Buffering Principles for Mobile Multimedia over IP

Masters Thesis
in Information and Communication
Technology

by

Johan Foldøy

Grimstad, May 28 2001

Abstract

This thesis suggests a test-bed for reviewing queue algorithms suitable for IP based mobile multimedia services. Subjective performance quality obtained in the test-bed was analysed using a network performance monitor tool.

In order to evaluate possible services for use in the test-bed, a comprehensive survey was conducted. The survey revealed that a variety of categorization methods for electronic services exist. However, none of them were targeted at mobile multimedia services. Based on the different methods a framework of six service classes were proposed. Together, these classes provide a useful and easy-to-navigate overview of both existing and future mobile services.

Most services on the Internet use TCP for end-to-end message transfer. It is reasonable to expect services in IP based mobile systems to use this protocol as well. TCP offers reliable connection management and is known as an extremely trustworthy protocol when used on wired links. Running TCP over wireless links is another story, though. Selected documents and reports that discuss TCP in wireless environments are evaluated, and proposed solutions are commented.

Two queuing algorithms were implemented in a router. Streaming video from a server on the Internet was routed to a host inside the selected test-bed. Run through Microsoft Media Player, the perceived quality of picture and sound was described. This description was then matched to a captured data flow from the same streaming session.

The comparison did not reveal strict relations between subjective experience and objective measurements. Possible explanations for this are discussed at the end of the thesis document.

Preface

This thesis project is part of the Master of Science (MSc) degree in Information and Communication Technology (ICT) at Agder University College. This last assignment is a closure on the education that lead to the title Master of Science.

I would like to use this opportunity to thank my supervisor at Ericsson in Grimstad, Lars Kjørsvik Schei, for valuable support. Thanks to Nils Ultveit Mo at Agder University College for useful assistance as well.

Last, but not least – thanks to my wife Helle for encouragement and understanding throughout the thesis period!

Grimstad,
May 2001

Johan Foldøy

Contents

Figures and Tables

1 Subject and basis for the thesis

1.1 Introduction

The term "Internet" and "mobile" have become part of almost everyone's vocabulary during the late 90's. Most people in the modern world have access to Internet either at work, home or both. Parallel to this evolution, network access while mobile has been a growing demand. Up to now, available technologies like Wireless Application Protocol (WAP) and Globale System Mobile (GSM) have not met this demand in a satisfactorily way. Universal Mobile Telephony System (UMTS) is designed to make mobile Internet access suitable for most services and bandwidth requirements. The UMTS Forum, an international organisation committed to the successful introduction and development of UMTS, recently published a report[1] which forecasts that revenues from 3G services worldwide will represent a cumulative market opportunity worth as much as $1 trillion from 2001 to 2010.

Among others, the report makes these predictions:

- Total operator-retained annual revenues of over $300 billion for 3G services by 2010
- In 2010 the average 3G subscriber will spend about $30 per month on 3G data services
- Non-voice service revenues will dominate voice revenues by 2004 and comprise 66% of 3G service revenues by 2010
- Asia Pacific represents the single largest total revenue opportunity - reaching $120 billion in 2010
- Europe and North America will provide the highest annual revenue per POP ($150 - $200 per year).

These predictions will only be met if users find the provided services interesting and worth paying for (in one way or another). Users will expect the same level of quality either they are mobile or connected to fixed lines. Such expectations challenge technical solutions in mobile systems. Effective use of radio resources and optimised use of signalling in the network are important goals for quality and economy.

1.1.1 The phases of Mobile systems

This is a simple classification of mobile systems development:

- 1st Generation: Nordic Mobile Telephone
- 2nd Generation: GSM
- 2.5 Generation: General Packet Radio Service (GPRS)
- 3rd Generation: UMTS

UMTS is a third-generation mobile system and the successor of today's GSM. International Telecommunication Union (ITU) is in charge for UMTS, which is also referred to as International Mobile Telecommunications (IMT-2000). Regional standardisation organisations throughout the world were invited to develop standards for the system. The proposal for UMTS's radio access, UMTS Transport Radio Access Network (UTRAN), came from European Telecommunications Standards Institute (ETSI). UTRAN was officially

[1] http://www.umts-forum.org/press/article064.html

accepted by ITU in the beginning of 1999, and ETSI started the development of UMTS, the European version of IMT-2000.

Several regional standardisation organisations joined ETSI in their work, and an international workgroup called 3rd Generation Partnership Project (3GPP) was established. This group commissioned the specification for version 1 of UMTS, referred to as "release-99". It is this version that will be deployed by operators during 2001[Ref. 1].

1.1.2 This thesis related to other work

This thesis document address several issues related to 3rd Generation Mobile Systems. It proposes a framework for service categorisation using six defined service classes. Hopefully, this framework can be useful for both vendors and operators when choosing future migration paths.

TCP over wireless links has been subject to extensive research for the last decade. The IEEE research environment has done throughout investigations on how performance can be tuned to the better. This thesis document reviews two different proposals for better achievements.

The practical testing focuses on whether different queuing algorithms produce different quality for users. If this should be the case, further research will focus on objective measures of data flow, in order to find connections between perceived and measured effects.

1.1.3 Thesis document structure

Chapter one is an introduction to the focus of this thesis project. It explains background, limitations and methodology used.
Chapter two provides some information on the protocol structure of Open Systems Interconnection model. Important notions used throughout the thesis document are defined and explained.
Chapter three defines the term multimedia.
Chapter four is quite comprehensive. IP and categorisation of services, together with a view of traffic classes for UMTS are central parts of this chapter.
Chapter five is dedicated to TCP and its rate adaptation capabilities.
Chapter six discuss buffering and queuing algorithms.
Chapter seven investigates the proposed test-bed for testing queuing algorithms.
Chapter eight reports on test results and discusses the returned results.
Chapter nine is an overall discussion on the most important findings in the document.
Chapter ten concludes on obtained results, and proposes some future issues that should be investigated.

1.2 Buffering Principles for Mobile Multimedia over IP

This is the initial thesis definition specified by Ericsson:

For UMTS there is a strong believe that multimedia will become one of the most popular services among customers. By running these services over IP, or equivalent, all the way to the user's terminal, it gets easier to adapt services to mobile users. For the interworking between different networks there are Media Gateways that deals with different data rates and protocols. This is done using flow control, buffering and dropping of data packets.

The focus is on the effect buffering and dropping of packets has on different multimedia services. Is it possible to find a principle that delivers good real-time properties and at the same time makes other services run efficiently, or should there be different principles for different services? Is it possible to detect the users demands by looking at patterns in the data flow?

Goals and activities:

- Get an overview of different IP based services that are suitable for mobile users, and their demands of real-time transfer and bandwidth characteristics
- What kind of requirements do the rate adaptation functions of TCP/IP put on the communication channel?
- Discuss the use of buffering and dropping of data in the interworking between different rates, possibly with a more thorough survey of a representative selection of services
- Implement a test set-up that makes it possible to try different principles, maybe in the shape of a short program for the algorithm itself, together with existing software and different terminal adapters and Ericsson Access Server
- Run the selected services through the test set-up. Is there an objective way to evaluate the results?
- Run a demonstration of the test set-up together with a presentation of the results

1.3 Limitations

The thesis is not concerned with pure voice over IP techniques and results. No practical testing of the UMTS radio interface is done; discussions are based on public reports that address TCP/IP running over wireless links.
Various Quality of Service (QoS) techniques like Differentiated Services (DiffServ), Resource Reservation Protocol (RSVP) and Integrated Services (IntServ) are not considered in this thesis document.
Demonstration of the test-bed will be done after this thesis is written. It is thus not a part of the documentation.

1.4 Methodology

The results in this thesis project were mainly based on studies of relevant literature. Articles and technical reports from Institute of Electrical and Electronics Engineers (IEEE) formed the basis for most discussions. The Internet was also used extensible throughout the project period. Literature from lectures held at Agder University College and international Universities provided background for discussions on general networking principles. Private, electronic correspondence with scientists from the IEEE society gave useful tips on where to find relevant material.

All practical tests were performed using standard and publicly available computers and software. The final test-bed was set up in a shared network at Agder University College.

2 Open Systems Interconnection Reference Model

2.1 Introduction

This thesis uses concepts and notions derived from the OSI. It is therefore vital that the reader is familiar with how the model is constructed and how it should be interpreted.

2.2 History

Representatives of major computer and telecommunication companies developed the OSI reference model in 1983. It was originally intended to be a detailed specification of interfaces. Instead, the committee decided to establish a common reference model for which others could develop detailed interfaces that in turn could become standards. The International Organization for Standardisation (ISO) officially adopted OSI as an international standard.

2.3 Description

OSI is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementers so that their products will consistently work with other products. The model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly consistent in terms of keeping related functions together in a well-defined layer, most products involved in telecommunication make an attempt to describe them selves in relation to the OSI model. It is also valuable as a single reference view of communication that provides everyone a common ground for discussion.

2.4 Protocol stack

The boundaries between each layer and the functions performed by each layer have been selected on the basis of experience from earlier standardization activity. Each layer performs a well-defined function in the context of the overall communication subsystem. The layer operates according to a strict set of rules – a protocol – by exchanging messages, both user data and control information. Data are exchanged with a similar layer in a remote system [Fig 1].

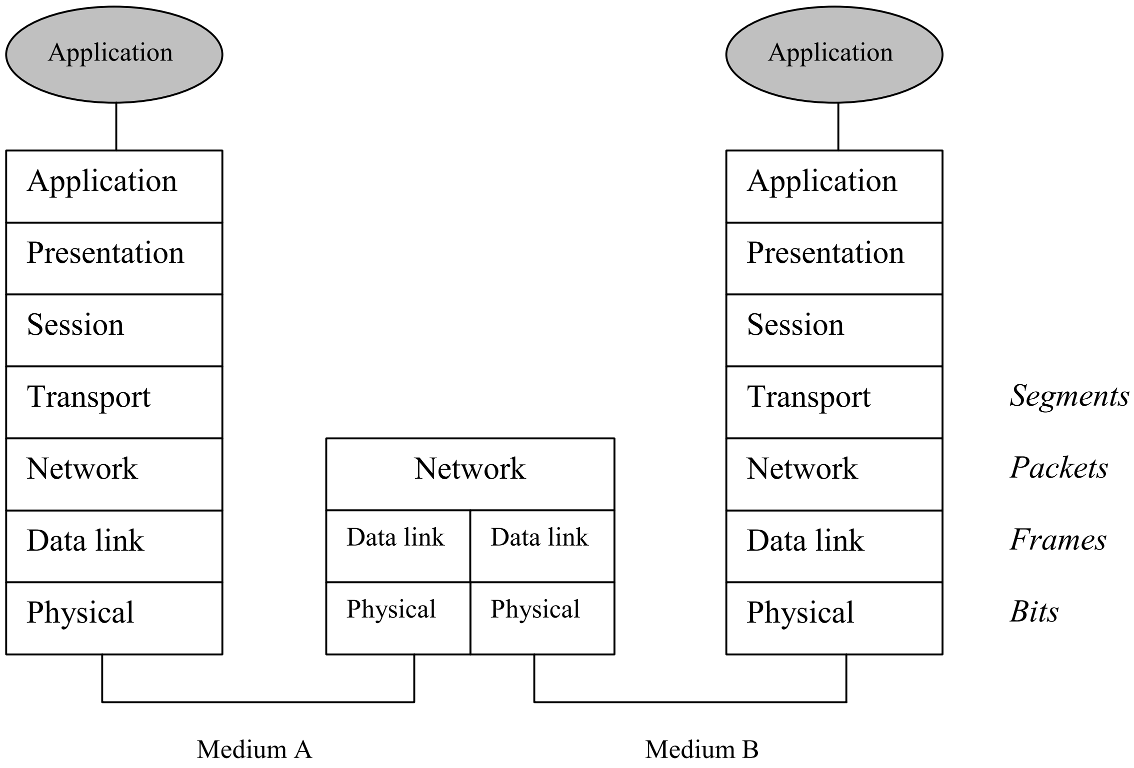


Figure 1 Open Systems Interconnection Reference Model protocol stack

The basic functions of each layer are, from bottom (layer one) to top (layer seven) [Ref 2]:

1. Physical layer: Mechanical and electrical network interface definitions.

2. Data link layer: Data link control (framing, data transparency, error control).

3. Network layer: Network routing, addressing, call set-up and clearing.

4. Transport layer: End-to-end message transfer (connection management, error control, fragmentation and flow control).

5. Session layer: Dialog and synchronization control for application entities.

6. Presentation layer: Transfer syntax negotiation and data representation transformations.

7. Application layer: File transfer, access and management, document and message interchange, job transfer and manipulation.

The lower right of figure one indicates name conventions for data at different layers.

3 Multimedia

3.1 Introduction

Multimedia is one of the hottest buzzwords in the technological community these days. The actual meaning of the word however, seems to be a bit vague. Reading both technical and non-technical articles, we find the term to be used in different contexts. One way to explain it would be "the use of multiple media to create a presentation", but from a technical point of view, this is not precise enough.

The objective of this chapter is to give a more thorough description of the term multimedia. It contains a description of different media, so the reader can get an idea of what formats and standards are most relevant up to now.

3.2 Media descriptions

Generally speaking, multimedia consists of four "pillars" [Ref. 3]. These are text, images, audio and motion picture (moving images without sound), as can be seen in Figure 2.

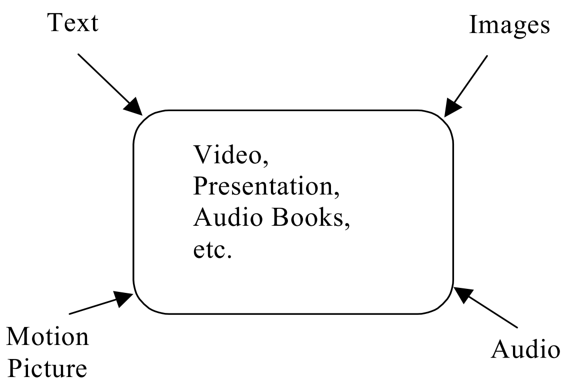


Figure 2 Basic media

Different ways of combining these four basic media evolves into more complex applications perceived by the user. One example is the combining of audio and motion picture into video. It also depends on the use of the files how they are stored and transmitted. A typical example would be the use of MP3 (MPEG-1 Layer 3). MP3 is a compressed (10-12:1) audio file, and its intended use is transmitting audio across the Internet. Due to its compression, transfer times are strongly reduced. But MP3 files cannot be put on a CD and played in a CD player. They only recognize Wave files, which is a different way of expressing the audio information within an audio stream, and is not compressed. Therefore it is not suitable for transmission across the Internet, but provides better quality than MP3.

3.2.1 Text

Most of the Internet today consists of text files. Apart from information in clear textual form, text is also used to describe media like motion pictures and audio (spoken text). Text files are deployed in many different formats. Most of these are proprietary which means that readers need special packets that can read the files. Two formats, however, are almost universal: ASCII and RTF.

ASCII (American Standard Code for Information Interchange) is the most common format for text files in computers and on the Internet. In an ASCII file, each alphabetic, numeric or special character is represented with a seven-bit number. Thus 128 possible characters are defined. ASCII is used in UNIX and Disc Operating System (DOS) -based operating systems, except for Windows NT, which uses a newer code, Unicode. The American National Standards Institute (ANSI) [Ref. 4] originally developed the ASCII format.

RTF (Rich Text Format) is a file format that lets users exchange text files independent of processors and operating systems. RTF defines control words and symbols that serve as common denominator formatting commands. When saving a file in the RTF format, an RTF writer that converts the word processors makeup to the RTF language processes the file. Upon reading, the control words and symbols are processed by an RTF reader that converts the RTF language into formatting for the processor that will display the document [Ref. 5].

3.2.2 Images

There are three popular methods of image storing: plain, compressed and lossy compressed.

The Bitmap (BMP) is an example of plain storage. It is the Microsoft Windows Bitmap format for Device Independent Bitmaps (BIBs). The format is very popular on Intel based PCs due to the support from Microsoft applications and optimisation for access from Intel processors. BMP is a fairly simple format and BMP readers are available on other systems as well [Ref. 6].

Graphics Interchange Format (GIF) is categorized as compressed storage. The GIF format is highly compressed and is designed to minimize file transfer times over the Internet. It should be noted that GIF only supports mapped images with eight bit colour. Unisys owns the compression algorithm used in GIF and GIF downloads and users are required to get a license from Unisys for use of their algorithm [Ref. 7].

The Joint Photographic Experts Group (JPEG) image format is a typical example of lossy compression. This format lets the user choose from a range of compression qualities (actually from a suite of compression algorithms) when creating or converting an image. It is then possible to make a trade-off between image quality and file size. Both JPEG and GIF are de-facto main standards for displaying images on the Internet [Ref. 8].

3.2.3 Audio

Audio data flow can either be stand-alone or they can accompany other media such as motion picture. From a general point of view, the most interesting part of this media is speech, music (notes and rhythm) or sound effects (gunshots, explosions etc.). For media companies this implies that the use of audio is focused around speech and retrieval of music, music on demand or video on demand. Digital stored audio can be found on many different devices including compact discs, magnetic discs, hard drives and optical discs.
Although the data rate for CD quality audio is small compared to the demands of video, it still exceeds the bandwidth of dial-up Internet connections, and longer recordings rapidly consume disc space. A three-minute song recorded in stereo will occupy more than 25 Mbytes [Ref. 9]. Hence, when audio is used in multimedia, and intends to be transferred across Internet, there is a need for compression.

The term compression can be further subdued into speech compression and perceptual based compression. There is a significant difference between these, and the reader should be aware of it. Speech compression has been in use since the 1960s when digital audio was introduced to the public telephone networks. The term companding (a compressed version of "compressing/expanding") is used to describe a technique where non-linear quantization lets higher levels be spaced further apart from the low ones. By doing this, quiet sounds are represented in greater details than louder ones.

Perceptually based compression is categorized as lossy compression. The idea behind this compression scheme is to identify and discard data that doesn't matter – sounds that are too quiet to be heard, or may be obscured by some other sound. To do this, a so-called psycho-acoustical model - a mathematical description of the way we hear and perceive sound – is used. The best-known algorithms that have developed are those in the MPEG standards. MPEG-1 and MPEG-2 are primarily video standards, but since most video has sound, they also include audio compression. Due to its success MPEG audio is often used on its own purely to compress sound, especially music.

MPEG-1 specifies three layers of audio compression, all based on perceptual principles. As the complexity of decoding increases from Layer 1 to Layer 3, the data rate of the compressed audio decreases. For stereo, these rates are 384 kbps at Layer 1, 256 kbps at Layer 2 and 128 kbps at Layer 3. The audio part of MPEG-2 is essentially identical to MPEG-1 audio, main differences being some extensions to cope with surround sound. It should also be noted that MP3 is not an abbreviation for MPEG-3 – as this format does not exists.

3.2.4 Motion Pictures

As for audio, there are different formats for storing motion pictures. Three well known examples would be Video Home System (VHS), Betamax and Digital Versatile Disc (DVD). For compressed digital video (audio + motion picture) two standards are considered more important than others. For digital video equipment intended for home and semi professional use is based on the DV standard, while studio equipment, digital TV and DVD are based on MPEG-2. Neither of these are a single standard. The main variations on DV – DVCAM and DVPro – are concerned with tape formats, but both uses the same compression algorithms and data streams as DV.

As stated in the previous section, the MPEG standards are a family of standards for both video and audio compression. MPEG-1 was designed for coding progressive video at a transmission rate of about 1.5 mbps. It has seen much use in Video-CD and CD-I (Compact Disc-interactive) media. MPEG-2 was designed for the coding of interlaced images at transmission rates above 4mbps. The standard is used for digital TV and DVD; backward compability lets an MPEG-2 player handle MPEG-1 data as well. A proposed MPEG-3 standard that was intended for High Definition TV (HDTV) was merged with MPEG-2 when it became clear that this standard met HDTV requirements [Ref. 10].

The MPEG-4 is a recently developed standard with quite ambitious goals. It addresses speech and video synthesis, fractal geometry, computer visualisation and means to interact with scene content [Ref. 11]. The standard is cantered around four basic ideas:

- Modular integration of images, text, audio and video objects.
- Transparent delivery of content from a variety of sources.
- User interactivity and identification.
- Protection of Intellectual Property Rights (IPS).

4 Survey of IP based services

4.1 Introduction

This section gives an overview of important matters regarding the Internet Protocol, known as IP. It also contains an investigation of different services and applications that are available to users of the current Internet, as well as services that are believed to be popular in the coming Universal Mobile Telecommunication System, UMTS.

4.2 The Internet Protocol

Data communications has proved that it is possible to build reasonably large networks by using bridges and switches. This is especially true for networks in a limited, geographical area, so-called LANs. The main problem of networks using this topology is their ability to scale and handle heterogeneity. Large, highly heterogeneous networks with efficient routing capabilities are referred to as internetworks [Fig 3]. These networks need other techniques in order to deliver reliable and effective services to its subscribers. The Internet Protocol is considered the key tool to implement this.

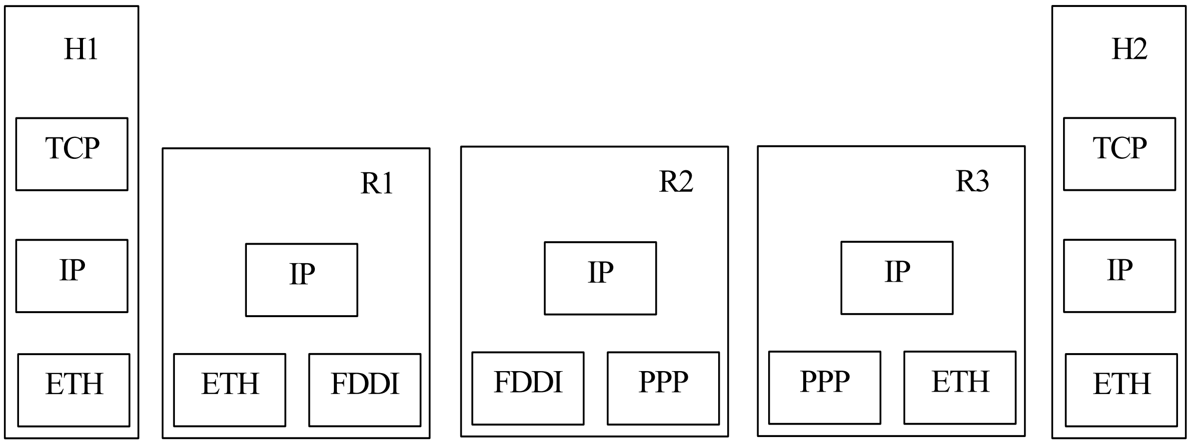


Figure 3 Model of a simple internetwork

4.2.1 History

Leonard Kleinrock at MIT published the first paper on packet switching theory in July 1961 and the first book on the subject in 1964 [Ref. 12]. Kleinrock convinced other researchers about the feasibility of communications using packets rather than circuits, which was a major step along the path towards computer networking. The other key step was to make the computers talk together. To explore this, Thomas Merrill and Lawrence Roberts in 1965 connected a TX-2 computer in Massachusetts to the Q-32 in California with a low speed dial-up telephone line. By doing this, they had created the first wide-area network (WAN) ever built. The result of this experiment was the realization that the time-shared computers could work well together, running programs and retrieving data as necessary on the remote machine, but that the circuit switched telephone system was totally inadequate for the job. So Kleinrock's conviction of the need for packet switching was confirmed [Ref. 13].

The Advanced Research Projects Agency (ARPA), later renamed Defence Advanced Research Projects Agency (DARPA), during the late 60's developed a packet switched network called the ARPANET. Computers were added quickly to the ARPANET during the following years, and work proceeded on completing a functionally complete Host-to-Host protocol and other network software. In December 1970 the Network Working Group (NWG) finished the initial ARPANET Host-to-Host protocol, called the Network Control Protocol (NCP). As the ARPANET sites completed implementing NCP during the period 1971-1972, the network users finally could begin to develop applications.

One shortcoming of the NCP was that it did not have the ability to address networks (and machines) further downstream than one destination switch on the ARPANET, and thus some change to NCP would be required. (The assumption was that the ARPANET was not changeable in this regard). NCP relied on ARPANET to provide end-to-end reliability. If any packets were lost, the protocol, and thereby any applications it supported, would come to an effective halt. In this model NCP had no end-to-end host error control, since the ARPANET naturally was the only network in existence, and it would be so reliable that no error control would be required.

It was then decided to develop a new version of the protocol that could meet the demands from a more complex network environment. This protocol would eventually be called the Transmission Control Protocol (TCP). While NCP acted more like a device driver, the new protocol would be more like a communications protocol.

TCP was from the beginning described as one protocol, which provided all the transport and forwarding services in the ARPANET, gradually termed Internet. The intentions were for the TCP protocol to support a range of transport services, from the totally reliable sequenced delivery of data (resembling a circuit model) to a datagram service in which the application made direct use of the underlying network service. This might imply occasional lost, corrupted or reordered packets.

However, the initial effort to implement TCP resulted in a version that only allowed for virtual circuits. This model worked good enough for file transfer and remote login applications, but it was also made clear that in some cases packet losses should not be corrected by TCP, but instead be left to the application to deal with. These experiences led to a reorganization of the original TCP into two protocols, the simple IP which provided addressing and forwarding of individual packets, and the separate TCP, which was concerned with service features such as flow control and recovery from lost packets. For those applications that did not want the services of TCP, an alternative called the User Datagram Protocol (UDP) was added in order to provide direct access to the basic service of IP.

This historical development has led to some important technical facts: The ability of IP to run over everything has clearly become one of its most significant characteristics. It is a fact that many of the technologies over which IP runs today did not exist when IP was invented. Up to now, no network technology invented has proven to bizarre for IP; a well-known joke among engineers is that IP can run over a network that consists of two tin cans and a piece of string.

4.2.2 Versions

In 1991 the Internet Engineering Task Force (IETF) began to look into the problem of expanding the IP address space. Since the IP address is carried in the header of every IP packet, increasing the size of the address implies a change in header it selves. This means a new version of the Internet Protocol - and consequently new software for every host and router in the Internet.

Scaling problems caused by the Internets massive growth was the prime motivation for expanding the address space. Another was the growth of routing table information needed in the Internet's routers. Although the IPv4 theoretically can accommodate close to 4 billion hosts (32 bit value), it will be exhausted long before this, due to the impossibility of achieving 100 % address utilisation efficiency [Ref. 14].

The work resulted in a proposal called IP Next Generation. This was later assigned the number 6, and is known as IPv6. Aside from an increase in address space from 32 to 128 bit, several changes were considered necessary from network designers point of view: support for real-time services, security support, auto configuration and support for mobile hosts. It is interesting to note that while many of these features were absent from IPv4 when IPv6 was designed, support for all of them has emerged in IPv4 during recent years.

As IPv6 was designed, it became clear that there should be a plan to move from IPv4 to IPv6. However, with the Internet so large and without a centralized control, it would be impossible to have a "flag day" when every host and router was shut down and started up again with a new version of IP inside. Today we are in the beginning of a transition period where some routers and hosts run IPv4 only, some run both IPv4 and IPv6 and some run IPv6 only.

4.3 Categorization of different services and applications

4.3.1 Introduction

This chapter gives an overview of IP based services that are suitable for mobile users. The list can never be complete, since services and applications for the mobile marked are under constant development. Each service is described and exemplified by an application that is currently in use by a certain number of mobile subscribers.

The reader also gets an introduction to different methods of categorizing well-known media services [Ref 15].

4.3.2 Methods of categorization

Lately it has been common to distinguish between different information and communication services by referring to their form (audio/motion picture/text etc) and how they are distributed (internets/public switched telephone networks (PSTN)/cable television). However, the development of convergence makes this a somewhat difficult division. It is clear that popular services combine different forms as well as the ability to be distributed across different infrastructure.

One way to categorize services is by their service level. It is then possible to divide between distribution, strict tele services and value adding services. Distribution refers to the underlying infrastructure, while tele services include distribution and applications offered to users. Value adding services include some sort of complemented services; an example of this is call senters.

A weakness in this type of service categorization is that you get little or none knowledge of the variations for each service level. This is especially true in the field of value adding services, where the diversity can be excessive. The aim of this chapter is also to provide an overview of such services.

Another solution is to categorize by communication modus. This means that services should be divided according to the number of senders and receivers. Broadcasting can be categorized as one to many, telephony as one to one. But the recent development in Information and Communication Technology (ICT) has led to several middelforms with a varying number of senders and receivers. This can be schematically described in the following figure [Fig 4]:

Recievers

Senders	One	Some	Many
One	Telephony Fax E-mail	Distant learning Video On Demand Pay Per View E-mail	Broadcast Homepage Webcasting Near Video On Demand
Some		Intranet Extranet Tele/video – conferencing Chat	
Many	E-commerce		Newsgroups Electronic bulletin

Figure 4 Different services categorized by communication modus

While this way of categorizing provides information about communication modus, it doesn't hint on what degree of interactivity each service provide to the end user. Second, it doesn't provide information about bandwidth and maximum delay requirements for each service. Third, it doesn't say what specific needs a service fulfils for the end user.

Another method would be to categorize services by demands of bandwidth. It is clear that services that implement an amount of motion pictures normally require a substantial bit of the available network resources, independent of underlying infrastructure. At the same time, more basic tele services like messaging, is far less demanding regarding underlying networks. However, it might be confusing and unpractical to draw fixed limits for what demands of bandwidth a certain type of services would require. Some services could very well be able to adapt and hence give consistent results over a wide variety of bandwidth range.

A related way of categorizing is to divide into groups according to their need to be transmitted in real-time. Speech is very sensitive of delay during transmission, whereas browsing web pages can accept a certain delay without serious compromising of user value. While this method is useful for finding services that applies well to the uncertain and rapid changing environment of mobile communication, it won't give straight information of how services can obtain real-time characteristics in a non-real-time environment.

It is also possible to divide between services by their extent of interactivity. Such interactivity assumes an element of two-way communication. Traditional broadcast services do not have interactivity, but the development of digital TV may include a return channel for ordering of programs, TV-shop items etc. Internet also presumes two-way interaction. It is a general opinion that the demand of capacity for downloading is greater than for uploading. Straight telephony however, is an example of an interactive service that assumes symmetrically distributed capacity.

The final method reviewed is the one where services are categorized by their functional character, or what user demands they try to fulfil. Based on this perspective, services can be put in four major categories:

- *Communications*: This category includes direct contact between individuals, like e-mail and traditional telephony. Technology has made it possible to evolve such services into communication within groups (one to some - some to some). Distant education, intranet and video conferencing are services that belong in this category.
- *Entertainment*: A major part of today's ICT services are aimed at satisfying people's need for entertainment and amusement. Traditionally these needs were fulfilled by TV and broadcasting, but during the last five years or so the PC has become one of the most important platforms for entertainment. Up to now, PC games have been bought or rented from specialized dealers, but there are indications that sales and distribution will move on to the Internet.
- *Information*: Traditionally this has been the most important issue for both broadcasted and printed media. Telephony also includes some information services; examples would be weathercasts, news and number information.
- *Transactions*: Such services include buying and selling merchandises using Internet or TV.

From a practical point of view it is often difficult or unpractical to separate between these four categories. The nature of convergence is that services may change form and get a broader field of function. Broadcasting, as we know it today, can clearly be categorized both as information and entertainment services. The development of interactive broadcasting implies that transaction and communication can be integrated into the medium. In addition to this, communication services will often present some kind of information or entertainment value.

The above discussion indicates that there are no definitive lines to draw when choosing methods of service categorization. None of the mentioned methods divides uniquely between different services. It is also clear that none of the methods intercept all relevant aspects of each service. Nonetheless, this survey gets useful because it illustrates which qualitative or technical matters that may become important in a categorization of IP-based services.

4.3.3 American market

The UMTS Forum estimates that there will be 190 million mobile users in the North American marked by 2005, rising to 220 million by 2010 [Ref 16].

USA is behind both Asia and Europe when it comes to use of mobile Internet. However, recent market studies shows that 20 % of today's mobile Internet users never had used or been connected to fixed Internet before [Ref 17]. It is also a fact that the American market is watching, learning and developing from what happens in Europe regarding the introduction of GPRS and UMTS. Strategy Analytics, an American consulting firm, did a market research in December 2000 regarding what kind of new services users would require [Fig 5]. The survey did not focus on issues like how much the customer would like to pay for the service.

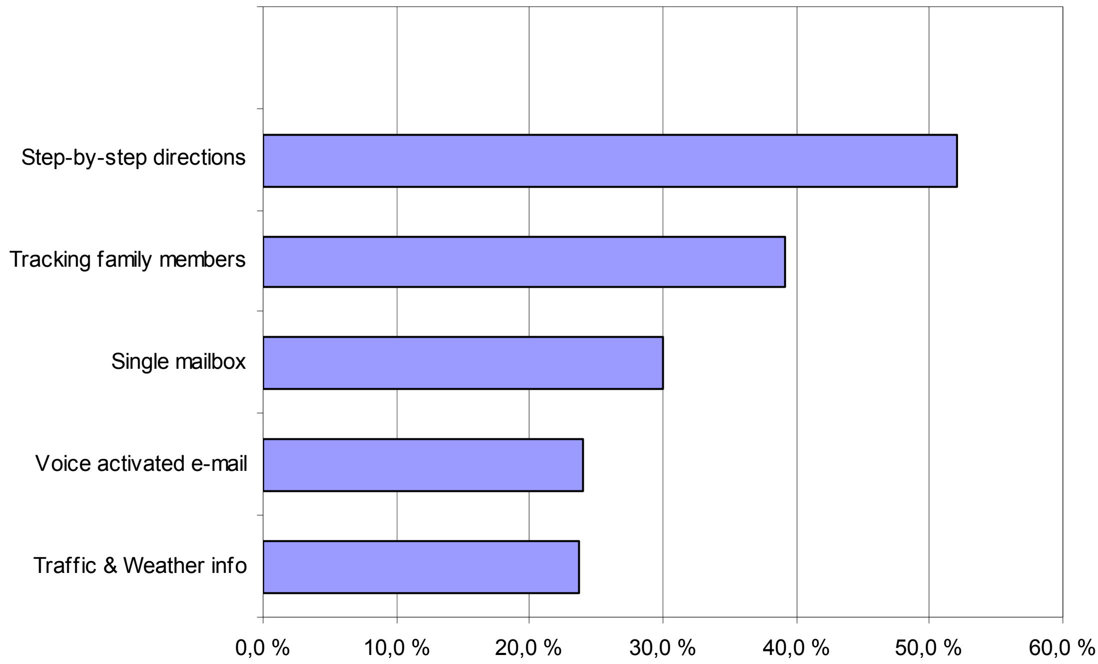


Figure 5 The overall interest in new mobile services, December 2000

As can be seen, the survey does not include entertainment services like games or mobile commerce (m-commerce). Neither is the survey concerned with technical problems and solutions, like communication protocols and coding schemes. It merely indicates which services are likely to get attention from the users. An immediate comment to the results would be that the high percentage of car ownership in the US might explain the high popularity of step-by-step direction applications.

4.3.4 Asian market

The UMTS Forum estimates that there will be 400 million mobile users in the Asia Pacific by 2005, rising to 850 million by 2010 [Ref 18].
NTT DoCoMo in Japan, the world's largest mobile telecommunications operator, in February 1999 launched a mobile system called i-mode. This system enables users to access Internet services via their cellular telephones, and reached 23 million users in May 2001. Being packet based, users are charged for amount of downloads, not time spent on the air. Since i-mode is a packet based technology, and hence is somewhat related to third generation mobile systems like UMTS, it can be interesting to study what services are successful and which are not. To assume that a similar pattern of use will evolve in both Europe and US when third generation systems are rolled out might be wise. Both service providers and operators could get useful information by looking at the development in the Asian market, and then be better prepared

for what could happen in their own domestic market. The following figure [Fig 6] is a summary of information gathered from Ericsson's web site [Ref 19].

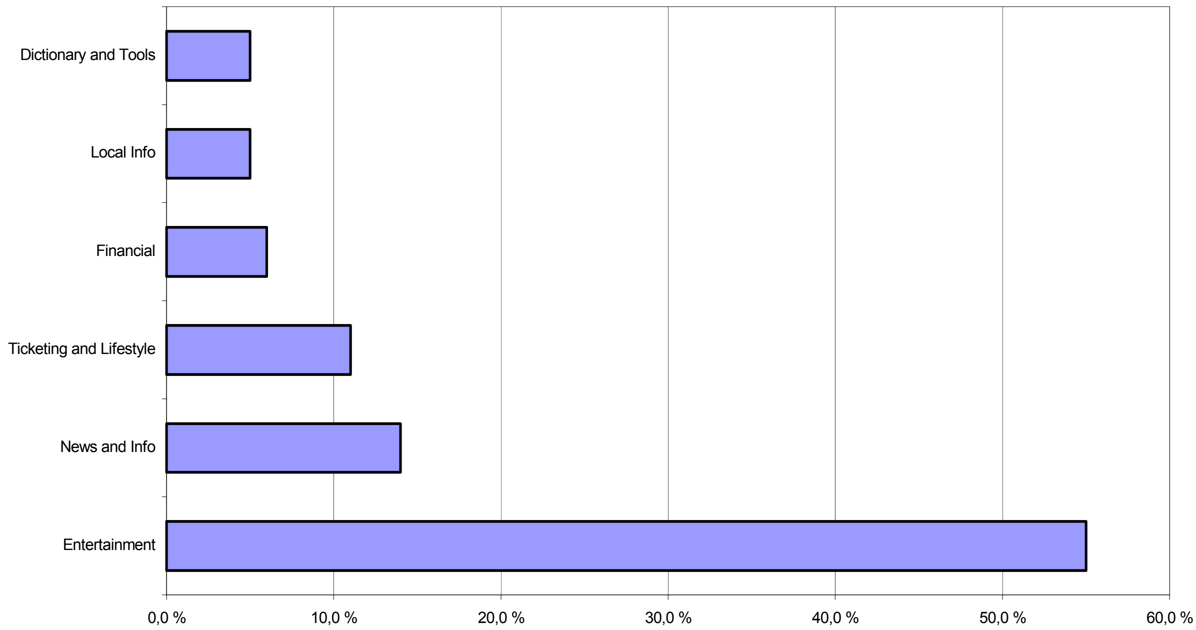


Figure 6 Contents accessed via i-mode in Japan, 2000.

The reader should note the high popularity of entertainment services. They are about four times more popular than news and information services. Also note the relative small use of financial and local information. UMTS operators and content suppliers should be aware of this when developing user services.

4.4 Mobile services

4.4.1 Introduction

This chapter is focused around what kind of services mobile Internet users are offered from various service providers. Third generation mobile systems are not yet introduced to the market, so a survey like this focus on services that may become available, in addition to those that are already implemented.

4.4.2 Top-level categorization of mobile services

In the table below possible categories of services for mobile users are listed. The table is not concerned with demands of real-time transfer and other technical matters; such issues are dealt with in following chapters.

From the previous chapter, "Methods of categorization", it should be clear that there are several ways to organize a survey of mobile services. It all depends on what the author wants to express and which characteristics he wants to emphasize. The purpose of this chapter is to give an easily accessible overview of what services are offered to users, now or in near future.

To provide an easy and understandable overview over mobile services, it is useful to define some main categories. Such categories may differ according to who or what organisation did the categorisation. Research [Ref. 20] reveals that most network operators, application vendors and system suppliers support an approach like this [Tab 1]:

Table 1 Top-level categorisation of mobile services

Service	Description	Comments
Interactive Media	Multimedia Messaging Service (MMS) lets users send pictures, motion pictures and audio independent of platform (GSM, GPRS, and UMTS).	Will this be the next "killer application"? "It will give meaning to GPRS", says A. Vanjopi, vice president Nokia Mobile Phones.
Entertainment	Both operators and suppliers cooperate with vendors like Sega, Nintendo and Sony.	Nokia predicts that four out of five mobile users in 2005 will use mobile games.
Positioning	A users location compared to other users, or a required destination.	This is clearly another key service. NTTDoCoMo says: "Among the most important services in our new 3G system FOMA".
M-commerce	The ability to use your Mobile Station (MS) as an electronic wallet.	Suppliers like Ericsson work close with Visa and MasterCard, trying to get their services prepared for mobile use.
Transportation	The MS can be used to check in prior to air travels.	This service is strongly connected to M-commerce.
Engine (Machine-to-Machine)	Cars and houses are equipped with artificial intelligence. They can then communicate with their owners and provide important information	Fantasy and market demands are the only limiters.

A listing of six main categories like what is done here, gives both advantages and drawbacks. At first glance it looks easy and user friendly, but a closer view reveals that things are a bit more complicated. Several of these categories are linked close together. Interactive Media and Entertainment may contain the same type of applications; this is true for Positioning and Transportation as well. To draw strict and rigid borders between these categories is difficult and sometimes inconvenient.

Nevertheless, since this chapter is focused around giving readers an outline of existing/possible services, the chosen layout is used to further investigate possibilities in each of the six main categories.

4.4.3 Interactive Media

Interactive Media is probably the most awaited service class. Most advertising for UMTS focuses on how customers can use interactive media to communicate with friends and relatives through video and e-mail.

Table 2 A selection of interactive media services

Service	Description	Comments/status
E-mail	Receive and send e-mail messages on the MS.	New handset with larger display makes it easier to view long messages.
Voice-mail	People can leave messages when the MS is switched off. It can also be used as a personal Dictaphone and audio post-it note.	This service is already implemented in some PSTN and GSM networks.
Notification	Subscribe to updates in selected fields of interest, e.g. music, fashion, economy etc.	SMS provides such services today, but only text based.
Video (conferencing)	There is a strong believe that new handsets will include a small digital camera. Users can then send pictures and short movies to friends, family and colleagues.	Video on MS is probably the most aforesaid application for third generation mobile systems. Most marketing involves a sequence where people view video on their handset.
Medicine/Rescue	Rescue units can use a MS to consult specialists when they reach an emergency site.	Applications like this are under development.
Calendar	Access your friends, family and colleagues calendars to find when they are busy and when they have time off. Schedule meetings with co-workers and business associates.	This should be a useful tool in the business world – since people tend to bring their job with them instead of being physically restricted to a single area.
Address books	Exchange address books with friends and colleagues. Download address books from your company's intranet.	Such applications would also help relatives to keep in touch with each other.

4.4.4 Entertainment

Various entertainment services are considered potential "killer applications". Mobile users seem willing to pay for simple services like logos and jokes etc delivered by SMS. Operators are eager to make the most out of this, due to the potential income generated by such services.

Table 3 A selection of entertainment services

Service	Description	Comments/status
Television Broadcasts	The user can follow his or her favourite show e.g. when travelling.	There are technical and economical challenges related to applications like this.
Radio	Use the MS to tune in to programs of interest.	Suppliers of mobile equipment already implement this.
Games	A vast amount of games should be available, both for downloading and on-line playing. Time spent on public transportation can be used to play against fellow passengers.	Operators and vendors must find a way to make money on such services.
Video	Users can subscribe to daily video clips from any area of interest, e.g. hidden camera, celebrities and so on.	A evolution from today's SMS services, where you can subscribe to daily jokes, riddle of the day, quotations etc.
Horoscope	Download your weekly horoscope.	
News and weather reports	Important news can be delivered to the MS. If the user is in a high capacity area, the transmission can include video and speech.	The news can be of either local or global interest. It would also be an effective way for local/national governments to distribute important information.

4.4.5 Positioning

Localized information and positioning services are expected to be two of the most important driving forces for the mobile Internet. According to a new report from the analysis firm The Strategics Group, the market for such services will be worth more than USD 80 billion in Europe alone over the next five years.
"By 2003, mobile positioning services will be a part of daily life. All cellular operators will offer their customers these services within a few years," says Göran Swedberg, marketing manager for Ericsson's Mobile Positioning System (MPS) [Ref. 21].

Table 4 A selection of positioning services

Service	Description	Comments/status
Locations	The MS can provide information on where friends and colleagues are located. It can also project a map of the area where, and guide the user to restaurants, public transportation or places of special interest.	Ericsson has implemented a position system that works in today's GSM phones. These services are concerned with matters of privacy – subscribers must be able to deny other users to trace their location.
Fleet management	To use resources efficiently, it is important that taxi, car rental and trucking companies know where their vehicles are.	Systems for fleet management have been in the market several years. Traditionally, such systems are based on GPS to provide geographical information.
Emergency	Emergency desks can trace the origin of incoming calls. This can save valuable time in the beginning of rescue operations.	This is possibly one of the most important location services. The Estonian operator Eesti Mobiil uses it to determine the location of people calling emergency numbers from their MS.
Security	Security guards working alone would benefit from the fact that their employers can trace their movements. Personal assault alarms that reveals location, can lead to faster response from police and rescue teams.	
Tracing	Third generation networks can be used to trace stolen vehicles, boats and railway wagons.	Traditionally such services have been based on satellite networks.

4.4.6 M-Commerce

One of the most important obstacles for operators, vendors and payment companies to overcome, is to unite on the subject of mobile commerce standards. Despite the establishment of several standardisation bodies, these tree industries are still failing to agree on an open-system payment solution. Visa is accused of slowing down the adoption of a more open payment standard, while operators are warned against implementation of small-scale mobile commerce payment solutions. Either way, it's clear that major steps should be taken to get payment solutions off the ground together with the introduction of GPRS and other 2 + generation technologies.

Table 5 A selection of M-commerce services

Service	Description	Comments/status
Payments	Your MS becomes your digital wallet.	As described above, there are important strategies (mostly related to security) to agree upon before the MS lets you pay for gasoline or groceries.
Account inquiries	Access your bank accounts independent of location and time.	The widespread use of net bank services implies that users should be interested in these applications.
Banking	Apply for loans, change bank connection or administrate accounts via the MS.	Users should be able to apply for loans, get immediate feedback, and eventually buy what tempts them.
Stock quotes	Have stock quotes and other trading information delivered instantly, buy and sell shares directly from the MS.	SMS can deliver stock quotes today, based on ticker symbol.
Road-toll	Each time your car passes through a road-toll zone, the fee is deducted from your account.	International agreements should make it possible to extend such services throughout Europe.
Tickets	Use the MS to order tickets for concerts, sport events, cinemas and different kind of plays.	Ordering and payment of tickets are close related, and a system that takes care of both should benefit both users and organizers.

4.4.7 Transportation

Cities and urban areas all over the world experience growth in populations. This puts an increasing pressure on all kinds of public transportation, transport of goods and general traffic coordination. If technical capabilities in third generation mobile systems are exploited and taken advantage of, local and national governments should have an efficient tool at hand for deploying new and efficient solutions to these problems.

Table 6 A selection of transportation services

Service	Description	Comments/status
Arrival control	If the bus/train/subway is late, passengers are given an estimate on when to expect the transport to arrive.	This service should fit well in the micro-cell concept of third generation mobile systems. Small cells provide better accuracy for estimate of time and velocity.
Taxi reservations	Taxi companies can immediately locate where customers call from, this should result in faster response from unoccupied taxies.	This service might be categorized as positioning, but it's also connected to M-commerce.
Check in at airports	As you enter the airport your MS automatically connects to the check-in system and confirms your reservation.	Identification, check-in and payment can be done in one operation, if the user agrees and the system permits it.
Direct payment.	Pay and register as you enter public transportation – no need to fiddle with coins.	This should make public transportation more effective and let the driver focus on driving. Also a useful feature for foreign visitors, they don't need to concern with local currency.
Traffic reports	Users can subscribe to traffic reports related to their current location.	Important messages regarding closed roads and accidents can be broadcasted to all on-line users in the area.

4.4.8 Engine

Machine-to-machine communications are growing fast. Throughout the computer and tele communications society there is a broad consensus on the following scenario: within the next five years, Internet connections between machines will outnumber the connections made by people [Ref 22]. Currently techniques like Dual Tone Multi Frequency (DTMF), SMS and Circuit Switched Data (CSD) are adequate to cover the needs of communication in wireless machine-to-machine applications, but as GPRS gets implemented in networks round the world, it will also be the first natural step for machine-to-machine communications. GPRS enables applications using machine-to-machine communications to be "always-on" so that a greater amount of data can be sent if and when needed. As a result, new types of applications and businesses will appear.

Table 7 A selection of engine services

Service	Description	Comments/status
Refrigerator	Call your refrigerator and get information on what is absent and what is not. It can then e-mail a replenishment order to the nearest supermarket.	Electrolux in cooperation with Ericsson made the first intelligent refrigerator, the Electrolux Screenfridge. Internet access from the kitchen is a reality for 50 Danish homes participating in a test program.
Locks and security	Electronic locks can be controlled via the MS, they can alarm police and owner if someone tries to break in.	The extended address space in IPv6 is expected to cope with a future where every toaster is connected to the Internet.
Car maintenance	The car automatically makes an appointment with the preferred service station when service is needed or due.	If a car suffers an engine problem, it can download emergency programs and reconfigure settings. It may then be able to drive to the nearest service point.
Alarms	Sensors in an elevator can set off alarms at the inspection centre if something fails.	This would minimize the need for human supervision in every building that has an elevator.
Self diagnose	Vending machines tell their owner when it is time to refill soft drinks.	The Coca Cola Company implements this service in the US.

4.5 Mobile services demands of real-time transfer and bandwidth characteristics

4.5.1 Introduction

As mentioned in chapter 1.1.1, 3GPP has specified "release-99" as the basis for the implementation of UMTS. This release defines four traffic (QoS) classes [Ref. 23]. These are Conversational, Streaming, Interactive and Background. The main difference between them is how delay sensitive their traffic is. Conversational and Streaming class are intended to carry real-time traffic flows. Interactive and Background class should be used by traditional Internet applications like WWW, Email, Telnet, FTP and News.

4.5.2 Conversational class

Telephony speech is the most well known application in this class. But with the introduction of multimedia capacity in network and coming mobile stations, a number of new applications will require this scheme. Two examples would be voice over IP and real-time video transfer tools. Real time conversation is always performed between single users or groups of human end-users. One should note that this is the only scheme where the required characteristics are strictly given by human perception.

Real-time conversation scheme has some special characteristics: The transfer time must be low due to the conversational nature of the scheme. Simultaneously the time relation (variation) between information entities (i.e. samples, packets) in the stream must be preserved in the same way as for real time streams. The maximum transfer delay is specified by the human perception of video and audio conversation. Therefore the limit for acceptable transfer delay is very strict, as failure to provide low enough transfer delay will result in intolerable lack of quality. The transfer delay requirement is therefore both significantly lower and more stringent than the round trip delay of the interactive traffic class.

Fundamental characteristics for this class are:

- The time relation between entities in the stream must be strictly controlled and preserved.
- Low and stringent delay is the key to achieve adequate end-user quality.

4.5.3 Streaming class

When the user views real-time video (motion pictures + audio) through the MS, the scheme of real time streams applies. A real-time data flow always aims at a human destination. It is a one-way flow from the source to the client.
This scheme is relatively new in data communication. It has also grown very popular among Internet users. Streaming raising several new requirements in both telecommunication and data communication systems. Its characteristics is that the time relations between information entities within a flow must be preserved, although it does not have any requirements on low transfer delay, like the Conversational class has.

Naturally, delay variation of the end-to-end flow must be limited. If not, time relations between entities in the stream get impossible to keep good enough. But since the stream normally is time aligned at the receiving end (in the user equipment), the highest acceptable delay variation over the transmission media is given by the capability of the time alignment function of the application. Acceptable delay variation is thus much greater than the delay variation given by the limits of human perception.

Fundamental characteristics for this class are:

- Time relations between information entities must be controlled. Relations are not so strict as for the Conversational class.

4.5.4 Interactive class

When the end-user, either a machine or a human, is on line requesting data from remote equipment (e.g. a server), this scheme applies. Examples of human interaction with the remote equipment are: web browsing, account inquiries, server access, and on-line games. Examples of machines interaction with remote equipment are: polling for measurement records and automatic data base enquiries (tele-machines).

Interactive traffic is a communication scheme that generally can be characterised by the request-response pattern of the end-user. The message destination will expect the response within a certain time. RTT is therefore one of the key attributes. Another characteristic is that the content of the packets must be transparently transferred, with low bit error rate.

Fundamental characteristics for this class are:

- Request-response pattern that depends on the perceived RTT.
- The payload content must be preserved.

4.5.5 Background class

When the end-user, typically a computer or a MS, sends and receives data-files in the background, this scheme applies. Examples are background delivery of E-mails, SMS, download of databases and reception of measurement records.

Background traffic is a communication scheme characterised by the fact that the destination is not expecting the data within a certain time. When the connection is established, it may take some time before the data is transferred, relative to real-time traffic. The scheme is thus more or less delivery time insensitive. As with data in the Interactive class, the content of the packets must be transparently transferred (with low bit error rate).
Fundamental characteristics for this class are:

- The destination is not expecting the data within a certain time.
- The payload content must be preserved.

4.5.6 Summary

The following table [Tab. 8] is extracted from 3G TS 23.107 [Ref. 23, p. 16]. It sums up the different traffic classes defined in UMTS "release-99" and gives examples of typical applications for each class.

Table 8 UMTS traffic classes

Traffic class	Conversational class. Real-time telephony.	Streaming class. Real-time for streaming services.	Interactive class. "Best effort" for interactive services.	Background class. "Best effort" for background services.
Fundamental characteristics	The time relation between entities in the stream must be strictly controlled and preserved. Low and stringent delay is the key to achieve adequate end-user quality.	The time relation between entities in the stream must be strictly controlled and preserved. Relations are not so strict as for the Conversational class.	Request-response pattern that depends on the perceived RTT. The payload content must be preserved.	The destination is not expecting the data within a certain time. The payload content must be preserved.
Example application	Telephony	News and weather forecast	Web browsing	E-mail

5 TCP rate adaptation related to the radio link

5.1 Introduction

This section discusses problems that arise when standard Internet protocols such as TCP are used over wireless links. Actual testing in a wireless test bed is outside the scope of this thesis; statements in the following chapters are based on results from various scientific institutions and personnel.

5.2 Adequate transmission theory

The following sub chapters provide basic knowledge needed when TCP over unreliable wireless links is discussed. It is assumed that the reader is familiar with the concept of wireless communications.

5.2.1 Wireless vs. fixed data transmission

There are major differences between wireless and fixed data access/transmission. Some of these differences are caused by natural and physical restrictions related to the nature of radio and mobility, and are to a certain degree independent of the type of technology implemented. In addition to economic related differences, there are three main technical areas of particular interest:

Bit error ratio (related to layer one and layer two in the OSI-stack)

- The bit error ratio over radio links is in the order of several magnitudes poorer than what is experienced over fixed links
- Errors on the radio link have completely different distribution compared to errors over fixed links
- Underlying expectations of bit error ratios from higher layers will not always be fulfilled

Delay

- Delays in mobile systems tend to be much longer than in fixed systems
- Delays in a mobile system may vary (applies to circuit switched connections as well)
- Underlying expectations of small and constant delay from higher layers will not always be fulfilled

Bandwidth

- Bandwidth in mobile systems is a nature given and nature limited resource. Mobile systems will generally not provide as much bandwidth as a fixed system
- The access bandwidth in packet switched mobile systems may vary a lot and also very fast
- Underlying expectations of a stable access bandwidth from higher layers will not always be fulfilled

Each of these three characteristics influence the performance of higher level protocols in wireless systems. TCP, that resides on the transport layer in the OSI reference protocol [Fig 1], is optimised to run over channels with predictable and stable characteristics.

5.2.2 Physical layer characteristics

The delivery delay [Fig 7] for a block of data is the time between sending the first bit at the sender and receiving the last bit at the receiver. It consists of transmission delay (the data size divided by link transmission speed) and a fixed propagation delay (the time it takes a signal to cross the link).

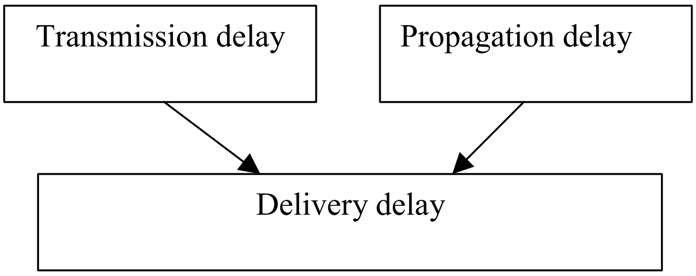


Figure 7 Delivery delay as a function of transmission and propagation delay

Wireless Local Area Network (WLAN) and Cellular Telephone (CT) have propagation delays on their wireless links similar to those of wired links. Transmission delay usually dominates the resulting delivery delay in these systems, unlike geostationary satellites where propagation delay is the dominant factor. This should be taken into account for operators of third generation mobile systems, since satellites will provide coverage in areas with little population.

The error behaviour of wireless links varies with time in a system-dependent manner. Traditional cellular systems are affected by atmospheric conditions just as satellite links are, but they also suffer from multipath fading owing to terrestrial obstructions. In addition, mobility constantly changes the fading and interference of a link. Such bursty errors are hard to combat using coding and interleaving, especially when link performance changes. This rapidly changing error behaviour must be dealt with by error recovery mechanisms that react fast to environmental changes.

5.2.3 Link layer characteristics

All modern digital systems provide some type of frame delivery service. To incorporate them in the Internet, the most important requirement is to provide link layer software to encapsulate IP datagrams into link frames. By doing this, the link layer transforms a given physical link into a logical one, isolating higher layers from low-level details. IP employs these services to support end-to-end best effort delivery. It is further assumed that simple framing schemes offer reasonable performance, indicating that end-to-end performance is limited by the worst link on the path.

Traditionally, the Internet model has delegated error recovery to end-to-end layers. This is done to avoid duplication of efforts, simplify link layer design and avoid error recovery overhead on applications that do not need it. As long as errors are rare, performing recovery only end-to-end is sensible. With the more error prone wireless links this strategy may be challenged. It seems that localized error recovery is potentially faster and more adaptable to link characteristics. If applications and protocols implement their own recovery schemes, some researchers [Ref. 24] argue that the transport layer may retransmit packets in parallel with the link layer, wasting available bandwidth.

5.2.4 Communication modes

It is vital to understand general differences between different communication methods used in tele and data systems. This is a short summary of the most important dividing lines:

- Circuit vs. packet switch connections: This is a clear dividing line for how communication is implemented. It should be noted that a packet switched connection on layer **n** might be circuit switched on layer **n-1**. What is circuit and /or packet switched also depends on the point of view. For layer one (physical) and layer two (data link) it may not always make sense to categorize in circuit and/or packet switching.
- Synchronous vs. asynchronous: Whether the communication is synchronous or asynchronous is another important division. Again, one should be aware that the question of synchronous/asynchronous depends on point of view.
- Connection oriented vs. connectionless: Circuit switched connections have per definition a set-up phase, a data communications phase and a disconnect phase. It is therefore correct to say that circuit switched connection are connection oriented by nature. Packet switched connections may have a set-up and a disconnect phase in addition to data communication phase. The set-up phase is used to establish a virtual circuit in the network. This virtual line makes it possible to use a very simplified addressing scheme for the packets. Since such a virtual circuit keeps information about addressing, taxation and priority in the nodes between sender and receiver, it is necessary to have a dedicated disconnect phase. An example of a packet switched connectionless protocol is the X.25 [Ref. 14, p.181]

  Packet switched connections can also be established without intermediate nodes setting up a virtual circuit. Such an approach demands that every packet carries complete information regarding addressing, priority, taxation and so on. IP is an example on a connectionless protocol.

This discussion leads to two important conclusions:

- Connection oriented communication need information in all intermediate nodes.
- Connectionless communication does *not* need information in intermediate nodes.

The last conclusion suggests that a link failure might not have any serious effect on communication if it is possible to find an alternative route around the failure and update forwarding tables in routers accordingly. Such a feature is important to the history of datagram networks. One of the important goals of the ARPHANET was to develop network technology that would be robust in a military environment. In such, one would expect links and nodes to fail because of attacks such as bombing. It was the ability to route around failures that led to a datagram-based design.

5.2.5 Mobile systems and Internet connectivity

Tradition digital mobile systems offer modest transmission bandwidth (less than 10 Kbps), small frame size and circuit mode operation, all because of their voice-oriented design. Data services are provided by including in each frame data handed to the link layer from higher layers rather than by a voice encoder. The lower transmission rate and longer rage of cellular systems compared to WLANs lead to higher delivery delays. But, they are also more predictable because there is no contention on the link.

For optimal voice performance most mobile systems use short frames that suffer from loss of one to two percent [Ref. 25]. If the physical layer is able to randomise these errors, e.g. through interleaving, most audible voice quality degrading is avoided.

Connections that terminate outside a traditional mobile system need a special interface to connect to other networks. Such an interface is called an interworking function (IWF) [Fig. 8]. To communicate with analog telephones or modems, the IWF transforms digital data or voice frames from the mobile system to analog waveforms. To communicate with ISDN systems the IWF performs rate adaptation and frame conversions. This way the IWF provides mobile devices with an end-to-end circuit abstraction, and at the same time hides from the wired network all physical details of the connection with the mobile host.

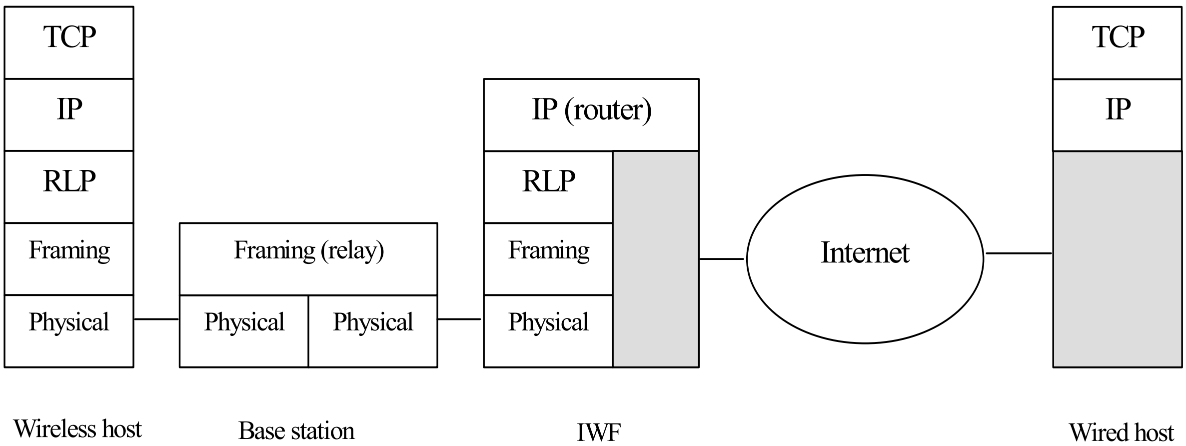


Figure 8 Mobile systems and Internet connectivity

Figure 8 [Ref. 24, p.57] gives a general view on how data transmission in traditional mobile systems like GSM is implemented. Network components that deal with network management, databases for mobility management, user profile handling and equipment authentication are not shown. The wireless host/MS can choose to run a fully reliable link layer protocol called Radio Link Protocol (RLP) or not do so. This is also referred to as to operate in non-transparent or transparent data mode, respectively.

RLP is a High-Level Data Link Control (HDLC)-derived protocol that uses selective Automatic Repeat Request (ARQ) for which the frame size has been optimised for the GSM radio link. In the scheme depicted in figure 8 RLP is used to improve reliability and to encapsulate IP datagrams into link frames. When standard IP services are provided on top of RLP, the wireless host can use any transport protocol (e.g. UDP or TCP) to communicate with other Internet hosts. RLP shuffles data in transparent mode, so it has no notion of what a Point-to-Point Protocol (PPP) frame or an IP packet is.

5.3 TCP as an effective transport protocol over wireless links

This chapter focuses on fundamental issues concerned with the use of TCP over unpredictable and changeable wireless links. The discussion in chapter 5.4 is based on a small selection from the vast amount of research done on the subject during the last decade.

5.3.1 Transport protocols

Chapter 4.2.1 gives information on the background for TCP. Readers may note that the initial use of TCP was related to military computer communication requirements, but as computer networks evolved at government and civilian levels it became clear that these environments needed reliable and standardized protocols as well.

TCP is a connection-oriented, end-to-end reliable protocol, designed to fit into a layered hierarchy of protocols that support multi-network applications. The TCP provides for reliable inter-process communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. Few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from hard-wired connections to packet-switched or circuit-switched networks [Ref. 26].

UDP is another choice of transport protocols used on the Internet. It provides a best effort delivery service without flow, congestion or error control. Higher layer protocols or applications may provide such services instead. UDP has been a popular protocol in LAN-based services, since wired LANs are very reliable and rich on bandwidth. Hence, error and congestion control are not crucial in these networks. UDP is also used for real-time applications like video conferencing, both in intra- and internetworks.

TCP normally achieves a lower throughput than UDP. This is not only due to the extra 12 bytes of overhead per data segment (the difference between header sizes), but also because acknowledgements (ACKs) share medium with forward data.

5.3.2 Network and environmental challenges related to TCP

The widespread use of optical fibre in long-distance networks has led to decreasing error rates. Simultaneously, the increasing traffic loads over the Internet has made congestion the dominant loss factor in modern data communication. Congestion is when a network element has so many packets queued that it runs out of buffer space and has to start dropping packets. The term contention, however, describes the situation where multiple packets are queued because they compete for the same output link.

Flow control and congestion control is one of the main tasks of transport protocols. The TCP protocol therefore implements some sophisticated methods to avoid link congestion. At the same time, it should make the most out of available network resources to utilize link capacity.

TCP faces several situations where its ability to adapt to changing condition is put on test:

- TCP connections are likely to have widely different round-trip times (RTT). RTT might depend on both distance between hosts, and time-dependent strain in intermediate networks. In addition, TCP must support simultaneous connections with very different performances. Timeout mechanisms that trigger retransmission must therefore be highly adaptive.
- Packets can be reordered as they cross the Internet. For slightly out of order packets this is not a problem, since sequence numbers make it easy to rearrange order. Worst case is that the IP's time to live (TTL) field expires, and the packet is thrown away. TCP assumes a Maximum Segment Life (MSL); currently recommended setting is 120 seconds [Ref. 14, p.374]. But IP

does not strictly follow this value; it is simply a conservative estimate made by TCP on how long a packet might live. TCP must thus be prepared for very old packets to show up at the receiver, possibly confusing adaptation algorithms.

- Almost any kind of computer can be connected to the Internet. This makes the resources dedicated to any one TCP connection highly variable, since one host potentially supports hundreds of TCP connections at the same time. TCP must have a mechanism that each side uses to learn what resources (e.g. buffer space) the other side can apply to the connection.
- The sending side of a TCP connection don't know what links must be crossed to reach the destination. A host connected to a fast Ethernet might send at 10 Mbps, but out in the network a 1.5 Mbps link must be transversed. If data generated by many sources tries to get through the same link, problems of network congestion may appear.

Next subject is to investigate what tools TCP uses to manage and maintain a reliable end-to-end connection.

5.3.3 Rate adaptation

TCP transmits data from the application layer as a byte stream. The data are divided into segments, called packets at the IP layer [Fig. 1]. The receiver generates an ACK for every segment – but only if it was received correctly (i.e. no feedback on segments received in error). ACKs are cumulative. They tell the sender up to which sequence number data has been correctly received in-order; duplicate ACKs (DUPACK) are generated for every segment received out of order.

If a TCP sender detects "packet loss", this is interpreted as an implicit signal for congestion. Packet loss because of damage is assumed to be rare, in accordance with the discussion in chapter 5.3.1

Two mechanisms have been specified for error recovery: a timeout mechanism and a fast retransmit algorithm. For the latter, the sender does not wait for a timeout, but retransmit an outstanding segment if three DUPACKs for the same sequence number are received. The timer mechanism at the sender is adaptive to the RTT and its variation, as discussed in the previous chapter.

As long as neither three DUPACKs nor a timeout signal is present, a TCP sender probes for bandwidth. This means that the sender continuously increases the load onto the network to detect its limits. When this limit is detected (threshold reached), the sender increases the load in a linear instead of an exponential way. Figure 9 is a float diagram that explains how this is implemented in TCP.

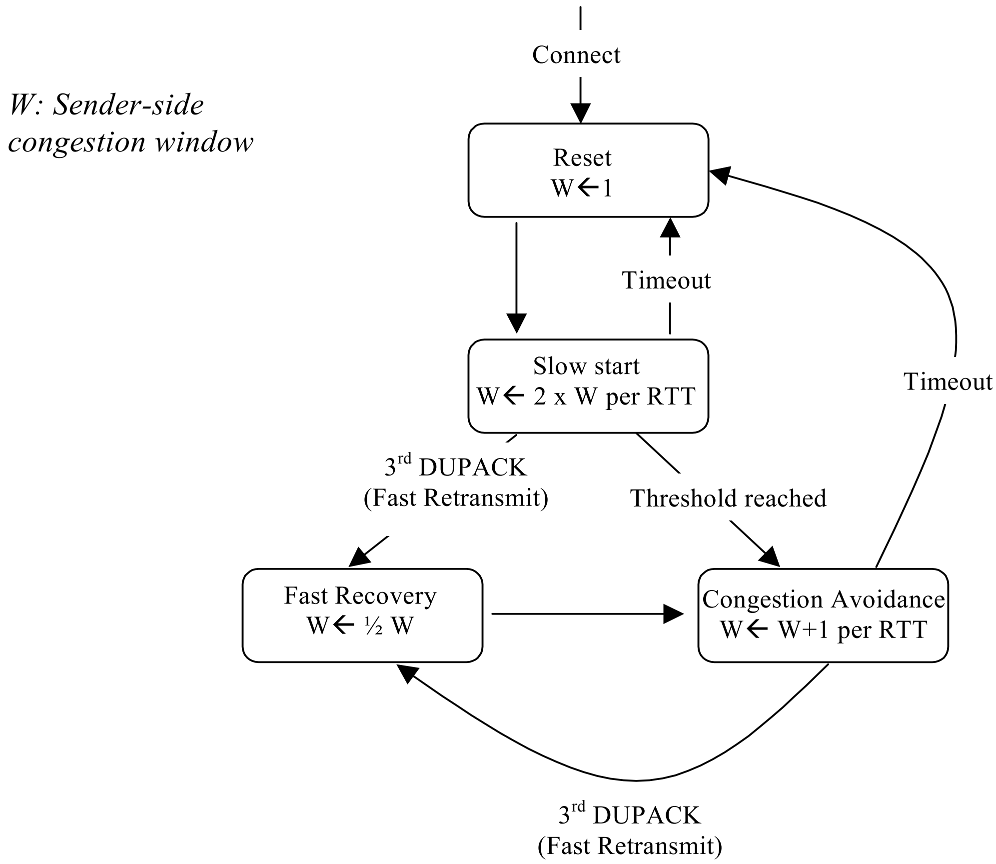


Figure 9 TCP congestion control diagram

A sender can never have more packets (assuming Maximum Segment Size (MSS) equals Maximum Transmission Unit (MTU) minus TCP and IP headers for the link) outstanding than the minimum of the window advertised by the receiver and the sender-side congestion window (W). This minimum equals the load a sender is allowed to put onto the network per RTT.

The following figure [Fig 10] shows a constructed situation of a TCP data flow. In the Slow-start phase the connection gets a timeout and is reset. It then moves to Slow-start and increases exponentially until the detected threshold is reached. From there the connection enters Congestion Avoidance phase and a linear growth in load per RTT. The Fast Retransmit and Fast Recovery incidents are not shown here.

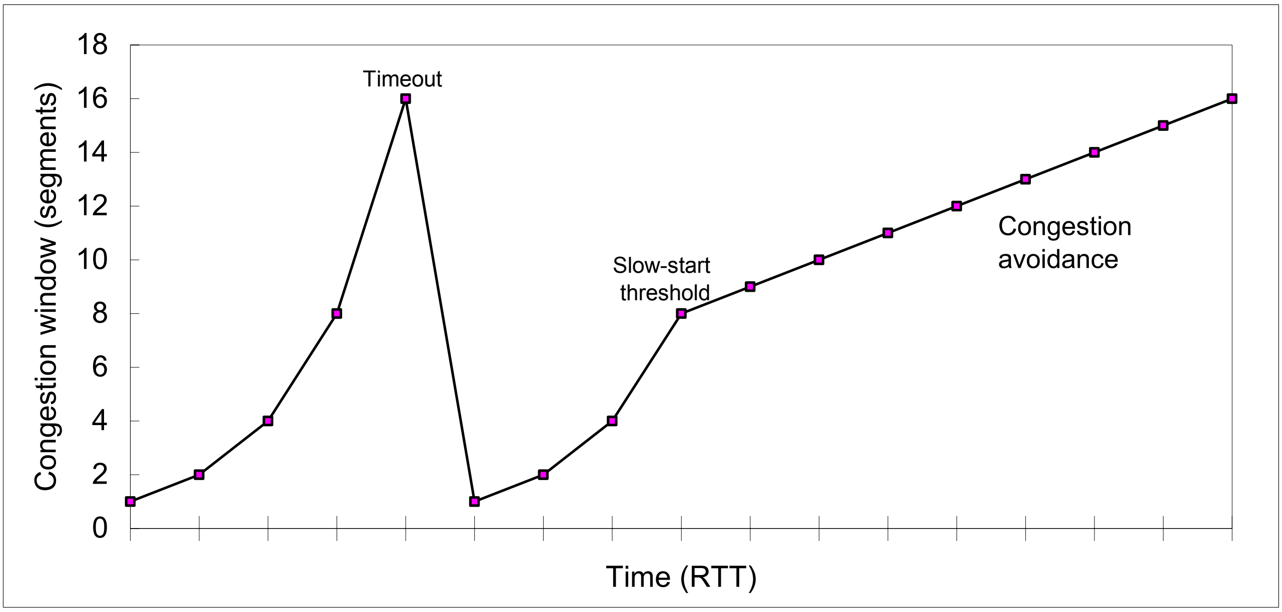


Figure 10 TCP congestion window behaviour

It is important to note that what is described in the figure is a connection where network capacity is the only limiting factor. The congestion window alone limits the offered load; local resources or the lack thereof at either side are not taken into account. Neither are negative effects from other connections sharing the same link.

5.3.4 A word of caution

TCP has been a rather fluid protocol over the last few years, especially in its congestion control mechanism. This has led to different names for different implementations/releases, like TCP Tahoe, TCP Reno and TCP Vegas. Tahoe corresponds to the original implementation of TCP and includes all of the mechanisms described above except fast recovery. Reno adds this mechanism along with an optimisation known as header prediction – optimising for the common case that segments receive in order. Reno also supports delayed ACKs (acknowledging every other segment rather than every segment). The Vegas is a takeoff of these earlier implementations of TCP.

Unless noted, this thesis refers to TCP Vegas whenever the term TCP is used.

5.4 Review of TCP, article 1

This first article reviewed is titled "Internet Protocol Performance over Networks with Wireless Links". It is written by G. Xylomenos and G. C. Polyzos, University of California, and appeared in IEEE Network, Vol 13, August 1999, pp. 55-63.

5.4.1 Scope

The major problems discussed in this article are related to the use of standard Internet protocols over wireless links. Link characteristics for wireless systems are reviewed on the basis of studies of commercial available WLANs and mobile systems. The future of wireless systems and new challenges for protocol evolution is discussed as well.

The article starts with a general summary of wireless systems characteristics. Differences and similarities between WLANs and several mobile systems (GSM, IS-54 and IS-95) regarding physical and link layer is established. The article then talks about UDP and TCP; background and vital operation features are explained.

5.4.2 Research, test procedures and reported results

The authors rely on both own [Ref. 27] and others research. References to other IEEE research are dominant throughout the article. Among others, the authors refer to an article named "A Trace-Based Approach for Modelling Wireless Channel Behaviour" [Ref. 28]. This article speaks of the relationship between UDP/TCP packet size and the resulting Frame Error Rate (FER) in both WLAN and cellular systems [Tab 8 shows cellular system]: reducing frame size (for UDP traffic) from 1400 (equals 1448-byte Ethernet frames) to 1100 bytes halves the measured FER. It also reports that TCP transfer over the same WLAN using 1400 bytes data segments gives a throughput of only 1.25 Mb/s out of the 1.6 Mb/s available on the link – i.e. a throughput reduction of 22 percent caused by only 1.55 percent FER. Xylomenos and Polyzos argue, " This is due to TCP frequently invoking congestion avoidance mechanisms that reduce its transmission rate, even though the losses are not due to congestion" [Ref. 24, p.59].

Table 9 Cellular link performances [Ref. 24, p.60]

User payload (bytes)	Frame Error rate (%)	Overhead (%)	Available Bandwidth (%)
81	7.763	5.814	86.874
167	14.924	2.907	82.603
339	27.620	1.453	71.328
683	47.612	0.727	52.008

Xylomenos and Polyzos also describe differences between WLAN and mobile systems when sending TCP traffic across the network. Mobile systems experience one to two percent FER, quite large for the short frames typically used. Due to the small frame size, the link layer segments IP datagrams into multiple frames. Compared to the example above, a UDP packet with 1400 bytes payload + 28 bytes UDP and IP header would be segmented into 68 link layer frames. Because of interleaving before transmission, frame errors are less bursty in mobile systems. On receiving the interleaved bits, errors are spread over multiple frames so that the embedded error correction code of each frame can recover its content. This results in less audible speech degrading, but adds considerable delay since several frames must be received prior to reverse interleaving and decoding. For example, frame delivery delay is 100 ms for IS-95 mobile system [Ref. 24, p.59].

Xylomenos and Polyzos focus on the consequences when multiple wireless links are transversed. This is the situation when users on distinct WLAN or mobile networks communicate with each other. Through calculations they show how the error rate increases, and argues "… the increased losses lead to more frequent use of TCP congestion avoidance algorithms. Reduced transmission rates due to mistaking wireless errors for congestion causes under utilisation of wireless links" [Ref. 24, p.60].

Transport layer solutions for wireless links are discussed. To avoid unnecessary end-to-end recovery, it is suggested to split TCP connections at pivot points. A router connected to both wireless and wired links is such a point. Different entities of TCP would then execute over each kind of link. As a result, losses at wireless links do not trigger end-to-end recovery. The authors seem to accept such TCP modifications as a general idea – but not unconditionally: "Schemes that only deal with hand-off problems modify the transport layer only at the endpoints, but they still face performance degrading due to wireless errors" [Ref. 24, p.61]. Xylomenos and Polyzos claims that new transport protocols compatible with TCP are needed to maximise performance over wireless segments. However, they are still aware that TCP modifications may violate its end-to-end semantics – applications that need this reliability must then use additional protocols above TCP. The problem is that applications don't know when these semantics are violated.

Alternatives to TCP modifications are mentioned. Link layer protocols operating over wireless links can use local recovery mechanisms to hide losses from higher layers. Mobile systems use RLP to enhance link reliability in this way (chapter 5.2.5). Another approach is to add some link layer functionality to IP (which belong to the network layer) to take care of local recovery. Xylomenos and Polyzos reports: "The local error recovery module snoops on all IP datagrams to gather TCP data and acknowledgement information. Buffered TCP data that need to be retransmitted are inserted into the data stream transparently to the receivers" [Ref. 24, p.61]. They support the conclusion that TCP and link layer coupling reduce link layer overhead and avoidance of conflicts between local and TCP retransmission.

At the end of their article the two authors dwell on future wireless solutions. The concept of hierarchical cellular systems are explained; going from macro all the way to pico cells and corresponding satellite, terrestrial and in-building coverage. Such systems will allow both horizontal (as in today's GSM) and vertical handoffs. In the process, steps must be taken to avoid TCP mistaking handoff-losses for congestion. Such hierarchical systems introduce challenges. With pico cells, the numbers of handoffs will necessary increase. Since user bandwidth will be higher, more data can potentially be lost during handoffs. Xylomenos and Polyzos argue that fast, localized error recovery is the most efficient way to deal with these challenges. "Adapting to these link-specific variations is much easier locally, at the link layer, where the peculiarities of each medium is known" [Ref. 24, p.62].

5.4.3 Conclusions and proposed further research

This chapter sums up what issues Xylomenos and Polyzos feel are important to investigate further. Shortcomings in existing protocols force the following proposed enhancements:

- The use of non-transparent RLP to mask wireless errors is not adequate for all transport protocols and applications. UDP-based real-time traffic is an example of this. Modifications at the transport layer are generally difficult to implement in hierarchical mobile systems. The authors propose a multiprotocol approach to wireless link enhancements.
- Wireless link losses can be efficiently recovered using local mechanisms. Still, higher layers must handle handoff pauses and handoffs to different networks. The authors claim "… we need to deal with each wireless link problem at the appropriate layer."
- Multiple transport protocols should be supported on the Internet without having to worry about specific link details. This is possible if link layer protocols can support flexible and general services instead of specific transport protocols. Xylomenos and Polyzos state that "protocol layering and isolation should be adhered to during design."
- The extension of wireless links will eventually cause many protocols to review their assumptions and modify their mechanisms. Adaptive mechanisms, flexibility and advanced link layer services will become a key feature for emerging protocols. "Lower layer protocols could provide functionality to support future higher layer needs."

These points can be summed up using the words of the authors: ”We advocate a more synergistic approach between layers, where generic end-to-end requirements are supported by customized local mechanisms. We believe that making interfaces between protocol layers richer in information content could pave the way for smarter and better performing future Internet protocols” [Ref. 24, p.62].

5.4.4 Comments

The authors do no actual testing in this article. They rely mostly on others research in the area, and only once do they refer to own research. To depend and base conclusions on what other researchers have accomplished, should be done with caution. The accuracy of referred test-beds is not discussed in the article. This makes it difficult to validate the basis for bandwidth calculations etc [Tab. 8].

TCP and its behaviour in mobile systems are not examined in context of a specific digital system. Although the authors explain certain characteristics of IS-54 (TDMA) and IS-95 (CDMA), it is hard to see connections between theoretical background and presented calculations. Real-world scenarios studied by important scientists (R. Ludwig, B. Rathony et al) are not mentioned in the article. Some of these studies show that TCP throughput over e.g. GSM is mostly ideal and that spurious timeouts are rare. Xylomenos and Polyzos do not discuss these findings even if they tend to disagree with each other on important points.

It is suggested that one should deal with wireless link problems at the appropriate layer. This can be an effective method for combating weakness at local parts of the path. However, it is vital that the semantics of TCP are accomplished. Applications must be able to rely on TCP and that it performs in accordance with its expected reliability.

Several approaches above and below the transport layer are suggested. Local recovery mechanisms can be used to hide losses from higher layers. But actual interactions between e.g. TCP and RLP are not studied or referenced. The discussion is instead focused on tasks of each individual layer. Without specific measurements it is hard to agree on suggestions for richer information content among layers.

5.5 Review of TCP, article 2

The second article reviewed here is titled “Improving Performance of TCP over Wireless Networks” and is written by Bikram s. Bakshi, P. Krishna, N. H. Vaidya and D. K. Pradhan, Texas A&M University. The article appeared in “IEEE Proceedings of the 17th International Conference on Distributed Computing Systems 1997. May 1997, pp. 365-373 [Ref. 29].

5.5.1 Scope

In this paper the authors study the effect of:

- Burst errors on wireless links
- Packet size variation on the wired network
- Local error recovery at the base station
- Explicit feedback by the base station

Everything is related to the performance of TCP over wireless networks. Experiments are performed using the Network Simulator (NS) from Lawrence Berkeley Labs.

The article starts with a description of TCP over wireless links. How TCP reacts to losses are described in details. A summary of previous approaches to the problems described is also given, before the authors discuss their own findings in light of relevant theory.

5.5.2 Research and background

The authors refer to both their own publications [Ref. 30] as well as others research on the subject. As for the article reviewed in chapter 5.4 most references are to IEEE material. This seems to be a very usual approach among IEEE members and researchers.

Bakshi et al state the following about TCP and a wireless environment: ”Errors on wireless links tend to be frequent and bursty, and are highly sensitive to direction of propagation, multipath fading and general interference. In a wireless network thus, TCP will encounter packet losses that may be unrelated to congestion” [Ref. 29, p. 366]. The authors also explain the possible disadvantage of using small frame size over wireless networks, namely that each packet gets fragmented when transmitted. “Loss of a single fragment over the wireless link will initiate error recovery and congestion control mechanisms at the source, causing noticeable performance degration” [Ref. 29, p. 366].

The article does not consider handoffs when performance in wireless networks is discussed. The authors are only interested in the performance of TCP (for bulk data transfer) in the presence of losses. This fact should be taken into account when analysing and verifying the results reported in the article.

Two performance metrics are of interest for Bakshi et al in this article:

- Goodput – the ratio of useful data received at the destination and total amount of data transmitted by the source.
- Throughput – the ratio of total data received by the end user and connection time.

Based on these, the authors propose two approaches to improve TCP performance. They are packet size variations and explicit feedback, and form the basis for the performed research.

Bakshi et al present a summary of previous approaches to the subject. Some of these approaches are discussed, and what they find to be shortcomings are identified. For the split-connection approach the following is reported: “… it violates the semantics of end-to-end reliability. This is because acknowledgements can arrive at the source even before the packet actually reaches the intended destination. Secondly, this approach requires a lot of state maintenance at the base station” [Ref. 29, p. 366].

The authors refer to proposals on incorporating a transport layer aware agent (snoop agent) [Ref. 31] at the base station. Such an agent would cache TCP packets destined for the mobile host and perform local retransmission when losses are detected through DUPACKs and timeouts. Bakshi et al comments: “However, a timeout can occur at the source, and congestion control procedures invoked, while the snoop agent is trying to resend lost packets to the mobile host. Moreover, both snoop and the split-connection approaches do not perform well in the presence of bursty losses on the wireless links” [Ref. 29, p. 366].

On the concept of Channel State Dependent Packet (CSDP) link level scheduling [Ref. 32] the authors find reason to comment. CSDP can be used if multiple TCP connections share the same link, which naturally will be the case in future mobile systems. It is claimed that scheduling protocols like round robin provide significant performance improvements over FIFO. Bakshi et al declare: “The main limitation of this approach is that the performance improvements achievable depends mostly on the accuracy of the channel state predictor. The problem of source timeouts exists in this approach too”.

5.5.3 Test procedure

Opposed to the work reviewed in chapter 5.4, Bakshi et al performs real simulations to support their findings. The actual testing and simulation performed in the project was done using the Network Simulator (NS) from Lawrence Berkeley Labs. The simulator is available by anonymous FTP from ftp://ftp.isi.edu/nsnam. The authors did incorporate extensions to simulate wireless links and evaluate the performance of their proposed schemes. NS is built using C++ and Tcl/Tk and can simulate various flavours of TCP available today for wired networks. Bakshi et al uses TCP-Tahoe throughout their simulations.

NS requires a set of parameter settings in order to provide usable and consistent results. The authors present the settings chosen for their model; the most important ones are reproduced here:

- Error model: A burst error model characterized by a two state Markov model is used to implement errors on the wireless link. In each state (good or bad), bit errors are Poisson-distributed with a certain Bit Error Rate (BER). The transition between the two states is also Poisson-distributed with a certain mean transition rate.
- Overhead: A packet from the wired network of length W bytes becomes 1.5 W bytes after addition of extra overhead due to framing, error correction, segmentation and synchronization.
- Bandwidth: Symmetrical, 19.2 Kbps is assumed. Effective link bandwidth after FEC etc have been removed equals 12.8 Kbps.
- Delay: Transmission delay and propagation delay are the main delay components. Only one connection is assumed being served by the base station, MAC delay is therefore assumed to be negligible.

The following simulation set-up was implemented in NS:

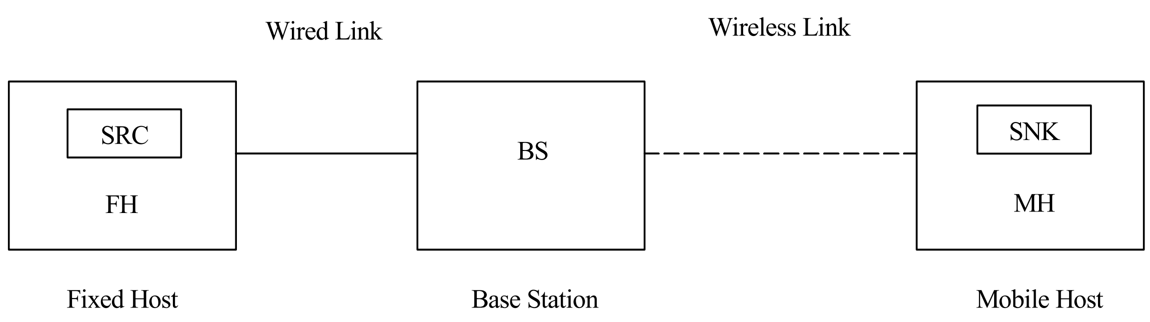


Figure 11 NS simulation set-up

There are three nodes in the model: a fixed host (FH), a base station (BS) and a mobile host (MH). A wired link running at 56 Kbps connects FH and BS, while the wireless link (19.2

Kbps) connects BS and MH. Since only bulk data transfer is of concern for the authors, a TCP source (SRC) and a TCP sink (SNK) is embedded in FH and MH respectively.

5.5.4 Reported results

The authors explain advantages of small MTU in wireless networks. A key point is that the probability of packets getting corrupted during transmission over wireless medium is reduced. With packet fragmentation, end hosts are relieved from worrying about the size of their data segments even though intermediate links may have largely different MTU sizes. Routers at the end of such intermediate links are responsible for fragmenting and reassembling the packet. Bakshi et al states: "... dropping or corruption of a single such fragment will result in the whole packet being dropped. The source would then have to retransmit the entire packet causing more fragments to litter the network and compound congestion problems" [Ref. 29, p. 368].

TCP and its sensitivity to error characteristics of the link are explained. It is showed that TCP is also affected by packet size on the wired network. If Path MTU Discovery (PMTU) is used to decide packet size for a TCP connection, the size will be chosen equal to the smallest MTU (usually the wireless link) along the path. The authors do not believe this to be an optical approach: "Our results indicate that for most error conditions, the optimal packet size differs from the MTU on the wireless links as well as the default IP datagram size (576 bytes)" [Ref. 29, p. 368]. This could be implemented by maintaining a fixed table at each base station. A particular wireless link error characteristic is mapped to the "good" packet size for that specific error characteristic.

A simple experiment illustrates the effect of explicit feedback on TCP. Bulk data is transferred from the fixed to the mobile host. Basic TCP, local recovery and explicit feedback are used – results are presented as figures that show packet throughput vs. time. From the presented results, it seems evident that explicit feedback from base station to source is more efficient than basic TCP and local recovery. On the problem of redundant retransmission, Bakshi et al propose: "This problem will not arise if TCP implementations use very coarse timers. But recent proposals advocate the use of finer granularity timers, as this will increase the sensitivity of the source TCP to congestion on the network" [Ref. 29, p. 369]. Current TCP implementations have a timer granularity of 300-500 ms, for the experiments covered by the report a granularity of 100 ms is used.

Based on their testing, the authors claim: "Explicit feedback from the base station can completely eliminate the possibility of timeouts occurring at the source, while the wireless link is in a bad state" [Ref. 29, p. 369]. The authors hence wish to explore the possibility of using existing feedback mechanisms for improving TCP performance.

- Explicit Congestion Notification (ECN): An Internet Control Message Protocol (ICMP) sources quench is a host's way of informing the source about congestion in the network. The notification is sent to the source after a packet has been dropped, or if the host anticipates packet dropping based on existing congestion conditions. This way congestion is predicted and measures may be taken before dropping occurs. Interestingly, the results found by Bakshi et al show otherwise. They claim that a situation where the host times out due to unacknowledged packets, will not get better by source quench messages from the base station. Packets already on the network will still time out. (It will of course reduce the flow of new packets and reduce their probability of timing out). To cope with this problem, Bakshi et al propose a mechanism to update the TCP timer at the source: "This will essentially thwart the source's attempt to invoke congestion control in response to delays on the wireless link" [Ref. 29, p. 370].

- Explicit Bad State Notification (EBSN) is the proposed solution for the situation described above. EBSN would cause the previous timeout at the source to be cancelled and a new timeout put in its place. This should be based on the existing estimate of RTT and variance: a large timeout value could cause deadlock and no timeout at the source; a very small value could make the source timeout before the next EBSN arrived. EBSN are sent to the source after every unsuccessful attempt by the base station to transmit packets over the wireless link. The correct timeout value at the source is readjusted upon the receipt of the first new ACK from the MH after any EBSN messages received. Even though a new ACK may cause a large variation in RTT calculations, it is a good indicator that the link has quit bad state and entered a good state. Ultimately, it should also result in reduced variance. The conclusion is: " … using EBSN prevents unnecessary timeouts at the source thus decreasing RTT variance and improving throughput" [Ref. 29, p. 371].

Packet size on the wired network was varied from 128 to 1536 bytes. Each run involved a 100 Kbytes file transfer from FH to MH. Mean length of good period was 10 sec, bad period ranging from 1 to 4 secs.
The results seem clear. For a given packet size, throughput increases as the length of the bad period decreases. For each bad period length there is an optimal packet size that delivers the maximum throughput. The "right" size yields about 30 % improvements in throughput compared to basic TCP. For TCP using EBSN, the results are even clearer. In the absence of timeouts there are no redundant retransmissions from the source. The throughput now increases with increase in packet size, since performance is no longer sensitive to fragmentation over the wireless link, and larger packets perform better.

5.5.5 Conclusions

Bakshi et al concludes: "Results show that an optimal packet size can lead to about 30 % performance improvement over a non-optimal packet size. We have shown that the optimal size varies with error conditions on the link. To prevent timeout while the base station is performing local recovery, an explicit feedback mechanism called Explicit Bad State Notification is proposed. When the source receives an EBSN the timeout value is reset – source timeouts will not interfere with local recovery" [Ref. 29, p. 373]. It is observed that TCP using EBSN provides up to 100 % performance improvements over basic TCP in wide-area wireless networks, and up to 50 % performance improvements in local-area wireless networks. The authors claim that these results are for a very conservative error model for wireless links. They expect greater improvements for wireless links having higher BERs.

5.5.6 Comments

As opposed to G. Xylomenos and G. C. Polyzos, Bakshi et al base their conclusions on simulation and testing. The tests are not performed in a live and running wireless environment, but implemented in a software simulator. Even though such a simulator can be a very powerful tool to verify general principles, it is extremely difficult to capture the true performance in "real-world" systems. NS is based on wired network behavior. Other research communities have added support for wireless schemes later. NS is not a polished and finished product, rather the results of on-going efforts of research and development. These factors combined suggest that performance obtained from the simulator should be met with healthy skepticism.

The article does not consider handoffs in wireless systems. Handoffs are considered a major challenge for end-to-end transport protocols like TCP. In future mobile systems where high capacity is an important goal, cell-sizes will be smaller and the number of handoffs between bases stations will increase. It is not discussed how this would influence the solutions proposed in the article.

The chosen simulation setup is simple with only one fixed host, one bases station and one mobile host. This gives an environment that is easy to control and with few possible error factors. In accordance with this, the error model implemented is fairly easy. With only two states – good or bad – it is difficult to foresee how the proposed algorithms would react to a more sophisticated environment.

The idea of finding optimal packet sizes for each wireless link seems to be a good one. The results obtained from testing supports this. However, one might see problems about keeping lists of different error characteristics in each base station. Theoretically, each user in the mobile cell could experience individual error distinctiveness. As the users move around, their link quality may change also. To adjust packet size continuously for each connection could put a solid strain on the base station's performance and maybe challenge its main functionality.

6 Buffering and dropping of data packets

This section investigates some existing and proposed algorithms deployed in network nodes to reduce consequences of congestion. The discussion is a natural extension of what has been reported in the previous chapter, “TCP rate adaptation related to the radio link”.

6.1 *Introduction*

Most readers of this thesis document are probably aware of the rapid advances in the area of wireless communications and the popularity of the Internet. Existing and future packet data services over wireless links, are gaining importance. Most of these applications make use of reliable end-to-end transport layer services provided by TCP.

It has been explained how TCP reacts to packet losses. The size of the congestion window is dropped, congestion control or avoidance mechanisms are initiated, and the retransmission timer is reduced. These measures result in reduced load on intermediate links and thus controlled congestion in the network.

However, when packets are lost for reasons other than congestion, these measures lead to unnecessary reduced end-to-end throughput. To cope with this, schemes have been developed to ease the effects of non-congestion related losses on TCP performance over high-loss links.

There are two different approaches to improve performance in such systems. The first approach hides any non-congestion related losses from the TCP sender. This way, no changes to existing sender implementation is required. Local problems are dealt with locally, and the transport layer does not need to be aware of individual link characteristics. Such mechanisms are investigated in chapter 6.2. The second approach attempts to make the sender aware of wireless links, and realize that some losses are not due to congestion. The sender can then stay away from invoking congestion control algorithms when there is no need for it.

6.2 *Algorithms in use*

This section contains a selection of congestion control schemes used in TCP/IP networks. The selection embraces both source and router/gateway based schemes.

6.2.1 Background

TCP’s congestion control algorithm is necessary and powerful but is not capable of providing good service inn all situations since the network is treated like a black box. Some control mechanisms are needed in the routers to complement the end system congestion control as well.

6.2.2 First In First Out (FIFO)

The idea of FIFO queuing is simple: The first packet that arrives at a router is the first packet to be transmitted. The amount of buffer space at each router is finite, so when a packet arrives and the queue is full, the packet is discarded. This is done without regard to which flow the packet belongs to or how important the packet is. This action is sometimes called tail drop, since packets that arrive at the tail end of the FIFO are dropped.

FIFO with tail drop is the simplest of all queuing algorithms. It is also the most widely used in Internet routers today [Ref. 14, p.459]. This simple approach to queuing pushes all responsibility for congestion control and resource allocation out to the edges of the network. Because of this, TCP must take responsibility to detect and respond to congestion.

A simple variation of basic FIFO queuing is priority queuing. Each packet is marked with a priority; the mark could be carried e.g. in the IP Type of Service (TOS) field. Routers then implement multiple FIFO queues, one for each priority class. The router transmits packets out of the highest priority queue if it is nonempty, before moving on to the next priority queue. Packets are still managed in a FIFO manner inside each priority.

6.2.3 Fair Queuing (FQ)

FIFO does not separate packets according to which flow they belong. This is a problem at two levels: First, congestion control algorithms at the source may have a problem to control congestion with so little help from the routers. Second, since the entire congestion control mechanism is implemented at the source, and FIFO does not provide a means to control how well the sources adhere to this mechanism, an ill-behaved source (flow) can capture a large fraction of the network's capacity. Inside Internet an application that do not use TCP may bypass its end-to-end semantics. The application is then able to flood the Internet's routers with its own packets, thereby causing packets from other applications to be discarded.

FQ is designed to address this problem. The idea of FQ is to maintain a separate queue for each flow currently being handled by the router. These queues are then serviced in a round-robin manner. When a flow sends packets to quickly, its queue fills up. When such a queue reaches a particular length, additional packets belonging to that flow's queue are discarded. This way, a given source cannot increase its share of router capacity at the expense of other flows.

FQ does not involve the router telling the traffic sources about the state of the router or in any way limit how fast a source sends packets. It should be used in conjunction with an end-to-end congestion control. FQ simply segregate s traffic so that an ill-behaved source do not interfere with those that faithfully implement the end-to-end algorithm.

6.2.4 Weighted Fair Queueing (WFQ)

WFQ is another variant of FQ. The key point of WFQ is that it allows a weight to be attached to each flow (queue). This weight logically specifies how many bits to transmit each time the router services that queue. Doing this, the router efficiently controls the percentage of the link's bandwidth that the flow will get. Simple FQ gives each queue a weight of one, which means logically only one bit is transmitted from each queue each time around. The result is that each flow gets 1/n of the bandwidth when there are n flows. With WFQ, however, one queue might have a weight of two; a second a weight of one and a third queue might have a weight of three. If each queue always contains a packet waiting to be transmitted, the first flow will get one-third of the available bandwidth, the second will get one-sixth and the third will get one-half of the available bandwidth.

WFQ could also be implemented on "classes" of traffic, and not just in terms of flow. Such classes would be different in some other way than the simple flows discussed so far. One possibility is to use the TOS field in the IP header to identify classes, and allocate a queue and a weight to each class.

Note that a router performing WFQ must learn what weights to assign to each queue. This can be done either by manual configuration or by some sort of signalling from the sources. For the latter, router handling would move towards a reservation-based model. But simply assigning a weight to a queue is still a rather weak form of reservation because these weights are only indirectly related to the bandwidth the flow receives.

6.2.5 Stochastic Fairness Queuing (SFQ)

SFQ is a simple implementation of the fair queuing algorithm family. It is less accurate than others are, but it also requires fewer calculations while being almost perfectly fair.

SFQ is not quite deterministic, but is reported to work fairly well. Its main benefits are that it requires little CPU and memory. 'Real' fair queuing requires that the kernel keep track of all running sessions.

The key word in SFQ is conversation (or flow), being a sequence of data packets having enough common parameters to distinguish it from other conversations. The parameters used in case of IP packets are source and destination address, and the protocol number.

SFQ consists of dynamically allocated number of FIFO queues, one queue for one conversation. The discipline runs in round robin, sending one packet from each FIFO in one turn, and this is why it's called fair. The main advantage of SFQ is that it allows fair sharing the link between several applications and prevents bandwidth take-over by one client. SFQ however cannot determine interactive flows from bulk ones.

6.2.6 Token Bucket Filter (TBF)

The Token Bucket Filter (TBF) is a simple queue that only passes packets arriving at rate in bounds of some administratively set limit, with possibility to buffer short bursts.

The TBF implementation consists of a buffer (bucket), constantly filled by some virtual pieces of information called tokens, at specific rate (token rate). The most important parameter of the bucket is its size, that is number of tokens it can store.

Each arriving token lets one incoming data packet out of the queue and is then deleted from the bucket. Associating this algorithm with the two flows - token and data - gives us three possible scenarios:

1. The data arrives into TBF at rate equal the rate of incoming tokens. In this case each incoming packet has its matching token and passes the queue without delay.
2. The data arrives into TBF at rate smaller than the token rate. Only some tokens are deleted at output of each data packet sent out the queue, so the tokens accumulate, up to the bucket size. The saved tokens can be then used to send data over the token rate; if short data burst occurs.
3. The data arrives into TBF at rate bigger than the token rate. In this case filter overrun occurs -- incoming data can be only sent out without loss until all accumulated tokens are used. After that, overlimit packets are dropped.

The last scenario is very important, because it allows to administratively shaping the bandwidth available to data, passing the filter. The accumulation of tokens allows short bursts of overlimit data to be still passed without loss, but any lasting overload will cause packets to be constantly dropped.

6.2.7 Random Early Detection (RED)

Sally Floyd and Van Jacobson invented RED in the early 1990s. It is designed to work together with TCP, which detects congestion through timeouts or DUPACKs. RED implicitly notifies the source of congestion by dropping one of its packets.

The RED gateway/router calculates the average queue size using a filter with an exponential weighted moving average. The average queue size is compared to two thresholds, a minimum and a maximum threshold. When the average queue size is less than the average threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are dropped, or if all sources are cooperative, this ensures that the average queue size does not significantly exceed the maximum threshold [Ref. 33].

When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked (dropped) with some probability **P** [Fig. 12]. This probability P increases slowly when the average queue size moves toward the maximum threshold, and equals 1 when maximum threshold is reached. The reason behind this is that when average queue size gets this big, dropping a few packets every now and then will not work – more drastic measures are needed. Hence, all arriving packets are dropped.

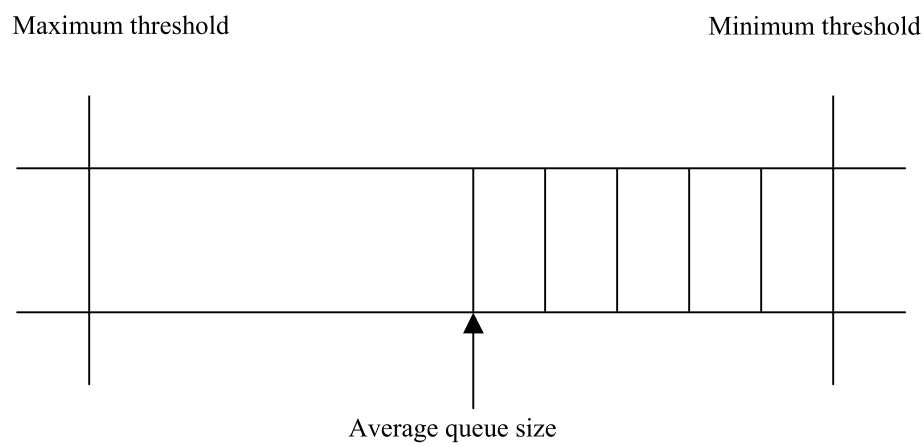


Figure 12 RED thresholds on a FIFO queue

The probability P is actually a function of both average queue size and how long it has been since the last packet was dropped. It is registered in a variable called **count** how many packets have been queued (not dropped) while the average queue size has stayed between the two thresholds. P slowly increases as count increases, this way the probability for a packet drop increases as time since the last drop increases. This makes closely spaced drops less likely than widely spaced drops. This extra step in calculating P was introduced by the inventors of RED when it was revealed that without it, packet drops were not well distributed in time, but occurred in bursts. Since packet arrivals from a certain connection are likely to arrive in bursts, this clustering of drops results in multiple drops for a single connection. It has been discussed in this thesis how only one drop per RTT is needed to cause a connection to reduce the window size, and how multiple drops might send it into Slow-start.

The desired behaviour for RED is that it should drop a small percentage of packets when average queue size exceeds Minimum threshold. The should cause a few sources to reduce their window size, which in turn will reduce the rate at which packets arrive at the router. If things go well, average queue size will decrease and congestion is avoided. The queue length is kept short, while throughput remains high since few packets are dropped.

RED operates with a queue size that is averaged over time. Its instantaneous queue size may very well be much longer than the average size; packets arriving when this is the situation are dropped, since there is nowhere to put them. When this happens, RED operates in tail drop mode. One of the goals of RED is to prevent tail drop behaviour if possible.

The random behaviour of RED adds an interesting aspect to the algorithm. Since packets are dropped randomly, the probability that RED drops a packet from a particular flow is roughly proportional to the share of bandwidth this flow gets at the router. A flow that sends a large number of packets provides more candidates for random dropping. Because of this, one might claim that there is some kind of fair resource allocation built into RED [Ref. 14, p. 480].

7 A suggestion of test set-up

7.1 Introduction

This chapter gives information on how selected queue strategies was tested on some services. It describes the test-bed and how it was set up and configured. A discussion on how different test scenarios were evaluated is included.

7.2 Test-beds investigated

7.2.1 Testing and its background

This Masters Thesis is not only concerned with a theoretical survey. Although the research is an important tool to discover what is considered core problems in the field of interest, its main purpose is to provide basis for a realistic investigation. A proper and reliable test-bed can in most cases confirm or weaken proposed solutions to a problem.

In this thesis the goal was to test different ways data packets are handled when processed in intermediate nodes. Different strategies should be employed on one or more services to see if and how they influence the end-user's experience. If possible, one should also find an objective way to evaluate the effects.

7.2.2 The original proposal for test-bed

The initial problem description has no fixed suggestion on how to implement a suitable test-bed. However, it proposes that one might base the testing on different terminal adapters, existing software and Ericsson Access Server. The idea was to connect a PC (running as gateway/router) to the Internet through modem and Access Server, modify data passing through the PC, and send the data to a machine running selected services.

This suggestion for a test-bed was adapted and put on trial. To keep things simple and flexible, the mentioned Access Server was disregarded. Instead, a local ISP provided Internet connection. The following figure describes the proposed test-bed [Fig. 13].

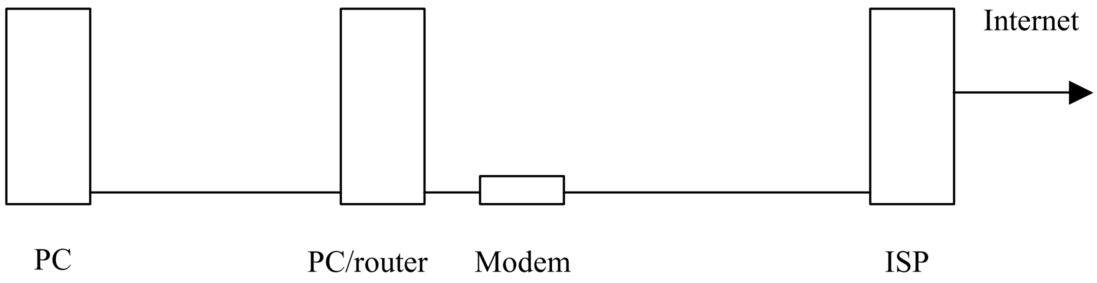


Figure 13 The original test-bed

How to actually modify the data passing through the router was not immediately clear. Several schemes were discussed with the assigned supervisor at Ericsson, for example the possibility to use DOS-related commands to route and alternate the data stream.

7.2.3 Network Simulator (NS) 2

NS is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS began as a variant of the REAL network simulator[2] in 1989 and has evolved substantially over the past few years. Currently NS development is supported through Defence Advanced Research Projects Agency (DARPA) and Collaborative Simulation for Education and Research (CONSER) at University of Southern California.

NS is a popular and powerful tool for simulating various network scenarios. QoS strategies like Differentiated Services (DiffServ) and Integrated Services (IntServ), multicast routing, different versions of TCP and congestion control as well as several queuing strategies can be simulated. This is reflected by the fact that many research projects use the simulator to examine network related issues[3].

Simulators have both advantages and drawbacks. Despite the cost of custom simulators, wide-area test-beds, and small-scale lab evaluations they all have some important advantages. Since such equipment use real code, experiments run in test-beds or labs automatically capture important details that might be missed in a simulation. However, building test-beds and labs is expensive, reconfiguring and sharing them is difficult, and they are relatively inflexible. Software simulators on the other hand, usually don't require expensive licenses. In addition, one might find valuable help on the Internet through mail-lists and discussion forums. Freeware simulators tend to create active and strong groups of supporters; this makes it possible to verify own simulations and results through comparison with other users.

It was considered to use NS to test various queuing strategies throughout this thesis. Due to the support for queuing algorithms in NS, it is possible to try each algorithms effect on data passing through the router. NB also has a visualisation tool called Network Animator (Nam). Nam lets the user view network simulation traces and real-world packet traces. It supports topology layout, packet level animation and various data inspection tools. Thus, one can actually see how and when packets are put into and taken out of queues. Dropped packets are visualized as well.

7.2.4 Selected hardware and software

Both PC's used in the test-bed [Fig. 13] were standard commercial products. The table below shows their most important components [Tab. 10].

Table 10 Key components in test-bed PC's

Components	PC running as router	Pc running multimedia services
Processor	Pentium-II 266 MHz	Pentium-II 266 MHz
Memory	130 600 MB RAM	130 600 MB RAM
Network adapter	3Com EtherLink XL 10/100 PCI TX NIC (3C905B-TX)	3Com EtherLink XL 10/100 PCI TX NIC (3C905B-TX)
Operating System	Red Hat 7.0 Linux, kernel version 2.4.2	Microsoft Windows 2000 5.00.2195

[2] http://www.cs.cornell.edu/skeshav/real/overview.html
[3] http://www.isi.edu/nsnam/ns/ns-research.html

The original test-bed [Fig. 13] also involved a standard 28.8 Kbps Ericsson modem interconnecting an analog phone line and the serial port of the router.

7.2.5 Problems encountered

The equipment shown in Figure 13 was collected and assembled. The router and PC running multimedia services were interconnected through a crossed network cable. Unfortunately there were problems when connecting to the ISP. A connection was established, the login procedure was successful but the Point-to-Point-Protocol (PPP)-daemon terminated after a few seconds. A Linux PPP-script called vwdial was edited and tested too, but it returned the same result.

It was decided to install NS. NS can be installed either as a single package, or one can download only desired/required tools. For simplicity, the all-in-one package was chosen. The downloading and installation seemed to go well, and its built-in verification tool did not return any errors[4]. Some simple test scripts were included in the distribution. These can be used to test the simulator after installation. The scripts run, returned some simulation parameters to the screen, but the Nam tool never started. This made it difficult to verify if and how the simulator handled incoming parameters.

7.2.6 Summary

As indicated in the previous sections, Linux was chosen to deal with routing issues. Linux is freeware, and has tools that can handle advanced networking schemes. Chapter 7.3 gives information on how some of these tools were used to configure the final test-bed.

DOS was early in the project period a possible candidate to handle routing, queuing and data flow influence. DOS is suitable if one wants to shuffle data between serial ports. It may be able to detect certain patterns in the data flow (e.g. the start of PPP frames etc), but to simulate specific queuing algorithms demand a more sophisticated strategy. Based on this DOS did not qualify for use in the test-bed. It could not provide the necessary tools to implement RED, TBF, SFQ and other queuing algorithms.

NS seems to be a powerful tool for testing different network configurations. Its strength lies in the possibility to combine wired and wireless networks in a virtual test-bed, and to study protocol interactions when data travels through different network nodes. Another strength is the ability to implement queuing algorithms (NS supports a flavour of such algorithms) on user defined data flows. What it cannot do is to act as real router in a real test-bed. NS is still a simulator and cannot use its built-in algorithms to alternate data flowing through a PC. This way it is not really suitable for testing together with mobile services. A user of such services would then not experience any changes for different queuing algorithms.

7.3 The chosen test-bed

7.3.1 Test-bed

The next figure describes how the final test-bed was put together. It differs from the proposed test-bed [Fig. 13] by exchanging the modem interface for a network adapter similar to the one described in [Tab. 10]. The network adapter connecting the router to the ordinary PC was called Eth1, while the network adapter connecting the router to Internet is was called Eth0.

[4] See http://www.isi.edu/nsnam/ns/ns-tests.html

These names are referred throughout the installation description, so the reader should make a mental note to distinguish them from each other.

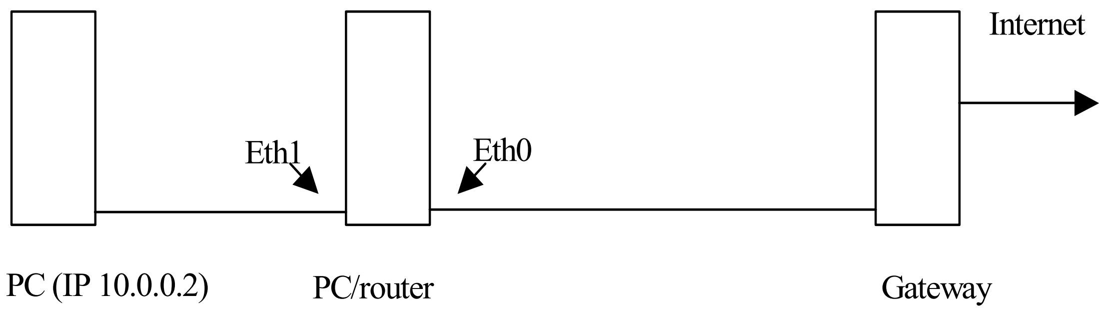


Figure 14 The chosen test-bed

The test-bed was not installed in a separate test environment. It ran in a live student network together with other PC's and network nodes, sharing Domain Name Server (DNS) and gateway to Internet with other computers.

7.3.2 Network Address Translation (NAT)

NAT is the translation of an IP address used inside one network to a different IP address known within another network. One network is called the inside network and the other is called the outside network. Typically, a company will map its inside local network addresses to one or more global (valid) IP addresses. The process is reversed for incoming IP packets. This has several advantages. Security is ensured since outgoing and incoming requests must go through a translation process. Requests can then be qualified, authenticated or matched to a previous request. NAT also limits the number of global IP addresses that a company needs. A single IP address can be used in its communication with the Internet.

IP masquerade is the tool used by Linux to implement NAT. A very hands-on approach to IP masquerading can be found in "Linux Network Administrators Guide, 2nd ed."[Ref. 34]. The chosen test-bed for this thesis project [Fig. 14] implements masquerading like this: The PC routes its IP packet to the masquerade router, which identifies this connection request as requiring masquerade services. It accepts the packet and allocates a port number to use, substitutes its own IP address and port number for those of the PC, and transmits the packet to the destination host. The destination host believes it has received a connection request from the Linux masquerade router and generates a reply packet. The masquerade router, upon receiving this IP packet, finds the association in its masquerade table and reverses the substitution it performed on the outgoing IP packet. It then transmits the reply packet to the PC.

7.3.3 NAT set-up

To enable NAT (masquerading) in a router is quite straightforward. First, make sure to enable masquerading when configuring the kernel. For the kernel mentioned in [Tab. 10], this is done by choosing "Y" in the section called "Networking options", for the choice "Network packet filtering (replaces ipchains)". When the kernel is compiled and installed, typing this command sets up masquerading:

```
[root@badjoke ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Any data passing out through eth0 will automatically be masqueraded by the Linux router. To list the rules created, use the -L argument. The router returns this when prompted:

```
[root@badjoke ~]# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --   anywhere            anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

This tells that all protocols from all sources are target for the masquerade rule.

7.3.4 Iproute2 and Traffic Control (TC) tools

Iproute2 is a powerful network tool that comes included in Linux kernel 2.2.x and up. For other kernels it is available via ftp://ftp.inr.ac.ru/ip-routing/iproute2-2.2.4-now-ss.tar.gz.

Traffic Control (TC) is a sophisticated system for bandwidth provisioning included in Iproute2. It supports various methods for classifying, prioritising, sharing and limiting both incoming and outgoing traffic. As an example, one can use the following command to display the interfaces in the router:

```
[root@badjoke /etc]# ip link list
1: lo: <LOOPBACK,UP> mtu 3904 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
    link/ether 00:a0:24:4d:a2:ab brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc cbq qlen 100
    link/ether 00:a0:24:a9:36:ad brd ff:ff:ff:ff:ff:ff
4: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
5: gre0@NONE: <NOARP> mtu 1476 qdisc noop
    link/gre 0.0.0.0 brd 0.0.0.0
```

Here we see each interface and some of their settings. The current MTU of each interface, queuing discipline (qdisc) with buffer size (if any), link type and MAC address. Note the absence of IP addresses. Iproute2 disconnects the concept of "links" and "IP address". With NAT, IP addresses are kind of irrelevant because of their translation when passing the router. The reader should also note the different queuing algorithms for eth0 (pfifo_fast, equal to FIFO) and eth1 (CBQ, a super class algorithm).

8 An evaluation of suitable multimedia services

8.1 Introduction

The purpose of this chapter is to see if and how different queuing algorithms influence the way a user perceives selected multimedia services. The focus is on subjective quality and whether it is possible to find objective methods to distinguish between "good" and "bad".

8.2 Services

8.2.1 Issues to consider when selecting services for testing

One of the goals for this thesis project was to identify possible IP based multimedia services. The services should be chosen because of their suitability or interest for mobile users. The research revealed a large number of appropriate multimedia services. Which service to use when testing queuing algorithms was not obvious. First of all the service(s) should be well known and familiar to most users. An experimental service might not behave in a steady and consistent way. Different platforms and network conditions could alter its performance. It was vital to locate a service that behaved in a consistent manner even in a changing environment.

The selected service should have the potentiality to run without major problems in a future mobile system like UMTS. However, such systems were not publicly available when this document was written. Qualified guessing and technical specifications formed the basis for final choose of service.

8.2.2 Selected service

An important part of this thesis was to see how TCP behaves in a wireless environment. Earlier discussions pointed out how TCP starts congestion avoidance mechanisms even if the problems are not related to congestion at all. To achieve a smooth performance from IP based multimedia services, it seems realistic to expect some kind of buffering at the mobile host. Based on these assumptions, the service (application) chosen to test queuing algorithms is streaming video.

Streaming video is per definition a multimedia product since it consists of moving pictures and audio. This was discussed in chapter 3.2. Streaming is also on of four traffic classes defined by 3GPP for UMTS "release-99". Streaming is very popular on the Internet. It is used in a vide variety of settings, from entertainment (e.g. "funny videos") to news (e.g. interviews and important happenings). Many radio stations broadcast their shows on the Internet as well. It also is likely that users will demand streaming services, audio or video, when they are mobile. Based on this, streaming video is a valid candidate for testing.

8.3 Queuing algorithms

8.3.1 Using Class Based Queuing (CBQ) for bandwidth management

Linux kernels 2.2.x and up comes with everything needed to do advanced bandwidth management. It is possible to manage bandwidth in ways comparable to highly advanced (and expensive) dedicated bandwidth management systems.

TC has to basic components called filters and queues. Filters place traffic into queues, and queues gather traffic and decide what to send first, send later, or drop. There are several flavours of filters and queues, as explained in chapter 6.2. The most common filters are fwmark and u32, the first makes it possible to use the Linux netfilter code to select traffic, and the second allows one to select traffic based on ANY header. The most notable queue is Class Based Queue (CBQ). CBQ is a super-queue; this means that it contains other queues (even other CBQs).

8.3.2 Selected algorithms

CBQ supports all algorithms mentioned in chapter 6.2. Thus, one stands free to implement the algorithm of choice into the kernel. To narrow down the selection of settings for the test-bed, it was decided to test only two algorithms against each other. This makes it easier to control factors that can affect the test results. The fewer factors and settings one must fiddle with, the better.

SFQ and PFIFO were selected for testing. PFIFO is a Linux edition of the FIFO algorithm, and employs tail dropping when the buffer run full. As mentioned, FIFO is the most used algorithm in routers today.

8.3.3 Installation of selected algorithms

This shows how CBQ was used to limit the bandwidth on eth1 (interface for the PC running streaming video):

First, traffic control is set up using the Iproute2 tool TC:

```
[root@badjoke ~]# tc qdisc add dev eth1 root handle 10: cbq bandwidth
10Mbit
avpkt 1000
```

This configures the queuing discipline for eth1. “root” denotes that this is the root discipline. Further, its handle is “10:”. CBQ is selected, and we tell the kernel to allocate 10 Mbit for the selected network adapter, and that average packet size is around 1000 octets.

Then the root class is generated. All others descend from this class.

```
[root@badjoke ~]# tc class add dev eth1 parent 10:0 classid 10:1 cbq
bandwidth 1250000 rate 1250000 allot 1514 weight 1mbit prio 8 maxburst 20
avpkt 1000
```

“parent 10:0” denotes that this class descends from the root of qdisc handle “10:”. “classid 10:1” names the class. “1250000” describes how many bytes this class reserves. 1 250 000 equals 10Mbit, so it completely fills the available device. MTU (plus some overhead) is specified as 1514 bytes. “weight 1 mbit” is a tuning parameter.

Next to generate is a test class:

```
[root@badjoke ~]# tc class add dev eth1 parent 10:1 classid 10:100 cbq
bandwidth 1250000 rate 16000 allot 1514 weight 1600 prio 5 maxburst 20
avpkt 1000 bounded
```

This class only gets 16000 bytes per second (equal to 128kbps). This is the key feature of CBQ. The term “bounded” denotes that test class must not exceed 128kbps.

Finally, the kernel gets information on what queuing algorithm to use for the test class:

```
[root@badjoke ~]# tc qdisc add dev eth1 parent 10:100 sfq quantum 1514b
perturb 15
```

Then the kernel must be told what protocol and destination to match with the test class (id 100:10):

```
[root@badjoke ~]# tc filter add dev eth1 parent 10:0 protocol ip prio 100
u32 match ip dst 10.0.0.2 flowid 10:100
```

To turn off these filters and queues is easy. Simply replace the word "add" with the word "del" in each command line.

8.4 Subjective quality and differences

8.4.1 Streaming video target

The streaming video file used in the testing can be found at http://www.ousland.no/. In the lower left part of this site there is a menu called video. The length of the video is about one minute. Based on their Internet connection, users can choose among three versions: 56kbps, 100kbps or 300kbps. When no filter or queue is installed for eth1 (no bandwidth restrictions), there are some differences among the versions: The one with highest bit rate provides a cleaner and more natural view. The sound is also somewhat "fuller" and more natural than for the smaller versions.

Windows Media Player version 6.4.09.1109 played the selected file. The 56kbps video version was used throughout the testing.

8.4.2 Test procedure

Filters can be turned on or off while the router shuffles data to eth1. It is not necessary to stop and start the streaming video for filters to have effect. However, switching between different queuing algorithms is more complex. Although the selected algorithms need few settings to make them operational, it is difficult jump from one algorithm to the other while the video runs.

Direct comparison between the algorithms was difficult. Clearly, some kind of reference was needed to compare each algorithm up against. Running the video without any filtering before each change made this possible.

The test procedure can be summed up like this:

- The streaming video was started. No filtering or queuing was enabled.
- After 10 seconds filtering was turned on. Either SFQ or PFIFO started to process the incoming data stream. Simultaneously, a packet-tracing tool called Ethereal was started at the PC running streaming video. For info on Ethereal, see chapter 8.5.2.
- The video ran until the observed results were perceived as consistent.

The procedure was then repeated for the other algorithm.

8.4.3 Perceived quality with SFQ enabled

After 10 seconds the filter with SFQ algorithm included was enabled at the router. Immediately, there was no change in quality. Both picture and sound were unaffected. It was difficult to see any changes.

After 30 seconds (i.e.20 seconds after the filter was enabled), things happened to the streaming video. The picture began to flick and freeze in an uneven way. Shortly after the picture froze completely. Media Player changed its status from "running" to "buffering"(lower left corner of the Media Player window). Its buffer size increased from 0 to about 60 % before anything happened. The video ran normal for a few seconds before it started flickering and freezing again. A complete freeze occurred again, and Media Player started to buffer. This sequence repeated itself throughout the rest of the video clip. A certain effect was noticeable in the periods just before the video froze completely. Horizontal lines appeared in each video frame. These lines seemed to blur the middle part of the affected frames.

The associated audio halted when the picture froze 30 seconds after start. There were short bursts of audio while the picture flickered. Naturally, there was no audio when Media Player filled its buffer. The audio returned just before the picture did. Next, both audio and picture froze simultaneously. Again, audio returned slightly prior to the picture. In addition, synchronisation between audio and picture was rather poor. The synchronisation problems continued throughout the video clip.

8.4.4 Perceived quality with PFIFO enabled

The PFIFO algorithm was enabled by these commands:

```
[root@badjoke ~]# tc filter del dev eth1 parent 10:0 protocol ip prio 100
u32 match ip dst 10.0.0.2 flowid 10:100

[root@badjoke ~]# tc qdisc del dev eth1 parent 10:100 sfq quantum 1514b
perturb 15

[root@badjoke ~]# tc qdisc add dev eth1 parent 10:100 pfifo

[root@badjoke ~]# tc filter add dev eth1 parent 10:0 protocol ip prio 100
u32 match ip dst 10.0.0.2 flowid 10:100
```

After 10 seconds the filter with PFIFO algorithm included was enabled at the router. As with SFQ there was no immediate effect on the perceived quality. Both picture and sound were unaffected. It was difficult to see any particular changes.

After about 28 seconds something happened. The picture halted and came to a complete stop. Compared to SFQ, there was little or no flickering in the picture before it stopped. Media Player began to buffer, when the movie started its quality was surprisingly good. Subjectively, PFIFO offered better picture quality than SFQ in the "intermediate" periods. Another effect of PFIFO was that transitions from normal to halt were cleaner, or more "brutal". The sequence of buffering, running and buffering repeated itselv throughout the video clip.

The associated audio halted when the picture froze. Again, there was no audio when Media Player filled its buffer. For PFIFO, the audio seemed to return precisely when the picture started up again. Next, picture froze a short wile before audio did. As reported, this was not the case with SFQ. With SFQ enabled, audio and picture froze simultaneously. For the transitions from normal to halt video, there were no clear differences between PFIFO and SFQ. Synchronisation between audio and picture was a bit better with PFIFO. It was difficult to tell if the achieved sound quality was better or worse with PFIFO enabled.

8.5 Objective quality and differences

8.5.1 Introduction

A goal for this thesis project was to see if it is possible to objectively evaluate results from different queuing algorithms. If such a method should exist, it could be used to test existing algorithms against each other. This chapter evaluates a protocol analyser called Ethereal, to see if it can relate changes in data flows to changes in perceived user quality.

8.5.2 The Ethereal analyser

Ethereal is a free network protocol analyser for Unix and Windows. Ethereal is available from http://www.ethereal.com, the Windows version is a single, self-extracting file. It allows the user to examine data from a live network or from a capture file on disk. It is possible to interactively browse the data while capturing. Ethereal has several powerful features, including several filters and the ability to view the reconstructed stream of a TCP session. Ethereal ran on the PC where the streaming video was tested.

8.5.3 Test procedure

Ethereal was started together with the video clip. All data passing the network adapter were traced. Ethereal reported (relative to start) packet number and packet time. It also reported packet source and packet destination, along with protocol name and info on carried data. The intention was to note when the video quality changed, and search for any particular behaviour in the data flow around that time.

8.5.4 Evaluation

Appendix A shows output from Ethereal for the two experiments reported in chapter 8.4.3 and chapter 8.4.4. The figures in Appendix A.1 and A.2 are screen-prints from Ethereal's capture interface.

To study the captured traffic was interesting. If the problems reported from the hands-on viewing could be recognised as peculiar data flow behaviour, one might be able to link specific picture or audio problems to packet retransmission etc. Hence, protocol analysers like Ethereal could prove a valuable tool for quality estimations. Several possibilities would then emerge, for instance to dynamically change from one queuing algorithm to another based on patterns in the passing data flow.

The captured files did not reveal any specific patterns related to picture or audio problems. Packets coming from the server after the reported problems occurred (30, 28 seconds), seemed to arrive at a slightly lower frequency than the earlier ones. This could be due to queuing effects and buffer problems, but increased strain on the network outside the test-bed might cause it, too. Clearly, this is one of the disadvantages of running test in a public environment. One cannot identify the primary cause of delay without the use of more sophisticated equipment.

The protocol analyser used for tracing data flow into the PC running streaming video, did not reveal any specific, quality related information. The captured packets and their arrival rate did not supply sufficient information to draw unique conclusions about quality and queuing algorithms.

9 Discussion

Multimedia
The thesis document explains the concept of multimedia. The four basic media text, audio, images and motion picture can be combined in a many ways. As technology and new standards evolve, one must expect new multimedia services to appear. Different combinations of media put different levels of strain on underlying networks. Such combinations will probably have diverse requirements to delay, bandwidth and jitter for maximum performance. It is also a fact that the number of formats for each basic media is growing. Each format is again optimised to satisfy a specific need: transfer speed, storing size or best quality. Naturally, these factors influence each other. Smaller size usually means poorer quality. As a result, new formats try to improve on all factors without negative effects on a single one. The challenge is then to make different networks perform their best for all kinds of traffic.

Service categorisation
Several methods for categorisation of IP based services are reviewed. It is important to choose a valuable and useful reference point when categorizing these services. Each point of reference focuses on certain aspects in the reviewed services. For example, services can be categorized by their demands of bandwidth. Complex services require a larger amount of available network resources than simpler ones do. Still, a categorisation like this does not reveal all aspects of the targeted services. Some IP based applications adapt themselves to the at any time available network resources. They give consistent results for a vide variety of bandwidth range. Thus, there are no definitive guiding lines for choosing categorisation methods.
A central part of this thesis document deal with future and existing IP based services for mobile users. A categorisation with six different service classes is proposed. These are Interactive Media, Entertainment, Positioning, M-commerce, Transportation and Engine. Using the proposed framework has both advantages and drawbacks. It is user-friendly and easy to understand thanks to strict divisions between the classes. However, such strict borders are not always convenient. It may not be obvious in which class to place certain services – they may fit into more than one class dependent on point of reference. A service like automated road-toll payment may belong in M-commerce, Transportation or possibly the Engine class. Alas, there will always be arguments against or in favour of different categorization attempts.
3GPP “release-99” specifies a set of traffic classes. These classes are Conversational, Streaming, Interactive and Background. Different services uses different classes based on how delay sensitive each service is. Real-time data flows are supposed to use Conversational and Streaming, while traditional Internet services like WWW and e-mail should use Interactive and Background class. It appears that this classification strategy may not be too easy to combine with the proposed categorisation framework for mobile services. The proposed framework focus on possible services and is not concerned with technical requirement or restrictions related to the service in question. It is left to software designers how each service should perform under various conditions, and to define lower bandwidth limit for satisfactorily performance.

TCP
TCP and its performance over wireless links are closely investigated. Typical challenges TCP meet when crossing wireless links are discussed. So are the tools TCP uses to deal with them. Two articles that examine this subject are studied. One is titled “Internet Protocol

Performance over Networks with Wireless Links" (G. Xylomenos, G. C. Polyzos); the other is titled "Improving Performance of TCP over Wireless Networks" (B. S. Bakshi et al). Xylomenos and Polyzos argue that closer communication between protocol layers is vital for improved performance across wireless links. They advocate the use of local recovery to combat wireless link peculiarities. This way loss is hidden from higher layers at the end-points. Their proposals have at least one shortcoming. Extensive use of local recovery may influence on TCP end-to-end semantics. Applications will not know when these semantics are corrupted, and must use additional protocols above TCP to get end-to-end support.
Bakshi et al propose two approaches for improved TCP performance over wireless links: packet size variations and explicit feedback. Through simulations they show that an optimal packet size lead to 30 % performance improvements over a non-optimal packet size. Explicit feedback from the base station to the TCP sender seems to combat sender timeouts and hence redundant transmissions. One should be aware that these are results obtained from a software simulator. Live testing in a mobile network might yield other results. The simplified model used for testing may also not be sophisticated enough to ensure consistent results outside the simulator.

Queuing algorithms
Tests performed in this thesis project did not reveal any clear difference between the chosen queuing algorithms. Several reasons might explain why:
The settings for the CBQ filter class were too conservative. The sudden reduction in bandwidth might mask any differences for the two algorithms. Reduced throughput has such a severe impact on Media Player that eventual differences from one algorithm to the other are eliminated.
SFQ and PFIFO operate in quite similar ways. They do not differentiate enough to separate eventual effects on the chosen multimedia service. If queuing algorithms with more advanced handling procedures were chosen, one might experience a different result. But advanced algorithms include more settings to decide on, thus bringing increased number of factors into the overall equation.
To set up the test-bed in a public environment is probably not an ideal situation. Using a shared network to transport data from the server to the host probably influence the results. Strain on server, routers and network links usually varies with time. It is unlikely that every streaming session experience similar conditions. This probably influences the perceived performance. On the other hand, a mobile system is also subject to varying conditions. Mobile hosts must deal with such challenges too.
Streaming video may not be the best service for testing queuing algorithms. Media Player buffers incoming data before displaying the video. How Media Player treats data and which algorithms it uses, is not known. Applications like NetMeeting do not use buffering in the same way, so they might be better suited for testing. However, streaming services are probably more suited for use in mobile systems due to their buffering capabilities.

Ethereal was not able to reveal any information that positively connected subjective performance to objective measurements. Packet traces did not indicate why there were only subtle differences between SFQ and PFIFO. Installing Ethereal on both the router and the PC running streaming video would perhaps yield clearer results.

10 Conclusions

This thesis document proposes a framework for categorisation of IP based mobile multimedia services. The proposed framework should be useful for both vendors and operators of mobile systems. It provides a user-friendly overview of possible developments in the area of mobile services.

TCP often run into problems when crossing wireless links. Link problems not related to congestion often trigger congestion avoidance mechanisms at the TCP source. This results in under-utilisation of the link. Research has proposed several strategies to combat the problem. Local recovery, varying packet size and explicit feedback mechanisms are among the most accurate suggestions.

TCP is used for most services in today's data networks. It is likely that this trend will continue in future mobile systems like UMTS. Organisations like 3GPP and IEEE should therefore agree on how to optimise TCP performance in these new mobile systems.

The test-bed chosen for this thesis project did not differentiate strong enough between the tested queuing algorithms. Based on the test results, it was difficult to report on one algorithm's real-time performance over the other. Streaming video may not be the best multimedia service to test. Applications without built-in buffering would probably differentiate more when reviewing different algorithms. At least, there would not be succeeding buffers for the data to pass before entering the application.

The Ethereal network protocol analyser was used to trace incoming traffic on the host. Browsing the captured traffic files, it is not possible to discover specific flow patterns. Thus, no strong connection between how users perceive video quality and the way data arrive at the host was discovered. It is suggested to install Ethereal at the router as well – this way one might see how algorithms affect each packet passing through.

Future research should investigate how other queuing algorithms manage different multimedia services. The possibility to dynamically assign best algorithm to each service, and thus increase quality for users, must be investigated. A high and consistent service quality is probably of major significance if the high expectations to UMTS are to be met.

11 References

[1] J. F. Henriksen, *"En vurdering av aktuelle teknikker for tjenestekvalitet i et IP-basert stamnett for UMTS"*, Hovedoppgave ved sivilingeniørutdanningen i informasjons- og kommunikasjonsteknologi på HiA, p. 8, Oslo, January 2001.

[2] Halsall F., *"Data Communications, Computer Networks and Open Systems",* fourth edition, p.15, Addison-Wesley, Reading, 1997.

[3] Walshe D., "*Multimedia – not just a pretty name"*
Broadcom Eireann Research Ltd: Communicate volume 5 – no.2, December 2000
http://www.broadcom.ie/communicate/vol5/index.phtml

[4] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,211600,00.html

[5] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,214276,00.html

[6] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,211672,00.html

[7] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,213984,00.html

[8] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212425,00.html

[9] Chapman N., *"Digital multimedia"*, p. 396, Wiley, Chichester, 2000.

[10] http://whatis.techtarget.com/WhatIs_Definition_Page/0,4152,212603,00.html

[11] http://www.cselt.it/MPEG/documents/koenen/MPEG-4.htm

[12] L. Kleinrock, *Queueing Systems: Vol II, Computer Applications,* Wiley, New York, 1976.

[13] http://www.isoc.org/internet-history/brief.html#Introduction

[14] L. Peterson & B. Davie, *"Computer Networks – a systems approach"* second edition, p. 328, Morgan Kaufman, San Francisco, 2000.

[15] Statens Forvaltningstjeneste, NOU 1999:26, *"Konvergens – sammensmeltningen av tele-, data- og mediesektorene"*, kapittel 3.2.1, Elanders Norge AS, Oslo, 1999.

[16] UMTS Forum Report nr 8, *"The Future Mobile Marked – Global trends and developments with a focus on Western Europe*, p. 7, 1999.

[17] R. Simon, *"Cool U.S. Reception to Wireless"*, NY Times, 29 January 2001.

[18] UMTS Forum Report nr 8, *"The Future Mobile Marked – Global trends and developments with a focus on Western Europe*, p. 7, 1999.

[19] http://www.ericsson.com/press/archive/2000Q2/20000605-0005.html

[20] J. Bamrud, *"3GSM i Cannes: Teknologi-entusiasmen stor"*, ukeavisen Telecom nr 08/01, 01 Mars 2001.

[21] N. Sundstrøm, *"Let your phone show the way"*, Ericsson magazine Contact, issue 11 June 2000, 22 June 2000.

[22] http://www.ericsson.com/infocenter/news/Chat_Machines.html

[23] 3rd Generation Partnership Project, *"QoS Concept and Architecture"*, 3G TS 23.107 version 3.0.0, pp. 14-16, October 1999.

[24] G. Xylomenos, C. Polyzos, *"Internet Protocol Performance over Networks with Wireless Links"*, IEEE Network, Vol 13, Issue 4, p. 56, July 1999.

[25] P. Karn, *"The Qualcomm CDMA Digital Cellular System"*, Proc. USENIX Mobile and Location-Independent Comp. Symp, August 1993, pp. 35-39.

[26] J. Postel, *"Transmission Control Protocol"*, RFC 793, p.1, September 1981.

[27] G. Xylomenos, C. Polyzos, *"TCP and UDP Performance over a Wireless LAN"*, IEEE INFOCOM`99, Vol 2, pp. 439-446, Mars 1999.

[28] G. T. Nguyen *et al.*, *"A Trace-Based Approach for Modelling Wireless Channel Behaviour"*, Winter Simulation Conference, pp. 597-604, December 1996.

[29] Bikram s. Bakshi et al, *"Improving Performance of TCP over Wireless Networks"*, IEEE Proceedings of the 17th International Conference on Distributed Computing Systems 1997, pp. 365-373, May 1997.

[30] Bikram s. Bakshi et al, *"Providing Seamless Communications over Mobile Wireless Networks"*, IEEE Proceedings of 21st LCN Conference, Minneapolis, Oct 1996.

[31] H. Balahrishnan, S. Seshan, E. Amir, R. H. Katz, *"Improving TCP/IP Performance over Wireless Networks"*, Proceedings 1st ACM Conference on Mobile Computing and Networking, Nov 1995.

[32] P. Bhagwat, P. Bhattacharya, A. Krishna, S.K. Tripathi, *"Enhancing Throughput over Wireless LANs Using Channel State Dependent Packet Scheduling"*, IEEE Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer Societies: Networking the Next Generation, pp. 1113 – 1140, March 1996.

[33] S. Floyd, V. Jacobson, *"Random Early Detection Gateways for Congestion Avoidance"*, IEEE/ACM Transactions on Networking, pp. 397 – 413, August 1993.

[34] O. Kirch, T. Dawson, *"Linux Network Administrators Guide, 2nd ed"*, O'Reilly & Associates Inc, Cambridge, March 2000.

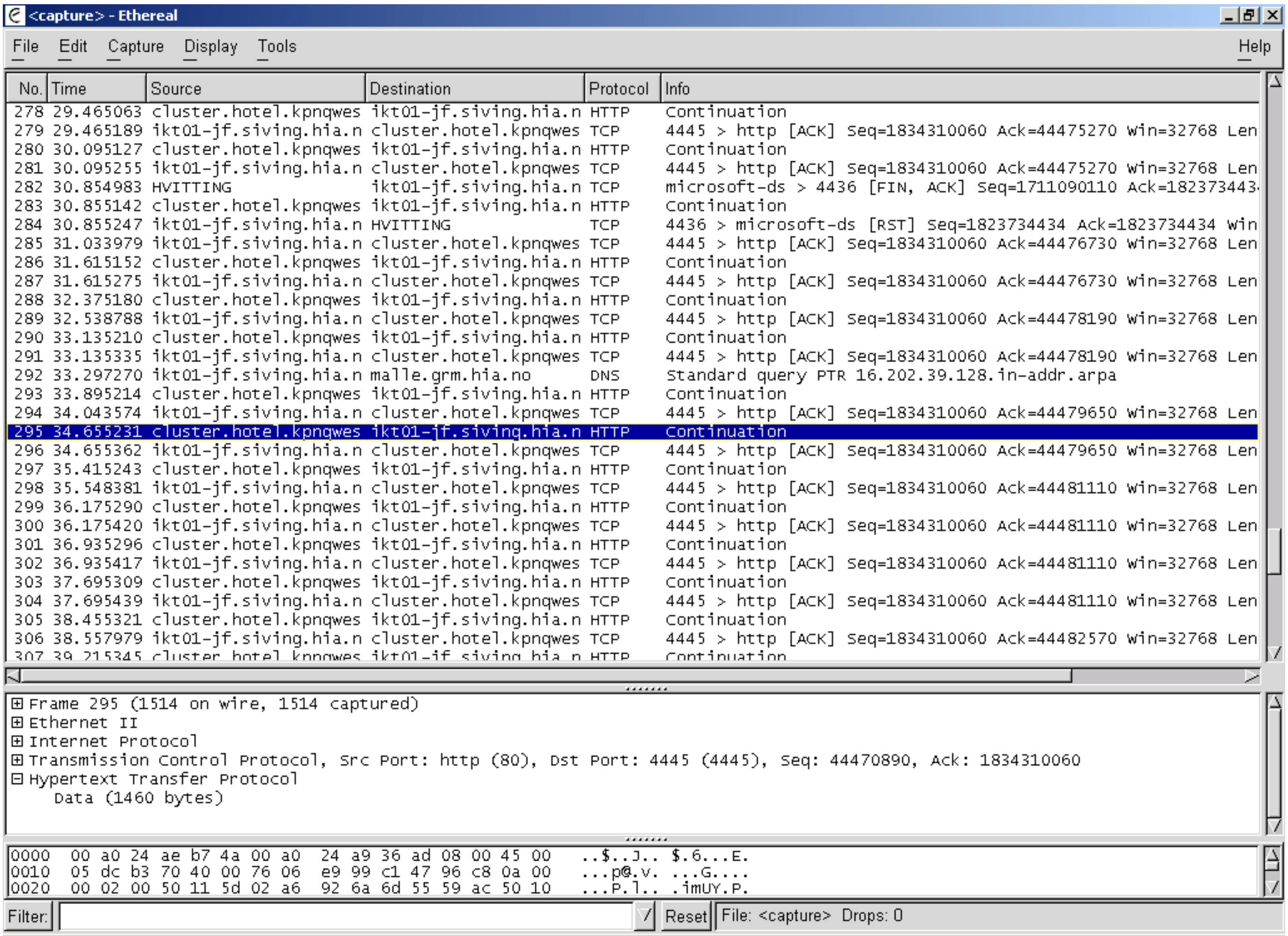
12 Abbreviations

3GPP	3rd Generation Partnership Project
ACK	Acknowledge
ARQ	Automatic Repeat Request
BER	Bit error Rate
BMP	Bitmap
CBQ	Class Based Queuing
CD	Compact Disc
CONCER	Collaborative Simulation for Education and Research
CSD	Circuit Switched Data
DARPA	Defence Advanced Research Projects Agency
DiffServ	Differentiated Services
DOS	Disc Operating System
DUPACK	Duplicate Acknowledgement
DVD	Digital Versatile Disc
ECN	Explicit Congestion Notification
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Control
FER	Frame Error Rate
FIFO	First In First Out
GIF	Graphics Interchange Format
GPRS	General Packet Radio Service
GSM	Globale System Mobile
HDLC	High-Level Data Link Control
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMT-2000	International Mobile Telecommunications
IntServ	Integrated Services
IP	Internet Protocol
IPS	Intellectual Property Rights
ISO	International Organisation for Standardisation
ITU	International Telecommunication Union
IWF	InterWorking Function
JPEG	Joint Photographic Experts Group
MMS	Multimedia Messaging Service
MP3	MPEG-1 Layer 3
MPEG	Moving Pictures Expert Group
MS	Mobile Station
MSL	Maximum Segment Life
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
Nam	Network Animator
NAT	Network Address Translation
NS	Network Simulator
NS	Network Simulator
OSI	Open Systems Interconnection
PSTN	Public Switched Telephone Network
RLP	Radio Link Protocol

RTF	Rich Text Format
RTT	Round Trip Time
SFQ	Stochastic Fairness Queuing
SMS	Short Message Service
TCP	Transport Control Protocol
TOS	Type of Service
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Transport Radio Access Network
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network

Appendix A

A.1 Packet trace with SFQ enabled



A.2 Packet trace with PFIFO enabled