it would be easier to develop a system that consists of applications that are more or less
separated from each other but that share some underlying functionality such as methods for
communicating with nodes and graphical user interfaces.

When designing a system that configures a set of routers simultaneously it is important to focus
on areas that are useful for the operators to use, especially tasks that are frequently performed on
a regular basis. Tasks that are complex to do manually should also be preferred over easier tasks
that are less prone to errors. The conclusion is that some parameters are more important to focus
on when developing a system for automation. [40]

5.3 **Drawbacks of automatic configuration**

The major problem if a set of routers in a network is to be configured automatically is that there
is most likely diversity among the routers, a diversity that makes it necessary to detect any
differences in syntax and node properties. Node properties in this case can be seen as the different
settings that are possible to apply on the node together with the services that a node supports.
When automating a task, such as network configuration, there are risks involved since control
over the network shifts from being at the network administrator to being at a computer with
access to the network. Even though the administrator has access to this computer there is a risk
that the administrator misses out on some updates in the network that may cause future updates to
be based on false information.

5.4 **Requirements for an automatic configuration system**

To conclude the system specification, there are a few parts that any system for automatic network
configuration must contain; the most important part is a dynamic approach to node
communication. It is possible to create a database containing all possible commands to any router
and map user input to the correct command; since there are hundreds, even thousands, of
commands this work will be extremely time consuming and must be renewed for each new
operating system in the network. If a standardized way of communicating with nodes can be used
it is possible to have one small system for parsing user input into commands that can be sent to
any node in the network. These standardized ways of communication are explained in chapter 4.1. [43]

The system must also be easily adaptable to fit customer demands, different network providers have different ways of working and thus they may want a system that supports their way of working and does not contain any, for them, unnecessary functions. This part is almost impossible to manage without any adaptation of the software for each user (in this case a network provider) and may also be subject for changes as new ways of working are adopted in an organization. The easiest way of adapting to demands from several organizations is to support all their needs, depending on the size of the system; it may still be foreseeable even if it is not adapted to the needs of a certain user. [43]

Depending on how network data is stored there might be a need for the system to collect network information from one or more databases that could all use different data formats, e.g. SQL or XML. If a system for automatic network configuration is to be deployed in a network, or in an OAM system, it is required that the system can read different formats and calculate useful information from data input. [43]

5.5 Network view

The systems discussed above have one thing in common: their need for creating configurations that are valid throughout the entire network in which they are applied. The definition of a valid configuration is however quite arbitrary, in one network there might be stringent rules for routing and scheduling while another network needs strict rules on other parameters. This calls for a system that reacts to user input and creates node configurations one way or another. The system needs to give operators an easier way of working while not prohibiting operators or administrators from configuring the nodes manually [40].

To provide a network view, of some kind, is essential for any system for automatic configuration, without this view the system will see the network node by node thus not being able to provide the important overview that administrators use when setting up rules for a network.

5.5.1 Consistency checks

The network view is achieved through collecting information that is stored internally in each router, i.e. the routing tables can be used, or externally in a centralized database. Collecting information from every node has the advantage that it is always up to date but requires calculations every time the network topology needs to be updated. The network topology is needed to set up a domain in which the checks that control the configuration for errors are used, these checks, also known as consistency checks, are responsible for detecting any error that might occur due to typos and configurations that gives the network a certain not permitted behavior. A consistency check is probably the most important part of a working system for configuration automation, without consistency checks it would be impossible to know that the network acquires the intended behavior. These demands add to the complexity of consistency checks since they must deny all changes that are not valid while it is flexible enough to support a wide variety of configurations.
The easiest, but also the most static, way to create a set of rules is to set up these manually using existing information about the network and knowledge about how these rules are to be set up. By knowledge about the rules it means that the network administrator must set up exactly how each and every configuration parameter can be created and how a specific value of one node corresponds to a parameter value at another node. A check of this kind could for example be that all packets from a certain sub network should be blocked. When a manually created set of rules is used there is a major drawback in the fact that the checks will only be able to perform tasks explicitly created by the administrator thus forwarding any logical errors made by the administrator into the network. It is therefore of utmost importance that the rules are correct and do not have any unwanted implications to overall network behavior.

The second approach to create the rules that checks the network for errors is to create an abstraction of the current network state and to use this state as basis for computation of what rules are present in the network and if there are any nodes that violate these rules. This approach is far more processing intensive since there is a need not only to create machine-readable syntax from configuration files, a task which in itself can be filled with obstacles due to the complexity of configuration language syntax, but also to get the full picture of the current configuration of all nodes. The steps in this process are easy to follow but require a lot of processing power. The first step is to download the configuration files from the nodes and to parse these into any syntax that is easy to read, for example XML. When the configuration files are in a format that is easy to read data mining can be used to discover behavior and anomalies in large data sets. When using data mining only a small set of the available data is used as training data, which is used to create the rules that are actually present in the network. When the training data is read it is possible to discover exceptions in single nodes that are possible violations to the rules found. [41, 36]

These rules, no matter how they are created, could be applied to various parts of the configuration and check whether the configuration on a single node is violating the rules or if several nodes are configured in a way that calls for a new set of rules, or a change in the existing set.

5.6 **Network state**

The network state is built by the configurations of every node in the network and it displays how all routers are interacting with each other. The combination of the configurations on all nodes in the network and it affects network behavior. When a configuration update is performed it is important to have information about the state of the network to calculate how the update will affect network behavior before actual configuration is done.

There are two ways to keep track of any changes of the network state, either to maintain a database that contains the configurations of all nodes in the network or to get the configuration from the nodes every time the network state is needed. Each method has its advantages and disadvantages: a database stored at a central server provides faster access to the information but requires hardware on which to store the database. Furthermore, a database requires a system for updating the database, probably a manual system through which the operator updates any information that is changed in the network. If the network itself is the database there is no need for separate hardware or a database that needs maintenance. However, this method requires more
processing power each time the network state needs to be calculated and this method will provide slower access to the information stored.
6 Automation of QoS parameter configuration

Quality of Service parameters may need to be changed at any time due to a change in network topology or because of introduction of a new service. Since there are a lot of nodes in a network that are affected by such a change, great savings could be achieved if the nodes in the network are updated automatically. The next section discusses an application that automates the configuration of QoS parameters in an Ericsson IP RAN.

6.1 Application properties

The application must define which nodes to configure and how these nodes should be configured. First, the user sets the rules, i.e. the configuration, that should be applied to the network. The rules that are possible to set are rules concerning scheduling types, number of queues, mapping between traffic types and queues. The application will take these rules and apply them to each of the nodes of interest. When a change in the configuration is done the application must also perform a consistency check to make sure that the applied changes won’t have any negative effects on the network that the user did not anticipate.

6.1.1 Scheduling

The scheduling principle used at any node can be strict priority, SP, fair queuing, FQ, or weighted fair queuing, WFQ. FQ can be considered as a special case of WFQ where all queues have the same weight, i.e. all queues are given the same capacity. There are other parameters that affect queuing, such as if random early detection, RED, is used or not as well as whether FIFO or priority queuing, PQ, should be used at the outgoing queue. When using RED, it is also important to state upper and lower thresholds of data rate that, when exceeded by any flow, are exposed to drops either randomly when exceeding the lower threshold or entirely when the data rate exceeds the upper threshold. [44]

6.1.2 Number of queues

How many queues each egress interface should use must not be consistent in all nodes but all nodes should preferably use the IP RAN recommendations that already exist. The application, and especially its consistency check, must take into account the different possibilities of every node. If some node puts restrictions on how many queues that are possible to create, this restriction has to be taken into account when performing consistency checks. The RAN routers, e.g. routers of the SmartEdge series and Extreme Networks Summit series, has a maximum of eight queues at each outgoing interface, thus putting a constraint on how much traffic classes can be separated [8]. There is therefore no need to separate traffic into more queues at other nodes since this separation won’t affect overall network performance. When having different number of queues in different nodes it is important to set the priority of each queue and traffic class, which is described further in next section.

6.1.3 Priority

For each outgoing queue in a network node it is important to separate traffic classes from each other for QoS to be reached. A traffic class consists of several flows created by different applications but that still put the same requirements on the network. In a RAN, most of the traffic can be divided in one of the four types, “speech”, “bulk data”, request/response applications” and
“network control and signaling”. This does not mean however, that these four classes of traffic are the only classes needed. The IETF has set configuration guidelines for DiffServ service classes in which traffic classes and their respective DSCP value are presented as stated in section 2.3.2 [45]. These values are shown in table Table 6-1, and give an overview of how to prioritize traffic classes; it is therefore not to be considered as a definitive solution that must be used in practice but merely a recommendation to how traffic classes can be prioritized with respect to their needs and demands.

<table>
<thead>
<tr>
<th>Service Class Name</th>
<th>DSCP Name</th>
<th>DSCP value</th>
<th>Application examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Control</td>
<td>CS6</td>
<td>110000</td>
<td>Network routing</td>
</tr>
<tr>
<td>Telephony</td>
<td>EF</td>
<td>101110</td>
<td>IP Telephony bearer</td>
</tr>
<tr>
<td>Signaling</td>
<td>CS5</td>
<td>101000</td>
<td>IP Telephony signaling</td>
</tr>
<tr>
<td>Multimedia</td>
<td>AF41, AF42, AF43</td>
<td>100010, 100100, 100110</td>
<td>H.323/V2 video conferencing (adaptive)</td>
</tr>
<tr>
<td>Conferencing</td>
<td>CS4</td>
<td>100000</td>
<td>Video conferencing and Interactive gaming</td>
</tr>
<tr>
<td>Real-Time Interactive</td>
<td>CS4</td>
<td>011000</td>
<td>Streaming video and audio on demand</td>
</tr>
<tr>
<td>Multimedia</td>
<td>AF31, AF32, AF33</td>
<td>011010, 011100, 011110</td>
<td>Streaming video and audio on demand</td>
</tr>
<tr>
<td>Streaming</td>
<td>CS3</td>
<td>011000</td>
<td>Broadcast TV &amp; live events</td>
</tr>
<tr>
<td>Broadcast Video</td>
<td>CS3</td>
<td>011000</td>
<td>Broadcast TV &amp; live events</td>
</tr>
<tr>
<td>Low-Latency Data</td>
<td>AF21, AF22, AF23</td>
<td>010010, 010100, 010110</td>
<td>Client/server transactions Web-based ordering</td>
</tr>
<tr>
<td>OAM</td>
<td>CS2</td>
<td>010000</td>
<td>OAM&amp;P</td>
</tr>
<tr>
<td>High-Throughput</td>
<td>AF11, AF12, AF13</td>
<td>001010, 001100, 001110</td>
<td>Store and forward applications</td>
</tr>
<tr>
<td>Data</td>
<td>DF (CS0)</td>
<td>000000</td>
<td>Undifferentiated applications</td>
</tr>
<tr>
<td>Standard</td>
<td>DF (CS0)</td>
<td>000000</td>
<td>Undifferentiated applications</td>
</tr>
<tr>
<td>Low-Priority Data</td>
<td>CS1</td>
<td>001000</td>
<td>Any flow that has no BW assurance</td>
</tr>
</tbody>
</table>

As shown in table Table 6-1, traffic flows can easily be divided into different classes that can be handled differently by the network without decreasing user experienced Quality of Service. A problem occurs however when traffic passes through a node that is not DiffServ compliant in the sense that it cannot read layer 3 data, but most solely rely on layer 2 information. The solution to this problem is to perform a mapping between DSCP and p-bits in a way that keeps the different classes separated and thus do not degrade system performance negatively. Mapping between DSCP and p-bits can be done in different ways; one way to do this is described in table B-1 in appendix B. [45]

6.1.4 Neighboring networks

Automation can only be done in a network that is controllable, i.e. any leased lines or neighboring networks that is out of control for the operator will be considered consistent to the network that is configured. Service Level Agreements, SLA, will most likely control all nodes
and links affected. When automating an IP RAN the situation is probably that the operator controls the entire network either directly using its own network or indirectly with strict SLA’s with other network operators.

6.1.5 User input
When using the application, it is not necessary to set configuration parameters according to all problems presented in previous sections. Each of the rules are set and checked for consistency independently of each other. The application will check if the input values are valid, e.g. if the scheduling principle applied is part of a predefined set of parameter values that are possible to use or that the number of queues wanted is within a fixed interval. This check is independent to consistency checks that are responsible for checking the validity of a setting on a network level. Continuing with the example of scheduling principle it is not necessary that a valid input, such as WFQ, is applicable on all routers. There might one, or a few, nodes in the network that are not able to cope with this setting, thus the system must be able to catch such problems and let the user apply a new principle for these, or all, nodes.

6.2 Consistency check
The consistency checks make sure that all new settings are valid and without affecting the network functionality in a way that was not intended. This is done by applying a set of rules that must be obeyed by all nodes in the network; together these rules will make sure that all nodes, and thus the entire network, will get the proper behavior. It is important to note that the proper behavior is the behavior that the network administrator intends to apply to the nodes. This makes it difficult to set up rules since they must be able to cope with any behavior wanted. The rules must also be able to manage a heterogeneous set of nodes that all have different characteristics.

6.2.1 Levels of consistency checks
A violation of a consistency check does not necessarily mean that the entire network will fail to provide basic services; instead some checks are of minor severity, which may only cause a slightly decreased Quality of Service, however still being satisfactory. In the case of setting a new principle for scheduling packets at an outgoing interface of a node the system must check for inconsistencies; however, when using different scheduling principles it is difficult to say whether or not a certain principle outperforms another since their different characteristics affect traffic differently. A consistency check that controls the scheduling principle must therefore be of the lowest severity level, a level at which only information about possible differences is sent to the operator.

Other consistency checks are far more important since a major part of network behavior is dependent upon these. For example are checks for mapping and classification very important since a minor error in one node can cause the network to behave in a way that is completely unacceptable for many, if not all, users. In this case the consistency check must be of the blocking level that makes it impossible to apply the change to nodes in the network. The system will not change the configuration provided by the operator but provide a configuration that would be accepted.
6.3 Use cases

A system that automatically configures QoS parameters in IP network nodes must be able to handle a number of demands posed on the application such as functionality and error correction. To show what the application is supposed to be able to perform, some scenarios are presented in which various parts of the application are used. The first scenario is a case where a network operator wants to set up consistent, e.g. equal queues, in an IP RAN that consists of SmartEdge and Extreme Networks Summit routers, Marconi OMS 1410 transport nodes and SIU’s. In this case the operator must first choose which queue type to apply on the nodes, SP, FQ or WFQ, and how many queues that each egress interface should have, one primary value that is preferred and one secondary value that is set if any node is unable to handle the primary choice. The operator must also set if RED should be used for congestion control and if so, also set the parameters that control the behavior of RED. This case is quite simple and the only consistency check needed is to control whether or not each node in the network is able to handle the number of queues wanted. However, there is a risk that neighboring nodes have another queue setting than the nodes selected by the operator, thus making it possible that two traffic classes that are separated in one node must use the same queue in another, thus possibly degrading the service for, at least, one of the classes concerned. Due to this problem, the tool must, if possible, check if there are any neighboring nodes that could handle the suggested number of queues, but that currently do not. The reason it might not be possible to check a neighboring node is if any other operator controls it.

The second case is when an operator wants to set priorities for queuing. Since there is a maximum of eight queues in the high performance routers there is no need for more than eight traffic classes even though there are more than eight traffic types. These eight different traffic classes have to be assigned a priority and also a marking to separate packets from each other based on their traffic class. This second case is closely bounded to the first case since the number of queues affects how traffic class aggregation can, or must, be done. It is also the case that when changing the number of queues a similar consistency check must be performed that changes the mapping between traffic classes and their queue. The easiest case is when the number of queues is the maximum value, e.g. eight in this case; then each traffic class will have its own queue. In the case of fewer queues the tool checks for existing rules that states how the mapping between traffic classes and queues should be done. The consistency check for this case must be able to find the rules that network administrators have set in order to find the mapping between traffic classes and their priorities; it must also do a check similar to the previous case, in which the possible amount of queues were checked, to find nodes that are unable to handle the suggested configuration.

All consistency checks that are performed must be able to detect misconfigurations due to several reasons; a problem that appears is that there has to be a lot of rules to detect all misconfigurations that may have appeared. To avoid this problem a generic approach towards creating rules for consistency checks would be preferred. However, what is a configuration error in one network might not be an error in a different network. This is due to network management and the fact that network administrators have created rules and configurations manually on a small network with few services and changing these rules as network size and complexity has grown. [46]
6.3.1 Network topology

The network used as a reference model for the cases is built entirely of products from Ericsson, with the exception of routers from Extreme Networks, i.e. the same products discussed in section 3. Since the objective of the application is to work flawless in a lab environment and to be seamlessly integrated with Ericsson OSS-RC, there is no need to support products from other vendors that are not used in the Ericsson IP RAN solution. Products from competing manufacturers lack basic communication support in OSS-RC, hence the exception from such products.

6.3.2 Communication

When an operator changes the configuration on a network the network state must first be known in order to avoid unnecessary processing and time-consuming changes that only applies the current network state to each node once again. To get information about the network state the system must be able to find all nodes in the network and collect the data that is needed to provide a full picture from them, thus enabling the system to apply unnecessary configuration changes to any nodes. To collect data from the nodes the proper messages must be sent using Ericsson’s Operations Support System - Radio and Core, OSS-RC.

6.3.3 Details of the automation process

The system needs user input to which it can react, this input is to be seen as the rules and policies set up by the network administrator. The user input can be divided into two parts: what entities to configure and how to perform the configuration on those entities. To state what entities to configure the user simply gives the IP addresses, or a range of addresses, of the nodes to be configured. How the new configuration is supposed to look like is dependent on the second step where the user enters the parameters to configure, their name and the values of the parameters.

The system needs to save all user input temporarily and does this by saving the input in two lists, separated from each other, the first one containing information about the nodes in the network using the 4-tuple IP address, node type, parameter names and their respective values. The second list contains Boolean variables that states whether or not a certain consistency check should be performed depending on if the parameters that the check belongs to has been changed.

There are consistency checks created for all of the parameters possible to set; each check is called when the system has gained knowledge about the performance of the nodes. To get this information, the system logs on to every node and collects the data needed. This data contains information about the node type, its current configuration and the possibilities for configuration. When this step is done the 4-tuple of information presented above is able to be populated completely.

The consistency checks are now possible to call using the information needed for each check, this may vary independently of each other and only the information actually needed is sent to the check. The result of the consistency check can be that the new configuration is allowed without any detected errors or that an error of some kind has been detected. If an error has occurred the system must also detect the severity level for that certain error. Some errors are serious and cause the system to block the configuration from being carried out whereas other errors only create an information message sent to the operator.
Figure 6-1 shows the structure of the system, user input is calculated for errors and sent to a database from which all nodes are configured. Details of the process are further described in section 7.

Figure 6-1 - Structure of the configuration process
7 System design

To create a system for automatic configuration of QoS parameters there are some constraints and properties posed by the network to consider when setting up specifications for the system. Since there are only a few parameters that this system should be able to handle it is possible to focus on consistency checks; focus should preferably be on consistency checks no matter how many parameters the system should support, this to ensure that the system can be used in a commercial network. Quality of Service is not to be seen as an exact science and is, unlike most things in IP networks, not binary in the sense that it can be either right or wrong. For this reason, consistency checks for QoS parameters are very difficult to create to give an answer whether or not a setting is good or bad, it can only say if the setting is possible to apply at all, or if there are any extreme errors that will disrupt network behavior, i.e. they can tell whether or not two settings are equal or not. It is always the administrator who is responsible for the configurations in the network and a system for automatic configuration should therefore not tell the administrator what is good or bad as long as some critical properties are fulfilled. The system must also be general enough to be used by different operators that use different methods to achieve what they consider good Quality of Service, for this reason consistency checks must only detect configurations that obstruct basic rules to be obeyed. Examples of such basic rules can be forwarding priorities, which, when done improperly, can cause loss of large amounts of jitter sensitive traffic.

These properties calls for a system that is able to discover errors on nodes as well as differences in configuration among nodes in the network. The user must correct all node errors; otherwise the configuration will not be applicable at all. An error is in this sense to be seen as a value that is not on the right format or contains a value that is not valid. Input that is not intended by the user but still applicable and otherwise correct will not be detected as an error.

The system can be described with Figure 7-1, which shows how data is collected from the user, checked for errors and sent to the network, all configuration updates are made after all checks have been completed to avoid causing inconsistencies during the configuration process. All input is parsed into its correct data type that is to be used throughout the process and checked for errors, these checks are further described in 7.3. When all checks are complete, and all errors have been corrected, objects containing the new configuration and the IP addresses to the nodes that should be configured are sent to AIPCM, which in turn applies the new configuration to each node.
Netconf capabilities as well as QPIM and CIM are methods that all can be used for a system that automates configuration of network nodes. When using Netconf it is possible to create a capability that sets a number of parameters using only a single command. The methods mentioned are only ways of communicating with hosts, there is also a need for some kind of intelligence in the system, intelligence that controls that only the correct nodes are configured properly and also that adjacent nodes are not affected. This means that not only the nodes that are configured must be checked; it is essential that adjacent nodes are checked in order to find possible misconfigurations. The intelligence of an automated configuration system must consist of a few parts in order to work in an operational environment

1. A database containing all entities in the network
2. A database containing the current configuration
3. Rules for how configurations can be combined between sub networks
4. Rules for how configurations can be combined between entities in the same sub network, and finally
5. Methods for communication with nodes in the network.

The solution to the first point has a rather simple; most likely does such a database already exist available to access by an administrator. This database must contain all necessary management information such as IP addresses, machine type and software version. An important note is that any upgrades, physical or logical, must not only be made in the already existing database but also in the database that the system for configuration uses. If this second database is not updated properly the possibility to cause a failure of the configuration system is created since the new configuration is compared to a non-valid configuration. Preferably, there should only be one database, containing both network information and the configuration data. The database
containing the current configuration may be stored entirely on a separate storage media but can also be collected from all nodes when necessary. How this is done depends on how easy configurations can be read from entities in the network and the cost of data storage. Since data storage is more or less free there is no need to collect configurations every time a change is to be performed. Factors like how often changes are done, the number of machines in the network and the complexity of parsing router data into data that is possible to store in a database are also important when deciding on how to store network information. When collecting configuration data from the network there is no risk that the system uses information that is out-of-date.

The third requirement that is posed on the system is how configurations can be combined between a set of routers in one part of the network and a set of routers in another part, i.e. how two sub networks, or network domains, affect each other. The problem is that if an administrator wants to reconfigure a set of nodes, other parts of the network risk not to be taken into consideration. This problem may create a network in which a small subset of the nodes are configured in a way that does not conform to the configuration of the majority of the nodes, hence there is a risk that the Quality of Service is degraded for traffic traveling through this sub network since there is no verification of the QoS settings between the set of nodes. A result of this is the need to check not only the nodes that were actually reconfigured but also their adjacent nodes; otherwise there might be links on which the network changes its behavior considerably, thus affecting traffic behavior in an unforeseeable manner.

When checking two sub networks for possible differences in configuration, the database containing all nodes must be used to find the links on which there might be inconsistent QoS settings, for this reason the database must not only contain the nodes that are currently reconfigured but also adjacent nodes. It is hard to overview the possible results of two differently configured networks exchanging delay- and jitter-sensitive traffic, but most likely will the service level decrease to some extent. When the network suffers from congestion, such misconfigurations may have an especially large impact on overall performance since there may be a difference in how traffic is queued, forwarded and even classified.

One problem that might occur is related to timing; during configuration the system might believe that the first node configured has gotten a faulty configuration since its settings may differ extensible from those of other nodes. It is therefore important not to make any conclusion too soon if a consistency check has been violated. The easiest solution to this problem is to avoid setting up the new configuration at the nodes before consistency checks on network level are finished.

On the border between two separate networks, for example a RAN and a core network, there might be different rules for how QoS should be used; therefore violations will most probably occur when configuring any node connecting to another network. It is unlikely that a RAN connects to another network except at some specific routers or through an RNC or BSC. This behavior of the network topology, which might be present in some but not all networks, requires that the system is possible to leave some inconsistencies since they may not be able to be solved.
7.1 Communication between devices in the RAN

In order to communicate with a node in an IP network there must be an underlying system that is responsible for the actual physical data transmission, and more important to set up how data transmission is done. Since a system to send information to routers has been developed at Ericsson it seems reasonable to use that system in order to avoid copying work already done. There are however some drawbacks using the Ericsson based system. For example, there are only a limited number of nodes supported by the system, which makes it impossible to communicate to some nodes using this system. At the moment the system does not support any nodes that are not manufactured by Ericsson or any of its subsidiaries. The system, OSS-RC, is based on Java but it also includes XML, CIM, common management information, CMI and the managed object format, MOF, to set up a structure for nodes and their properties. Unfortunately this system contains a very large amount of files that makes it rather difficult to get a picture on how all files are connected, both to each other and to a physical network node. The intricate design of OSS-RC suggests that it would be a better solution to create a system that does not use OSS-RC for communication. On the other hand, the approach taken should also be considered from the view as a business case and not only as entirely academic. From this point of view it would be more interesting with an application that is integrated into OSS-RC since network operators use it on a daily basis, a stand-alone application risks being poorly maintained and not spread to those who might want to use it.

The physical update of a new configuration is done differently depending on the node that is updated. Figure 7-2 shows how the processes that takes place when a SIU is reconfigured. What happens is that a new XML-file is created based on the user input. This xml-file is sent to a database from which the SIU can retrieve it at any time. In order for the node to get the new file it needs to be told that there is a new file to get, this is done by sending an encrypted message to the SIU that tells it to get the new file from the database. The SIU is now able to connect to the database, get its new configuration and perform all changes necessary to complete the configuration process.

![Figure 7-2 - Configuration update of a SIU](image)

7.2 Storage of network information

An up to date database that contains all Quality of Service settings on all nodes in the network must be maintained continuously to avoid using a database that is out of date which can cause errors messages when not needed and vice versa. This approach minimizes however the need to send data through the network which is an advantage especially during hours with heavy traffic load when it can be useful to use information from the network without stressing it further. There is also the possibility to let the network be the database; in this case the application must gather
the information necessary before any further actions can be made. This approach requires less intelligence from the system but more data transmission and calculations during configuration updates. Furthermore, there is no doubt that the database will be correct and up to date. These properties combined with each other make it suitable to let the network store the information that is needed to check configuration updates for errors, i.e. the amount of data needed to be stored can be decreased to an absolute minimum.

7.3 Design specification

As discussed, the process begins with the user entering the management IP address of all nodes to configure; this is done in the window displayed in Figure 7-3. In this window it only possible to set these addresses and forwarding is disabled until any value is entered. When pressing “Next”, the system checks that only valid IP addresses are entered before showing the next window.

The object containing the IP addresses must be reachable from anywhere within the event creating thread of the application since these addresses are needed continuously during the configuration process, for example when data from AIPCM is needed.

In this second window, shown in Figure 7-4, the user enters the DSCP and priority bit values to map to each queue. If any queue is not given any value, no packets will enter this queue, and thus it remains unused. When input is finished, the system sends input to methods responsible to check that neither DSCP nor P-bit values are out of range or duplicated. Regarding priority of traffic in a network it is essential that one traffic type does not have a higher priority than another traffic type in one node and a lower priority in another. Therefore, the system must check that for
any pair of entered values, they have the same relative priority in all nodes adjacent to the nodes reconfigured. The values of the DSCP classes can be considered being well known to network operators; hence those values can be used here.

![Automatic QoS configuration](image)

**Figure 7-4 - Map DSCP and P-bit values to queues**

The window where DSCP and P-bit values are mapped to queues also contains a checkbox to set if the user wants to create a mapping between DSCP and P-bit values. This information is valuable for nodes that are not able to handle any of these two values. When this box is checked, a new window shown in Figure 7-5 opens. This window is similar to the previous, since very similar input should be entered here. The new window is needed since it should be possible to map several DSCP and P-bit values to one queue, but there must only be one P-bit value mapped to any single DSCP value. Therefore, the data entered in Figure 7-4 may not be enough to create a consistent mapping, e.g. if there are three DSCP values and two P-bit values mapped to one queue it is impossible to say which P-bit value any of the DSCP values should be mapped to.

When this mapping is entered, the system must check that all values are within the allowed range the same way as done when the data from the previous window is to be checked.
The last window that handles user input contains buttons where the user can set which scheduling principle to use. Since the system might be used for reconfiguring nodes, these nodes might already use some type of scheduling that the user does not want to change. Therefore, this step is possible to skip without choosing any of the buttons. There is also no need for checks since only one predefined value is possible to choose, and thus there is no risk that the user inserts any values that causes errors in the nodes or network. This window is shown in Figure 7-6.
The last window of this configuration wizard is used to inform the user that the data input process is complete and that the nodes will be configured when the user presses the “Finish” button. When the user reaches this window everything that the system must do before configuring nodes is already completed. The only thing that happens when the process is finished is that all data is pushed to AIPCM for further transmission to nodes. It is still possible to browse backwards through the wizard, but all other buttons are disabled. This last window is shown in Figure 7-7.
There is one more function in this wizard, the “Skip checks” button; this button is needed if the user may want to override any consistency checks. Unless this is always possible, control of the network behavior will switch from the user to the system, which is not intended and may eventually decreasing the use of the application.

Functionality to retrieve all QoS parameters from AIPCM may not be present but this document will also describe how to extend the existing system to adapt to an application for automatic configuration of QoS settings, these recommendations are found in section 7.4.

If an error is detected, no matter the reason, the window for user input is not updated with a new window; instead the same window is displayed together with an error message. The cause of the error must also be displayed to the user to make it possible to adapt the input correctly. For each QoS parameter a certain function is called in which the user input is compared to the configuration setting, at the nodes where changes are to be performed. A comparison to settings at the router is needed for parameters that vary from one router to another such as possible number of queues at a port or the scheduling principle that can be used. Some parameters are more static in the sense that there can only be a certain set of values, i.e. the DSCP values range from 0 to 63 and the priority bits are in the range 0 to 7. In this latter case there is no need to get data from a network node since the allowed input is already stored in the application.

If all steps are carried through and user input is given at all steps the entire process can be described using Figure 7-8. This figure shows in what order all input is given and also what input
that must to be checked for errors. This is not important for all parameter since some input can be chosen from a predefined set of values, which make it correct without the need of error control.

Figure 7-8 – All steps of the automation process

The mapping between DSCP and P-bit values is only entered if the user makes an active choice to enter this information, otherwise this part of the process is skipped, hence the two branches at the end of the process.
7.3.1 Consistency checks

The controls that make sure that the wanted configuration is possible can be divided into two categories, controls that check for errors at a single node and controls that check for errors between a set of nodes. The checks can be further divided into two subgroups, checks that calls functions in AIPCM and checks that only use functions built into the application. When calling AIPCM the system requires some information that is specific for a single node whereas checks that can be completely built-in into the application handles logical rules that must be followed by all nodes.

Each control that passes the input from the system on to AIPCM discovers the type of the current node that is to be configured and calls the correct node specific method for error checking. In AIPCM it is also possible to access the correct consistency checks and messages without further processing, e.g. if the correct processes for the correct node are called the correct consistency will also be called automatically.

7.3.1.1 Node level

The output from the controls do not have to be processed any further if no errors have occurred, the system will however not throw away any data before finished running. If errors have occurred, the system, and the user, must take the proper actions. Since the system does not know what is right or wrong the operator must change any input that turned out to be wrong. The system can aid the user by presenting any errors that have occurred may it be a typo or a logical error, to the user and allow for changes of the erroneous variable.

There are four main checks that must be done for each node in the network; these are, with no relative order, to check that the wanted number of queues and scheduling principle are possible to apply and also that the wanted DSCP and P-bit values versus queues show a strict priority between different traffic classes. The first two checks require integration/data exchange with AIPCM, and thus must the IP address of the node be a part of the input to the check. Except the IP address the parameter value, and the parameter, is the only needed data input. The input is in fact a vector of IP addresses to make it necessary only to call this function only once. The functionality required from AIPCM is to get the type of node from an IP address and then call the correct functions depending on the input. The required output from AIPCM is the user input, unaltered, as well as the possible range of values. This output requirement can be changed to contain only a Boolean variable that states if the change is ok or not, and if it is not ok the possible values, or value ranges. What approach to take depends on if there are any gains on processing times and capacity etc.

For checks that do not require any data exchange with AIPCM there is also no need to use IP addresses. The consistency checks that are responsible for controlling that prioritization is done correctly only require the sought mapping between DSCP and P-bit values and the outgoing queue, which is the priority. The input to this check is the relative order of DSCP and P-bit values, data that can be compressed into two arrays. The possible output is a Boolean variable that states whether or not the wanted priority is applicable; and in the case of an error, what values that must be corrected.
7.3.1.2 Network level

When a parameter has been checked for node internal errors, the same parameter is checked on a network wide level to make sure that the new configuration does not have any large negative impact on overall network behavior. These checks are more complex since they check dependencies of parameters according to relationships of nodes. The relationships can be found either by asking each node for its neighbors’ management IP addresses or by storing the relationships in a database. The approach chosen for this system is to use the latter due to its simplicity. Using this information it is possible to see how data is forwarded in the network, and thus how a configuration in one node affects traffic in another node.

The system now checks the network for possible “errors” in the configurations, as Quality of Service is not something that is right or wrong, an error is in this case merely a way of describing a pair of configurations that, when compared to each other, show significant differences. The differences between configurations are causes for problems; in a network when all routers are configured the same way there is little risk that any errors occur as long as all nodes are correctly configured.

In order to check for possible inconsistencies throughout a network, or a part of a network, there must be several different methods to do so. The approach that tends to be simpler is to have one function for each QoS parameter; this approach also provides a better forward compatibility since any algorithm can be changed easier. All these methods are built up in the same way but there are some differences in the impact of the result. The method that checks for inconsistencies among the mapping between DSCP and queues must stop any configuration that sets the mapping to switch the priority for two traffic classes between nodes on a path. Such an inconsistency must never be present in a network where traffic prioritization is important.

These methods, whose purpose are described above, must also be described in a way to make sure that a programmer with full access to the entire program, not just access to read the source code but also the permission to change it, can set up the application properly. There are some classes that are specific for this application, and they are described in later sections.

When the process begins there is a lot of data exchange between subsystems of the application due to the need for scalability. The application can be divided in four main parts, one responsible for the user interface, one for the connection to AIPCM, one responsible for consistency checks and a last part that makes sure that all threads are executed in order. This last part calls all other methods to create the GUI, check input for errors and send the input to AIPCM, most methods called do however call other methods, this to make analysis, upgrades and troubleshooting easier. With all methods named and commented properly this approach can hopefully assist software developers to improve the application.

Figure 7-9 shows what happens when a user sets new DSCP values to queues at a node, if several nodes are configured the same steps will be carried through once for each node. First, the input is checked for errors at node level. Feedback is given to the user if any value is not applicable. When the new setting is compared to the setting on neighboring nodes, AIPCM must be called to retrieve their current configuration of DSCP values from all affected nodes. The values on each pair of nodes are compared with each other, if any errors are detected the user will receive
feedback from the system and a chance to correct the input. If no errors are detected, the new configuration will be sent to AIPCM.

![Diagram of processing DSCP values, detailed](image)

There are two boxes in Figure 7-9 that show where the application connects to AIPCM, unfortunately, not all functionality required is currently supported by AIPCM. This support is crucial for the application and it should hopefully be present soon. The functionality wanted from AIPCM regards transferring user input from the local computer to the nodes in the network as well as collecting some data from the nodes.

### 7.4 Extensions to AIPCM

There are some extensions necessary to make to AIPCM in order for the prototype created to work according to its requirements; extensions needed due to the lack of functionality for controlling IP based nodes in AIPCM. These extensions are discussed as well as how the application created will suffer if AIPCM is not extended accordingly.
A vital requirement for automation is a database in which all nodes together with their current configuration. As previously discussed it is also possible to get the current configuration from all nodes prior to beginning the configuration process, which will increase process time but reduce the need for data storage and the possibility for using configurations that are out-of-date. A network provider may however have products in the network that are not supported at all in AIPCM, such as routers from Cisco or Juniper, which requires a database anyway, in such cases the system will be made easier if all data is possible to collect from one single source. All calls to AIPCM from the system are done via “accessAIPCM.java”, which can be changed to call a database or each individual node. As long as the type of the objects returned are not changed the system itself will not notice any difference. The methods that should be updated are “getIPFromRelation”, “getDSCPFromIP” and “getPbitFromIP”.

There is another method in “accessAIPCM” that may need to be updated, this method is “setConfig” that needs to call each node and set the new configuration. Depending on how communication is done to an individual node, as discussed in section 4.2, the correct method in AIPCM must be called with the new configuration as input. Before this has been updated, there is currently no possibility to set any new configurations due to this lack of communication. Regarding the individual parameters discussed, AIPCM must be updated to support all these parameters, otherwise AIPCM will not be able to create the correct messages and send them to the nodes.
8 Discussion

Today, there are no systems for automatic network configuration that are able to handle the diverse network that an IP RAN is. Most work done in this area is focused towards automating configuration of routers from Cisco since their products are very common in high-speed IP networks. The performed work still presents some important ideas to consider during development of a system for use in an IP RAN. A configuration system should be built with two main parts are vital for the system to work, one part that sets up the boundaries for the data to apply to nodes and one part for communication in the network. The first part must handle all exceptions that can be anticipated from user input, it is not reasonable that input control should be done by the user itself. If that was the case usability of the system would rapidly decrease. This part of the system is also responsible to check that the input does not affect the network negatively by using consistency checks. For Quality of Service consistency checks tend to be rather arbitrary since one service level can be reached with different approaches. Therefore these checks are merely a support to the user in that they find differences in configuration files across the network and give feedback to the user if any inconsistencies have been found.

The second part of the system is responsible for pushing the final configuration to all nodes. A system that does not contain well established, standardized, ways of communicating with nodes must use the built in CLI of every router. It is possible to create a system that sends exact commands via SSH but it requires lot of work, such a system would require a much effort only to parsing user input into OS dependent command.

The two parts of the system must be able to interact with each other continuously throughout the process since data from the network is needed to calculate if, and how, the new configuration affects overall network behavior. When data is collected from the network during the process it is guaranteed that no data is out-of-date, if wrong data is used as input to any check, the output from that check will also be wrong.

When these two parts coexist it is possible to develop both parts independent of each other, thus making it possible for someone to focus on developing and extending the system that checks the input without having to keep data transmission in mind. Since the two parts of the system must interact there is however still a need for the developers to work in the same direction.

Automating QoS parameters could be very relevant for network operators, depending on how much effort is put on reconfiguring QoS parts without a system for automation, large savings can be made, both in time and money. Since configuration is done faster when automated there is a shorter time slot during which there are several different QoS settings, thus not leaving the network open for unnecessary failures.
9 Conclusion
This thesis discusses the concepts of automatic IP network configuration with focus towards QoS parameters in the Ericsson IP RAN solution. A vast majority of the work is however applicable on any IP network. The reason for automating the configuration process is that there is a large risk that errors are introduced in the network due to mistakes by the network operator when routers are configured manually using the built-in CLI of a router. In a network with several different routers using different operating system, and thus different syntax for commands, this risk increases. With errors in the network the risk that any error has a large impact on the service level increases, hence the need for control of user input. Automation is not difficult to achieve per se, there are however several things to consider to make a configuration system work in a live commercial network. The paradox with automated configuration is that the operators want to have control over their work while the system requires some of this control in order to give the operator the intended support. If both there demands are to be fulfilled it is needed that the operator has control over all data input in the sense that the control of the system can always be overridden. This is especially important for a system that handles Quality of Service parameters. Other systems for automatic configuration, that do not concern Quality of Service, may need to block the new configuration due to some factor that can be calculated by the system.

How these checks should be designed is one of the most important success factors for this type of systems, if these checks do not conform with the approach to configuration of the network provider there is a risk that do not provide the aid that is intended. It is however easier to adapt the way of working of the network operators than the design of the system, especially if the system is to be integrated in AIPCM which is used by several organizations.

It is possible to implement a system for automatic configuration of IP network routers using OSS-RC and AIPCM for creating and transmitting the files and data messages that are needed to configure each individual node. There is however some work needed in AIPCM to support the configuration of QoS parameters; at the moment there is little support for IP configuration at all in AIPCM. When AIPCM is extended enough it is possible to implement the prototype made into AIPCM to enable automatic configuration of QoS parameters in IP RAN.

The prototype developed is able to compare the given input with the constraints on that particular parameter and also to check if there are any problems with the new configuration when it is compared to the configuration of adjacent routers of the nodes reconfigured.
10 Future work

The work presented here can be extended in various directions; there are some main areas though in which this thesis can be used as a foundation for further development. These main areas are AIPCM, QoS and configuration automation. Regarding AIPCM there is some work needed in order to make automation possible at all, this should however be done as AIPCM is continuously developed and more support for IP nodes should be implemented. Since DiffServ only uses eight bits of the IP header it is not possible to extend the work in this direction but other parameters that are connected to QoS should be evaluated, it is also possible to implement IntServ in the system. It is possible to apply deterministic calculation to router performance; if this is done it is possible to create a system that is possible to calculate how a new configuration affects the network instead of leaving these calculations to the operator. This case also makes it possible to create a network in which inconsistencies can be handled easily since the system would be able to calculate optimal configurations. This latter idea must be considered as very difficult to apply in a commercial network.

There are some minor extensions that can be made to improve the prototype developed, for example would it be a good idea to extend all consistency checks on network level to be applied on a certain predefined domain and calculate network performance on each individual domain.

The work presented here can also be used to extend the application, or to create an entirely new application, that focuses on other areas of IP networks, such as routing or security. It is possible to automate any task, which would make configuration in heterogeneous IP networks much easier, and with less risk of errors, than it is today.
References
[12] Nichols et al, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474
[14] Heinanen J. et al, Assured Forwarding PHB Group, RFC 2597
[16] Braden R. editor, Requirements for Internet hosts – Communication layers, RFC 1122
[17] Almquist P., Type of service in the Internet protocol suite, RFC 1349
[18] Ramakrishnan K. et al, The addition of explicit congestion notification (ECN) to IP, RFC 3168
[24] SIU-01 Product Description, Ericsson AB, 2009-09-08, 181/1551-LZA 701 0001 Uen
[26] Holmgren T, OMS 1400 Optical Multiservice Metro-Edge, Ericsson AB, 2009-02-23, 221 02-CSA 113 007/1 Uen
[27] IP RAN and MINI-LINK, Ericsson Internal presentation
[34] Snir Y. et al, *Policy Quality of Service (QoS) information model*, RFC 3644
[42] Brown A., Hellerstein J., *Reducing the cost of IT operations - Is automation always the answer?*, HotOS X, Tenth workshop on hot topics in operating systems
Appendix A - Abbreviations

ACL Access Control List
AF Assured Forwarding
AIPCM Abis over IP Configuration Management
AS Autonomous System
BER Bit Error Rate
BSC Base Station Controller
BTS Base Transceiver Station
CIM Common Information Model
CMI Common Management Information
CLI Command Line Interface
CRC Cyclic Redundancy Check
CS Class Selector
DF Default Forwarding
DiffServ Differentiated Services
DS Differentiated Services
EF Expedited Forwarding
IETF Internet Engineering Task Force
FQ Fair Queueing
FIFO First In First Out
GSM Global System for Mobile communications
GUI Graphical User Interface
IntServ Integrated Services
IP Internet Protocol
IPv4 Internet Protocol version 4
ISP Internet Service Provider
ITU International Telecommunication Union
LAN Local Area Network
MIB Management Information Base
MOF Managed Object Format
OAM Operations, Administration and Management
OSS Operations Support System
OSS-RC Operations Support System - Radio & Core
OS Operating System
PHB Per-Hop Behavior
PQ Priority Queue
PSTN Public Switched Telephone Network
QoS Quality of Service
RAN Radio Access Network
RBS Radio Base Station
RED Random Early Detection
RFC Request For Comments
RNC Radio Network Controller
RSVP Resource Reservation Protocol
SDH Synchronous Digital Hierarchy
Automatic configuration of QoS parameters in IP RAN

Henrik Hjalmarsson

SIU Site Integration Unit
SLA Service Level Agreement
SMI Structure of Management Information
SNMP Simple Network Management Protocol
SP Strict Priority
SSH Secure Shell
TCP Transmission Control Protocol
TDM Time Division Multiplexing
TOS Type of Service
UDP User Datagram Protocol
VLAN Virtual Local Area Network
VoIP Voice over IP
WCDMA Wideband Code Division Multiple Access
WFQ Weighted Fair Queueing
XML Extensible Markup Language