Helsinki University of Technology, Networking laboratory
Teknillinen korkeakoulu, Tieteverkkolaboratorio
Espoo 2005

IRoNet - final technical report

Mika Ilvesmäki, Raimo Kantola, Marko Luoma, Markus Peuhkuri, Samuli Aalto, Pasi Lassila and Heikki Hämmäinen

http://www.netlab.hut.fi/tutkimus/ironet/

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking laboratory

Teknillinen Korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Tietoverkkolaboratorio

Distribution:
Helsinki University of Technology
Networking laboratory
P.O.Box 3000
FIN-02015 HUT
Finland

Tel. +358 9 451 2461
Fax +358 9 451 2474

Contents

1 Introduction **1**

2 Prototyping **2**

2.1 Executive summary 2
2.2 Task goals 2
2.3 Task results 4
2.3.1 Prototype platform 4
2.3.2 Time control 6
2.3.3 Differentiated Services based network architecture 8
2.3.4 Policy control functionalities within DiffServ routers 10
2.3.5 Background traffic classification engine - LVQ pack 11
2.3.6 Real-time traffic classification engine - SIP signalling proxy . 12
2.3.7 Centralized routing engine 14
2.3.8 Delay adaptive scheduling 14

3 Traffic measurements and classification **17**

3.1 Executive summary 17
3.1.1 FUNET measurements 17
3.1.2 Traffic measurement analysis and classification 18
3.1.3 Analysis of FUNET data for traffic engineering purposes . . 18
3.2 Task goals 18
3.2.1 FUNET measurements 18
3.2.2 Traffic measurement analysis and classification 18
3.2.3 Analysis of FUNET data for traffic engineering purposes . . 19
3.3 Task results 19
3.3.1 FUNET measurements 19
3.3.2 Traffic measurement analysis and classification 20
3.3.3 Analysis of FUNET data for traffic engineering purposes . . 21

4 Traffic and network modeling **23**

4.1 Executive summary 23
4.2 Task goals 23
4.2.1 Multi-level TCP models 23
4.2.2 Dimensioning formulas for elastic data traffic 23
4.2.3 Insensitive load balancing 24
4.3 Task results 24
4.3.1 Multi-level TCP models 24
4.3.2 Dimensioning formulas for elastic data traffic 25

4.3.3 Insensitive load balancing . . . 25

5 Routing **26**
5.1 Executive summary . . . 26
5.1.1 Survey of different QoS routing approaches . . . 26
5.1.2 Multi Class Routing - MCR . . . 26
5.2 Task goals . . . 27
5.2.1 Survey of different QoS routing approaches . . . 27
5.2.2 Multi Class Routing - MCR . . . 27
5.3 Task results . . . 27
5.3.1 Survey of different QoS routing approaches . . . 27
5.3.2 Multi Class Routing - MCR . . . 27
5.3.3 Routing convergence and restoration . . . 28

6 Network Economics **30**
6.1 Executive summary . . . 30
6.2 Task goals . . . 30
6.3 Task results . . . 31

7 Conclusions **32**
7.1 Competence and Knowledge Building . . . 32
7.2 Technological Applications and Scientific results . . . 33
7.3 Deliverables . . . 33
7.4 Customer Satisfaction . . . 34
7.5 Benchmarking . . . 34
7.6 Educational Impact / Dissemination . . . 34
7.7 Wider societal impact . . . 35
7.8 Securing Possibilities of Networking Research . . . 35

IRoNet publications **36**

Chapter 1

Introduction

IRoNet studied the additional intelligence that is needed in the IP network in order to support Quality of Service. IRoNet scope covered the packet forwarding plane, the control plane and the management plane functionality needed in an IP network in order to fulfill the operator's popular vision of All-IP network and to provide a QoS enhanced Internet. The project used mathematical modeling of the forwarding plane mechanisms and the behavior of traffic streams, simulations of protocols and algorithms, traffic and performance measurements and prototyping of the mechanisms and algorithms particularly in the control and management planes.

This report is the final technical report to TEKES on the IRoNet–project that was carried out in the networking laboratory of the Helsinki University of Technology from 2002-2004. The report gives an overview of the completed and ongoing research and aims to give a summary of all the results that have been presented in the numerous publications. In addition, we aim to report the increase in knowledge in various areas of networking during the IRoNet–project The report is divided into five major and important areas of research:

1. Prototyping,
2. Traffic classification and measurements,
3. Analytical and theoretical work on traffic and network modeling,
4. Quality of Service Routing
5. Network Economics

All in all, IRoNet-project produced 48 different types of publications, including one doctoral thesis, one licentiate thesis, 11 master's thesis and 20 papers in international conferences. In addition, the IRoNet-project increased the quality and quantity of research and teaching in the Networking Laboratory at Helsinki University of Technology.

Chapter 2

Prototyping

2.1 Executive summary

Based on work presented in [1, 2] differentiation of traffic should be done based on application characteristics and that the number of scheduling classes should be between three and five.

In [3] the architecture and the performance evaluation of the policy agent in PC architecture was presented. The results were further elaborated in [4]. In addition we worked on synchronization [5] and produced several technical reports [6, 7, 8, 9, 10, 11] describing the functionalities of the proto-environment. A delay-based scheduling algorithm for adaptive provisioning and its evaluation by simulations and measurements in a prototype router network has been published in several conferences [12, 13, 14, 15].

The prototype -task was very software oriented. It produced roughly 50000 lines of code as shown in Table 2.1.

2.2 Task goals

Management of network resources and users is becoming more and more difficult for the network operators. This is largely due to changes that have happened in networks and the way the networks are used. Traditionally networks have provided a pure best effort service with a static binding of the user location and the provisioned access resources. However, this model is slowly changing. The change is partly due to integration of voice communication services and mobile terminals. Mobile terminals[1] require adaptivity from the network with respect to user profiles. These profiles describe the service offered to the particular user and should be installed to the point where the user is located when using the network service. Voice

[1] In this context mobile terminal means laptop or any other device which could be used to access network resources from variable locations but not moving while connected to the network.

Table 2.1: Software produced in the prototype –task

Software	Designer(s)	Code lines
Policy Control Agent	Piia Töyrylä	3896
Policy Server modules	Jari Huttunen and Piia Töyrylä	871
ALTQ confguration GUI	Ondres Fialka and Jari Huttunen	5340
Flow classifier and aggregator	Sampo Kaikkonen and Marko Luoma	26348
SIP proxy control module	Emanuel Schmid	1475
Centralized routing server	Sampo Kaikkonen	615
HPD scheduling algorithm	Antti Paju	3082
Testing tools	Jari Huttunen	932
Modified ALTQ stat	Jari Huttunen	5709

communication on the other hand requires strict delay control from the network. Legacy IP networks do not provide delay control. This control can be achieved with differentiation of packets e.q., separation of voice packets from the stream of packets to a separate service class.

Differentiation can also be used to achieve other goals than delay control of VoIP traffic. It can be used to minimize the interference that is natural for different transport (UDP/TCP) and application protocols (file transfers vs web browsing). Differentiation is then more related to general management of network resources. With good planning, differentiation can be a tool for a network provider to maximize the carried traffic on a given infrastructure. However, planning is in a key role in this case. Planning is the decision process in which a selected set of traffic components (applications) are grouped together. This aggregation can be done based on past knowledge of traffic components or based on the current situation. Past knowledge is gained from the measurements of network traffic. Measurements are conducted in a manner such that properties of traffic components can be analyzed with a clustering algorithm. If the aggregation is done based on the current situation, the knowledge of the traffic components is gained from the signalling between the network and the users. This is, however, not a viable solution for the majority of traffic components and should not therefore be the only means to gain knowledge of the traffic.

Having proper differentiation methods and knowledge of the traffic components are not enough to provide quality of service for the traffic. We also need multi-class scheduling algorithms to make the actual difference between service classes. Multiclass scheduling is a two-fold action: i) it provides a tool to isolate one traffic component from another, ii) it can deteriorate network performance if there is a mismatch between the offered traffic and provisioned class resources. In real networks there is always a mismatch in scheduling properties. This is due to the fact that offered traffic in packet networks is never smooth nor has it any meaningful steady-state during the contention situations. Making a scheduling algorithm that matches the resources to the offered traffic is a task that has also pros and cons. Matching the resources to the offered traffic means that the network tracks the offered traffic and provides resources to traffic components needing it. Without policies in the network the operation returns the good old FCFS operation which provides absolutely no quality. However, with a realistic policy this can work. The policy can be for example sequential i.e., secure delay of voice traffic after which keep tolerable delay for web surfing but give as much capacity as possible for file transfers. How to implement this is then another question. We have devised an adaptive scheduling algorithm supporting the policy of proportional delays between traffic classes.

To summarize, our goal was to create a prototype network which has the following two major capabilities:

1. Roaming policies for the users
2. Delay control for real-time communications

These goals where targeted with the following actions:

1. Development of prototype platforms
2. Time synchronization of networked systems
3. Differentiated Services based network architecture
4. Policy control functionalities within DiffServ routers
5. Background traffic classification engine
6. Real-time traffic classification engine - SIP signalling proxy
7. Centralized routing engine
8. Delay adaptive scheduling

A descriptive picture of our network and its components is presented in figure 2.1. It clarifies the components, their interaction and what are the essential parameters that make the system adaptive in each region.

2.3 Task results

2.3.1 Prototype platform

Our prototype has two development platforms:

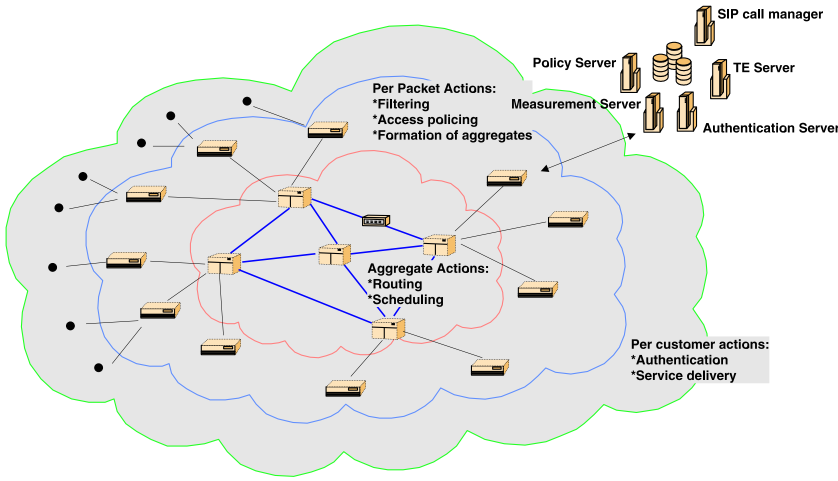


Figure 2.1: The big picture of our work

1. PC based routers
   - AMD 1.3GHz processor
   - 4 pcs 10/100 Ethernet interfaces
   - FreeBSD 4.5 operating system
   - alternate queueing (ALTQ) traffic management software
2. Necsom media switches (MS)
   - 10/100 Ethernet interfaces having
     - Motorola network processors
     - 128MB memory
     - Linux 2.14 operating system
   - Frame Synchronized Ring (FSR) backplane

The PC based platform is flexible and easy for developing new software features. However, the performance is bounded by the capacity of the PCI-bus and the central processor. The 33MHz PCI-bus offers an aggregated data rate of 133MBps. In case of network devices the same data passes the PCI-bus twice meaning that the actual performance of the PCI-bus is less than 66MBps (500Mbps). This is currently enough for our routers which use only four 10/100Mbps Ethernet interfaces. Processor performance limits the capacity in IP-layer operations of data traffic. In our case the processor is used for implementing all traffic management aspects: traffic classification, forwarding and queue management actions. With the PC-based routers the forwarding rate is typically less than 100kpps which is roughly 50Mbps when using small IP-packets, and 1.2Gbps when using large IP-packets. The capacity of the PCI-bus allows only 40kpps when large IP-packets are used.

The Necsom platform is not flexible but it scales to performance values which are not possible by using single bus/processor architecture. The FSR backplane

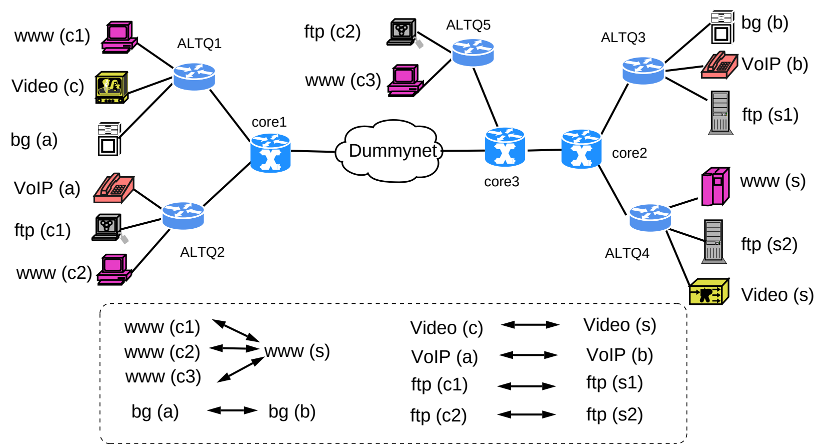


Figure 2.2: Measurement network topology.

in media switches has the capacity of 1Gbps which is used by line cards. Each line card is actually a processing card having a Motorola PowerPC based network processor and a separate operating system. Traffic classification, forwarding and scheduling is distributed across router interfaces and, therefore the architecture does not have a bottleneck. Software development in this architecture is, however, difficult due to the lack of development support within the router itself. Debugging of software errors can not be done as there are no means to take system snapshots or even print error information to any external device. Therefore, only a subset of features have been developed for the Necsom media switches.

Devices from both of these platforms form a network with the total of 16 routers. However, when all of the software functionalities are tested only the PC-based systems are used. This is due to the lack of a policy control agent in media switches. The current network topology (only PC routers) is presented in figure 2.2.

Traffic generation in the test network is based on open source software and dedicated measurement devices. Traffic components used in testing are: VoIP, QuickTime Streaming Video, WWW, FTP and Raw UDP. Because our test network is located in one rack we need to emulate realistic network delays. This is an important aspect when stateful TCP traffic is generated into the network. We do this by adding additional network impairment emulators (Dummynet) to the links on the network.

As a result of general test network procedures we have developed two functional network prototype environments where we have integrated all the software components which are explained in the following sections. Also a technical report [9] explaining the test network operations and measurement methods was produced.

2.3.2 Time control

Accurate clock synchronization is needed for studying e.g., a packet flow through the prototype routing network as a function of time. Commodity PC hardware, while applicable as prototype router platform, does not provide accurate time-keeping nor time stamping because of low-cost components and a general purpose design.

Dedicated network traffic analyzers can locally measure a packet flow against time with high accuracy using a specialized and high-cost design. They do, however, often lack the possibility of performing measurements between two physically separate locations. Packet time stamping, when done on integrated circuit level, provides a reliable way to monitor network packet flows. An operating system level implementation of time stamping on PC hardware introduces several error factors which may affect obtained results. In the worst case scenario some packets seem to be received before even being sent because of the clock offset between two computers. It is evident that accuracy and stability of monitoring device's clock is crucial when aiming for reliable results. Accurate timing information can be applied e.g. in measurement based control algorithms or in measuring the exact behavior of some mechanism or some network level control property.

On PC hardware the up keeping accuracy of the clock is mainly dependent on used computer architecture, design and operating system, and also on the load the CPU and other system resources are experiencing. Network Time Protocol (NTP) utility provides methods for synchronizing computer's clock to the accuracy of tens of microseconds if the pulse-per-second signal is enabled. If synchronization is realized over the network, the achieved accuracy may decrease to the level of a few milliseconds. The main obstacle of determining the exact inter-computer clock offset is the fact that there is no physical signal which phase could be measured or compared between computers. Therefore only upper limit estimates can be obtained for the inter-computer clock offset. This is caused by the software-oriented nature of computer's system clock up keeping.

Our study shows that it is not possible to reach better inter-computer synchronization than tens of microseconds even if sub-microsecond accurate pulse-per-second signal is provided for each unit via a serial port. Developed time synchronization system is shown in figure 2.3. Therefore, we have developed additional hardware that would enable sub-microsecond clock synchronization accuracy even if off-the-shelf PC hardware is used.

To minimize processing latency, temperature and CPU load dependent error factors we have developed a card into a PCI bus which contains 100 MHz VCO, PLL, two high resolution counters and two direct interfaces to a GPS receiver. The main idea of this approach is to dynamically adjust the time increment on each timer interrupt by measuring the elapsed time between two successive timer interrupts. By forcing the system clock time to elapse correctly on each interrupt there is less need for actual synchronization. Needed modifications are implemented on the Linux-2.4.20-NANO kernel. With this new system we can achieve stable and good estimation for clock error, see figure 2.4 for an example of time error plot in case of two successive compilations for the purpose of loading the CPU. This work has produced a Master's Thesis [5] and three versions of HW/SW for GPS based time synchronization system.

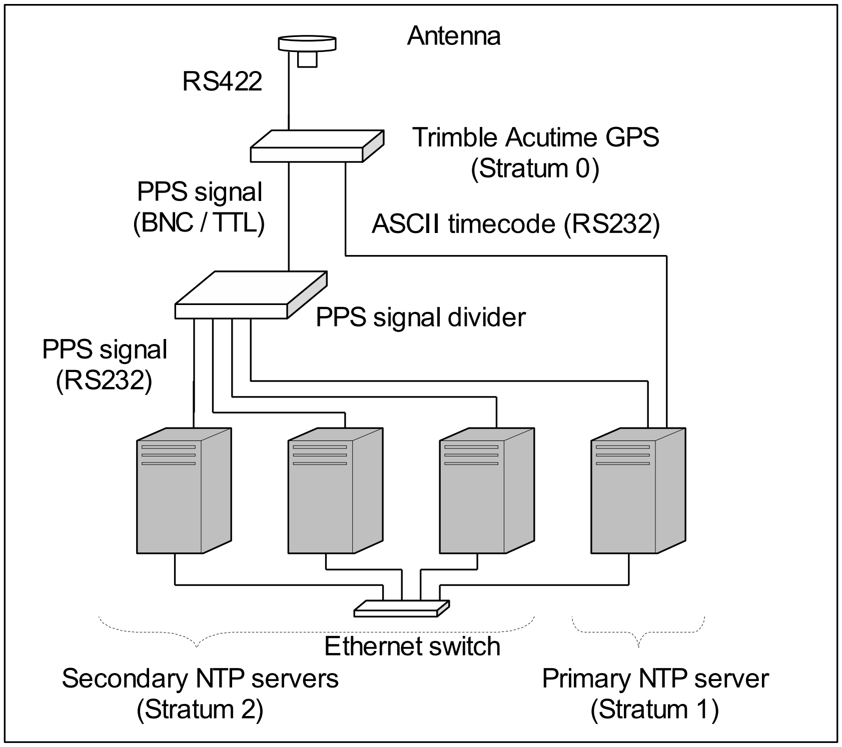


Figure 2.3: Architecture of GPS based time synchronization unit

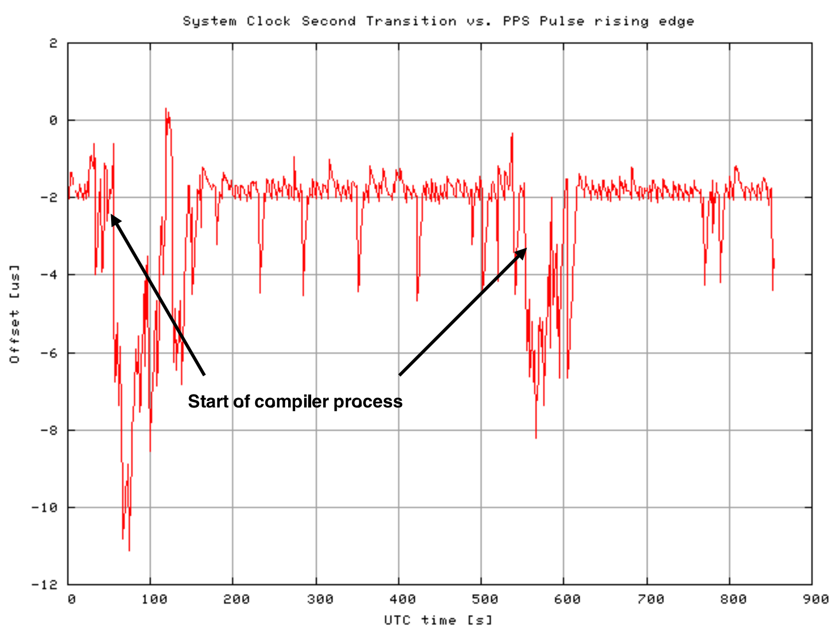
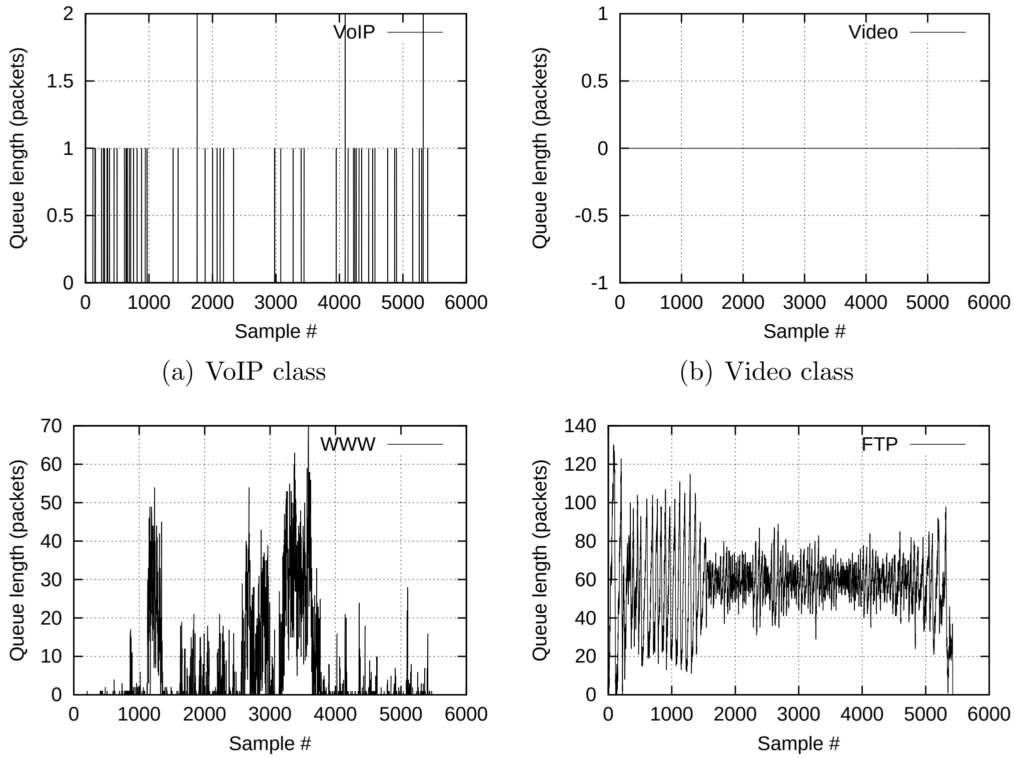


Figure 2.4: Clock offset from the PCI-based synchronization system in heavy I/O situation

2.3.3 Differentiated Services based network architecture

We were interested to see how traffic should be differentiated in order to maximize network utility. Utility is defined as the maximum utilization while still maintaining the minimum service characteristics for different applications. Differentiation



(a) VoIP class

(b) Video class

(c) WWW class

(d) FTP class

Figure 2.5: Queue lengths for traffic classes in the four class model at the bottle neck link

has an effect on this utility by controlling the interference that happens between different applications. There is an inherent interference in the traffic caused by the mixture of i) real-time traffic with strict delay requirements and bursty non-real-time traffic, and ii) short and long TCP flows. This interference has different magnitudes depending on the way differentiation is done. Differentiation requires several scheduling classes which, depending on the scheduling algorithm, cause lower performance on cases when there is a mismatch between provisioned resources and offered traffic.

This issue was investigated in Master's Thesis by Jari Huttunen [1]. In this thesis different differentiation methods along with levels of differentiation were studied in a prototype environment based on PC routers running the FreeBSD operating system and the ALTQ package to produce DiffServ functionalities. In this work we came to the solution which supported our previous simulation studies e.g., differentiation should be done based on application characteristics and that the number of scheduling classes should be between three and five. Three in the case when only real-time and non-real-time traffic are separated (+ extra class for network control traffic). Five in the case when we differentiate between small and large packets size flows on the real-time side, and short and long flows on the non-real-time side. The scheduling algorithm used in this study was CBQ which is one form of policy controlled multiclass scheduling. Queue lengths of different scheduling classes are shown in figure 2.5 These results were further elaborated in

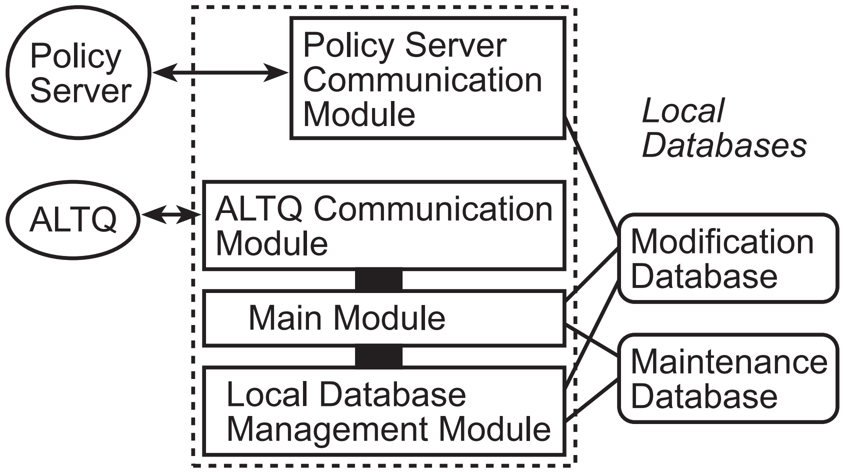


Figure 2.6: Architectural overview of Policy Control Agent

a conference paper [2].

2.3.4 Policy control functionalities within DiffServ routers

Communication between routers and centralized management servers in our adaptive, policy-based network prototype is based on a management software called a Policy Control Agent residing in every edge router. The Differentiated Services functionalities in the network are realized with the ALTQ traffic management software and the purpose of the Policy Control Agent is to automatically reconfigure ALTQ when the traffic conditions change. The Agent software communicates with the Policy Server databases which contain the information on the configuration changes, and reconfigures ALTQ according to the parameters it receives from the Policy Server. The Agent also maintains local databases which store the current ALTQ configuration parameters and the new, not yet installed parameters. The architecture of the agent and its internal communication is shown in figure 2.6.

The Policy Control Agent has an important role in the implementation of user mobility. When the user logs in to the network, his SLA profiles are installed into the edge router he is connected to and when he leaves the network, the profiles are deleted from the edge router. Thus, it does not matter through which edge router the user logs in to the network, he always has his own SLA profiles available. The Agent software takes care of installing and deleting the profiles in the edge router, according to the information it receives from the Policy Server.

There are clear benefits in using the Agent instead of reconfiguring the routers manually: the configuration time is only a small fraction of the manual configuration time and no misspellings occur. For the undistorted operation of the network, it is crucial to keep the reconfiguration time as short as possible. In the performance tests, we measured the time-consumption of different ALTQ reconfiguration operations, parameter data handling and the management of the local databases.

Although the results did not reveal any fatal bottlenecks, they showed clearly that the inefficient implementation of the local databases and the number of context switches between the Agent and ALTQ need improvement. An efficient implementation of such policy control is one prerequisite for implementing an IP network that supports both user mobility and quality of service.

The routers in the network prototype are built on general PC hardware which naturally affects the overall performance of the routers. However, the measurements proved that the Agent, although being the first working prototype implementation, is capable of fast configuration and is therefore ready for network-wide testing and further development.

The development of the Agent produced one Master's Thesis [3] and one conference paper [4]. In the Master's Thesis the architecture and performance evaluation of the agent in the PC architecture was made. The results were further elaborated on the conference paper. In addition, the operation of the network prototype and the Agent at the same time has been presented in two public demonstrations during the IRoNet project.

The policy server and our configuration tool for managing policies was documented in two technical reports [8] and [10].

2.3.5 Background traffic classification engine - LVQ pack

The idea of a background traffic classification engine is to provide filters which are inserted into the network routers with the aid of a Policy Server and Policy Control Agents. Our background traffic classification engine is based on the theoretical work carried by Mika Ilvesmäki in his PhD thesis [16]. Theoretical system developed in his study was put in a real test network by integrating two tools:

1. Coral Reef measurement and analysis software from CAIDA
2. LVQ PAK from HUT[2]

Coral Reef is a measurement and analysis software developed for general Internet traffic analysis. It contains modules for traffic capturing from different low level interfaces, such as Berkeley packet filters or DAG cards, and traffic analysis/reporting modules. We use only a single (heavily modified) analysis module from the Coral package and pass the data from Coral into the LVQ software which does the actual classification. LVQ stands for Linear Vector Quantizing and is one of the methods of supervised learning developed by Teuvo Kohonen.

The whole system is implemented so that traffic within the core network is split to the analysis computer which gathers the data for a predefined amount of time after which it runs the traffic analysis. As a result of the analysis, a list of protocol

[2] *http://www.cis.hut.fi/research/lvq_pak/*

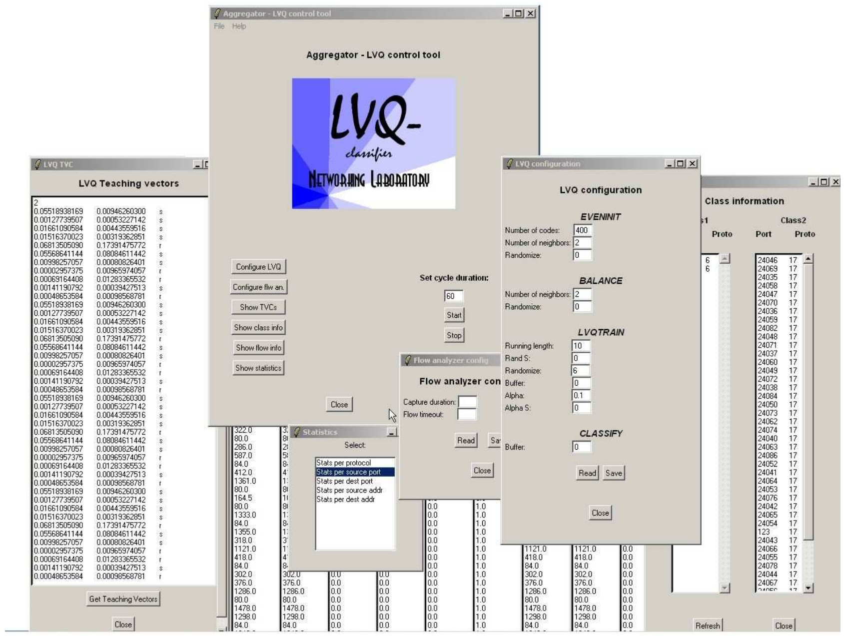


Figure 2.7: A snapshot from the GUI for LVQ

and port numbers are generated for each traffic class. These lists are injected into the Policy Server which then serves these lists to routers with the aid of the Policy Control Agent. This process runs in loops so that Coral is measuring the traffic continuously and LVQ-software calculates the class lists after each interval has finished.

Management of the LVQ pack software and the whole system was simplified with the development of two software tools:

1. A GUI for generating ALTQ configuration files[10]. This GUI can also be used for analyzing the content of the Policy Server in general.
2. A GUI for controlling the LVQ pack configuration files[11]. This GUI makes it possible to create teaching vectors and other configuration data, seen in figure 2.7

2.3.6 Real-time traffic classification engine - SIP signalling proxy

The idea of a real-time traffic classification engine is to provide the possibility for high quality VoIP calls within the network. This is accomplished by having: i) a high priority class which is only used for this type of communication and ii) a real-time classification engine which intercepts SIP messages within the network. A simplified diagram of the operation is shown in figure 2.8. The arrows are labelled

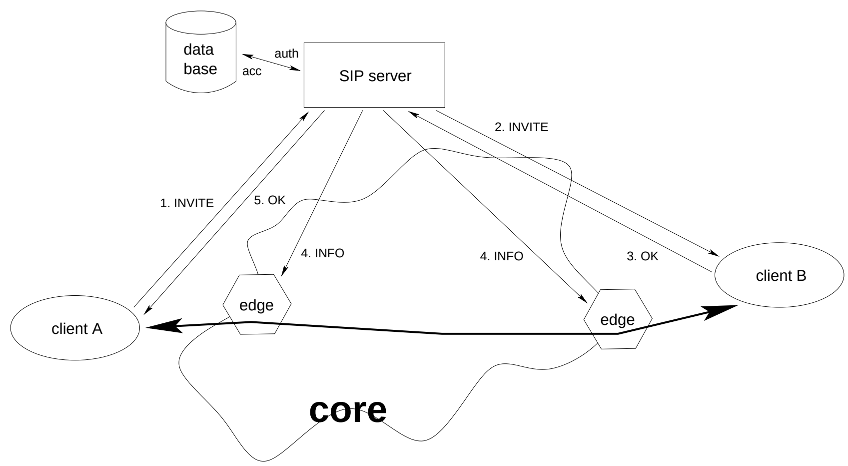


Figure 2.8: Network overview.

and numbered according to a SIP INVITE transaction. Short description of the actions follows:

- First the calling client sends an invitation to the central SIP server, which authenticates (proxy authentication) and accounts the call to a MySQL database.
- The invitation is then forwarded to the called client that either accepts or refuses the call.
- This answer message has to pass the SIP server again, which, if the call is accepted, generates an INFO message for each involved edge router. The INFO message contains the necessary data to install the filter rules that enable the actual call between calling parties and assign the proper treatment to the voice flow.
- Finally the answer of the called client is sent to the caller to complete the transaction.

The data needed in the edge router is collected from the SIP header and the SDP (Session Description Protocol) packet in the body of the first INVITE message and the OK reply. The name of the profile the filter belongs to is the same as the username, which can be found in the authentication digest of the INVITE message. The ID string for the installed filter is the call ID, which can be found in the header of all SIP messages related to this call. All network configuration data (source and destination IP address, source and destination port) are transferred in the SDP packet and read from there. All this data is locally stored.

To end the call the SIP server sends another pair of INFO packets to the edge routers when it receives a BYE message. The necessary data to remove the filters (profile name and filter ID) are found in the package header (Call ID) and in the locale database (username). This sequence is also stored into the database for billing purposes making it possible to use time based charging for VoIP calls.

A technical report [7] about the implementation and configuration of SIP based real-time classification was produced. Performance evaluation and scalability analysis are to be done later in following projects.

2.3.7 Centralized routing engine

The goal of a centralized routing engine is to create a system where routes for each and every router are calculated on the centralized routing engine which has complete knowledge about network status and offered traffic. This makes it possible to calculate routes with other algorithms and criterions than in conventional distributed routing. This has been implemented with two main components: i) zebra routing software on network routers, and ii) QRS routing simulator in the centralized routing engine.

The idea is to have zebra running in every router of the network so that the calls by the zebra daemon to the routing socket write-function is disabled. That means that the zebra itself cannot modify the routing tables. However, other functionalities of the zebra are running and it therefore creates and updates the OSPF Link-State DataBase. The LSDB is read periodically to the QoS Routing Simulator (QRS) which calculates the routes for every router. These routes are sent to each router which install these routes to the routing tables using the routing socket interface.

To be able to calculate routes that have distinctive characteristics, other than distance knowledge of the network and its devices are needed. QRS was developed for studies in quality of service routing where each routers capabilities are important in determining the best possible route. We have not used these capabilities in this work but provided a tool in which these capabilities can be edited in an efficient manner. Snapshot from the tool is in figure 2.9. To make routing really adaptive these parameters should be partly inferred from the network status. However, this is possible in our system due to measurement based scheduling which keeps track of the situation on each link.

This work is partly finished, but the final assessment on the scalability of the system has not yet been done. Nevertheless, we can conclude that the system is operable and produces routing tables for a network of 50 routers in a fraction of a second. This system has a limitation that it can only handle networks having a single OSPF area i.e. all links and their status must be known to the simulator.

2.3.8 Delay adaptive scheduling

The main goal of this research was to develop a delay-based scheduling algorithm for adaptive provisioning and to evaluate it by simulations and measurements in a prototype router network. Work on adaptive scheduling has been published in several conferences [12, 13, 14, 15].

First, we compared the performance of static and adaptive provisioning methods

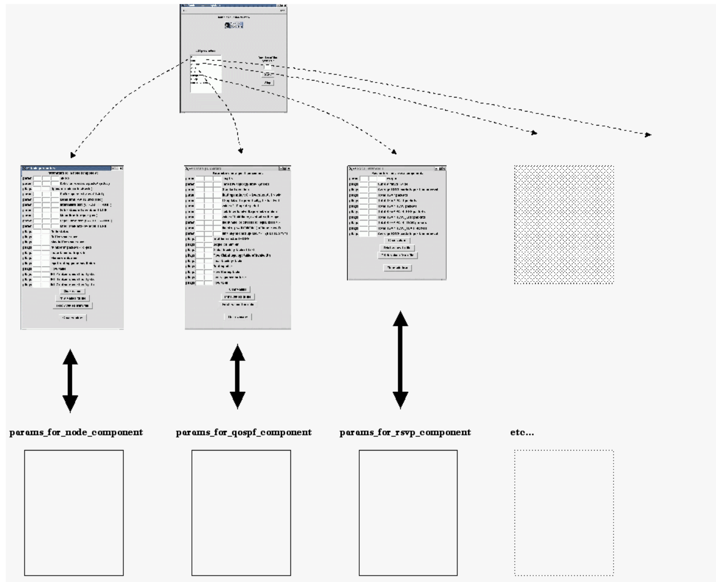


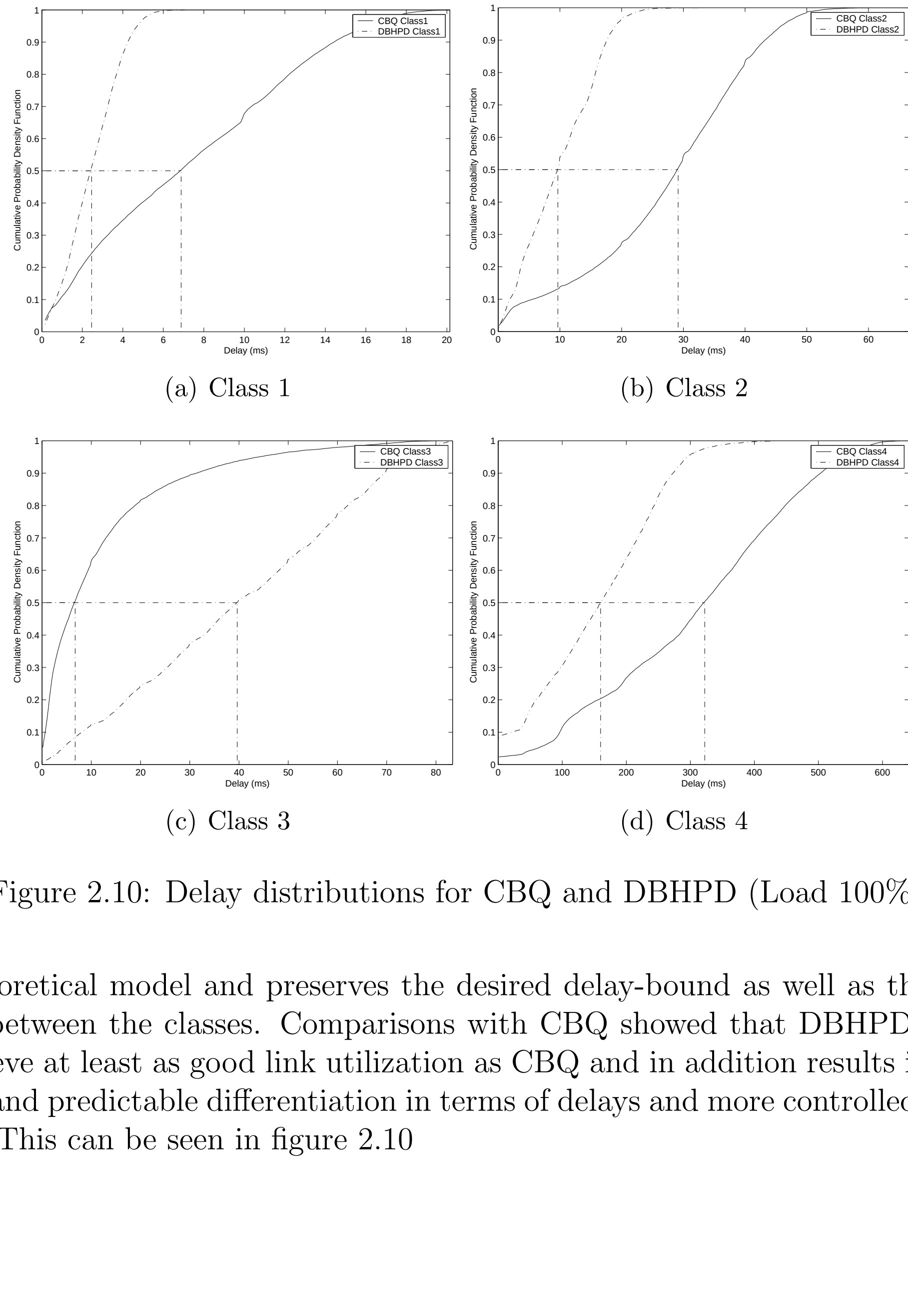
Figure 2.9: Tool to edit node parameters in QRS

with ns2-simulations in order to see what kind of performance advantage can be achieved by adaptivity. For the static provisioning case we used capacity based Deficit Round Robin (DRR) and for adaptive provisioning delay-bounded Hybrid Proportional Delay (DBHPD). Next, we proposed three new packet delay estimators for an adaptive, delay-bounded HPD (DBHPD) scheduling algorithm in the DiffServ context: a simple Exponential Weighted Moving Average (EWMA) estimator, an EWMA estimator with restart (EWMA-r) and the EWMA based on proportional error of the estimate (EWMA-pe). We compared these estimators with the original, simple sum estimator with ns2-simulations using several traffic mixes. Finally, we developed the first working implementation of the DBHPD algorithm in a FreeBSD-based ALTQ prototype router and presented measurement results of the DBHPD implementation with FTP, HTTP, Video Streaming and VoIP traffic in underload, overload and heavy overload conditions.

We showed that DBHPD algorithm operates well according to the theoretical model and preserves the desired delay-bound as well as the delay ratios between the classes. We also compared DBHPD to an existing Class-Based-Queueing (CBQ) algorithm that is widely used both in research and in the industry.

According to ns2-simulations performed in a network setup the delay-bounded HPD algorithm is better able to achieve the targeted provisioning goal than the static DRR algorithm regardless of the load level, application mix or queue management method used. The simulations performed with the new estimators showed that the simple sum and EWMA estimators often lead to false scheduling decisions. On the other hand, the EWMA-r and especially the EWMA-pe estimator provide good estimates of the packet delay regardless of the traffic mix.

The implementation results proved that the algorithm operates well according to



(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

Figure 2.10: Delay distributions for CBQ and DBHPD (Load 100%)

the theoretical model and preserves the desired delay-bound as well as the delay ratios between the classes. Comparisons with CBQ showed that DBHPD is able to achieve at least as good link utilization as CBQ and in addition results in much better and predictable differentiation in terms of delays and more controlled packet losses. This can be seen in figure 2.10

Chapter 3

Traffic measurements and classification

3.1 Executive summary

3.1.1 FUNET measurements

Traffic measurements are invaluable tool to study application, protocol and network behavior for private networks and for the Internet. The largest shortcoming has been unavailability of long traffic traces. Most of available traces are one hour in length and some of those are quite old not reflecting current internet traffic.

In the measurements area we invested into traffic capture installation now connected to FUNET on 2.5Gbit/s [17]. The capture software has been debugged and we are now storing clean IP traffic traces on full speed as the packet are transferred on FUNET links. We are making the arrangements to archive the traces on CSC servers and analyze them on CSC computers. Care is being taken to make sure we comply with the data privacy regulations. We ensure confidentiality by removing user data and by anonymizing lowest byte of IP addresses in the capture equipment prior to intermediate storing. During the project we were not sure that these arrangements would be possible due to data privacy reasons and therefore we achieved a major milestone. Our plans are to continue to store a significant amount of data and analyze it on a regular basis.

The measurement system on FUNET 2.5 Gbit/s backbone link became finally operable, after both technical and policy-related problems were solved. The measurement system itself has operated flawlessly, only temporal problems with analysis and storage computers has caused loss some of statistical measurement data. At the time, there are more than 70 days worth of packet traces stored on CSC archive server using 8 TiB of storage capacity.

3.1.2 Traffic measurement analysis and classification

IRoNet saw the maturing of our traffic classification methodology based on analysis of traffic measurements and application of the learning vector quantization (LVQ) algorithm. The work could be said to have started with the licentiate thesis of Markus Peuhkuri [18] which analyzed our laboratory traffic. This work culminated in Mika Ilvesmäki's doctoral thesis [16]. Prior to the doctoral thesis we published a conference paper [19] and also opened up a new research direction in analyzing flow inter-arrival times [20]. Together with the prototype network –task, we developed a methodology to synchronize hardware clocks in a network [5].

3.1.3 Analysis of FUNET data for traffic engineering purposes

Riikka Susitaival and Ilmari Juva analyzed the FUNET traces from the traffic engineering point of view. The analysis took place in Autumn 2004 and resulted in a conference paper [21] that appeared in Spring 2005. After the end of the IRoNet project, the analysis has been continued and deepened in Networking laboratory's other research projects.

3.2 Task goals

3.2.1 FUNET measurements

The initial idea of these measurements emerged in connection without work on QoS Routing. We understood that the feasibility of QoS or Class based routing depends on two aspects. One is the efficiency of the Routing mechanisms for their purpose as such. Another is the variability of Internet traffic in time and space. If the variations in space take place in timescales on which routing can react to those changes, routing can help in assuring the quality or balancing the load of the network. If the variations are of shorter duration, then only nodal mechanisms can be used.

3.2.2 Traffic measurement analysis and classification

We expanded the research by utilizing more and different analysis methods and examining other measured properties from the packet stream. Our main focus in the latter will concentrate on packet inter-arrival times and packet length distributions.

3.2.3 Analysis of FUNET data for traffic engineering purposes

For Internet traffic engineering purposes, it is important to characterize traffic volumes typically over 5-minute intervals. Based on measurements made in a local network at Lucent in winter 1999, Cao et al. [1] proposed a moving IID Gaussian model for the characterization of 5-minute traffic volumes, with a power-law relationship between the mean and the variance. Our goal was to analyze the Funet traces gathered in Summer 2004 and investigate the validity of the moving IID Gaussian model and the proposed mean-variance relationship when the measurement interval is varying from 1 second to 5 minutes.

3.3 Task results

3.3.1 FUNET measurements

Towards the end of the project all administrative and legal issues were solved and we started to analyze the data. During IRoNet we established a functional measurement infrastructure (see also [18] for methods on compressing securely the traffic traces) that is able to produce us with vast amounts of traffic data. This will enable us to continue using up-to-date traffic measurements in our research. The architecture of the measurement setting is shown in Figure 3.1

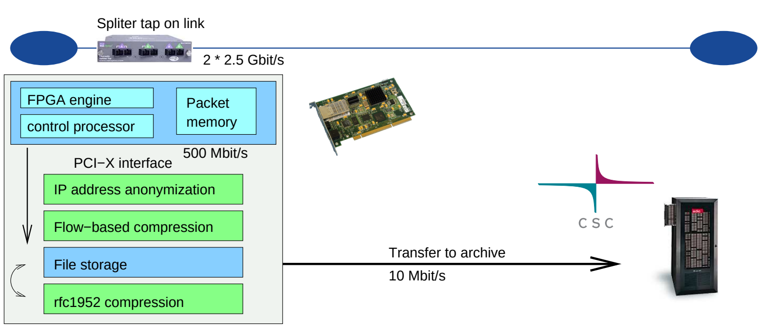


Figure 3.1: Measurement architecture

As measurement system was deployed on late phase of IRoNet project, not much of analysis were yet completed, but provides a good starting point for future projects. The measurement task provided valuable input for future measurement projects:

- The measurement device must have sufficient local storage to buffer possible networking problems or problem with backend analysis and storage servers.

[1] J. Cao, D. Davis, S. V. Wiel and B. Yu, "Time-varying network tomography," *Journal of the American Statistical Association*, vol. 95, pp. 1063–1075, 2000.

Current system has local capacity for three to four days that is not enough if severe hardware problems took place on analysis.

- Archival system must be balanced to support sustained demand more than 100 GiB daily and close to 1 TiB weekly volume. There must be enough disk storage to buffer temporal overloading of near-line storage.

The measurement activities on gigabit-class networks are very storage-intensive: various analysis need a large working area for data analysis starting from terabyte-class disk systems. Analysis needs also a large physical memory to keep all data structures in memory.

3.3.2 Traffic measurement analysis and classification

The work done in [18], in addition to basic traffic analysis results, produced a method to compress securely traffic packet traces. Some results of the traffic analysis are shown in Figure 3.2.

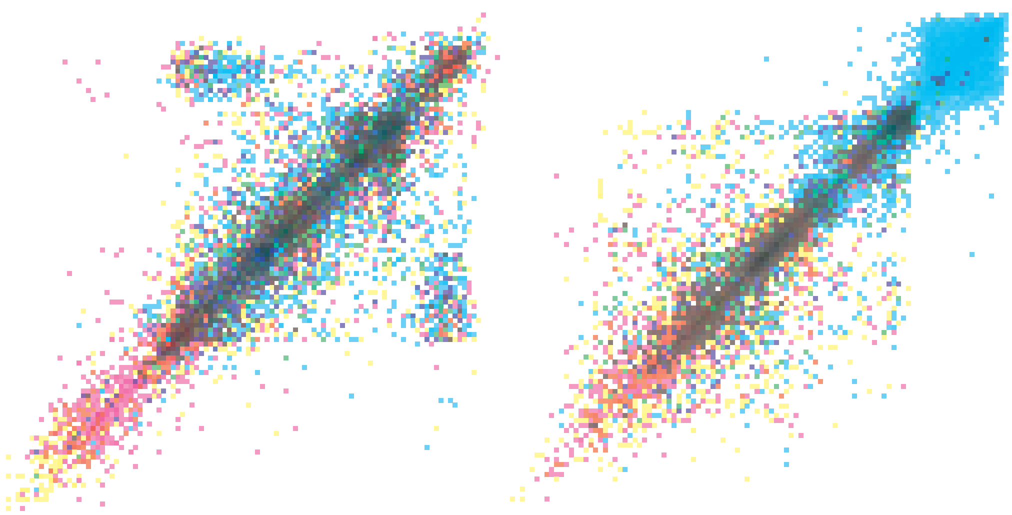


Figure 3.2: Bandwidth of probe pair in phase density plot. Probe intervals are represented by colors: cyan is for flows less than 10 seconds intervals, magenta for 10–60 s and yellow for 1–5 minutes. The darker the color component, the more there are occurrences. Dataset SRC on the left (110 B/s – 4 MB/s) and DST on the right (220 B/s – 5.4 MB/s), logarithmic scale.

Our traffic classification method produced in Ilvesmäki's doctoral thesis in [16] provides a clear methodology for grouping applications that seem to behave in a similar manner into a class. As an example, from Figure 3.3 we can see that packet or flow data as stand alone measurements do not reveal the application behavior. One can also observe that similar applications tend to position themselves in the packet/flow space the same way in different network environments.

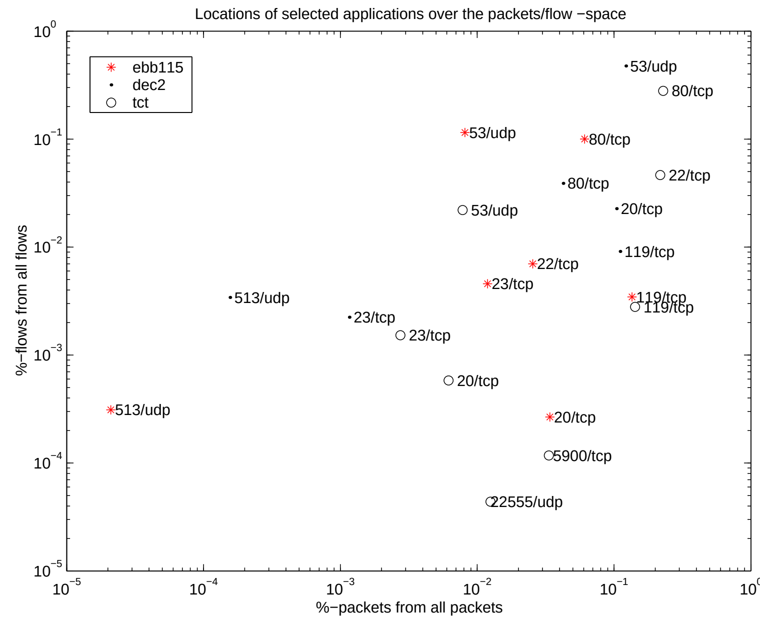


Figure 3.3: Application locations in logarithmic packet/flow -space

3.3.3 Analysis of FUNET data for traffic engineering purposes

As a result, we found that the Gaussian assumption is much more justified with current core link rates. The mean-variance relationship seems, indeed, to follow a power-law with exponent approximately equal to 1.3 in the FUNET data set. However, the IID assumption concerning the standardized residual was not verified, but we found a clear positive correlation between adjacent 5-minute volumes, and only slightly weaker negative correlation for traffic volumes with distance 20-30 minutes. Figure 3.4 shows the variances as a function of the mean at the log-log scale, along with the lines depicting the best functional fit for original one second sample interval, and for aggregated measurements with $\Delta = 30, 60, 300$ s.

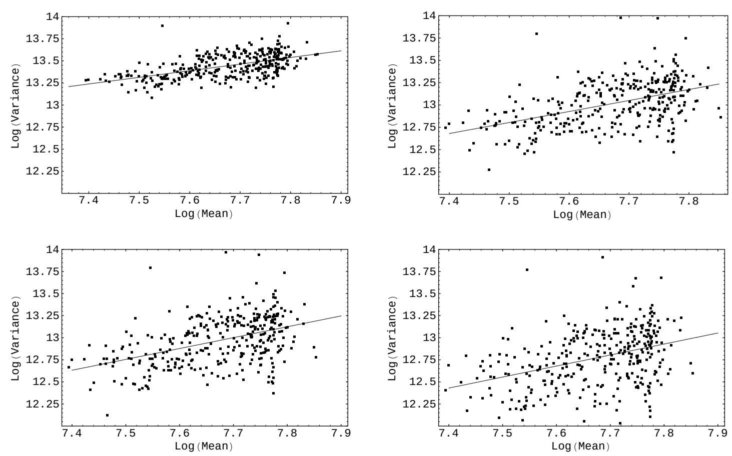


Figure 3.4: Mean Variance relation scatter plot and best functional fit for FUNET data for sample intervals of one second (top left), 30 seconds (top-right), 60 seconds (bottom left) and 300 second (bottom right).

Chapter 4

Traffic and network modeling

4.1 Executive summary

From the point of view of a user, the performance of data networks, such as the Internet, carrying elastic traffic is manifested at the flow level. Performance analysis at the flow level involves modelling a system where flows arrive randomly and have random lengths, and the simultaneously on-going flows share dynamically the resources of the network. In this task several studies have been made related to the flow level performance of data networks. The results have been published as three conference papers [22, 23, 24] and one technical report [25]. The research has been carried out by Pasi Lassila, Juha Leino, and prof. Jorma Virtamo.

4.2 Task goals

4.2.1 Multi-level TCP models

In the Internet, TCP is responsible for realizing the sharing of resources in the network. Processor sharing (PS) models for TCP behavior nicely capture the bandwidth sharing and statistical multiplexing effect of TCP flows on the flow level. However, these 'rough' models do not provide insight into the impact of packet level parameters (such as the round trip time and the buffer size) on, e.g., throughput and flow transfer times. In [22] and [23], the aim was to develop combined packet and flow level models for the mean file transfer delays of TCP flows sharing the bandwidth of a single bottleneck link.

4.2.2 Dimensioning formulas for elastic data traffic

The recently introduced notion of balanced fairness, generalizing the single link PS model to a fair sharing model covering the whole network, provides an elegant abstraction of the dynamic bandwidth sharing at the flow level that can be used to approximate even the fairness realized by TCP. For models based on balanced fairness, the performance can efficiently be computed for networks of realistic size. In addition, these models lead to robust performance measures that are completely independent of detailed traffic characteristics, which is a very desirable feature. This property is called insensitivity. In [25] our target was to utilize the models based on balanced fairness for dimensioning IP access networks, where the dimensioning criterion is based on the mean file transfer delay (or equivalently the mean throughput) of a customer (user).

4.2.3 Insensitive load balancing

More theoretically oriented work on insensitive routing and load balancing (traffic splitting) based on the notion of balanced fairness was done in [24]. In this work, the aim was to extend the notion of balanced fairness and define routing and resource allocations of different flow classes such that insensitivity of the system is still retained.

4.3 Task results

4.3.1 Multi-level TCP models

The results in [22, 23] show that it is possible to derive simple analytic models for these delays that are reasonably accurate across a wide range of system parameters (RTTs and buffer size). We developed PS type models, but they were modified to take into account impact of packet losses and different RTTs. The models were validated against simulations using the ns2 simulator.

Figure 4.1 illustrates the results. The left figure depicts the dynamic behavior of the mean number of flows in the system as a function of time, when the system starts empty, i.e., it gives the length of the transient period. The solid line is our analytical model (under the assumption of exponential file lengths) and the blue line corresponds to its validation with simulations using exponential file lengths. The fit is reasonably good. The other lines correspond to simulations with Pareto-distributed file lengths with different parameters showing how the initial transient becomes longer for heavy-tailed file lengths. The right figure illustrates the accuracy of the stead-state model in a system with access rates of 1 Mbps and the bottleneck link rate equals 10 Mbps, and a buffer size of 10 packets. The solid lines depict simulated results and the solid lines are from our models. The figure gives the mean file transfer delay as a function of load, and shows that accuracy at high loads is better for large RTTs.

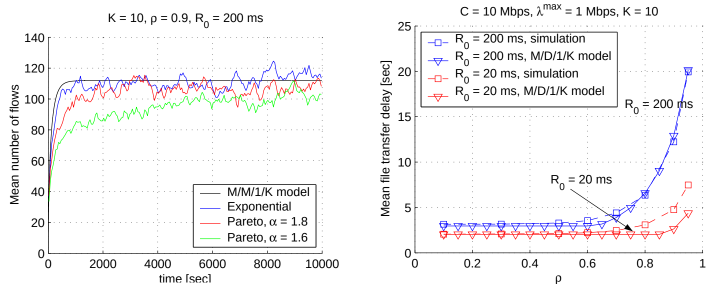


Figure 4.1: Length of the transient period for different distributions (left) and the mean file transfer delays for various parameters (right).

4.3.2 Dimensioning formulas for elastic data traffic

Network dimensioning for circuit switched networks based on the celebrated Erlang's formula has long been a well understood problem. However, dimensioning for elastic data traffic is still an area that is not very well developed. In [25] we defined the dimensioning problem for a tree-type access network as an optimization task where the link capacities are chosen such that costs are minimized subject to per-flow constraints on the minimum flow level throughput. These throughputs were calculated utilizing the robust flow level models based on the notion of balanced fairness.

4.3.3 Insensitive load balancing

In [24] we considered the specific case of a data network where each flow can be routed on one of a set of alternative routes. By using the linear programming formulation of MDP theory we were able to analyze optimal routing policies that utilize the full global state information. In the ordinary LP formulation of MDP theory, the global balance condition appears as a linear constraint on the decision variables. In order to retain insensitivity, we imposed stricter detailed balance conditions as constraints. As a further extension, the MDP-LP approach allows joint optimization of the routing and resource sharing, in contrast to the earlier work where the resource sharing policy was required to be separately balanced and fixed in advance. According to our first experiments, the advantage given by global state information is in this case negligible, whereas the joint routing and resource sharing gives a clear improvement. The requirement of insensitivity still implies some performance penalty in comparison with the best sensitive policy.

Chapter 5

Routing

5.1 Executive summary

5.1.1 Survey of different QoS routing approaches

Quality of service in IP networks requires many different mechanisms that may work in different timescales starting from packet level, flow and session level or on a timescale determined by routing convergence time. As background research and preliminary studies of the existing research in regards to QoS routing and its applications in DiffServ-networks, several Master's thesis were completed during the project [26, 27, 28, 29, 30, 31, 32, 33, 34].

5.1.2 Multi Class Routing - MCR

We explored the approach of QoS routing to aid in achieving quality differentiation. Our approach is called Multi-Class Routing (MCR) [35, 36, 37, 38, 39] in a DiffServ type of architecture, to maximize the performance of each class. In MCR each class has its own routing table that is produced using a routing algorithm and metric that is determined for that particular class. Different classes can use different metrics and different routing algorithms. In addition, since the links are common for all classes, distribution of link state information is a common task for all classes.

In addition to QoS routing, work has been done on analyzing MPLS traffic engineering mechanisms and load balancing [40, 24] and studying the convergence of intra-domain routing [41].

5.2 Task goals

5.2.1 Survey of different QoS routing approaches

The goal of this task was to survey different Quality of Service routing approaches based on existing literature.

5.2.2 Multi Class Routing - MCR

The study of multi class routing (MCR) has concentrated on two aspects: performance and scalability. The so-called inter-class effect is serious if there is no protection of classes against each other. Thus, it is difficult to meet performance criteria of all classes unless some protective measures are taken. In particular under single class best effort shortest path routing it is easy for high priority classes to destroy the performance of a lower class.

5.3 Task results

5.3.1 Survey of different QoS routing approaches

The results of this task were reported, among other thesis, in Ilmari Juva's MSc thesis [29]. The main points of the survey were as follows. Different path selection problems with one or two metrics were formalized in a systematic way as optimization problems, and solution algorithms were discussed. In addition to path selection, which is just one piece in the whole QoS routing, different strategic approaches, like the choices between source routing and distributed routing, or between pre-computation of paths and on-demand computation, were addressed. Thirdly, some problems arising from QoS routing were identified, and possible solutions were discussed. Finally, different evolutions of QoS routing found from the literature were reviewed.

5.3.2 Multi Class Routing - MCR

First, we found that a routing approach, for example MCR, can largely alleviate the inter-class effect. Through simulation study, our algorithm is proved to be simple and effective. We also implemented a partial prototype of multiclass routing based on Zebra. The prototype allowed exploring the needed modifications to OSPF in order to support MCR and to study the problems of co-existence of MCR with single class routing. The prototyping also provided input to the evaluation of different approaches to MCR implementation i.e. distributed static, distributed dynamic and centralized. The prototype is able to produce a routing table for each class. However, due to the monolithic nature of the kernel, only one routing table can be active at a time. It remains a problem for further study how to best implement the rest of the MCR. Also we aim to explore other routing approaches for solving the inter-class effect.

Figure 5.1 shows an example use of the MCR in a differentiated services network.

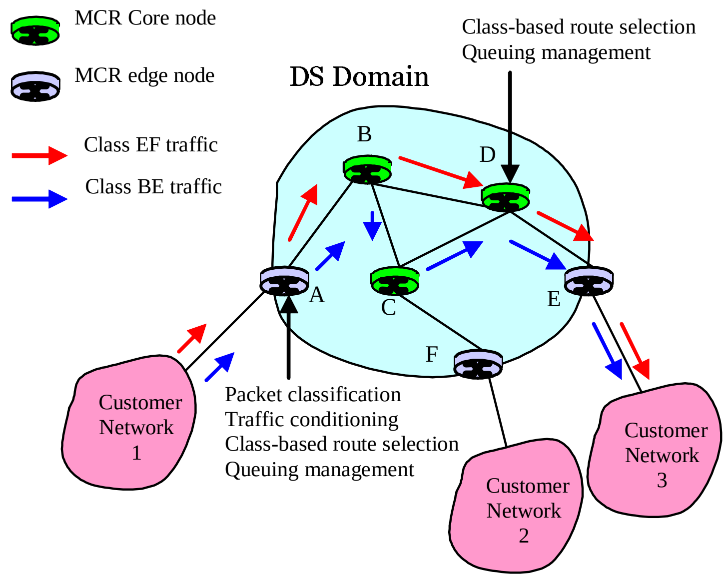


Figure 5.1: An example of the MCR DiffServ network

5.3.3 Routing convergence and restoration

In QoS aware networks, routing scalability should be carefully examined. In our study, we explore several approaches to solve the scalability problem. We see that MCR tries to optimize network performance and quality provided to users. Any recalculation of routes requires that all routers converge to coherent routing tables based on the same link states. If link states depend on traffic volume, they will change often. However, routing convergence takes time. During the period of convergence packets can be lost or they may loop for a long time creating a long tailed delay distribution. If routing recalculation for optimization purposes takes place too often, packet loss increases and the QoS seen by the users will degrade. Also the goodput of the network will decrease. As a result, there is an optimal frequency for performing recalculation of routes for optimization purposes. The gains of routing must be balanced against packet loss. The problem of convergence can be tackled by fine tuning the routing protocol (with OSPF this allows to come from tens of seconds with default timer values to just below one second in convergence time), by better router architectures (we have tested some commercial routers that seem to achieve 100 ms convergence time) or by developing new link state distribution algorithms.

In the last category, we studied a new link state update algorithm called Fish Eye - Link state update (FS-LSU) in order to apply QoS routing into a large-scale network without causing too much cost. (The term Fish Eye routing is adopted from an Ad hoc routing algorithm.) The algorithm mainly aims to reduce the distribution cost of link state. Another approach, that we have studied, looks at two different routing scenarios, pre-computation and on-demand, to study their impact on network performance with the increase of network size. Our preliminary study shows that the scalability problem can be alleviated with the help of these approaches but not completely solved. In a paper we also suggested using fast multicast for distributing link state changes.

Another important question in improving the quality provided by IP networks to users is fast link restoration from failures. One commercial approach is manual configuration of alternative label switched paths in MPLS networks. For example in case the network is supporting a large number of VPNs, clearly this approach scales poorly. In a paper we suggested a multipath routing protocol that creates multiple paths to each destination. Thus in case of failures, on detection restoration is immediate.

Chapter 6

Network Economics

6.1 Executive summary

Our network economics task focused on designing, implementing and testing the first prototypes of a mobile operator business game called MOB. The result is documented as a master's thesis [42] and a conference paper [43]. In addition, a smaller effort was put on studying the pricing for IP service quality using the evolution of voice service as an example. The result is documented as a conference paper [44].

6.2 Task goals

The MOB business game was developed to initiate a process for collecting, structuring and disseminating techno-economic understanding of new mobile services. Originally two mechanisms were planned for creating new understanding:

1. Techno-economic research tasks (independent of MOB) and
2. Interactive game sessions with senior telecom experts (based on evolving MOB prototypes).

The main target group for dissemination was the telecom students at Helsinki University of Technology. Since developing a business game is a big effort, the assumption was to use the existing general-purpose business game by Juuso Tyli as a stepping stone.

The pricing of IP service quality, and voice quality in particular, was studied to better understand the user's willingness to pay for service quality. This pre-study was based on literature, open interviews, and constructive scenario building.

6.3 Task results

The first MOB prototype built by Juha Kokko was ready in December 2003 and it was tested with students in game sessions run as part of a graduate seminar. Two other business games were also used in this seminar in order to benchmark MOB. Results are documented in Kokko's master's thesis [42]. The student feedback and comparison with other games enabled fast creation of the second MOB prototype which was then tested with the students of Networking Business Course in April 2004. At this point the MOB tool and the related teaching process were mature enough to get very good feedback from students. By now, MOB has become an established teaching tool after several game sessions with players ranging from young students to senior industrial telecom experts. MOB has clearly achieved the original goal of becoming a delivery platform for research results. However, the other goal of using MOB sessions for creating new techno-economic understanding does not seem plausible [43].

Our pre-study on voice-over-IP service identifies three main evolution paths: PSTN-like voice (public connectivity), PBX-like voice (enterprise connectivity), and IM-like voice ("tribes" connectivity). Each path is IP-based, but differs significantly regarding the requirements for pricing and quality. The PSTN-like IP voice is based on open standards and connectivity between IP and PSTN worlds, which includes similar requirements for pricing and technical quality. The PBX-like IP voice is a feature-rich internal voice service for enterprises and replaces the old PBX. The necessary voice quality in closed networks can be reached using over-dimensioning, but the issue of mobile access remains open. The IM-like IP voice exploits the best-effort nature of Internet and avoids any network-based quality features as well as pricing for quality. All three paths are rapidly adopting the benefits of wireless access. The demand forecasting of these three paths was left for further study.

Chapter 7

Conclusions

7.1 Competence and Knowledge Building

We have built competence in internetworking research allowing us to verify any new result or an idea in packet forwarding, that have been achieved by simulations, in the proto environment in a few weeks. In this area this competence is unique in a Finnish University and rare in Europe. The prototyping laboratory uses PC hardware on the edge and dedicated open LINUX routers in the core. The software base includes Zebra, ALTQ on FreeBSD and Linux Traffic Control. Due to extensive measurements, we have gained additional insight into the workings of different QoS mechanisms and identified and corrected bugs in them.

We refined analysis of traffic measurements and subsequent traffic classification methodology using an approach that is independent of the particular protocols that are used to carry the traffic. This is important because our methods are insensitive to encryption applied by users and because it may be politically incorrect to make traffic management decisions based on content of messages.

We have conducted extensive simulations in the areas of packet forwarding and routing. In IRoNet, and partially in its sister projects COST279 and FIT, we have developed a simple and robust adaptive load balancing mechanism, which based on on-line link load measurements takes gradual corrective actions to redistribute the traffic in the network. Our overall routing architecture includes Multi-Class routing (MCR) and fast convergence. We have developed some initial ideas on millisecond convergence of link state intra domain routing.

Summarizing we can say that any QoS architecture is against the alternative of over-provisioning. Better QoS techniques especially suitable for different access networks are needed. Even a single household or a single user will want to minimize the disturbance its different applications cause to each other. In the core, over-provisioning may be more cost efficient.

7.2 Technological Applications and Scientific results

Our traffic classification method [16] provides a clear methodology for grouping applications that seem to behave in a similar manner into a class, assign one DiffServ code point to the class and apply one per-hop-behavior to all applications in one class. We believe that traffic classification on such premises can be a multipurpose tool for network operators trying to better understand what is happening in their networks and trying to make network management decisions based on the gained knowledge.

Our prototype classifies applications with similar traffic behavior into a class. It assigns appropriate class treatment properties to each of the applications used by each user of the network. The prototype is able to change the users traffic filters on the fly. This is much better than what most commercial systems can do.

The purpose of grouping similar applications into one class is to reduce the disturbance quite different applications can cause to each other and achieve more predictable quality of service for each application. In IRoNet we have found additional evidence in favor of the approach both using additional simulations and by measurements in the prototyping environment.

In IRoNet we continued to develop the Quality of Service Routing Simulator (QRS) and used it in particular for the study of different link state distribution algorithms. We also integrated the QRS with our prototype.

Our focus in end-to-end modelling has been on developing analytical models for performance evaluation of TCP bandwidth sharing. We have developed models that are based on so called processor sharing (PS) taking into account the impact of packet losses and different Round Trip Times. The models have been validated against simulations using the ns2 simulator. Our modified PS models are able to increase the accuracy in predicting the performance of TCP.

In cooperation with the FIT project we have worked on the problem of insensitive routing and load balancing (traffic splitting) based on the relatively new notion of balanced fairness (BF). BF leads to performance measures that are completely independent of detailed traffic characteristics, which is a desirable feature.

7.3 Deliverables

This final report has a list of all publications on different forums that were produced in IRONET. The report also has a list of the substantial amount of software that was produced for prototyping purposes. An example is a modification to the LINUX kernel that makes the Linux traffic control useful (earlier the TC was there but the way it used to communicate with the Network Interface card made it largely useless). Some of the software are at least valuable tools for research.

Some may also be useful in an operative context in an ISP.

7.4 Customer Satisfaction

Many ideas that we pursued in our packet forwarding and routing prototyping work have been adopted for commercial use by Tellabs. The measurement probe connected to FUNET continues to give us valuable traffic information. It also provides operational benefits to CSC. In IRoNet we created a Mobile Operator game in which business teams play the Mobile Internet market. This work lead to two spin-off projects during 2004.

7.5 Benchmarking

IRONET focused on an area that is technologically rather mature and in which the research and development leadership clearly resides in the US from the inception of IP technology till the present day. The area is also flooded with a lot of research that has a loose relationship with the realities of networking. By focusing on an experimental approach we tried to make sure that we verify our theoretical results at least in a prototype. In some spear head areas we did world class (and certainly unique in Europe) research in IRONET. IRONET has been instrumental in raising the overall level of networking research in the Networking Laboratory onto a significantly higher level where we were when the project was started. We believe that we will be better recognized for our contribution in the International Research community when we publish all the software that was produced in IRONET. The fact that a Doctoral thesis was finalized in IRONET also speaks of the level of the work.

7.6 Educational Impact / Dissemination

Several courses of the Networking laboratory greatly benefitted from IRoNet. A special course on IP network measurements and measurement analysis is starting in the spring of 2006 at TKK. The mobile operator business game is used in teaching of Networking Business at TKK and as a vehicle for discussion in research. IRoNet results have been disseminated in annual project seminars to which people were invited through the NETS mailing lists.

7.7 Wider societal impact

IRoNet-project was carried out during a time when the telecommunications and ICT industry went through a significant change in their working environments. This change and its effects were pondered in [45, 46, 47, 48]. Some of the publications [46, 47, 48] were written largely based on discussion in the Broadband thematic group in the NETS program in which we took an active part.

7.8 Securing Possibilities of Networking Research

Our understanding of the nature of network traffic is a prerequisite for developing efficient algorithms for network nodes. We are now collecting traffic trace information from FUNET. We remove the content of the packet and store only headers after scrambling the IP addresses that can be viewed as user identification information. The traces are owned by CSC and access to them is limited by a contract that each researcher signs with FUNET. By this arrangement be ensure the interests of user protection against privacy violations. The telecom privacy law however is open to different interpretations. We see it important for the future of any experimental work on developing better mechanisms for networks to make such provisions in the law and the instructions on applying the law that the kind of work we are carrying out is allowed to continue.

IRoNet publications

[1] Jari Huttunen. Measurements on differentiation of internet traffic. Master's thesis, Networking Laboratory, Helsinki University of Technology, January 2005.

[2] Marko Luoma and Jari Huttunen. Differentiation of traffic on access networks. In *Proceedings of MSAN 2005*. IEEE, June 2005.

[3] Piia Pulkkinen. Design and implementation of a policy control agent for a differentiated services router. Master's thesis, Networking Laboratory, Helsinki University of Technology, May 2004.

[4] Piia Pulkkinen and Marko Luoma. Performance measurements of a policy control agent for a differentiated services router. In *Proceedings of IASTED PDCS 2004*, pages 902–907, November 2004.

[5] Antti Gröhn. Clock synchronisation of a computer test network (testiverkon kellosynkronointi). Master's thesis, Networking Laboratory, Helsinki University of Technology, October 2004.

[6] Jari Huttunen. Database implementations. Technical report, HUT, Networking Laboratory, August 2002.

[7] Emanuel Schmid. Connection signaling for a dynamic VoIP network with SIP. Technical report, HUT, Networking Laboratory, August 2004.

[8] Jari Huttunen. MySQL database support forALTQ GUI. Technical report, HUT, Networking Laboratory, February 2004.

[9] Jari Huttunen. Network measurements - tools and methods. Technical report, HUT, Networking Laboratory, January 2005.

[10] Ondrej Fialka. GUI for generating ALTQ configuration files. Technical report, HUT, Networking Laboratory, August 2004.

[11] Sampo Kaikkonen. Development of a GUI for LVQ pack. Technical report, HUT, Networking Laboratory, January 2005.

[12] Johanna Antila and Marko Luoma. Scheduling and quality differentiation in differentiated services. In *Proceedings of MIPS2003*, pages 119–130, November 2003.

[13] Marko Luoma and Johanna Antila. Differentiation of the internet traffic. In *Proceedings of ICN2004*, pages 505–512, February-March 2004.

[14] Johanna Antila and Marko Luoma. Adaptive scheduling for improved quality differentiation. In *Proceedings of MIPS 2004*, November 2004.

[15] Johanna Antila and Marko Luoma. Robust delay estimation of an adaptive scheduling algorithm. In *Proceedings of QoS-IP 2005*, February 2005.

[16] Mika Ilvesmäki. *On traffic classification and its applications in the Internet.* Doctoral dissertation, May. 2005.

[17] Markus Peuhkuri. Mittauksia FUNETin runkoverkosta. *@CSC*, (2):25–26, 2005.

[18] Markus Peuhkuri. Internet traffic measurements - aims, methodology and discoveries. *Licentiate thesis*, Helsinki University of Technology, 2002.

[19] Mika Ilvesmäki, Raimo Kantola, and Marko Luoma. Traffic differentiability based on packet and flow per application –analysis. In *Proceedings of Globecom 2003*. IEEE, IEEE, November-December 2003.

[20] Evgenia Daskalova, Mika Ilvesmäki, and Raimo Kantola. Analysis of flow inter-arrival time distributions. In *Proceedings of IASTED Euro-IMSA*, February 2005.

[21] Ilmari Juva, Riikka Susitaival, Markus Peuhkuri, and Samuli Aalto. Traffic characterization for traffic engineering purposes: Analysis of FUNET data. In *Proceedings of NGI 2005 (to appear)*, 2005.

[22] Pasi Lassila, H. van den Berg, M. Mandjes, and R. Kooij. An integrated packet/flow model for TCP performance analysis. In *Proceedings of ITC-18*, pages 651–660, Berlin, Germany, August 2003.

[23] Pasi Lassila and M. Mandjes. A multi-level TCP model with heterogeneous RTTs. In *Proceedings of Networking 2004*, pages 52–63, Athens, Greece, May 2004.

[24] J. Leino and J. Virtamo. Optimal load balancing in insensitive data networks. In *Proceedings of QoS-IP 2005*, pages 313 – 324, February 2005.

[25] Pasi Lassila and Jorma Virtamo. Access network dimensioning for elastic traffic based on flow-level QoS. Technical report, Networking Laboratory, Helsinki University of Technology, Mar. 2005.

[26] Peng Zhang, Raimo Kantola, and Samuli Aalto. QoS routing for diffserv networks: Issues and solutions. Technical Report n/a, Networking Laboratory, Helsinki University of Technology, 2002.

[27] Tyh-Dar Fan. Policy management and its relationship to QoS mechanisms and case studies. Master's thesis, Networking Laboratory, Helsinki University of Technology, 2002.

[28] Wenpeng Zhou. Provision and route optimisation in diffserv networks. Master's thesis, Networking Laboratory, Helsinki University of Technology, 2002.

[29] Ilmari Juva. Analysis of quality of service routing approaches and algorithms. Master's thesis, Networking Laboratory, Helsinki University of Technology, mar. 2003.

[30] Xiaole Bai. Investigation and analysis of optimal multi-class routing algorithms for IP networks. Master's thesis, Networking Laboratory, Helsinki University of Technology, 2003.

[31] Yi Zhou. Investigation on the fish-eye state algorithm in context with QoS routing. Master's thesis, Networking Laboratory, Helsinki University of Technology, 2003.

[32] J. Agustina, Peng Zhang, and Raimo Kantola. Performance evaluation of GSM handover traffic in a GPRS/GSM network. In *Proceedings of IEEE ISCC2003*. IEEE, IEEE, June 2003.

[33] Haifeng Zhu. Comparison between pre-computation and on-demand computation qos routing with different link state update algorithms. Master's thesis, Networking Laboratory, Helsinki University of Technology, 2004.

[34] Marcin Matuszewski. Peer-to-peer music distribution. In *Proceedings of ICTSM12*, July 2004.

[35] W. Zhou, Peng Zhang, Xiaole Bai, and Raimo Kantola. A QoS based routing algorithm for multi-class optimization in diffserv networks. In *Proceedings of IEEE HSNMC'2003*. IEEE, IEEE, July 2003.

[36] Yin Wang. Adding multi-class routing into the diffserv architecture. Master's thesis, Networking Laboratory, Helsinki University of Technology, November 2004.

[37] P. Zhang, X. Bai, and R. Kantola. A routing scheme for optimizing multiple classes in diffserv networks. In *Proceedings of IEEE HPSR'2004*. IEEE, IEEE, 2004.

[38] Yin Wang, Raimo Kantola, and Shuping Liu. Adding multi-class routing into the diffserv architecture. In *Proceedings of the Advanced Industrial Conference on Wireless Technologies (ICW05) Conference*. CPS, May 2005.

[39] Yin Wang. User manual for MCRv1 prototype. Technical Report n/a, Networking Laboratory, Helsinki University of Technology, n/a.

[40] Riikka Susitaival and Samuli Aalto. Providing differentiated services by load balancing and scheduling in MPLS networks. Technical Report TD(03)03, COST279, Jan. 2003.

[41] Marcin Matuszewski, Raimo Kantola, and Risto Sarala. OSPF convergence and its impact on voip. In *Proceedings of ATNAC 2004*, 2004.

[42] Juha Kokko. Mobile operator business game. Master's thesis, Networking Laboratory, Helsinki University of Technology, January 2004.

[43] H. Hämmäinen, M. Tallberg, and J. Töyli. A business management simulation of mobile service competition. In *Experimental Interactive Learning in Industrial Management*. IFIP WG5.7, November 2004.

[44] T. Smura and H. Hämmäinen. The role of voip: Future evolution paths of voice communication. In *Proceedings of 1st International CICT Conference*, November 2004.

[45] Raimo Kantola. Tietotalouden kriisi on monisyinen. Talouselämä 34, October 2003.

[46] Raimo Kantola. E-talouskisassa ei ole dopingtestejä. Tietoyhteys 4/2004, 2004.

[47] Raimo Kantola. Näivettymisestä uuteen telepolitiikkaan. Helsingin Sanomat 15.11.2004, 2004.

[48] Raimo Kantola and Nisse Husberg. Pursiainen puolustaa pysähtyneisyyttä. Tietoviikko 14.10.2004, 2004.