ESPRIT PROJECT EP28425

Broadband Integrated Satellite Network Traffic Evaluations

Deliverable 1.1

Application and User Behaviour Characterisation

Status / Version :	**DELIVERABLE / FINAL**
Date :	**September 30, 1999**
Distribution :	**Public**
Code :	**BISANTE/DEL11**
Author (s) :	**M .Marot, G. Kotsis (Ed.)**

Abstract: Starting with a classification of relevant services and applications, a survey of existing techniques used for traffic source modelling is given. The need for new approaches which are capable of modelling the user behaviour is identified. Some seminal work in this direction is reported based on which a layered modelling approach is proposed. A section on solution techniques rounds off the deliverable.

To foster dissemination of results, this revised version of DEL11 does no longer include the BISANTE Consortium confidential data on the chosen case studies, which are reported in an additional, confidential deliverable (DEL11-CS).

The BISANTE Consortium consists of :

Thomson-CSF Communications	Partner	France
Netway	Partner	Austria
Solinet	Partner	Germany
University of Vienna	Associated Partner	Austria
University of Surrey	Associated Partner	United Kingdom
Institut National des Télécommunications (INT)	Associated Partner	France

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

The trend towards distributed computing and emerging interactive multi-user applications puts new challenges on the underlying network infrastructure. As opposed to conventional computing and data communications operating with alphanumeric data, multimedia computing due to the type and nature of multimedia data exhibits whole new classes of system requirements with respect to capturing, storing, streaming, transmission, synchronisation and presentation, commonly referred to as Quality of Service (QoS) requirements.

To understand the interdependencies among the delivered quality of service and the workload and network characteristics, **performance evaluation studies** are necessary. Performance measurements are one way to address this problem, but are limited to existing environments.

When a network (or components of a network) does not exist, it is necessary to construct a **model** of the network in order to evaluate its performance. If **analytical** studies are tractable, they are preferred, but they are often too complicated, because the network is too large or the known approximations are too simple, or for other reasons. That is when **simulation** comes into play, which permits to have some good estimates of the performance criteria, with given confidence intervals if the simulation is run for a sufficiently long time. But even in cases where the network already exists, it can be necessary to run simulations in order to study its performance under new load conditions or the effects of changing some characteristics of the network : adding memory in the switches by dimensioning the buffers, optimising the routing strategy, etc. The simulation of such “What-if” scenarios is necessary in network capacity planning, tuning, and optimisation.

The actual need when simulations are run is to get good **traffic models**, where the quality of a model is often described in terms of its representativeness, flexibility, simplicity of construction, compactness, or usage costs. It is well known that the present models fail to meet these requirements, because of their simplicity or because they are too complicated and not tractable even by simulations (some fractal models for instance). On the other hand, the nature of the traffic is strongly related to the behaviour of the user sharing the resources of the network, and a traffic modelling strategy cannot be designed without considering the modelling of the user behaviour.

As the **behaviour of the users** is strongly related to the kind of **services** they use, services should first be classified. Following an ITU-T proposal, a classification is given which is built by considering the level of interactivity of the user, the time requirements of the service - real-time or not -, and the number of users involved in the service (point-to-point or multi-point connections). By interacting with these services, users trigger the generation of data flows on the network whose accurate modelling is the ultimate goal of the user behaviour and traffic models as said above. A great interest is then taken in the description of the **application** spaces for three environments : the Internet, mobile networks and lastly, CSCW (Computer Supported Collaborative Works). These applications give a hint about those which can be often used. When choosing the applications to model, typical applications will be taken from these application spaces. As the qualitative and quantitative characteristics of the data flows are depending on the coding scheme used by the applications, it is necessary to consider coding schemes for different types of data. In

some cases, the use of a specific coding standard can generate more or less data. For the example of a video flow, the use of a coding implementing some compression techniques will generate more or less data than the use of a coding implementing an other algorithm which has not the same compression ratio, or the data flow can have a given distribution with a given coding standard and an other with an other standard. The reader can find in appendix a presentation of the existing main coding standards. It includes an overview of text and document, graphic, audio/video codecs and security codecs.

An **overview on the traffic source modelling** is given, in which a summary of known mathematical techniques is presented. It includes conventional Markov models, fluid models, linear stochastic models (such as ARMA, ARIMA, and so on), and more complicated models such as self-similar ones, and lastly pointers to papers written on traffic generator studies and implementations.

All those models do not explicitely take into account the actual traffic source, namely the user interacting with the application. Motivating a **user-behaviour oriented** modelling framework, Section 4 starts with a review of user modelling techniques. Ways of characterising and classifying users are given, as well as data analysis techniques available to extract interesting knowledge about users.

An approach is proposed, which aims at **combining** the **traffic source modelling** techniques discussed in Section 3 **with knowledge on application characteristics and user behaviour**. The approach is based on a layered framework, representing at the top level the users, their sessions, the applications used within a session, the commands available within an application, the services activated by the commands, and finally the physical resources to which the load generated from the service is imposed to.

In Sections 3.3 and 3.4 we will give an overview of the methodology and framework, a detailed discussion will follow in D1.2.

The **hierarchical framework** can be transformed into more simplified models by aggregation of lower layers. At the highest level of aggregation, the model is reduced to a simple on-off source ommitting packets according to a distribution derived from the aggregation of sub-layers. A detailed discussion of aggregation techniques will be given in D1.3 (Generation of User Profiles).

When applications and services are defined and classified, when overviews of source modelling techniques, user modelling techniques and integration methodologies are given, it remains to present the **solution techniques** used to resolve performance problems or to evaluate the network performance. This is the aim of the last part, Section 5. Concerning the solution techniques, performance criteria of interest are identified. The models used in network performance studies are also presented, as well as the methods : when are analytical methods, aggregation techniques, measurements, etc. important ? Models are often of the same type as those used in other domains such as user behaviour modelling, but their applications are different.

2. INTRODUCTION

There is no denying among network professionals that network simulation is an extremely potent tool in network design/implementation, potentially allowing engineers to discover problems before actually setting up the network. For this approach to be fruitful, the network mechanisms must be correctly modelled, but also representative traffic sources must be designed. This later requirement is clearly the most subject to caution, as most network planners use empirical methods to set up sources that ‘’should’’ prove coherent with the flux generated by the future users. One of the most important output of the BISANTE Project will then be the precise characterisation and modelling of traffic sources.

Traffic sources regroup two different aspects : the users, which are the ultimate source of traffic on networks, and the services that are called by these same users, that generate traffic on their own. User behaviour characterisation, in terms of traffic flow, is beyond our scope and will be reviewed in another Deliverable (1.2) in this document, we are only concerned with the characterisation of services.

This characterisation should take into account the vast array of different services at user’s disposal today. For this reason, we shall begin to give a global survey of the matter, before going into details.

As part of the BISANTE project, we have to model the characteristics of the applications used on nowadays networks. To save time, it would be of great interest to identify first the main classes of applications that exist, for the simple reason that similar applications must have similar behaviours. Also, we are not to restrict ourselves to existing tools, but we may also have a look at applications that are too new to be used right now, but look interesting enough so that they may become standard in the future.

2.1. GENERAL SURVEY OF SERVICES

What follows is a widely-used classification of services provided on networks. The most basic distinction is between interactive services, involving only two end-stations, and distributed services, that involve far more end users; but even those two classes can be further divided as well.

2.1.1. INTERACTIVE SERVICES

As said above, interactive services offer one user the possibility to interact with another. The kind of interaction provided is another matter that results in very specific situations (and thus in very different network traffic flows) that will be detailed below.

- **Conversational**

Applications belonging to this class offer a bi-directional service of communication between two entities in real time, without any need to save data for later transmission.

Most of the time, this results in exchange of voice data, which means that this traffic has to be synchronised (to avoid packets arriving in random order) and fast (so as to avoid lengthy delays between packets, which result in very poor sound quality for the end users. Generally, it is considered that 20 ms is the delay above which the human ear begins to notice the delay, and thus the poor quality of service offered).

Although the communication is bi-directional, the flow of data can be unidirectional (ex : a camera recording images).

Examples : telephony over IP, visiophone.

- **Messaging**

This kind of applications offers a bi-directional service of communication between two entities, but not in real-time, so this means that the data to be transmitted has to be saved locally and effectively be sent later. Also, these applications offer typically a number of services like mailboxes or messages editing.

As for now, messaging services treat only text data, but with the recent gain in popularity of multimedia applications, some services have begun to offer multimedia transfer flows as well. If text transfer does not require much in terms of network characteristics (except a reasonable degree of reliability of course), multimedia transfer does require a lot of synchronism, not only between voice packets, but also between voice and video packets. Also, this kind of traffic requires a lot of bandwidth as multimedia files are typically huge.

Examples : E-mail (possibly multimedia).

- **Retrieval**

This kind of applications offers bi-directional, near real time, and asymmetric sessions. That means that one of the entity (called client) takes the initiative of launching all the sessions or terminating them. The other end entity (called a provider) only reacts to the user’s commands.

There is no ‘’standard’’ kind of traffic associated with those services. The range of data offered by providers covers the whole spectrum : text, binary data, still images, video, audio, multimedia (audio + video, and possibly text mixed in as well). As already said, the network requirements are as a consequence vastly different ranging from reliability to synchronisation mechanisms and speed of transmission.

Examples : database consultation, WWW.

2.1.2. DISTRIBUTION SERVICES

Applications belonging to this class still act on bi-directional sessions, but there are more than one session (and two entities) concerned. The entities are separated into two groups : the providers and the users. Now there can be two different ways of communicating between these groups :

- **Without Individual Control**

In this particular case, all the providers can do is to send data to a user that requests it. All a user can do is to request data. However, the user has no liberty other than initiating the session. Once it’s done, he receives the data and cannot act upon it (e.g. he cannot filter the data, or select one particular piece of it).

Generally, these services offer audio or multimedia files to users. As already stated, that means that problems of synchronisation and speed can arouse. It is further enhanced by the fact that in this particular case the user has absolutely no control on the traffic flow, so if the quality of service goes below a certain level of tolerance, all he can do is to improve the situation is to stop the transmission, and begin it all over again later.

Example : video on demand, radio broadcast.

- **With Individual Control**

In this case, a user is not restricted to an open/close-session operating mode, but he can interact with the data he is receiving, by selecting a particular piece of it or modifying it.

The traffic flows requirements are the same as in the previous situation, but in this case the user has far more control on the situation and can react more accurately on perceived QoS. This means that even under bad transmitting conditions, the user can choose to reduce its own demands and to select which kind of transmission he still wants to operate. This way, there is still data exchanged, but its quantity and quality have been adapted to network conditions by the user.

Example : distributed databases.

2.1.3. COLLECTIVE SERVICES

Although this situation involves more than two entities as seen in the preceding example, the major difference is that the session is collective, and not made up from a set of dual sessions. This kind of services encompasses all the resources-sharing tools (files, file systems, ...) as well as software components needed to build virtual workspaces.

Resources-sharing services usually involve only text or binary transfer, but virtual spaces are typical services where multimedia data is exchanged (voice + video + text), and at a greater rate than usual. This is due not only to the fact that multimedia files are typically large, but also that they are generated by numerous users and in situation that is theoretically ‘’in real time’’, so exchanges have to be frequent and quick. For these reasons, this particular kind of services is in practice the one that offers the worst QoS.

Example : videoconference, NFS (or similar file systems-sharing tools).

2.2. APPLICATIONS

In the preceding section, we have introduced the various kinds of services found nowadays on communication networks, and the resulting traffic flow characteristics. Let us now give an overview of the applications.

In the context of this deliverable, an application is viewed as a means to provide services to the user.

From a modelling point of view it is necessary to describe the qualitative and quantitative characteristics of applications to be able to construct a model at the appropriate level of abstraction. In addition, for each application the possible types of user interaction within the application and their impact on the resulting traffic will be discussed.

2.2.1. APPLICATIONS ON THE INTERNET

The Internet is the latest product of the growing trend toward a truly global economy. This inter-networking of computers is part of the same phenomenon that is creating borderless markets across a wide variety of industries.

To compete effectively in this global economy requires organisations to adopt a degree of openness and accessibility to which they may not be accustomed. Businesses already using the Internet are succeeding because they embrace an open culture that best leverages the competitive advantages offered by the Internet.

Of course there are drawbacks to such openness. Competitors may learn more about products and services of each other, for instance. Computer hackers may break into corporate networks over Internet connections. But the value of improved communications with the outside world far overshadows any perceived risks in being connected to the Internet.

On the one hand the Internet is full of information and more and more services are running on it, on the other hand the rapidly growth of the Internet is attended with lost of overview.

Today the Internet is no longer a text-based field: Animated Images, TV, Videoconferences, Web-cams and much more are provided for entertainment, to cut costs or to get information faster.

Some services like „Gopher“, „WAIS“ and „Archie“ were pioneers and are decreasing in use while new technologies like „Push“ or „Intelligence Agents“ are getting more popular or failed the goal.

To get the necessary information you can choose a more passive information retrieval like subscription to a mailing-list, push channel, news- or stock ticker, or a more active way of searching with the available search engines. This could be difficult since that will become complex if you use the advanced search criteria and also to choose one probe for your information seeking.

E-commerce and Internet advertising takes advantages of the new security standards and make the Net a marketplace.

How services like Internet phone or Internet Fax servers help someone to spend money lays in everyone’s decision.

Communities like IRC Channels, Web-rings, Virtual Places and more seem to have the power to style a new culture of information exchanges, private meeting places or simply the ability to presents someone’s live, knowledge or on some homepages even the picture of a cat.

2.2.1.1.THE WORLD WIDE WEB

The World Wide Web is a distributed, hypermedia-based Internet information browser. It presents users with a friendly point and click interface to a wide variety of types of information (text, graphics, sounds, movies, etc.) and Internet services. It is possible to use the Web to access FTP archives, databases, and even almost every server.

The exact definition for the World Wide Web (popularly known as the Web) varies, depending on whom you ask. Three common descriptions are:

- A collection of resources (FTP, http, telnet, Usenet, Search Engines and others) which can be accessed via a web browser.

- A collection of hypertext files available on web servers.
- A set of specifications (protocols) that allows the transmission of web pages over the Internet.

You can think of the Web as a worldwide collection of text and multimedia files and other network services interconnected via a system of hypertext documents. HTTP (*HyperText Transfer Protocol*) was created in 1990, at CERN, the European Particle Physics Laboratory in Geneva, Switzerland, as a means for sharing scientific data internationally, instantly, and inexpensively. With hypertext a word or phrase can contain a link to other text. To achieve this a programming language called HTML was developed, that allows you to easily link you to other pages or network services on the Web.

This non-linear, non-hierarchical method of accessing information was a breakthrough in information sharing and quickly became the major source of traffic on the Internet.

The basic elements of the World Wide Web are:

- HTTP (Hypertext Transfer Protocol) - the set of standards used by computers to communicate and share files with each other.
- URL's (Uniform Resource Locator) - the "address" of a resource (file or directory) on the Web.
- HTML (Hypertext Markup Language) - the programming "tags" added to text documents that turn them into hypertext documents.

Homepages holds on Websites carry all fundamental Links to necessary Resources. Also referred to as a web page. The starting point of a Web presentation and a sort of table of contents for what is at the website, offering direct links to the different parts of the site.

2.2.1.2.FINDING AND SHARING INFORMATION AND FILES

- **FTP**

An acronym for *File Transfer Protocol* -- a very common method of transferring one or more files from one computer to another. FTP is a specific way to connect to another Internet site to retrieve and send files. FTP was developed in the early days of the Internet to copy files from computer to computer. With the advent of the World Wide Web, and web browser software, you no longer need to know FTP commands to copy to and from other computers

Web Upload Tools are specialized FTP Clients that facilitate the management of Web sites by uploading and/or mirroring Web documents using the File Transfer Protocol. In other words, if you manage a Web site by uploading files to a remote location, you can use one of these applications to expedite the transfer process. Several of these tools will also allow you to mirror your site at multiple locations using automatic synchronisation capabilities.

- **Archie**

The Archie system was created to automatically track anonymous FTP archive sites, and this is still its primary function. The system makes available the names and locations of some Million files at some thousand archive sites. Archie's User Access component allows you to search the "files" database for these filenames. When matches are found, you are presented with the appropriate archive site name, IP address, the location within the archive, and other useful information. You can also use Archie to "browse" through a site's complete listing in search of information of interest, or obtain a complete list of the archive sites known to that server. The Archie server also offers a "package descriptions" (or "whatis") database. This is a collection of names and descriptions gathered from a variety of sources and can be used to identify files located throughout the Internet, as well as other useful information. Files identified in the whatis database can then be found by searching the files database as described above. Nevertheless, Archie becomes more and more history.

- **Search Engines – Intelligent Agents**

The field of Internet resource discovery tools is one of the most dynamic on the Internet today. There are several tools in addition to those discussed here that are useful for discovering or searching Internet resources. Software packets called „Internet Agents" are more comfortable and can be advised to do their task in the background.

Internet Agents automate many routine and often time-consuming tasks, giving you more time to have fun surfing the internet. A sampling of agent tasks includes filtering and downloading Websites, automatically downloading product updates and bug fixes, and alerting you when Web sites have been updated. While internet agents may not be the perfect panacea for keeping up with the constant evolution of your favourite products and Websites, they do go a long way towards helping you stay as up to date as far as possible with the latest information the internet has to offer.

A search engine is an Internet tool that allows you to search through vast amounts of information quickly. It gives users the ability to search and retrieve information from remote databases. Most databases contain text based information like newspapers, documents, journals, etc. but the part of sources hold movies, pictures and music are steady growing.

An extremely wide variety of diverse types of information in an easy to use menu-driven interface. Servers link information from all around the Internet in a manner that can be transparent to the user. (Users can easily discover the source of any piece of information, however, if they wish.) For example, the links databases of every type, applications, white pages directories, sounds, and pictures. Since most servers are linked to other gophers, if you can get to one, you can get to many.

It takes as input keywords which it will use as its search criteria. A keyword can be a single word like dog or a more defined phrase such as Scottish terrier. It will then search the entire list of resources that you specify and find all those text documents that match.

Meta-Search engines have the ability to use the different power of different machines to find the information you asked for.

Searches are not limited to either one or very few keywords. One can use Boolean logic in the searches, but it is important to be as precise as possible in order to limit the scope of the answers.

- **Telnet**

The use of this term as a verb, as in "telnet to a host" (the remote login over a network) means to establish a connection across the Internet from one host to another. All that is needed is an account on the remote host to be able to login to it once you've made a connection. However, some hosts, such as those offering white pages directories, provide public services that do not require a personal account.

2.2.1.3. Communication on the Internet

- **E-mail**

Short for electronic mail, e-mail consists of messages, often just text, sent from one user to another via a network. E-mail can also be sent automatically to a number of addresses.

Electronic mail has grown over the past twenty years from a technical tool used by research scientists, to a business tool almost as common as the fax machine. Today, electronic mail is primarily utilised only within a company's local area networks. Productivity gains from the use of intra-company email are significant and it has become indispensable to many organisations. However, internal communications represent only a small fraction of overall business activity. Thus, an opportunity exists to derive similar efficiencies from our external communications.

- **Mailing Lists**

Mailing lists are a way of having a group discussion by electronic mail. Also used to distribute announcements to a large number of people. A mailing list is very much like a conference on a bulletin board system, except the conversation comes to the e-mail box. Each time a subscriber of the list posts a reply to the conversation, it is distributed to the e-mail box of every subscriber of the list. All of this traffic is automated and managed by programs called mailing list managers (MLM's) or mail servers.

Mailing lists are the most basic form of Internet conferencing. They can be public or private and, unlike Usenet newsgroups, which require additional software to run, all you need to participate is an e-mail address.

A mailing list is said to be "unmoderated" if all of the messages sent to the list are automatically forwarded to each member of the list. In a "moderated" list, all messages are sent first to a list moderator, who makes decisions about which postings will or will not be sent to everyone on the list.

If many people are on a mailing list, the traffic in the e-mail box can be overwhelming. One way to deal with this is to subscribe to the "digest" version of the list (not all mailing lists have digest versions). In a digest version, postings are collected into a single file and distributed to the list on a regular basis (usually daily). In this way you receive only one big file at regular intervals rather than hundreds of small ones everyday.

- **Newsgroups**

Electronic discussion groups consisting of collections of related postings (also called articles) on a particular topic that are posted to a news server which then distributes them to other participating servers. There are thousands of newsgroups covering a wide range of subjects. It is necessary to subscribe to a newsgroup in order to participate or to track the discussion on an on-going basis. Unlike with a magazine or newspaper, subscribing to a newsgroup is free of charge.

Newsgroups are found primarily on *Usenet*. Usenet is the collection of computers that participate in a global conferencing system that make newsgroups perhaps the largest distributed bulletin board system in the world. Newsgroups are one of the oldest and most widely used services on the Internet.

Some newsgroups are "moderated," which means that a person decides which postings will become part of the conversation. Most are unmoderated, which means that any posting sent to the list is automatically added to the group.

- **IRC (Internet Relay Chat)**

Next to e-mail and surfing the Web, chatting is the most popular pastime on the internet. The applications in this category are designed to help you chat in style. The capabilities of these clients range all the way from one-to-one text-based communication clients to the electronic version of those little yellow sticky notes to full-duplex, two-way audio and video-based conferencing tools. Additional types of specialized chat clients includes Internet Phones, IRC Apps, and Virtual 3-D Chat Tools.

IRC is an acronym for *Internet Relay Chat*, a program that allows you to carry on "live" conversations with people all over the world by typing messages back and forth across the Internet. You can talk in groups or in private with only one person. IRC consists of "channels," which usually are devoted to specific topics. Anyone can create a "channel" and any message typed in a given channel is seen by all others participating the channel.

There is no limit to the number of people who can chat on a particular channel. Because of this unique feature, IRC channels have served as unofficial "news" sites during times of crisis, such as the Gulf War and the 1994 southern California earthquake. Mostly, though, you will find the same thing as on in newsgroups - people talking about things they are interested in.

- **Push Technology**

Supports automated distribution of reports ("push") via the Internet and internal intranets. It seems to leak in customer acceptation.

Online News Clients deliver a variety of news-related services, including up-to-date information on stocks, sports, business, entertainment, technology, weather, and more. These applications typically operate like stock tickers in that they display current information in a banner that scrolls across the screen. Screen savers, Web pages, custom interfaces, and audio messages are often used as well. Information is updated automatically at set intervals (usually via push technology) so that one is never out of touch with what is going on in the world.

2.2.1.4. Multimedia – Broadcasting – Real Audio

Using more than one type of media simultaneously, like text with sound, moving or still images, music, etc. for TV, entertainment, conferencing, presentations and so on.

Streaming tools utilise advanced multimedia technology to deliver audio and/or video content to you while it is being downloaded, as opposed to after it has completely been downloaded.

Real Audio is an innovative method of livening up web pages through the inclusion of audio. Traditionally the use of sound on web pages has been constrained by the need to use bulky file formats. The disadvantage of these file formats is that they are often very big and can take ages to download before they are even ready to be played.

Real Audio converts these files to a special format which "streams" sound allowing the listener to enjoy the presentation after only a few seconds. Effectively what happens is that as soon as you request a Real Audio / Video file - usually by following a link - the first 2 or 3 seconds of the file is downloaded to your computer. This immediately starts to play and as it does, the Real Audio stream continues to download the rest of the file. By the time the first few seconds of content have been played, the next few seconds are ready, hence creating a continuous "stream". Another advantage is that Real Audio / Video is highly compressed and therefore takes up less space on your web-server - a positive advantage when you have limited web space.

2.2.1.5.INTERNET COMMERCE

The world of electronic commerce is about to explode. Soon there will be approved security mechanisms to enable online user authentication, the use of digital "signatures" and the secure encryption of credit card numbers (SET) and purchase order information. Offering your goods and services for sale via the World Wide Web is a natural extension of contemporary programs that will result from these imminent technology developments.

By effectively using the Web to display products and services through multimedia presentation techniques, it is only natural to enable the user to "click here" to order the product. Orders can be automatically transferred to your central business systems for order processing, inventory management, billing and shipping. In the process, the electronic mail ID of the customer is also gained and other information for future pinpoint marketing activities.

A company can achieve tremendous savings by automating elements of its order processing. Companies in a wide variety of industries have already done so. As the reliance on paper transaction documentation continues to migrate to electronic systems, even more cost savings will result. Greater numbers of these transactions will inevitably move to the Internet and be deployed on a business-to-business basis.

This trend will continue to accelerate as it’s possible to apply new technology to electronic commerce. Traditional order-taking staffs already are able to handle more inquiries in less time relying to this technology. In time, more businesses will redirect these employees into more income- generating positions, as this new way of buying and selling continues to expand.

2.2.2. APPLICATIONS CHARACTERISTICS FOR MOBILE NETWORKS

2.2.2.1.GSM (GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS)

Many services (35) are included with GSM. The GSM supplementary services (call forwarding, call barring, multi-party calling, ...) are ISDN-like services that are not inherent to speech or data. They will not be listed in this document.

- Speech

This is the most important service offered by GSM. The main speech service provided by GSM is telephony. Emergency calling is a distinct service allowing a mobile user by dialling 112 to reach the nearby emergency service. Group 3 fax is also supported by use of an appropriate fax adapter. Voice mailbox is another speech service: the caller can directly access the mailbox of a mobile subscriber and deposit a voice message. It will be stored and can be heard when the mobile user wants to retrieve it.

- Data

Data services between GSM and the Public Switched Telephone Network (PSTN) users are based on the PSTN data services. We are only considering here the Plain Old Telephone System (POTS). In such systems, audio modems are used for data transmissions. Unfortunately, the speech bearer of the GSM radio interface is encoded in a way optimised for speech. This is not suitable to the signal used with audio modems. Hence, to still be able to use this service, the communication between the mobile terminal and the network is fully digital. Then, an audio modem placed at the interworking point takes care of the communication with the PSTN user. Up to 9600 bit/s full duplex communication can take place. GSM has the advantage that a single subscriber can be called through multiple numbers, one per service.

Data can use either the transparent service, which has a fixed delay without guarantee of the data integrity, or a non-transparent service, which guarantees data integrity through an Automatic Repeat reQuest (ARQ) mechanism, but with a variable delay. The data rates supported by GSM are 300, 600, 1200, 2400, 4800 and 9600 bps.

Data services between GSM and ISDN users should work very well since GSM has been designed to ease the integration of GSM in an ISDN system. However the limit of 9600 bit/s of GSM data transmission is low compared to the 64 kbit/s transmission capability of ISDN.

Data services between GSM users are one of those exposed in the two previous paragraphs. There is no GSM independent network. The network shared between two GSM users is either PSTN or ISDN (only ISDN in the future).

A unique feature of GSM, the Short Message Service (SMS) is a bi-directional service for sending short alphanumeric (up to 160 bytes) messages. SMS can also be used in a cell-broadcast mode, for sending messages such as traffic updates. Messages can be stored in the SIM card for later retrieval.

2.2.2.2.DECT (DIGITAL EUROPEAN CORDLESS TELECOMMUNICATIONS)

The DECT timescale coincides with the introduction of ISDN services. The implementation of all ISDN basic rate services will be possible within DECT. The services are different from one environment to another. DECT classify four working environments:

- Residential use
- Public use
- Small (business) system use
- Large (business) system use

For each one, basic capabilities and enhancement features are defined.

1. Speech

As for GSM, the speech services are the first aim of DECT. The speech quality (ADPCM encoded) is similar to this of the fixed network. An emergency service is also provided with DECT.

2. Data

The applications of DECT data terminals fall into two categories:

- Primarily static, using DECT as a cordless drop-line bearer to a high-speed back-bone
- Primarily portable, for entirely new applications made for DECT:
  - multimedia (e.g. voice and fax) mail terminal;
  - note-pad;
  - access to data bases;
  - ultra-light, small PC emulator;
  - cordless ISDN videophone

DECT supports the group of general teleservices:

- remote terminal service;
- batch file transfer;
- real-time file access;
- generic ISDN connection based services.

The bearer services have to use the spectrum efficiently. Hence, DECT offers bearers which are well matched to the needs of the above services. These needs, for the first 3 services are listed in Table 1 : Teleservices requirements. The last service (generic ISDN) is a (2B+D) and requires a full rate of 144 kbit/s for full duplex.

Application	Link Establishment Time	Transaction Duration at Full Rate	Full Rate
Remote Terminal			
• Text	50 ms	100 ms - 5 s	10 - 20 kb/s
• Graphics	50 ms	500 ms - 10 s	24 - 128 kb/s
Batch File Transfer			
• Light	1 - 5 s	1 - 30 s	32 kb/s
• Heavy	1 - 30 s	5 - 1000 s	64 kb/s
Real-time file access			
• Slices	50 ms	200 ms - 2 s	64 - 256 kb/s
• Chunks	500 ms	1 - 10 s	64 - 256 kb/s

Table 1 : Teleservices requirements

Asymmetric bearers using a fixed ratio (i.e. 10:1) are supported since they better characterise the flow of information of a data service than the symmetric one.

2.2.2.3.GPRS (General Packet Radio Service)

The GPRS is a set of new GSM bearer services that provides packet mode transmission within the PLMN and interworks with external networks. It is not a requirement that the GPRS be an extension of or an emulation of one or any of these non-GSM services or networks.

The GPRS shall not prevent the user's operation of other GSM services.

The GPRS allows the service subscriber to send and receive data in an end-to-end packet transfer mode, without utilising network resources in circuit switched mode.

GPRS enables the cost effective and efficient use of network resources for packet mode data applications e.g. for applications that exhibit one or more of the following characteristics:

- intermittent, non-periodic (i.e., bursty) data transmissions, where the time between successive transmissions greatly exceeds the average transfer delay;
- frequent transmissions of small volumes of data, for example transactions consisting of less than 500 octets of data occurring at a rate of up to several transactions per minute;
- infrequent transmission of larger volumes of data, for example transactions consisting of several kilobytes of data occurring at a rate of up to several transactions per hour.

Within the GPRS, two different bearer service types are defined. These are:

- Point-To-Point (PTP);
- Point-To-Multipoint (PTM).

Based on standardised network protocols supported by the GPRS bearer services, a GPRS network administration may offer (or support) a set of additional services. This is outside the scope of this deliverable, however, a number of possible PTP interactive teleservices include:

- retrieval services which provide the capability of accessing information stored in data base centres. The information is sent to the user on demand only. An example of one such service in the Internet's World Wide Web (WWW);
- messaging services which offer user-to-user communication between individual users via storage units with store-and-forward mailbox, and/or message handling (e.g., information editing, processing and conversion) functions;
- conversational services which provide bi-directional communication by means of real-time (no store-and-forward) end-to-end information transfer from user to user. An example of such a service is the Internet's Telnet application;
- tele-action services which are characterised by low data-volume (short) transactions, for example credit card validations, lottery transactions, utility meter readings and electronic monitoring and surveillance systems.

Some examples of teleservices which may be supported by a PTM bearer service include:

- distribution services which are characterised by the unidirectional flow of information from a given point in the network to other (multiple) locations.

Examples may include news, weather and traffic reports, as well as product or service advertisements;

- dispatching services which are characterised by the bi-directional flow of information from a given point in the network (dispatcher) and other (multiple) users. Examples include taxi and public utility fleet services;
- conferencing services which provide multi-directional communication by means of real-time (no store-and-forward) information transfer between multiple users.

Capabilities that may be offered together with the PTM bearer services include:

- geographical routing capability, which provides the ability to restrict message distribution to a specified geographical area;
- scheduled delivery capability, allowing store-and-forward type services to specify a future delivery time and a repetition rate.

It is possible to include these capabilities as part of the service request (i.e., as part of the packet). Some operators may offer PTM services only together with these capabilities.

☞ GPRS service description

There are two categories of GPRS services:

- Point to Point (PTP) services,
- Point to Multipoint (PTM) services.

The PTP service provides a transmission of one or more packets between two users, initiated by a service requester and received by a receiver.

There are two PTP services:

- PTP Connectionless Network Service (PTP-CLNS);
- PTP Connection Orientated Network Service (PTP-CONS).

The PTM service provides a transmission of packets between a service requester and a receiver group.

There are three PTM services:

- PTM Multicast (PTM-M);
- PTM Group Call (PTM-G);
- IP Multicast (IP-M).

For PTM-M and PTM-G the data transmission is restricted to the members of a receiver group currently located within a geographical area. Both the receiver group and the geographical area are specified by the service requester.

The geographical area addressing mechanism is not applicable to IP-M.

☞ Service characteristics

Subscriber profile : The subscriber profile holds subscription information about services and other parameters that have been assigned for an agreed contractual period. A subscription is required to allow a subscriber to initiate a PTM-M data transfer operation. No subscription is required to receive PTM-M messages. It shall be possible to validate a service request against a service subscriber's subscription profile.

Quality of Service (QoS) : GPRS shall include the functionality to increase or decrease the amount of radio resources allocated to GPRS on a dynamic basis. The criteria used to decide on dynamic changes of the GPRS part of the radio resource should not be specified. Within GPRS the dynamic allocation of the radio resource for bursty or lengthy file transfer applications shall be such that it can be controlled by the network operator. The defined QoS parameter values, assume the user is at a location with acceptable GSM-/GPRS-coverage and refer to and are valid for normal network operating conditions or, as in the case of the service precedence parameter, regulate how the network shall handle abnormal conditions.

Service precedence (priority) : The service precedence indicates the relative priority of maintaining the service. For example under abnormal conditions (e.g. network congestion) packets which may be discarded can be identified. The following precedence levels are defined:

- High precedence: Service commitments will be maintained ahead of all other precedence levels.
- Normal precedence: Service commitments will be maintained ahead of low priority users.
- Low precedence: Service commitments will be maintained after the high and normal priority commitments have been fulfilled.

Reliability : The reliability parameter indicates the transmission characteristics that are required by an application. The reliability class defines the probability of loss of, duplication of, mis-sequencing of or corruption of SDUs.

Table 2 : Reliability classes lists the three classes of the data reliability.

Reliability class	Lost SDU probability (a)	Duplicate SDU probability	Out of Sequence SDU probability	Corrupt SDU probability (b)	Example of application characteristics.
1	10^{-9}	10^{-9}	10^{-9}	10^{-9}	Error sensitive, no error correction capability, limited error tolerance capability.
2	10^{-4}	10^{-5}	10^{-5}	10^{-6}	Error sensitive, limited error correction capability, good error tolerance capability.
3	10^{-2}	10^{-5}	10^{-5}	10^{-2}	Not error sensitive, error correction capability and/or very good error tolerance capability.

Table 2 : Reliability classes

(a) To protect against buffer overflow or a protocol malfunction, there is a maximum holding time for each SDU in the GPRS network after which the SDU is discarded. The maximum holding time depends on the protocols used (e.g., TCP/IP).

(b) Corrupt SDU probability: the probability that a SDU will be delivered to the user with an undetected error.

Delay : GPRS is not a "store and forward" service - although data is temporarily stored at network nodes during transmission - thus, any delay incurred is due to technical transmission characteristics (or limitations) of the system and is to be minimised for a particular delay class. The delay parameter thus defines the maximum values for the mean delay and 95-percentile delay to be incurred by the transfer of data through the GPRS network(s). The delay parameter defines the end-to-end transfer delay incurred in the transmission of SDUs through the GPRS network(s).

This includes the radio channel access delay (on uplink) or radio channel scheduling delay (on downlink), the radio channel transit delay (uplink and/or downlink paths) and the GPRS-network transit delay (multiple hops). It does not include transfer delays in external networks.

Delay Class	Delay (maximum values)			
	SDU size: 128 octets		SDU size: 1024 octets	
	Mean Transfer Delay (sec)	95 percentile Delay (sec)	Mean Transfer Delay (sec)	95 percentile Delay (sec)
1. (Predictive)	< 0.5	< 1.5	< 2	< 7
2. (Predictive)	< 5	< 25	< 15	< 75
3. (Predictive)	< 50	< 250	< 75	< 375
4. (Best Effort)	Unspecified			

Table 3 : Delay classes

Throughput : The throughput parameter indicates the user data throughput requested by the user.

Throughput is defined by two negotiable parameters:

- Maximum bit rate
- Mean bit rate (includes, for example for "bursty" transmissions, the periods in which no data is transmitted.)

The maximum and mean bit rates can be negotiated to a value up to the Information Transfer Rate value.

It shall be possible for either the MS or the network to re-negotiate the throughput parameters at any time during a session.

Scheduled repeated transmission : The scheduled repeated transmission parameter indicates the time of the first transmission, the number of transmissions and the interval between transmissions requested by the user.

QoS profile - PTP and PTM-G : The subscriber´s QoS profile for the PTP and PTM-G services consists of the following parameters which are negotiated or set to default values:

- service precedence (priority),
- reliability,
- delay,

- user data throughput.

QoS profile - PTM-M : The subscriber´s QoS profile for the PTM-M service consists of the following parameters which are negotiated or set to default values:
- service precedence (priority),
- reliability,
- user data throughput.
- scheduled repeated transmission.

2.2.2.4.TELESERVICES

In the following table, a set of mobile network's applications with a short comment for each one is presented. Additionally audio and video application's characteristics are mentioned for GPRS.

Service	Sub-service	Sub-sub-service	Explanation
Telephony	Speech		Real time 2 ways speech conversation
	Emergency call	Priority call	Allowed to proceed before any other call
		Pre-emptive priority call	Necessary resources will be released if necessary to set-up the required communication
		Access priority	Allows users to have different priorities
	Teleconference	Group call	2 way point to multipoint communication
		Acknowledged group call	Idem but the presence of the called parties is checked before setting the multipoint call
		Include call	Allows addition of one or more users to existing comm.
	Voice-Band Data	Modems	Modems of 14400 bps or more
		Facsimile	Group 1, 2 & 3 fax support
Sound			Transmission of sound between 7kHz and 20kHz
Video telephony	Video Telephone		2 ways speech and image with min of 64kbps throughputs
Teleaction Services			Low bit rate (2 kbps) with quick response
Message Handling Services	Short Message Service		Provides a means of sending messages of limited size using a Service Centre
	Paging		Textual messages from a Service Centre and a mobile
	Voice Mail		Enable calling users to record a voice message
	Fax Mail / Store & forward		Store to be sent & retrieve received fax to/from a fax. Paging may alert a user of the presence of an incoming msg
	Electronic Mail		Cover X.400 mail facilities
Telefax			Group 4 fax support
Teletex			Allow the exchange of text with coded info for text layout

Table 4

- *Interactive audio.* For example 64 kbits/s PCM coded, possibly multi-channel audio. In telephone applications the propagation delay should preferably be less than the order of 100 ms.

- *Distribution audio.* Up to 384 kbits/s, e.g. Musicam coded stereo. Propagation delay not critical, but the impact of Cell Delay Variation (CDV) should be studied further. Propagation delay should be below 200 ms.
- *Interactive Video.* 64 kbits/s to 384 kbits/s using H.261. Less than 64 kbits/s using MPEG-4.

2.2.3. COMPUTER SUPPORTED COLLABORATIVE WORKS

GSM and DECT are the most advanced digital mobile networks that have already been established. GPRS is one kind of packet switched data technology, which is being developed for GSM, and expected to be implemented in the 1999 – 2000 time frame. In the following sections the aforementioned mobile network application’s characteristics are being described.

2.2.3.1.CLASSIFICATIONS

Several classifications have been proposed using application functionality, internal architecture, geographical space, and time. In the following sections, we discuss each classification and present a brief description for each application type.

- Functionality

CSCW applications are being built for conversation support, workflow, co-authoring writing tools, drawings tools, audio/video conferencing, and meeting support/Group Decision Support Systems. Conversation/coordination applications are being developed to help persons interact with each other, the software acts as a moderator. Some artificial intelligence techniques are used for these applications. Supports interactive conversations, similar to talk, but using natural language processing techniques to keep track of the intentions for setting up the conversation. Workflow tools are used to alleviate paperwork in companies. These tools allow to annotate, send, and receive electronic documents. Some tools allow users to send/receive messages to/from traditional applications, applying the message-passing protocol concept. In general, these tools have some triggering rule mechanism to send messages whenever certain conditions have been met.

Co-authoring writing tools are multi-user editors, where one or more users can be editing the same document simultaneously. These tools should give the about the same functionality of a single user application, but should keep track of each user's operations (insert, delete, modify, etc.). The first CSCW program was NLA: On-Line System developed in Stanford Research Institute by Engelbart in the 1960's. Multi-user editors have been used to write technical papers among several authors.

Audio/Video conference applications allow two or more users to interact using audio, video, or audio/video. These applications can handle either analog or digital signals. Meeting support/Group Decision Support System normally include two or more of the above type of applications, such as audio/video conference and drawing tool.

- Internal architecture

The internal data architecture and process architecture commonly found in CSCW applications is :

Data Architecture. Many researchers have claimed that the single-user concept of separating the underlying information from its view is crucial for the development of multi-user applications. This implies that there should exist at least two layers, such that the state information, also called model layer or underlying model, interact with its presentation. The interaction should be considered in both directions: a modification in the state information might reflect a modification in the presentation, and an action at the presentation level might change some state information. Data architecture can also be classified as centralised, pure distributed, or replicated. In a centralised architecture, user process and data servers need to communicate whenever an update is sent by a participant. In a pure distributed data environment, persistent updates can be handled with well-known distributed algorithms, however a notification need to be sent to every participant. Replicated data is an identical copy at each user site. This approach requires an immediate update notification to each remote user to execute the same update. Several applications in this area take a replicated approach because the local response time is not affected. Crowley claims that a centralised approach is simpler since remote sites only need to intercept the input events, send these events to the central server, and broadcast any change to every site. This approach can preserver the required data consistency by serialising every update (transaction) in the central server. On the other hand, Crowley also claims that replicated architecture are more difficult to implement since it needs to: 1) propagate local input events to maintain identical local copies at each site and 2) use some concurrency control mechanism.

Process Architecture. The internal process architecture of CSCW applications can be divided into the following three categories: monolithic, client/server oriented, and pure distributed systems. As pointed out previously, groupware primarily goal is to share information. In general, sharing information is being done by some type of communication channel between the members of the work group. Monolithic systems are scarce since the trend is towards client/server architectures. Examples of this type of programs are MUD-Multiple User Dialogue (which rely on telnet for remote users), ALL-IN-1, and PROFS. The last two examples were developed several years ago. The second type of systems, client/server architecture, are being built for server(s) specialized in one or more services. In general, a server owns the device or information being shared by its clients. Client/Server applications can be divided even further in single server applications and multi-server applications. In a mono-user environment, the server acts as coordinator and supplier to its clients. Collage is an example of mono-server system. In Collage, all the users/clients are connected to one server. There is no server-server communication between multiple instances of the Collage system. The client-client communication is normally done through a central server, however some systems open user-user communication channels using the information from a name server. In the former case, the virtual topology is a star and in the latter case the configuration can vary dynamically from a totally unconnected to totally connected graphs. Multi-server applications can be built with specialized servers such as communication, name, printer, file servers. The communication channels between users/clients and servers vary as in the second case for single-server applications. Finally, pure distributed applications allow users to run processes and maintain data in several computer systems, transparently. The persistent data is not replicated, but the underlying information, that resides in main local memory at each user site, might be replicated, caching shared information to save network bandwidth.

It is conceivable any combination of data and process architectures described above. In practice, early systems used centralised data and process architectures. Presently, most systems are being built with client/server process architecture and replicated data architecture.

- Geographic: Local vs. Non-local

Groupware is also classified by the geographic location of the users. Local collaboration occurs in face-to-face meeting or message left at the same location, while non-local collaboration occurs at distant points. Programs as talk can be either local or non-local.

For example, an internal mailing system is local groupware while a teleconference is consider a distributed application. With the advent of Integrated Service Digital Networks (ISDN) and Broadband-ISDN, the difference between local and remote systems will probably not exist.

- Time: Asynchronous vs. Synchronous

In CSCW, synchronous or asynchronous term is defined for systems where two or more users share the same object at the same or different time, respectively. That is, synchronism is determined from the user's perspective, and not from the computers or network protocols involved.

Electronic mail is an example of an asynchronous system. The sender is writing the message, while the receiver is not aware of this situation. When the receiver sees the message, the sender is not aware that the message is being read. The two users are independent from each other. The notification of the arriving message and the message itself are the collaborative issues involved in this application.

Talk is an example of synchronous system, both users are interacting simultaneously with each other. The "shared screen" is updated with each character typed by both users. There is a mis-conception that synchronous systems are equivalent to WYSIWIS (what you see is what I see) applications.

Strictly speaking, talk is not a WYSIWIS tool since the two screens are different, even though they have the same information. Each user sees the message written by herself/himself in the upper section of the screen, while the lower section is used for the remote site messages.

WYSIWIS applications are a special case of synchronous applications, such that each sites share exactly the same view, and the input at any side will have the same effect in all the displays. To implement this behaviour, each side (computer/process) needs to change instantly its state no only by their local input but by an external, possibly remote, input, to keep the display consistent. For example, in a co-authoring program, if a user inserts a word, all the participants must see the new word immediately. This is refer as a fine grained synchronous system, opposed to coarse-grained where updates are performed in larger intervals. Scientists have identified that the last two classifications (time and locality) does not partition groupware applications. Most application will require an asymmetric approach, where some objects or entities can have a fine-grained synchronous behaviour, while other abstractions can be coarse-grain synchronous, asynchronous, or even unshared. For example, a card game as blackjack need to be modelled with an asymmetric approach. Each player can look at the open cards on the virtual table (synchronous) but only the player that owns a card should be able to look at the cover card (unshared).

2.2.3.2.EXAMPLES

The research community and software companies have developed software applications in all areas described above. The following sections describe briefly two or three applications for each area.

- Conversation/Coordination

Coordinator developed by Action Technologies, is a conversation application that keeps track of moves (question, opinion, etc.) made by each user, expecting that other users make a move in the same topic. Coordinator is an asynchronous system that uses e-mail messages, and even though it is described in the literature as a local system, there is no conceptual restriction to use it in a remote environment.

Argnoter is a synchronous tool developed by Xerox Parc Research Lab. The purpose of the tool is to keep track of the proposing, arguing, and evaluating phases of a conversation.

- Workflow

Lotus Notes from Lotus Development Inc., and Information Lens/Object Lens (basis of programming tool OVAL) from MIT are examples of this category. These applications are asynchronous systems that allow end-users to read messages sent to them, attach new annotations, send a version of the document to a group of people. Notes is currently the most popular and successful groupware product.

- Co-authoring writing tools

In the past few years several multi-user editors have been developed such as ShrEdit at University of Michigan and Comedia at University of Karlsruhe, Germany. In general, the co-authoring writing tools are synchronous programs, and some systems use the WYSIWIS technique. This technique is useful, for example to debug a program, where the participants are following the control flow of the program. Most editing collaborative tool use a relaxed WYSIWIS technique, where users can browse through the document freely, and they are aware that other persons are reading and modifying the file. ShrEdit is a WYSIWIS editor, while Comedia is a coarse-grain synchronous editor.

- Drawing Tool

Drawing Tools in multi-user environments are very useful for remote meetings (tele-meetings), allowing each participant to have access to a common drawing board. A person or group of persons can decide what drawings should be saved for the next meeting or for late-comers. In this way, late-comers can not only see the drawings in the board when the new user arrives, but can retrieve old drawings to keep up with the meeting.

Several drawing tools have been build such as GroupSketch from University of Calgary (basis for Groupkit programming tool); SketchPad from University of Karlsruhe, Germany; wb developed in UC at Berkeley; and ShowMe:WhiteBoard 1 from SunSolutions. All these programs vary their functionality from basic groupware support to complex multi-user drawing tools.

- Audio/Video conference

Examples of this category are *sd* (session director) from University of California-Berkeley that works with applications such as nv (video component) from Xerox Parc Research Lab, vat (audio component) from UC-Berkeley, and IVS (second video component) from INRIA.

ShowMe:Audio and ShowMe:Video and *Communique!2* and *Person-to-Person 3* are commercial products released recently. *ShowMe* has a single server internal architecture for SPARC stations. The audio/video information are digital signals that can be transferred using TCP/IP, ATM, or FDDI LAN/WAN networks.

Communique!, on the other hand, claims to be a multiplatform application under Unix and Windows/ NT operating systems. The functionality of this software includes: store/forward and record/playback video and audio digital signals, software compression algorithms (JPEG and MPEG), and allows users to display video, even if there is no video compression card installed in the machine.

These programs were developed mainly to meet requirements of synchronous systems, although some tools provide storing mechanism to play-back images and audio data. An image mail systems will be too expensive with the present disk technology and pricing.

- Meeting support/Group Decision Support Systems

This category includes applications such as *CoLab* from Xerox Parc Research Labs. This project developed three tools:

- *Boardnoter* drawing tool that allows remote users to share diagrams.
- *Cognoter* tool used to prepare presentations collectively, meetings are organized in the following three phases: brainstorming, organising, and evaluation.
- *Argnoter* a tool to prepare and evaluate proposals, is divided in: proposing, arguing, and evaluating phases.

2.3. IDENTIFICATION OF CASE STUDIES

Within the range of options spawned by the spectrum of applications and services identified above, we have chosen the following case studies to demonstrate the BISANTE traffic modelling approach:

1. WEB/HTTP: In response to the fact, that internet traffic contributes to a large amount of nowadays traffic, the first case study is dedicated to this type of traffic. Within the case study, several scenarios will be investigated: One scenario involves the evaluation of user behaviour in a closed environment (a student´s

computer lab) and the derivation of traffic characteristics. Another scenario analyses the traffic generation behaviour of a broader user community based on log file data of an internet service provider (NETWAY). The major goal in this case study is to identify classes of users for the construction of user profiles.

2. A/V Broadcasting Technology: The characteristics of Audio and Video traffic are a topic of major interest in the performance evaluation community. Numerous models for this type of traffic have been proposed in literature (see D1.1 for a survey). It is therefor reasonable to apply the BISANTE methodology to this type of traffic and to validate the results against other models. The main challenge is to find representative models characterising the audio and video streams and to be able to include user behaviour as feedback in those models (e.g. a user might change the quality of service requirements from a full screen color video to a window-sized black and white video, if the network capacity is decreasing).
3. Distributed Cooperative Work: Uniting remote workers in a virtual environment is the application of the future in telecommunication networks. In this case study, we will start with the evaluation of video-conference based communication services and collaboration tools such as shared editors. We will elaborate this case study in the second year of the project to also include mobile applications and communication services, focusing on GPRS.
4. Intranet Traffic: In the last case study, the type of traffic generated in a typical LAN-based office environment will be studied. As in such a scenario much more knowledge is available on user behaviour and applications, the derived models are expected to show a high degree of representativeness. We intend to use this case study to experiment with the different levels of detail of the modelling approach. A department of Thomson CSF will serve as a testbed for real measurements.

Further details on the case studies are given in DEL11-CS.

3. TRAFFIC SOURCE MODELLING

In this section, various stochastic models for traffic generation will be described. The intention of this description is to group the model types in such a way that models belonging to one type can be described by similar parameter sets. Creating models based on these types thus is done by identifying the appropriate values of these parameters.

When generating artificial network traffic, streams of requests can occur on several different levels of description. Such a stream *S* of request in essence is characterized by a sequence of observations

$$\ldots, X(t_{n-1}), X(t_n), X(t_{n+1}), \ldots$$

at time points

$$\ldots, t_{n-1}, t_n, t_{n+1}, \ldots$$

These observations now can describe, for instance, inter-arrival times between succeeding user commands at the user behaviour level or the inter-arrival times or sizes of data packets at the application or network level. Usually, the $X(t_i)$ are modelled by a family of random variables with known probability distribution function and time index *t*. If the set of possible values (the state space) is finite or countable, then the process is called discrete-state process, and a continuous-state process otherwise. The time index *t* may be finite or countable, yielding a discrete-time process $S = \{X_n\}_{n=0}^{\infty}$ or may take any value in a set of finite or infinite intervals, yielding a continuous-time process $S = \{X(t)\}_{t=0}^{\infty}$.

If the process describes the arrival of single discrete entities (packets, cells, commands,...), it is called *point process* , consisting of a sequence $T_0 = 0, T_1, T_2, \ldots, T_n, \ldots$ of arrival instants measured from the origin 0. An alternative description is given by *counting processes* $\{N(t)\}_{t=0}^{\infty}$, a continuous-time, non-negative integer valued stochastic process, where $N(t) = \max\{n : T_n \le t\}$ is the number of (traffic) arrivals in the interval $(0,t]$. Yet another description of point processes is given by interarrival time processes $\{A_n\}_{n=1}^{\infty}$, where $A_n = T_n - T_{n-1}$ is the length of the time interval separating the n-th arrival from the previous one. The equivalence of these descriptions follows from the fact that $T_n = \sum_{k=1}^{n} A_k$, and from the equality of events [JAG96]

$$\{N(t) = n\} = \{T_n \le t < T_{n+1}\} = \{\sum_{k=1}^{n} A_k \le t < \sum_{k=1}^{n+1} A_k\}.$$

3.1. RENEWAL MODELS

In a renewal process, the $X(t_i)$ are independent, identically distributed, but their distribution function is allowed to be general. Being independent means that the observation at time *t* does not depend on any observation in the past or future, thus the auto-correlation function for all lags $k \neq 0$ is equal to zero.

3.1.1. POISSON PROCESSES

A Poisson process describes the arrival of observations at certain points in time. The n-th interarrival time A_n is described by an exponential distribution

$$P\{A_n \leq \tau\} = 1 - e^{-\lambda\tau}$$

with mean arrival rate (mean number of arrivals per time unit) λ. The number of arrivals within an interval of length τ is described by a counting process satisfying

$$P\{N(\tau) = n\} = \frac{(\lambda\tau)^n e^{-\lambda\tau}}{n!}.$$

Creating a Poisson model means identifying the correct arrival rate λ.

3.1.2. BERNOULLI PROCESSES

Bernoulli processes are the discrete-time analog of Poisson processes. Arrivals can only take place at some time slot k. The probability of an arrival in such time slot is p, independent of others. The number of arrivals for slot k is binomial $P(N_k = n) = \binom{k}{n} p^n (1-p)^{k-n}$

The number of timeslots in between two arrivals is geometrically distributed with parameter p $P\{A_n = j\} = p(1-p)^j$, j being a non-negative integer.

Creating Bernoulli processes means estimating the correct arrival probability p.

3.1.3. PHASE-TYPE RENEWAL PROCESSES

Phase-type renewal processes [JAG96] denote arrival processes, whose interarrival times are modelled as the time to absorption in a continuous-time Markov process $C = \{C(t)\}_{t=0}^{\infty}$ with state space $\{0,1,...,m\}$. State 0 is absorbing, all other states are transient, absorption is guaranteed in finite time. To determine A_n, start the Markov process C with some initial distribution π. When absorption occurs (e.g. the process enters state 0), stop the process. The elapsed time yields A_n. Restart with the same initial distribution π to get A_{n+1}.

3.1.4. POSSIBLE APPLICATIONS OF RENEWAL PROCESSES

Renewal processes can be used to model arrivals, which are strictly independent from each other:

- The arrival of users to some company/computer facility.
- The arrival of network traffic packets, if the observed network traffic shows no auto-correlation.
- A stream of commands issued to an application, if no interdependencies on past results are observed.

3.2. MARKOV MODELS

A first step to describe dependencies between the $X(t_i)$ is given by Markov processes. A Markov process with discrete state space is called *Markov chain* [KLE75]. A set of random variables $\{X_n\}$ is called discrete-time Markov chain, if the probability that the next observed value (state) will be $x_{n+1} = j$ depends only on the current state $x_n = i$ and is given by p_{ij} . The dependency thus reaches back one unit in time and is also independent of the time, the process has spent in its current state (memoryless property). In the discrete-time case, the time spent in a particular state must therefore be geometrically distributed.

For a continuous-time Markov chain $X(t)$, the state changes can occur at any time t and the memoryless property demands that the time spent in a particular state *i* is distributed exponentially with parameter λ_i (depending on the current state *i*).

If traffic is to be generated by a continuous-time Markov chain, each jump from one state to the other (or possibly back to the same state) represents an entity arrival [JAG96]. If traffic is generated with discrete time Markov chains, each state *i* can correspond to *i* idle slots separating successive slots, and p_{ij} is the probability of a *j*-slot separation, given that the previous one was an *i*-slot separation.

A *Markov renewal process* $R = \{X_n, t_n\}_{n=0}^{\infty}$ is defined by the Markov chain X_n and its jump times t_n. The next state x_{n+1} and the next inter-jump time $\tau_n = t_{n+1} - t_n$ is completely determined by the current state x_n. Again, each jump signals the arrival of an entity. In contrast to ordinary Markov chains, interarrival times can be arbitrarily distributed [JAG96].

Markov processes can be used to model processes, where the observations depend on only the previous observed value:

- User behaviour: The next action is determined by the previous action plus maybe some return value (success, failure, ...)
- System/Network state change/failure
- Network traffic, if the observed traffic shows no or little auto-correlation.

In [CON96], a Markov chain approach is taken to model the length of Groups Of Pictures (GOPs) in an MPEG-1 encoded video. The GOP length is split into 2 time series, one being the original sequence and one being smoothed by a moving average filter of length W. Both time series are then quantized into *M* and *N* steps, yielding two discrete state time series. Combining these two chains into one yields a discrete time discrete state Markov chain with associated probability transition matrix.

3.3. MARKOV MODULATED TRAFFIC MODELS

In Markov-modulated models an explicit notion of state is introduced into the description of a traffic stream. An auxiliary Markov process is evolving in time and its current state controls the probability law of the traffic mechanism.

3.3.1. MARKOV-MODULATED POISSON PROCESSES (MMPP)

In this scheme, the current state k of a Markov chain M defines the currently used arrival rate λ_k of the modulated Poisson process. As the state of M changes, so does the rate [JAG96].

Creating MMPP means identifying all possible states of the system, together with the observed probability distributions and transition probabilities from one state to any other.

3.3.2. TRANSITION-MODULATED PROCESSES

Transition-modulated processes are a variation of the state modulation idea. The modulating agent is a state transition and can be described by a pair of states, whose components are the one before transition and the one after it.

Markov modulated processes are applicable, if the observed arrival stream changes its behaviour over time:

- Model network failures and their impact on the resulting network traffic.
- Model user behaviour: users might change their behaviour and with them, the traffic they create.
- Model empirical network traffic of different behaviour.

3.4. FLUID MODELS

If the number of arriving entities grows very large, each individual entity will add only negligible information to the traffic stream, just like the molecules in a water pipeline. As an example, the number of ATM cells per time unit might grow very large when sending high quality video information. When simulating such streams, the time granularity would be quite fine, and consequently simulation of all ATM cell arrivals would consume vast amounts of CPU time and main memory.

A fluid simulation would assume that the incoming fluid flow remains (roughly) constant over much longer time periods. Traffic fluctuations would be modelled by events signalling a change of flow rate [JAG96, KOB92, SEN89].

Possible Applications of Fluid Models are the following :

- ATM cells from near constant sources.
- TCP/IP traffic, if large volumes of data are transferred, and network conditions do no change.

3.5. LINEAR STOCHASTIC MODELS

The Markov property demands that the next observed state of a process can only depend on the current one, summing the whole process history into the current state. Autoregressive models define the next random variable X_n in the sequence as an explicit function of previous ones within a time window stretching from present to past. The distribution of the X_n is called marginal distribution, the autocorrelation function $\rho : N \to R$ yields for each $k \in N_0$, called the lag, the correlation coefficient of X_n and X_{n-k}.

Autoregressive models are suitable for modelling short-range dependencies, but fail to model long-range dependencies as often measured in VBR-coded video, web and Ethernet traffic.

A discrete-time stochastic process $\{X(t)\}_{n=0}^{\infty}$ is called *Gaussian process*, if for any finite set $\{t_1, t_2, ..., t_n\}$ of time points, the corresponding random variables $\{X(t_i)\}_{i=1}^{n}$ define a multivariate normal distribution. The marginal distributions thus consist of normal distributions.

The class of *linear stochastic models* [BOX70] has the form:

$$X_n = a_0 + \sum_{r=1}^{\infty} (a_r X_{n-r} - \beta_r \varepsilon_{n-r}) + \varepsilon_n, \; n > 0,$$

where the X_n are a family random variables, the α_r and β_r are real constants, the ε_n are zero-mean, IID random variables, called residuals or innovations, which are independent of the X_n. The most popular classes of linear stochastic models are called *AR(p), MA(q), ARMA(p,q)* and *ARIMA(p,d,q)*. The *ARFIMA(p,d,q)* models use a similar scheme, but are designed to yield fractal (self-similar) output.

3.5.1. Creating Linear Stochastic Models

- First, the exact type of model must be identified, i.e. whether it is AR(p), MA(q), ARMA(p,q) or ARIMA(p,d,q) [BOX70].
- Then, the parameters *p*,*q* and *d* must be identified. Usually, *p* and *q* are smaller than 2.
- Finally, the parameters α_i and β_j must be estimated [BOX70]. This can be done, for example, by using least-squares approximation [SHE98].

3.5.2. Possible Applications of Linear Stochastic Models

Because of their simplicity, AR(p) models are particularly well suited to model short-range dependencies:

- VBR coded video [JAG96]: Such videos produce a stream of frames of similar length, which can be modelled by autoregressive type models, while scene changes, causing a major burst, might be modelled by some modulating mechanism such as a Markov chain. In [ADD95], the mixture of two AR(1) traffic models is used to generate VBR coded video.
- Network traffic with rapidly decaying autocorrelation function.

Though linear stochastic models are members of the class of Gaussian models, the observed marginal distributions often differ from perfect Gaussian distributions by some skew.

3.5.3. THE DAR(P) MODEL

In [RYU96], a special autoregressive process, called DAR(p) (discrete autoregressive process of order p), is used to simulate VBR traffic and to measure the effectiveness compared to self-similar models. Let $\{\varepsilon_n\}$ be a sequence of IID random variables taking values in *Z*, the set of integers, with distribution π. Let $\{V_n\}$ be a sequence of Bernoulli random variables with $P(V_n = 1) = 1 - P(V_n = 0) = \rho$ for $0 \le \rho < 1$. For the DAR(p) process, ρ represents the first-lag autocorrelation. Let $\{A_n\}$ be a sequence of IID random variables taking values in $\{1, \ldots, p\}$ with $P(A_n = i) = a_i \ge 0,\ i = 1,2, \ldots, p$ with $\sum_{i=1}^{p} a_i = 1$. Let $S_n = V_n S_{n-A_n} + (1 - V_n)\varepsilon_n$ for $n \ge 1$, then the process $S = \{S_n\}$ is called DAR(p) process. This process has *p* degrees of freedom and can match up to the first *p* autocorrelations. It depends explicitly on the last *p* values. In [RYU96], it is also claimed that instead of taking into account all autocorrelations up to infinity, it is enough to take into account only a finite number up to an index called CTS (Critical Time Scale).

3.6. TRANSFORM-EXPAND-SAMPLE (TES) MODELS

Gaussian models assume Gaussian marginal distributions, yet real traffic measurements have revealed that this is not necessarily the case. More specifically, the observed marginal distributions often have heavier tails than Gaussian random variables. TES models [MEL93] capture both marginals and autocorrelations of empirical records. The method assumes that time series (such as traffic measurements over time) are available. It aims to construct a model capturing the empirical marginal distribution (by using histograms), the leading autocorrelations up to a reasonable lag and yielding output that resembles the observed records. TES is based on the following principles:

1. The Inversion Method: Let F be any distribution function and $U \sim Uniform(0,1)$. Then the random variable $X = F^{-1}(U)$ satisfies $X \sim F$. The sequence $\{U_n\}$ with uniform marginal distribution is thus transformed into a sequence $\{X_n\}$ with marginal distribution *F*. This principle can be expanded by using the empirical histogram of observed values instead of *F*.
2. Modulo-1 Arithmetic: Let *x* be any real number, then the floor operator $\lfloor . \rfloor$ is defined by $\lfloor x \rfloor$=max{integer $n : n \le x$ }. The modulo-1 operator $\langle . \rangle$ is defined for any real *x* by $\langle x \rangle = x - \lfloor x \rfloor$.
3. Iterated Uniformity: Let $U \sim Uniform(0,1)$ and let *V* be any real random variable. Define $W = \langle U + V \rangle$. Then $W \sim Uniform(0,1)$. Furthermore, let $U_0 \sim Uniform(0,1)$, and $\{V_n\}_{n=1}^{\infty}$ be a sequence of iid random variables with arbitrary marginal density f_V , and independent of U_0. The V_n are called innovations. Then the recursive scheme $U_n = \langle U_{n-1} + V_n \rangle,\ n > 0,$ is marginally uniform on $[0,1]$. Note, that the distribution of the V_n is completely irrelevant!
4. Foreground/Background Schemes: TES sequences consist of a background sequence $\{U_n\}_{n=0}^{\infty}$ of marginally uniform distributed random variables constructed

as described in 3, by using appropriate innovations V_n. The inversion method is then used to transform this sequence into a sequence $\{X_n\}_{n=0}^{\infty}$ by using the empirically observed histogram.

Possible Applications of TES Models are the following. TES models allow to include a variety of different autocorrelation functions, from slowly decaying, alternating in sign to oscillatory. Thus, TES models are well suited to model VBR coded video, Ethernet traffic and web traffic.

In [JEL96], a model for MPEG-1 traffic is given that splits the observed MPEG-1 traffic into two traffic streams.:

1. Slow time scale traffic: MPEG-1 traffic consists of I, B and P pictures. The used encoder in [JEL96] produces a sequence of IBBPBBPBBPBB sequences, called Group of Pictures (GOP). Over 8 such GOPs, the I, B and P sizes are averaged independently. The slow time scale traffic then consists of 8 IBBPBBPBBPBB sequences, where all I, B and P pictures have the same value.
2. Fast time scale traffic: This is the difference of the slow time scale traffic to the observed traffic.

The four random processes (I, B, P, fast time scale) then were modelled by using TES processes.

3.7. SELF-SIMILAR TRAFFIC MODELS

Empirical measurements of traffic have often shown the property of self-similarity, at least, if the traffic is high. A zero-mean, stationary time series $X = \{X_n\}_{n=0}^{\infty}$ can be *m*-aggregated to $X^{(m)} = \{X_k^{(m)}\}_{k=0}^{\infty}$ by summing the original time series over nonoverlapping blocks of size *m*. Then *X* is said to be *H*-self-similar if for all positive *m*, $X^{(m)}$ has the same distribution as *X*, rescaled by m^H:

$$X_n \stackrel{d}{=} m^{-H} \sum_{i=(n-1)m+1}^{nm} X_i = m^{-H} X^{(m)}$$

for all $m \in N$. Self-similarity can be described by the Hurst parameter *H*, for which $0.5 < H < 1$ holds, *H=0.5* indicating no self-similarity and *H=1* indicating perfect self-similarity. If the equality holds only for variances and autocorrelation function, the process is called second order self-similar.

There are several methods for estimating *H* from an empirical time series:

1. Variance-Time plot [LEL93, CRO97]: plots the variance of $X^{(m)}$ against *m* on a log-log scale. A straight line with slope $(-\beta)$ greater than -1 indicates self-similarity with $H = 1 - \beta / 2$.
2. R/S plot [LEL93, JAG96]: The rescaled range statistic R/S grows like a power law with exponent *H* for self-similar traffic.
3. Periodogramm method [LEL93, CRO97]: The shape of the power spectrum of a self-similar time series is a straight line on a log-log plot with slope $\beta - 1 = 1 - 2H$ close to the origin.
4. Whittle estimator: provides a confidence interval for *H* [LEL93, PAX96]. In [BER94], an S-PLUS program for the calculation of the Whittle estimator is

given. Before estimating, however, an appropriate stochastic model has to be chosen.

5. Wavelet estimator [ABR98, FEL98A]: In wavelet analysis the signal $x(t)$ is analysed by using a set of orthogonal basis functions $\phi_l^m(t)$, called wavelet functions, yielding the coefficients d_j^m (ref. to 3.7.4). Estimating $\hat{H}(j_1, j_2)$ for appropriate scalings j_1 and j_2 is done by plotting

$$\log_s\left(\hat{\Gamma}_x(2^{-j}\upsilon_0)\right) = \log_s\left(\frac{1}{n_j}\sum_m \left| d_j^m \right|^2\right)$$

against j, and applying linear regression. Here, $n_j = 2^{-j}n$ and υ_0 is an appropriately chosen reference frequency.

A confidence interval for H is given by

$$\hat{H} - \sigma_{\hat{H}} z_\beta \le H \le \hat{H} + \sigma_{\hat{H}} z_\beta,$$

where z_β is the $1-\beta$ quantile of the standard Gaussian distribution and

$$\sigma_{\hat{H}}^2 = \operatorname{var}\hat{H}(j_1, j_2) = \frac{2}{n_{j_1}\ln^2 2}\frac{1-2^j}{1-2^{-(J+1)}\left(J^2+4\right)+2^{-2J}}.$$

Self-similarity has strong influence on the resulting traffic and has the following properties:

- Long range dependencies: The autocorrelation function decays like a power law rather than exponentially.
- Heavy tailed distributions (like Pareto distribution) with infinite variance are observed. Extremely large values are more likely. The tail of a distribution is said to be heavy tailed, if it decays like a power law: $P[X > x] = 1 - F(x) = \hat{F}(x) \sim x^{-a}$. There are several methods to estimate the tail index α from given data:
  1. Plotting $\hat{F}(x)$ on log-log-axes [CRO99]: Plotted in this way, heavy-tailed distributions have the property that $\dfrac{d\log\hat{F}(x)}{d\log x} \sim -\alpha$, for large x. Linear behaviour in the upper tail is evidence of a heavy-tailed distribution.
  2. The Hill-estimator [CRO99]: The Hill estimator gives an estimate of α as a function of the k largest elements in the data set:

$$\mathrm{H}_{k,n} = \left(\frac{1}{k}\sum_{i=0}^{k-1}\left(\log X_{(n-i)} - \log X_{(n-k)}\right)\right)^{-1}$$

- Traffic bursts are observed. Such bursts in contrast to Poisson arrival processes with the same mean arrival rate will increase the mean waiting time and cell-loss probability due to buffer overflow drastically. Traffic bursts generally describe the ability of the process to stay below or above the average for a long time, and are strongly tied to large positive autocorrelations. There are some popular indices for burtiness [JAG96]:

1. Peak-To-Mean ratio (PMR).
2. Coefficient of variation for interarrival times $\{A_n\}_{n=0}^{\infty}$, c_A.
3. Hurst parameter (according to self-similarity) *H*.
4. Poisson traffic comparison (PTC).
5. Infinite server effect (ISE).
6. Index of dispersion for intervals (IDI), $J_A(n)$.
7. Index of dispersion for counts (IDC), $I_A(n)$.
8. The peakness functional $z_X[B]$.

Self-similar traffic has been observed in Ethernet [LEL93] and ATM traffic, telnet and ftp traffic [PAX95b], web traffic [CRO97] and VBR-video traffic [GAR94]. Artificial traffic of these types should thus also hold this property. The following sections will show some self-similar stochastic processes.

3.7.1. Fractional Brownian Motion

The zero mean Gaussian process $B_H(t)$ with Hurst parameter *H* is defined by

1. $E[B_H(t)] = 0$
2. $B_H(0) = 0$
3. $B_H(t+\partial) - B_H(t)$ is normally distributed $N(0, \sigma|\delta|^H)$
4. $B_H(t)$ has independent increments.
5. $E[B_H(t)B_H(s)] = \sigma^2/2\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right)$

$B_H(t)$ is exactly self-similar, perfectly determined by *H*.

In [HUE98], FBm is defined to characterise the number of arrivals in the time $(0,t)$:

$$N_t = mt + \sqrt{am}Z_t,$$

where *m* denotes the mean of the process, *a* is the coefficient of variation $Var[T]/E[T]$, and Z_t is the normalised FBm with Hurst parameter *H*.

Creating FBm traffic

Fractional Brownian Motion (FBm) [MAN68] can be created, for example, by the Random Midpoint Displacement (RMD) method [LAU95]:

1. Start with two end-points
2. Add one point in the middle of these two points, and displace it with a random term (which depends on *H*).
3. Add points between all existing points and displace them with random terms, until the desired number of points has been generated.

In [FLA92, SLN94], FBm is created by using wavelets.

Possible Applications of FBm

FBm can be used to model the sum or integral of self-similar traffic (as observed in network buffers, file sizes of audio/video streams, ...). Its increments/derivative can yield the self-similar fractional Gaussian Noise.

3.7.2. FRACTIONAL GAUSSIAN NOISE

The increments of FBm are known as Fractional Gaussian noise (FGn) [MAN68] and form a stationary process $G_H(t)$ with the following properties:

1. $G_H(t) = \frac{1}{\delta}(B_H(t+\delta) - B_H(t))$.
2. $G_H(t)$ is normally distributed with $N(0, \sigma|\delta|^{H-1})$.
3. $E[G_H(t+\tau)G_H(t)] = \sigma^2 H(2H-1)|\tau|^{2H-2}$ for $\tau >> \delta$.

Discrete time FGn also has the following autocorrelation function [JAG96]:

$$\rho_X(k) = \frac{1}{2}\left(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right), k \geq 1.$$

It can be necessary to truncate the FGn series, as negative values are possible.

Creating FGn traffic

In [PAX95a], an algorithm is given to efficiently create estimated discrete-time FGn. The Algorithm first generates an estimate of the power series $f(\lambda, H)$ of the desired traffic stream at the discrete frequencies $\lambda_j = 2\pi j/n, j = 1, \ldots, n/2$. Here, only the Hurst parameter H is necessary (for estimation see above). After some transformations, a sequence of n complex numbers is obtained, which is transformed back via the inverse Fourier transformation to obtain a sequence $\{x_n\}_{n=1}^{n}$. An instance of the algorithm is explicitly stated, programmed in the statistics language S.

In [BER94], an S-PLUS program creating FGn is given. Here, the series is generated by using the according covariances up to lag n and applying inverse Fourier transformation.

In [HUA95], an algorithm initially proposed in [HOS81] is briefly described.

Possible Applications of FGn

FGn are exactly second-order self-similar and are thus good candidates in modelling self-similar traffic:

- Ethernet [LEL93]
- ATM
- VBR coded video
- Web traffic, cache
- telnet, ftp

3.7.3. ARFIMA

Fractional ARIMA models (ARFIMA or FARIMA) [LEL93] are built on classical ARIMA models. $\{X_n\}_{n=0}^{\infty}$ is called an ARFIMA(p,d,q) process, if $\{\Delta^d X_n\}_{n=0}^{\infty}$ is an ARMA(p,q) process for some non-integer *d>0*. *B* is the Backshift-operator $B(X_n) = X_{n-1}$ and Δ^d can be represented by $\Delta^d = (1-B)^d = \sum_{u=0}^{\infty} \pi_u B^u$

with $\pi_0 = 0$ and $\pi_u = \frac{\Gamma(u-d)}{\Gamma(u+1)\Gamma(-d)} = \prod_{k=1}^{\infty} \frac{k-1-d}{k}, u = 1,2,...$

ARFIMA processes are asymptotically self-similar, if $0 < d < 0.5$, with Hurst parameter $H = d + 0.5$.

For large lags, the correlations of an ARFIMA(p,d,q) process are similar to those of an ARFIMA(0,d,0) with the same *d*.

Creating ARFIMA models

The fractional differentiating parameter *d* can be estimated from a previous estimate of the Hurst parameter *H* by using the above equation. After this, the observed time series must be fractionally differenced to yield a new time series

$$\{Y_n = (1-B)^d X_n\}_{n=0}^{\infty}.$$

For the new time series, an appropriate ARMA(p,q) model is then created [BOX70]. In [GAR94], an algorithm is given for the creation of ARFIMA(0,d,0) processes with arbitrary marginal distributions. The algorithm, though, is of complexity $o(n^2)$ and required 10 hours of CPU time for generating 171,000 points on an 1994 state of the art workstation.

In [BER94], an S-PLUS program for generating ARFIMA(0,d,0) series is given.

Possible Applications of ARFIMA models

ARFIMA models are similar to FGn, yet they are very flexible due to the natural correspondence to ARIMA(p,d,q) models and to their higher number of parameters.

In [GAR94], VBR coded video traffic is modelled with ARFIMA models.

3.7.4. WAVELETS

The above described stochastic models try to capture short- and long-term dependencies as observed in VBR video or Ethernet traffic. Wavelets now provide a means of transforming the original self-similar process into a new process with **much less** self-similar behaviour. For this new process, simpler models can be applied. Traffic is then generated first in the wavelet domain, and then transformed back into the time domain by applying the inverse wavelet transformation [SHE98, FLA92, SLN94].

The Wavelet Transform

Like in the Fourier transform, a the observed values $\{x(t)\}_{t=0}^{2^K}$ (for some integer K) of an equally spaced, discrete-time process are analysed according to a complete orthonormal basis of the Hilbert space $L^2(\Re)$ of all squared integrable functions [POS98]. The members $\phi_j^m(t)$ of this orthonormal basis are derived from a special function $\phi(t)$, the mother wavelet, by translation in the time domain, and scaling in the frequency domain [SHE98]:

$$\phi_j^m(t) = 2^{-j/2}\phi(2^{-j}t - m).$$

Here, the positive integer m denotes the translation index, while the positive integer j denotes the scaling index. The task of wavelet transformation is to find wavelet coefficients d_j^m such that

$$x(t) = \sum_{j=0}^{K}\sum_{m=0}^{2^{K-j}-1} d_j^m \phi_j^m(t) + \phi_0$$

holds for $0 \le t < 2^K$. This is called the *inverse* wavelet transform. The wavelet coefficients are given by

$$d_j^m = \sum_{t=0}^{2^K-1} x(t)\phi_j^m(t).$$

There are several popular mother wavelets. One, for example, is the Haar wavelet

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \le t < 1/2 \\ -1, & \text{if } 1/2 \le t < 1 \\ 0, & \text{otherwise} \end{cases} .$$

The corresponding Haar wavelets $\phi_j^m(t)$ are scaled and shifted versions of $\phi(t)$. For Haar wavelets, the corresponding wavelet coefficients are given by

$$d_j^m = 2^{-\frac{j}{2}}\left(\sum_{t=m2^j}^{(m+0.5)2^j-1} x(t) - \sum_{t=(m+0.5)2^j}^{(m+1)2^j-1} x(t)\right).$$

Though the wavelet coefficients have two indices, they can be transformed into a discrete-time process d_s by using a triangular scheme:

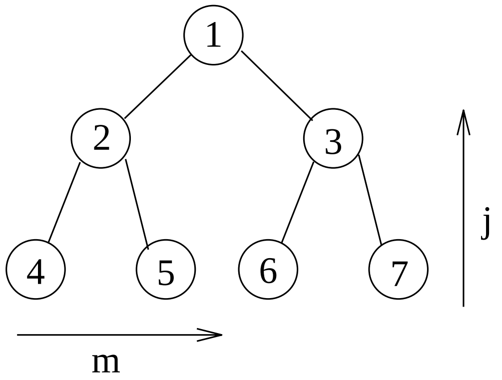


Figure 1

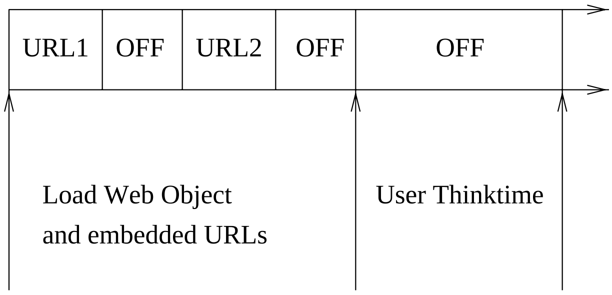


Figure 2

Furthermore, if the observed process consists of random variables, then the wavelet coefficients themselves are also random variables. Due to the one-to-one correspondence of the input process and wavelet coefficient process, the statistical properties of $x(t)$ are completely determined by the statistical properties of the wavelet coefficients.

Experiments show that the auto-correlation function of the new discrete-time process d_s decays much faster (exponentially) than that of the original (possibly self-similar) process. Thus, simpler models like Gaussian type models can be used to model this new process.

Creating Traffic with Wavelets

Network traffic is generated in the following way [SHE98]:

1. Sample $N = 2^K$ observation values.
2. Compute the wavelet coefficients d_j^m for this data.
3. Transform the indices to get the new process d_s.
4. Model d_s by a simple Gaussian type, stochastic process (*n*-th order Markov).
5. Create 2^K variates in the wavelet domain by using this stochastic process.
6. Create the required $x(t)$ in the time domain by applying the inverse wavelet transform.

As the complexity of the wavelet transform and inverse wavelet transform is of order $O(N)$, where $N = 2^K$ is the length of the time series, the complexity of the whole algorithm is $O(N)$. This makes the scheme very efficient!

Possible Applications of Wavelet Generated Traffic

Wavelets are capable of capturing both short-range and long-range dependencies [SHE98]. They are thus well suited for modelling Ethernet, ATM, VBR, telnet, ftp and web traffic.

3.7.5. ON / OFF SOURCES

A large number of superimposed heavy tailed On/Off processes [CRO95] can yield self-similar traffic as well. An On/Off process is either in state *On* or *Off*. We construct a time series by observing the number of On-processes at any time point. If On-times and Off-times are drawn from a heavy tailed distribution like the Pareto distribution with parameters α_1 and α_2, then the observed stochastic process is a self-similar fractional Gaussian noise process with $H = (3 - \min(\alpha_1, \alpha_2))/2$. On/Off-processes are mapped to network traffic in the following way:

- Each process corresponds to a workstation either being silent (Off) or sending data at a constant rate (On).

- Each process correspond to a web user, On-times are given by the web document transmission times and Off-times are the time intervals between the transmissions. This model can be refined by modelling active-Off times (time between the transmission of two files belonging to the same HTML document) and inactive-Off times (time between user actions) as well. Transmission times of files are a function of their length, thus the distribution of web file length has been shown to be heavy-tailed. Zipf's law connects this file length to the number of times, a file has been transmitted (file popularity).

In [RYU98], mixtures of fractal On/Off processes, called fractal point processes, are discussed.

3.7.6. POISSON-ZETA MODEL

A Poisson-Zeta model $PZ[\alpha, \rho]$ is a discrete time On/Off process, where the number of bursts at each time point *n* is given by a Poisson distribution with mean α. Each burst generates one cell (ATM) per time unit during its duration, the duration *l* of each burst has independent identical Zeta distributions $\{g_h\}_{h=1}^{\infty}$ (like a discrete version of Pareto) with parameter $1 < \rho < 2$. g_h is the probability that the burst will last for *h* time units. In [LIK95] it has been demonstrated that g_h is asymptotically self-similar.

3.7.7. DETERMINISTIC CHAOTIC MAPS

Deterministic chaotic maps are related to On/Off sources [ERR94]. Here, the driving sequence is derived from chaotic processes having the SIC (Sensitive dependence on Initial Conditions) property. In such processes, the observed trajectories severely depend on the starting point. Changes of these starting points have exponential effects on the observed trajectories. Traffic is produced by creating the stochastic processes x_n and y_n:

$$x_{n+1} = f_1(x_n), \quad y_n = 0, \quad if\ 0 < x_n \le d$$
$$x_{n+1} = f_2(x_n), \quad y_n = 1, \quad if\ \mathrm{d} < x_n < 1$$

for an appropriately chosen *d* and map functions $f_1(x)$ and $f_2(x)$. If x_n is above the threshold, then one traffic packet is generated. In [ERR94], two maps, the piecewise linear maps

$$x_{n+1} = \begin{cases} \dfrac{x_n}{(1-\lambda)} & 0 < x_n \le (1-\lambda) \\ \dfrac{x_n - (1-\lambda)}{\lambda} & (1-\lambda) < x_n < 1 \end{cases}$$

and the intermittency maps

$$x_{n+1} = \begin{cases} \varepsilon + x_n + c x_n^m & 1 < x_n \le d \\ \\ \dfrac{x_n - d}{1-d} & d < x_n < 1 \end{cases} \qquad \text{where } c = \frac{1-\varepsilon-d}{d^m}$$

are then defined.

3.7.8. Self-Similarity Through Aggregation

A more sophisticated process, yielding self-similar traffic through aggregation, is given in [LEL93] and [JAG96]. Let $\{I_k\}_{k=0}^{\infty}$ be a sequence of iid integer-valued random variables with asymptotic tail probabilities obeying the power law (for example Pareto) $P\{I_k \ge t\} \approx t^{-\alpha} h(t), \quad \text{as}\, t \to \infty,$

where $1 < \alpha < 2$ and *h* is a slowly varying function. Let $\{G_k\}_{k=0}^{\infty}$ be an iid sequence,

$$S_k = S_0 + \sum_{j=1}^{k} I_j, \quad k \ge 1,$$

independent of $\{I_k\}$, with $E[G_k] = 0$ and $E[G_k^2] < \infty$. Define the stationary sequence with an appropriately chosen S_0. Then define $W = \{W_k\}_{k=1}^{\infty}$:

$$W_k = \sum_{n=1}^{k} G_n \operatorname{Index}[(S_{n-1}, S_n]](k),$$

Index() being the index function. Construct M iid copies $W^{(1)}, \ldots, W^{(M)}$ of *W*. Then the process $W^* = \{W_k^*(M)\}_{k=0}^{\infty}$, given by

$$W_k^*(M) = \begin{cases} 0, & k = 0 \\ \sum_{n=1}^{k} \sum_{m=1}^{M} W_n^{(m)}, & k > 0 \end{cases}$$

behaves like FBm, provided that *k* and *M* are large and $k << M$.

3.7.9. THE M/G/∞ MODEL

In [PAX95b], an M/G/∞ model is stated, which is capable of constructing asymptotically self-similar traffic. Let $\{X_t\}_{t=0,1,2,\ldots}$ be the counting process denoting the number of customers in the M/G/∞ system at time *t*. If customers have a service distribution function *F*, then the autocorrelation function of X_t is

$$r(k) = \rho \int_k^\infty (1 - F(x))dx,$$

where ρ is the rate of the Poisson process of customers arriving at the system. If *F* is the Pareto distribution, then

$$r(k) = \rho \int_k^\infty \left(\frac{a}{k}\right)^\beta dx = \frac{\rho\alpha^\beta}{\beta - 1} k^{(1-\beta)},$$

and thus the process is asymptotically self-similar.

In [PAX95b], various aspects of telnet and FTP traffic in connection with M/G/∞ are discussed.

3.7.10. SUPERIMPOSING AR(1) PROCESSES

In [LEL93] it is stated that when aggregating many simple AR(1) processes, where the AR(1) parameters are chosen from a beta-distribution on $[0,1]$ with shape parameters *p* and *q*, then the superposition process is asymptotically self-similar. Also, the Hurst parameter *H* depends linearly on the shape parameter *q* of the beta-distribution. Obviously, creating the AR(1) processes can be done in parallel.

In [ADD95], the mixing of two AR(1) processes is used to generate ATM traffic.

3.7.11. SELF-SIMILAR MARKOV MODULATED

In [ROB97], self-similarity is simulated by using a Markov modulated discrete-time, discrete-state process. The proposed modulating Markov chain depends only on 3 parameters.

VBR, telnet, ftp, Ethernet, Web, ... are possible applications.

3.7.12. THE GBAR AND GBMA PROCESSES

The GBAR process [HEY97] is a Gamma-Beta autoregressive process. Let $Z_{i-1} \sim Gam(\alpha,1)$, $W_i \sim Gam((1-\rho)\alpha,1)$, and $B_i \sim Beta(\alpha\rho, \alpha(1-\rho))$ be independent, then

$$Z_i = B_i Z_{i-1} + W_i$$

is also $Gam(\alpha,1)$-distributed. The autocorrelation function of this process is geometric.

As is stated in [FRE98], triangular shaped autocorrelation functions can be derived from moving averages of Gamma processes. Any kind of autocorrelation can be modelled by weighting Gamma processes with Beta distributions, then applying the moving average filter to it (GBMA process).

In [FRE98], the GBAR and GBMA models are used to model the sizes of MPEG frames.

3.7.13. Spatial Renewal Processes

Spatial renewal processes [JEL96, DEV98] consist of two background processes, the first being a point process $T = \{T_0 \le 0, T_n, n \ge 1\}$, such that the inter-renewal times $T_n - T_{n-1}, n \ge 1$ are iid with distribution function $F_T(t)$. A second process $\{X_n\}_{n=0}^{\infty}$ consists of iid random variables with the desired marginal distribution as observed. These two processes together then yield the foreground process $Y_t = X_n$ for $T_n \le t < T_{n+1}$. If the desired autocorrelation function $\rho(t)$ for $\{Y_t\}$ is either given empirically or known analytically (for example, if we want to generate FGn, then the autocorrelation function is known for discrete points and must be extended to the set of real numbers), then we just have to construct $F_T(t)$ over equation

$$1 - \rho(t) = \mu^{-1} \int_0^t (1 - F_T(u)) du, \; t \ge 0$$

or equivalently

$$-\frac{d}{dt}\rho(t) = \mu^{-1}(1 - F_T(t)), \; \rho(0) = 1, \; t \ge 0,$$

where

$$\mu = \int_0^\infty (1 - F_T(u)) du.$$

In order to yield valid distribution functions, the used autocorrelation function must be a decreasing, concave-up function. **The constructed foreground process $\{Y_t\}$ will then have the desired marginal distribution and the required autocorrelation function!** In [DEV98], the distribution function $F_T(t)$ for FGn is stated explicitly.

3.7.14. Possible Applications of Spatial Renewal Processes

In [JEL96], SRP are used to model MPEG-1 encoded video streams. Other possible applications include Ethernet and ATM traffic, WAN and Web traffic.

3.8. MULTIFRACTAL TRAFFIC

In [FEL98B], the multifractal nature of WAN traffic is demonstrated. In contrast to monofractal (self-similar) traffic, where the local scaling behaviour is constant, multifractal traffic takes into account the changing local scaling behaviour over time. This local scaling behaviour is measured as the rate, at which the number of bytes/packets observed in the interval $[t_0, t_0 + \delta t]$ tends to zero as $\delta t \to 0$. In [FEL98B], this local scaling behaviour is calculated by using wavelet transforms. The multifractal property is then motivated by the cascading nature of WAN traffic (each trace consists of sessions, each session consists of traffic requests, each traffic request consists of TCP connections, each TCP connection consists of IP packets, ...).

3.9. OVERVIEW OF TRAFFIC GENERATORS

In this section, an overview of the bibliography for modelling and generating traffic of certain types is given:

3.9.1. VBR VIDEO TRAFFIC/MPEG

- AKY97: Geometric On/Off.
- CHO98: Periodic Markov modulated batch Bernoulli.
- CON96: Markov chain.
- FRE98: GBAR, GBMA.
- GAR94: ARFIMA(0,d,0).
- HEY96a: Discrete AR, Markov chain, Scene changes.
- HEY96b: Markov chain.
- HUA95: FGn, Arbitrary marginal distribution.
- ISM96: TES.
- JEL96: Multiple time scale TES, Spatial renewal processes.
- KRI97: FBm, DAR (Markov), Markov chain.
- LAZ93: TES.
- LAZ94: Generalized TES.
- LIE98: Leaky bucket, empirical envelopes.
- MCL91: Markov modulated On/Off.
- MEL92: TES.
- PAX95a: FGn.
- ROS97: Markov chains.
- RYU96: DAR(p).
- SHA97: TES.
- SHE98: Wavelets.
- TAR98: Spatial renewal processes.

3.9.2. ETHERNET

- ABR98: Wavelets.

- LEL93: FGn, ARFIMA(p,d,q), aggregation.
- ROB97: self-similar Markov modulated.
- RYU98: Fractal point processes.
- SHE98: Wavelets.

3.9.3. ATM

- ADD95: Superimposing two AR(1) processes.
- DAN98: On/Off sources.
- DIA98: heavy-tailed Renewal.
- FAN97: Poisson-Zeta (On/Off).
- JEN96: On/Off sources.
- NEA95, NEA97: FBm, mix of two AR(1), M/Pareto.
- TRY97: On/Off sources.
- ULR98: MMBP.

3.9.4. WAN, TCP, TELNET, FTP, NNTP, SMTP

- FEL98A: Wavelets, $M/G/\infty$.
- FEL98B: Multifractals
- HUE98: Poisson, MMPP, AR(1), Weibull, Pareto, FBm.
- PAX94: Interarrvial times, autocorrelation function.
- PAX95b: Pareto, $M/G/\infty$, Log-normal.
- RYU98: Fractal point processes.

3.9.5. WEB TRAFFIC

- BAR98b: On/Off sources, Zipf's Law.
- CRO95, CRO97: On/Off sources, Zipf's law.
- DIL98: Poisson.
- FEL98A: Wavelets, $M/G/\infty$.
- IYE98: ARIMA(p,d,q).

4. A USER BEHAVIOUR ORIENTED MODELLING APPROACH

Within BISANTE, the availability of reliable, parameterised user models is of utmost importance. These user models will, together with traffic models of selected applications, drive the simulation engine by creating realistic network traffic. Realistic here means that a large number of such users and applications should create a traffic that has similar characteristics as observed traffic.

Which characteristics to mimic is an important question and depends on the answers to find. If questions of general interest like the mean busy time of servers or the probability of rejections of incoming service requests are of interest, then simple characteristics like independent arrival rates and data size distributions might be sufficient. If, on the other side, more detailed results like queue size distributions or buffer overflows are to be derived, then stronger characteristics like the autocorrelation function of the observed traffic or the Hurst parameter should also be similar to the observed traffic.

Also, the level of detail, at which network traffic is described and generated, can vary, the description of each individual packet on the one hand being the most detailed, while on the other hand descriptions of data volumes per time unit (like fluid models) being a low detail description. Again, the questions to answer determine the level of necessary detail. If large numbers of users or long time periods are to be observed, less details will increase simulation speed and will make it possible to gain results that are otherwise too expensive to be computed.

The central entity to model will be the user, starting and stopping applications and services, jumping from one application to the other and issuing commands. Such commands have influence on the services and thus on the generated network traffic.

4.1. USER BEHAVIOUR MODELS

In this section, mathematical models capturing the behaviour of users will be discussed. These user models will generally use lower level models for generating streams of network traffic requests as discussed in the previous sections.

State Transition Diagrams : The user behaviour can be represent by a discrete-time or continuous-time, discrete-state stochastic process. The states represent actions or commands produced by the user, the transition from one state to another means issuing this action/command and thus causing network workload.

- *User Behaviour Graphs*

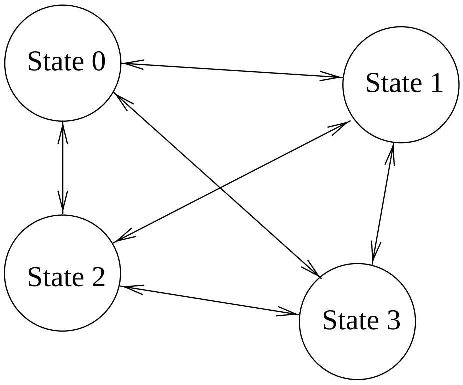


Figure 3

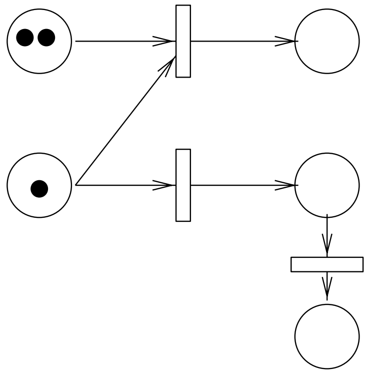


Figure 4

The states can represent single commands or classes of commands, commands of each class having similar characteristics [CAL90]. By assigning an *n*-dimensional resource vector to each command (for example, the CPU-time, the number of bytes transferred, ...), the commands can easily be classified by using the k-means or some hierarchical clustering algorithm.

- *States represent Stream of Commands*

By being in state *k*, the user can produce more than one command. This stream of commands can be modelled by any method explained in the previous sections.

Petri Nets : User behaviour can also be modelled by Petri nets. A Petri net is a graph with two different kinds of nodes. Places (usually represented by circles) can hold $n \geq 0$ tokens and thus represent the state of the system. Transitions move tokens from one place to the other and thus represent actions changing the state of the system. Petri nets can be solve analytically under certain constraints by using the embedded Markov chain (each state represents a different Petri net marking). The explosion of the number of possible states of this Markov chain often makes this analytical approach impossible in reality.

Petri nets are very flexible and can be applied for user modelling in the following ways:

1. One Petri net per user. In each Petri net, there is only one token. The place, this token is currently situated in, defines the state of the user.
2. One Petri net for all users. One token per user. The positions of the user tokens define their states.
3. By entering a new place, the token can produce one or several actions (streams of commands).

Probabilistic Attributed Context Free Grammars : The hierarchical nature of network workload leads to the use of context free grammars for its description [RAG95]. A context free grammar consists of a start symbol WL, a set of terminals (cannot be exchanged further) and non-terminals (can be exchanged further) and a set of production rules:

PR I: $WL \rightarrow aA_i \mid aA_i Z$

PR II: $Z \rightarrow aA_i Z \mid \varepsilon$

PR III: $A_i \rightarrow aA_i \mid \varepsilon$

In the upper example, *WL* denotes the start symbol, *a* and ε are terminals and A_i and Z are non-terminals. By using the rules, first the start symbol and later on other non-terminals are exchanged to non-terminals or terminals. If several rules are possible, one is selected stochastically. Additionally, Attributes are derived from appropriate distributions. This is repeated, until no further non-terminal exists. The result of this process is a stream of terminals, each attributed by a start time and a stop time. Each terminal represents some amount of work on its level of workload description. According to the hierarchy, terminals represent

1. User sessions
2. Open windows
3. Commands issued in these windows
4. Network requests issued by the commands
5. Network packets

A stream *S* of terminals might look in the above example might look like $S = \{aaaaaaaaaaa\varepsilon\}$, each *a* having a start and stop time and representing a session/window/command etc.

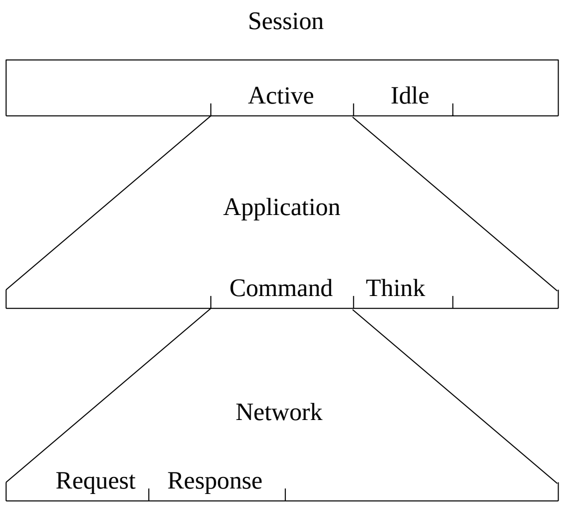


Figure 5

4.2. BEHAVIOUR BASED USER CLASSIFICATION

In this section, ways of characterising and classifying users will be discussed. General user models then can be parameterised to yield classes of models, each representing a user class.

4.2.1. USER BEHAVIOUR INDICES

This technique requires that the construction of a discrete-time Markov chain describing the possible user states $S = \{1, \ldots, M\}$ [RAG85]. User states can, for example, represent a certain class of commands issued, or a class of web documents loaded. There are two types of probabilities:

1. p_i denotes the probability of entering state i. This represents the static user behaviour.
2. p_{ij} denotes the probability of entering state j, provided that the current state is i. This transition matrix describes the dynamic user behaviour .

Classes of users are created by using the theory of entropy in information:

$$\text{User behaviour entropy (UBE)=} \sum_{i=1}^{M} p_i \log(p_i) / \log(M) .$$

$$\text{User behaviour mobility (UBM) =} \sum_{i,j=1}^{M} p_i \log(p_{ij}) / [M \log(M)].$$

For each user, one tupel (UBE,UBM) is created and plotted in the 2-dimensional plane. By applying clustering techniques, appropriate user classes can be constructed.

4.2.2. WEB SITE ANALYSIS

To build reliable user models, data from web site accesses has to be traced and analysed. To achieve this, several techniques for data sampling and grouping can be applied.

Clustering : Every recorded user can be marked with an n-tupel $(r_1, \ldots, r_n)$, each r_i representing a value of interest, for example, the average size of the downloaded HTML-files, the average sum of the sizes of all files being embedded into these files, the variance of these variables, the number of actions/commands of type i, ...

By using an appropriate clustering algorithm (k-means, hierarchical), the user classes can be constructed for each cluster.

In [YAN96], a subset of all HTML pages/files of a web server were taken and number from 1 to *n*. Each access to such a document was then marked by increasing the documents counter, yielding an *n*-dimensional vector for each session. Clustering techniques were then applied to find access patterns.

In [SAL87] and [SAL90], the contents of text documents (called terms) are used to assign *n*-dimensional vectors to each document. Let *N* be the number of documents, *t* be the number of terms under observation, then the vector

$$D_i = (w_{i1}, \ldots, w_{it})$$

is assigned to document D_i, where the term weights w_{ik} are given by

$$w_{ik} = \frac{tf_{ik}\log(N/n_k)}{\sqrt{\sum_{\text{all terms } j \text{ in } D_k}\left(tf_j \log(N/n_j)\right)^2}}.$$

tf_{ik} is the term frequency of term *k* in D_i, and n_k is the number of documents containing term *k*. The inner product of two such vectors yields a measure of distance for two documents, clusters of documents then can be found by continuously adding documents to clusters of smallest distance. Clusters then can be linked together by defining the smallest distance of two documents in different clusters.

Terms can be generated from all words in the HTML documents, excluding stop words and applying suffix stripping and stemming (retrieving word stems). To decrease the number of dimensions (=number of word stems), [VLA98] propose word correlations. The correlation $-1 \le corr_{ij} \le 1$ of two words T_i and T_j is near 1, if the existence of T_i in a document implies the existence of T_j in the same document. The reverse is not necessarily true, this correlation thus is not commutative! For example, the correlation of "applet$\rightarrow$java" was found to be one.

In [CHE97], two additional metrics for document distances are given. One is defined by the number of hyperlinks from document D_i to D_j related to the total number of hyperlinks in D_i, the other by one-step transition probability matrices. All three forms of similarity measures (content, link, state-transition) form the so-called *Generalized similarity analysis*.

In [VLA98], a neuronal network is used to apply generalised similarity analysis in order to build linked clusters.

In [PIR96] , each web page of the Xerox web locality is assigned a vector with the following components:

- Size. In bytes of the item
- Inlinks. The number of links into this document from the Xerox web locality.
- Outlinks. The number of links from this document to other documents of the Xerox web locality.
- Frequency. The number of times the item was selected during some sampling period.
- Sources. The number of times the document was selected as start point of a path traversal.
- CSIM. The textual similarity of this item to its children.
- Cdepth. The average depth of the item's children measured by the number of "/" in their path.

Temporal Analysis : At the page level, the time spent reading can be measured as the interarrival time between the request for the page and a subsequent page request [PIT97]. If a sufficiently large sample is gathered, reasonable statements can be made.

In [CAT95], user sessions were artificially separated, if the time between events was larger then 25.5 minutes, where the mean event inter-arrival time was found to be 9.3 minutes. Other log-file analysis programs use 30 minutes as session time-outs. Other metrics are the time, spent by the user on a specific site (session length) and the time between visits (inter-visit period).

Path Analysis : Instead of including the observed time into the model, in path analysis, the sequence of requested web documents, together with the posting of data to web servers is analysed.

- *Modelling Web Sites as Trees*

When designing sites with the shape of a tree [PIT97] (with one root page, each page then can have one or several ancestors and one or several children). With this information, each page is then marked with a certain depth within the tree (measured from the root), each visitor will then generate a path on the tree with minimal, maximal and mean depth. Also, the number of internal (are often used for navigation) and leaf nodes (often contain content) traversed can be measured. Entry points can be identified by looking at the difference between the sum of all the incoming paths to a page and the total number of requests made for a page. Large differences will indicate that visitors do not rely on the proposed site structure. Likewise, exit points can be identified by looking at the last elements of the logged paths.

- *Browsing Strategies*

Browsing strategies [CAT95] can be observed for users using web browsers. In such a scenario, a user will download a sequence of web pages, each being stored on a web server. By jumping from one web page to the other, the user might also jump from one web server to the other. One such path of length 3 might be, for example www.gatech.com to www.ici.com to www.ncsa.edu.com. By using pattern detection algorithms, general paths of length n and their frequencies $f(n)$ can be detected. By plotting $f(n)$ against n, a near-linear relationship with negative slope s has been found, the average slope being $s = -0.24$. This slope can then be used to construct classes of users with the same behaviour.

The same method for pattern recognition has been applied in [TAU97], this time to the requested URLs, in order to identify the longest repeated sub-sequences (LRSs) of page visitations. It is reported, however that LRS is not a good means for predicting web browsing.

- *Recurrence Metrics*

In [TAU97], several metrics and plots are given to understand, how users revisit site pages. This is important for the design of efficient history tools. The *recurrence rate* is defined (per user or session) as

$$R = \frac{\text{total URLs visited - different URLs visited}}{\text{total URLs visited}} \times 100.$$

The *vocabulary* is the set of unique URLs requested so far. On a per user or per session basis, this vocabulary can be plotted against the number of requests (as a monotonically increasing function). Depending on the slope of this graph, the authors identified several different strategies:

1. First-time visits to a cluster of pages (interconnected by URLs). Evident, if the slope is steep.
2. Revisits to pages. Evident for plateau areas.
3. Authoring of pages. Also showing plateaux. Authoring denotes the changing of existing pages and the following reloads to check, if the changes are as planned.
4. Regular use of web-based applications. Moderate slopes, consisting of Open URLs, Back and Forms activities. For example, when using knowledge retrieval tools on web basis.
5. Hub-and-spoke. People visit central pages (hubs), including a large number of links, and from there, following the links to new pages and immediately back again to the hubs.
6. Guided tour. Some pages and tutorials include standard links like next-page. People following these links, issuing a stream of Open URL requests.
7. Depth-first search. People follow links deeply before returning to the central page. This results in a stream of OpenURL requests, followed by a stream of Back commands.

The URL *visit frequency* can be plotted as a function of distance. In this sense, the percentage of revisits to some page, provided the same page has been accessed *n* steps before, is plotted over *n*:

- Reloading a page results in a revisit of distance one.
- Going back one page in the history results in a revisit of distance two.
- Clicking again on the back button results in a revisit of distance four.

By using these frequencies distributions, another user classification is possible:

1. Personal pages. These pages have been authored by the user typically for their own use. They include personal information and links to sites of their own interest.
2. Start-up page. Users set their own personal pages often as start-up pages. Administrators set start-up pages to support local groups of interest.
3. Organisation/other user's home page. This is the main page for an organisation or individual making information available via the web.
4. Index pages. Contain links pertaining a particular topic.
5. Search engines. Such as Lycos, Altavista, ...
6. Web applications. Some pages are close to web applications, like knowledge elicitation tools, a print queue query, corporate phone book query, dictionaries, etc.
7. Authored page. A page revisited as it is being developed.

A l*ocality* set is a set of interconnected pages that is visited successively. Consider for example a set of 4 pages that is accessed for 10 times, without accessing other pages. The set then has size 4 with duration 10. The number of observed locality sets of length *n* can then be plotted against *n*.

4.3. MODELLING METHODOLOGY

The techniques presented in Section 3 have proven to be useful in modelling traffic at the resource level, i.e. in terms of (flow of) packets. However, many modelling studies require the construction of source models at a higher level, namely a user-behaviour oriented view. For example, an ISP is interested in the effects on network utilisation considering an increase in the user population or a different mix of user classes.

To answer such questions, it is on the one hand necessary to have mechanisms to describe user behaviour. Seminal work in this direction has been summarised in sections 4.1 and 4.2. But there is a gap on how to map the user behaviour to the resource oriented description of traffic neede to drive network simulations. Some efforts have been proposed in literature proposing hierarchical models to bridge this gap. This is a natural solution, as also the system under study exhibits a hierarchical nature.

In BISANTE we also propose a methodology based on a hierarchical modelling concept, which should fulfill the following criteria:

- **Ease of Use**: The analyst should be able to use the traffic models developed within BISANTE without detailed knowledge on the statistical techniques incorporated into the models.
- **Flexibility**: The analyst must be able to construct the model at the desired level of detail. He or she must be able to either omit details if not necessary for the particular simulation study or extend the model to represent additional aspects of the system under study.
- **Re-usability**: A model developed for a particular case study should be re-usable in another case study without too much effort.
- **Complexity control:** Techniques must be available to simplify model construction and evaluation (possible at the costs of model accuracy).
- **Well-defined Interface**: The workload model must have a clear interface with the network simulation part.

Following these goals, we propose a modular, extensible framework for user-oriented traffic modelling as described in the following section.

4.4. OVERVIEW OF LAYERS

In this section, a layered approach for creating realistic network traffic will be described. Layers start from top to bottom and each layer adds another level of detail to the model. What to do next is passed from the highest layer down to the lowest by triggering actions (in the sense of streams of commands). If such an action is triggered for layer *i* by the next higher layer *i+1*, layer *i* starts triggering actions for the next lower layer *i-1* and waits for results from it. Results are then processed and, if necessary, passed on to layer *i+1*. The lowest layer in this stack generates the network traffic that is observed. If intermediate layers are missing, then they will be filled up by dummy layers.

The goal is to find sets of layer models that can be plugged onto each other to represent some typical user behaviour that has been either observed or that is projected to be seen in the future. The models will be created top down, starting always at the highest layer and then going down to the chosen level of detail. The more layers are added, the more detailed will be the traffic description and the better will be the result, leading also to more system events to more CPU time necessary to obtain the result. The lowest layer will generate network traffic, either at packet level or as network volume per time unit.

Many of the following models will consist of a finite (yet dynamically changing) set of states.

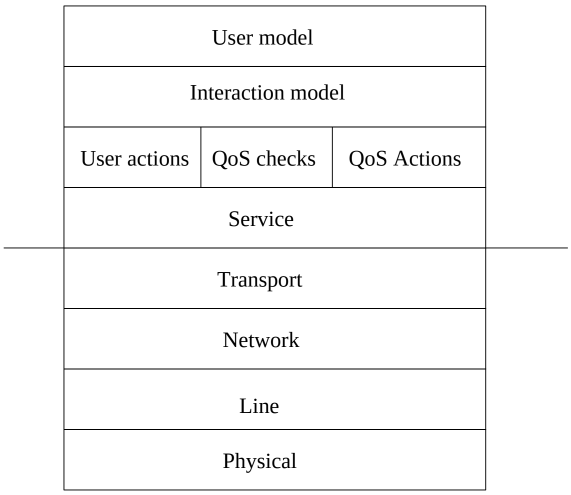


Figure 6: Model framework layers

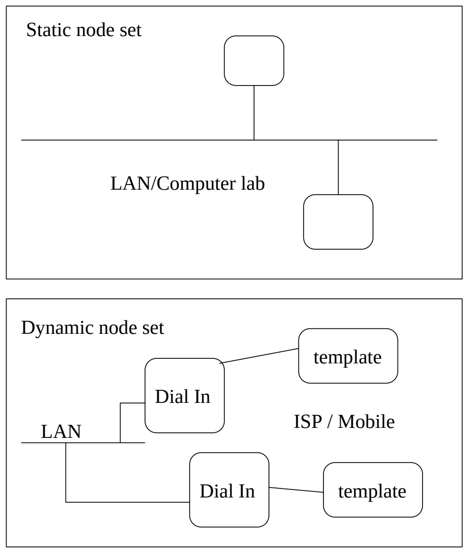


Figure 7: Static and dynamic network nodes.

As each layer can be used as the end layer, it must be able to generate network traffic or to pass actions down to the next model layer.

What to pass down of course depends on the modelled applications and services. In this sense, applications represent one or more services that the user can start, stop or modify.

4.4.1. Generating Events

Generating events of any kind, like terminal equipment, user model actions or user interactions is modelled by the interarrival time of such events and the type of the event itself. At each layer, thus a stream of events is produced. The stream of events can be influenced by results from lower levels (observed QoS), be stopped by itself (for instance, when deciding to close the application) or be started by higher layers. Lower layers may also stop higher layers (e.g. stop the application).

4.4.2. Terminal Equipment

Terminal equipment denotes the hardware that applications run on. They are also members of the network topology under observation. Terminal equipment is either created and connected to the network topology before simulation start, or are defined as templates. Template instances are then created at run time and are dynamically connected to the network topology at some predefined nodes (dial-in nodes). Terminal equipment can also move around and might be passed from one dial-in node to the other (handover).

4.4.3. USER MODELS

These models define the highest level of user behaviour. At a first stage it is assumed that for each created terminal equipment instance, one user is automatically created with it. User actions then start after some fixed or distributed time. User models have the following tasks:

- Start applications.
- Choose the application to use.

Note that it is only the application to use, not the service that is chosen. We denote that the number of possible states of the user may vary, as in principle, many applications can be started and stopped again.

Once, an application is chosen, the next lower layer starts generating actions or network traffic. Starting and choosing applications can either be done synchronous or asynchronous to these actions.

- Synchronous: Whenever the next lower layer decides to do the next user interaction, this event is passed up one level and the user model decides, whether this application is still chosen, or another application is chosen or started. If another application is chosen, the next user interaction is performed for the new application and the time for the next user interaction is calculated.
- Asynchronous: The times for starting/choosing applications are chosen from the user model. Whenever the user model decides to change the application, the next user interaction is performed and the time of the next user interaction is calculated.

User models are plugged directly on top of an interaction model. Note that user models can be very simple dummy models that just start an application (which in turn just starts a simple service). This way, each TE/user can create network traffic with very small effort at a low level of detail.

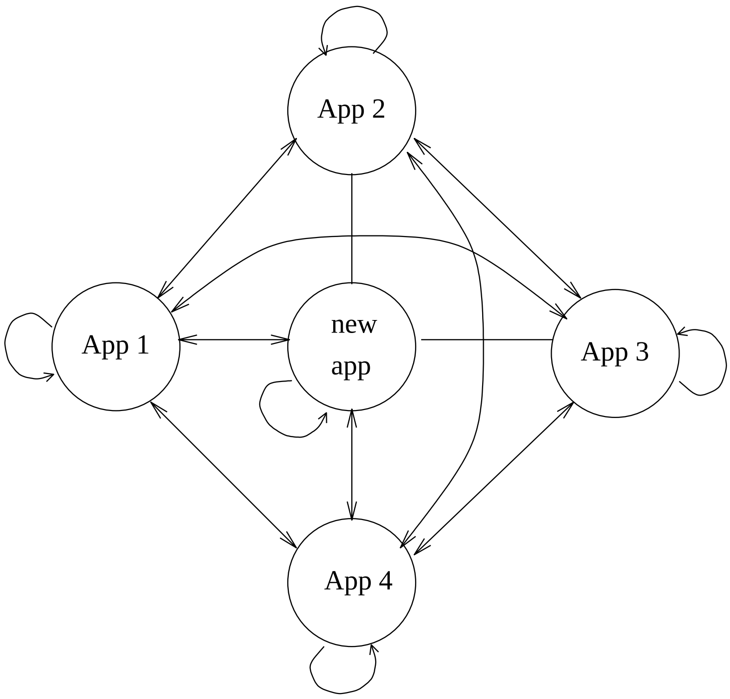


Figure 8: Applications on a TE.

This next lower layer will start generating commands for the application. If the user model is on top of a traffic generator, this next lower level will start generating traffic connections, packets or volumes of traffic.

4.4.4. INTERACTION MODELS

Interaction models define, how the user interacts with the chosen application. Basically, the user can perform one of the following interactions:

- Start a service
- Choose a service
- Stop the application

If the application is stopped, the user model must start another application or choose another one. The generation of user interactions can again be done synchronously or asynchronously to the user actions at the next lower layer. As long as a service is running, it delivers QoS results. Services can stop themselves after delivering the result (for example HTTP).

If a service is started, it generates network requests (packets, volumes) either directly, or by sending network requests down to the next lower layer. The observed QoS is part of the service and can be obtained by the next higher level.

Interaction models are plugged on top of user actions and QoS checks/actions. Note that interaction models can be very simple dummies, having nothing more to do than starting a service.

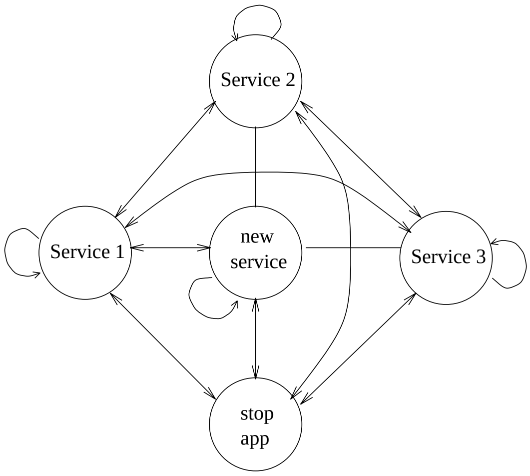


Figure 9: Services available in a running application

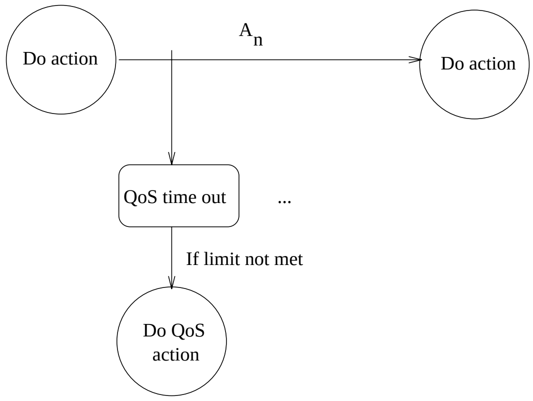


Figure 10: Service commands and QoS actions.

4.4.5. USER ACTIONS

If a service is chosen by the interaction model, the according user action model starts generating streams of commands for it.

Amongst the possible commands are:

- Change service parameters.
- Stop the service.

If a service is stopped, the interaction model must take over again and generate the next application interaction.

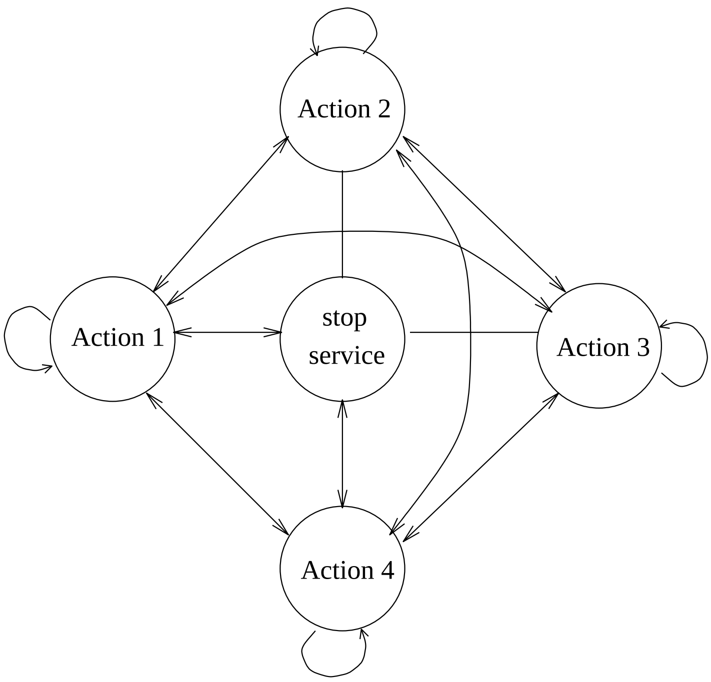


Figure 11: User actions.

User actions can be simple dummy models, which either do nothing or just stop the service after some time.

4.4.6. QoS Checks and Actions

Running services result in QoS descriptions. For each QoS description, a QoS level can be defined. QoS checks compare the observed QoS level to the requested QoS level. If the check fails, an appropriate QoS action is triggered and performed. QoS actions and user actions are at the same level.

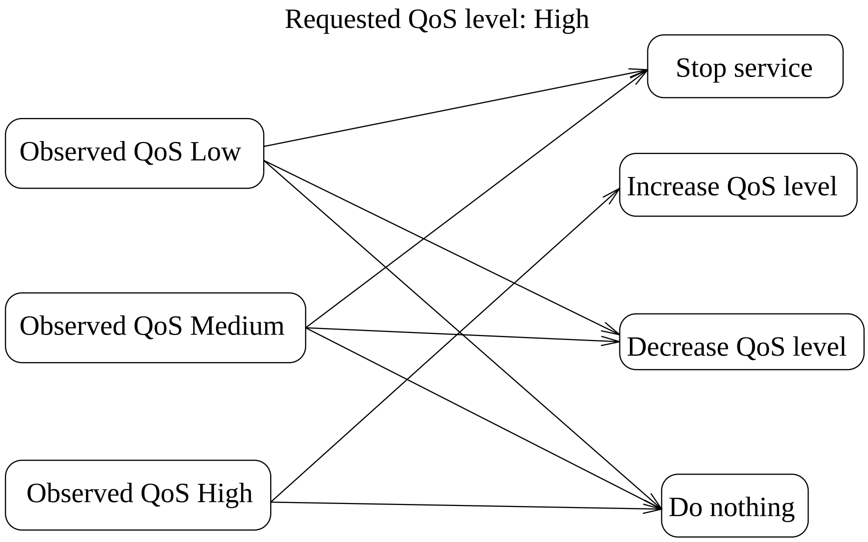


Figure 12: QoS checks.

QoS checks can be performed as single or periodic checks:

- Single: The check is performed only once, after a time out occurred. If the service has not been completed, the appropriate QoS action is triggered (for example the requested web document has not been downloaded)
- Periodic. The check is performed periodically, until the service stops.

4.4.7. Service models

Service models consist of parameterised traffic generators. Some parameters can be changed by higher level models and influence the QoS level that the higher level model chooses. Other parameters are fixed and describe a situation that can not be changed by users, like the distribution of file sizes at a web site.

Traffic requests can either be passed directly to the network or down to the next lower layer.

Services can stop themselves, it they have performed their task (like delivered the web document, played the video). Services can also be very simple dummy services, which just create low detail traffic on the network by using simple fluid models or renewal models like Poisson arrivals.

4.4.8. Lower Layers

Lower layers consist of

- Network: Link or physical layer, if traffic is generated directly from service layer
- TCP/IP: transport layer, network layer, link layer, physical layer
- ATM: AAL5

These lower layers provided either connections to other applications or are just entry points for the generated network traffic.

4.4.9. TYPES OF MODELS

The previous sections describe a framework of model classes and how to glue these model classes together. Yet it says nothing about the type of models to use. Its strongest suggestion is that each model is always in one well defined state, where the number of possible states may vary dynamically.

Modelling will be divided into two parts:

1. Modelling user and application behaviour: this part covers the top three levels of the approach. At each level, knowledge on user behaviour will be encapsulated into a model, which is mainly reponsible for triggering actions at the next lower layer and reporting status information to the higher levels. This mechanism can be driven by a very simple state-diagram, but more sophisticated techniques as discussed in 4.1 and 4.2 can be applied. Depending on the objective of the evaluation study, it might even be reasonable to "skip" higher layers, i.e. replacing them by dummy layers simply passing on the signals. For example, at the top level a dummy user model may simply start one dummy application, activating one service in this application.
2. Modelling service traffic: This model at the lowest layer will generate the actual network traffic. In principle, any type of model discussed in section three may be chosen for a particular type of traffic. The parameters of the model may be determined from the user behaviour models at the higher layers.

The next steps in work package one are to demonstrate this approach on selected examples and to identify a set of models which will be the basis for the knowledge base on user behaviour in the BISANTE modelling suite. In deliverable DEL12 – Knowledge Based User Behaviour Modelling – we will show how information on user behaviour can be incorporated into the proposed approach. Finally, DEL13 – Generation of User Profiles – will discuss how to encapsulate all information into re-usable modules, called user behaviour profiles.

5. SOLUTION TECHNIQUES

An overview of performance modelling approaches for parallel systems and communication networks is given in this part. Such approaches are useful to understand the behaviour of ATM switches, multimedia servers and Web servers. The following issues are considered:

- performance criteria
- performance models
- performance methods such as approximate mathematical solutions, simulations, measurements, aggregation methods and learning techniques.

Communication and computing systems are evolving very fast. Their growth, their economical importance, their need for competitiveness, lead to the need for dimensioning and for validation of their operation rules.

There are historical examples of new systems that were built too fast, their performance were not evaluated, and this led to a disaster. Let us mention the VASA. It was a war ship. The Swedish king wanted it to be built very fast. He wanted to impress enemies with a huge ship with two rows of cannons. The ship designers had noticed a bad balance but they did not dare to tell to the king that it was necessary to run tests. They were not able to compute stability models. The boat started and sank in the harbour.

Same story may happen for new distributed systems or communication networks.

When is it necessary to evaluate performance of parallel systems and communication networks ?

When a system does not exist and when designing it, it is necessary to dimension it and to make sure that it will operate properly For example, ATM (Asynchronous Transfer Mode) communication networks were designed so as to carry on the same support several traffics: voice, data and video images. In order to spare time, it is intended to use the silences of some traffics in order to carry the other traffics. These networks are asynchronous because in a frame, the position of the packets of each traffic is not reserved. It is easy to guess that this asynchronism may lead to an eventual gain but that congestion may be important. So performance studies were and still are essential before designing and dimensioning ATM networks. ATM packets are called cells.

Nowadays, ATM is standardised. On progress studies about ATM concern IP over ATM and ATM over satellite.

When a system exists, in order to improve its operation, it is necessary to study its performance.

Web servers exist and Internet principles consist in allowing everybody to develop new applications and to load the network. Performance may be very bad. The users will be regulating themselves, according to the performance they get. Nevertheless after having spent time and money in the design of a server, the manager will want to make sure that the server is operating properly.

Let us note that we are interested in general performance results. We are not much interested in traces, in performance results on ‘‘that” day and when ‘‘that” user is sending requests to the system. We are more interested in performance criteria which are **probabilities or expected values**. But probabilities or expected values are often estimated by time averages, when ergodic theorem stands. A key point is then the evaluation of the precision of the estimation and of the **confidence interval**. For example the probability that a resource is busy is estimated by the average of the busy time.

When this average is computed over a long enough period, the confidence interval of the average will be good enough.

5.1. PERFORMANCE EVALUATION CRITERIA

Let us define a few performance criteria that are interesting for computer and communication systems. Let us note that we are interested in traffic problems. Sometimes, programs are running on dedicated computers. Then, waiting times are not so usual and in order to get an estimation of performance, it is enough to count the number of operations in a code. This is wrong when dealing with load dependent performance and this is always the case when several users are allowed to share a resource.

Response time : *Response time is the expected value of the time between the arrival of a request and its completion.*

This time may include waiting times and service or computing times. When dealing with communication networks, response time will be the expected value of the time between arrival of a packet and time when the packet is successfully transmitted. It is called transmission delay. Its variations may be important. In ATM networks jitter is estimated from cell delay variation (CDV). When dealing with reading requests on a disk, response time will include waiting, searching for the data, extractions, transfer ...

Throughput : *Throughput is the expected value of the number of requests completed during a time unit.*

It may be in a communication network the number of packets transmitted in a time unit, it may be the number of data read on a disk.

Utilisation of a resource : *Utilisation of a resource is the probability that the resource is busy.*

When a resource is expensive this criteria is important.

Loss rate : *Loss rate is the probability that a packet is lost.*

There are many error rates that may be interesting.

Blocking rate : Blocking rate is the probability that no path is available, when trying to reserve a path, or probability that a packet cannot be sent in a finite capacity network.

5.2. PERFORMANCE EVALUATION MODELS

Need for a model : In order to evaluate the performance criteria, it is necessary to model the system:

- When the system does not exist, it is necessary to design a model.
- When the system exists, if we want to estimate performance criteria for many values of the operating parameters, without disrupting the system, it is better to use a model.
- When the solution of the model is not very simple, in order to solve it we shall have to use an approximate model that is more simple.

Queuing models : In order to deal with traffic problems, we mostly use queuing network models.

A **queue** is a mathematical object, in which there are customers and servers. Customers are receiving a service, but they may have to wait in case the server is busy giving a service to other customers.

Modelling strategy : The first problem is to define what is a **customer**. There generally are different tiers for the study, corresponding to different time scales and to different performance criteria.

For example, in ATM networks, there are different levels corresponding to different time scales. A call can be considered as a set of bursts. A burst is a set of cells (see Figure 13 : The 3 different levels).

When studying a switch at the cell level, an event is happening during one slot. During a call, there will be a huge number of cells, so, when studying a switch at the call level, the times that will be considered will be very large, when compared with those of a cell level study. It is not possible to consider at the same time two different levels of study, such as call level and cell level [KUH94], because an event at the call level implies too many events at the cell level.

The different levels will lead to different types of conclusions: While a cell level study will help dimensioning the switch, a burst level study will lead to source policing or congestion avoidance and a call level study will lead to Call Admission Control (CAC) and network planning.

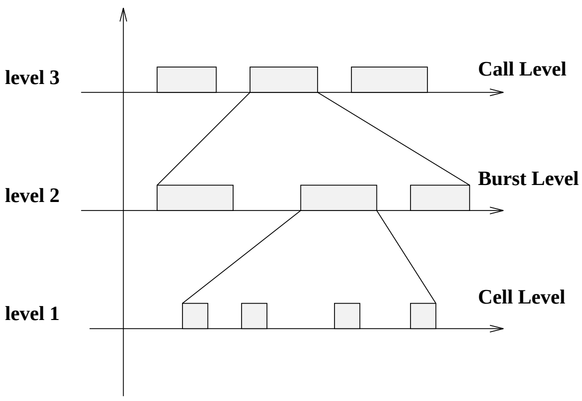


Figure 13 : The 3 different levels in ATM networks

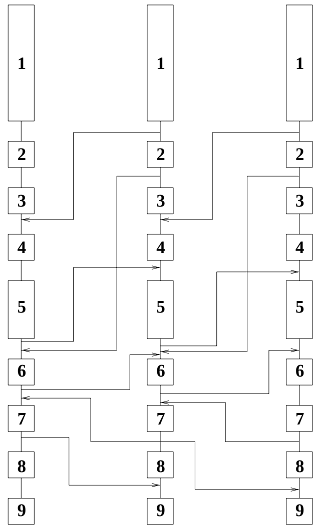


Figure 14 : A Task Graph

When the granularity of a customer is defined, in order to characterise the queue operation, it is necessary to define the arrival law, the service law, the number of servers, the scheduling rule, the capacity of the queue ...

When modelling a Web server, a queue may be a disk, the customers may be requests for a file transfer from the disk, the service times may be the transfer time, the scheduling rule may be FIFO.

Usually we try to get a model of the **architecture** and a model of the circulation of the customers. In fact for the Web server model we shall have a **task graph** [BEC86] and durations for each task and precedence relations between tasks (cf.
Figure 14 : A Task Graph).

These precedence relations will correspond to the user surfing. From the task graph analysis will be derived the service times and the transition probabilities for the customer movements.

Then the first important study that has to be done is: what is the **bottleneck** ? For example if the network interface between the server site and the outside Internet network is very low speed (it may be a 64 Kbits/s link) modelling the Web server is not very useful, the only performance study that has to be performed concerns the transfers on this link.

Markov chain models :

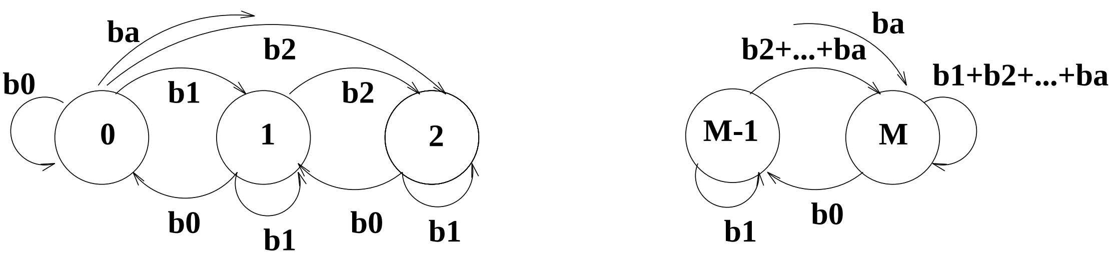


Figure 15 : A Markov chain

In order to solve the models, it is useful to extract memoryless processes and to describe the system with a Markov chain. The system has to be described well enough so that the description completely summarises all the pertinent past history as far as that history affects the future of the system. The eventual states of the system and the transitions between states will then be defined. The performance criteria will be computed, if the number of states in the Markov chain is not too large.

Figure 15 : A Markov chain presents a Markov chain modelling of a switching element with finite capacity [BEY91].

In case of long term correlations [LEL93], which occurs for example when there is a day or a week periodicity, an exact Markov model is not valid.

An approximate Markov model may nevertheless be used when the level of the study leads to studying the system during time periods that are short when compared with the correlation period.

Queuing network models : Figure 16 : Queuing network for a VOD server presents a queuing network that is modelling a VOD server [KAD98]. It is a Video On Demand Server where the performance strongly depends on the data accesses, on the way of sharing data into blocks, of placing data on disks and of prefetching data blocks in order to avoid breaks in the block transmission.

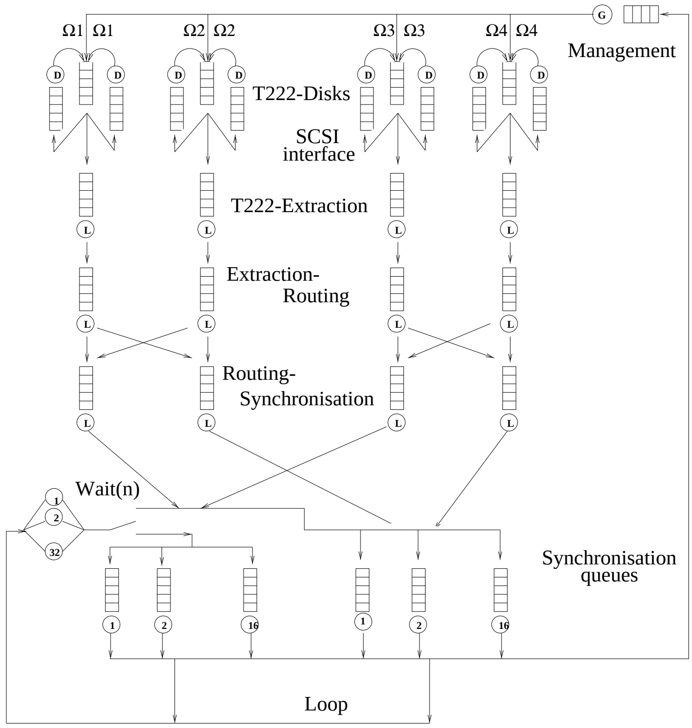


Figure 16 : Queuing network for a VOD server

Petri net models : In order to model synchronisations, Stochastic Petri net models are useful. Transitions are happening after a random time. Token transitions are random: there are branching probabilities. The imbedded Markov chain has to be solved, but it often is huge. Figure 17 : A pipe-lined multiplier model[BEC90] presents a Petri net model of a FPS multiplier.

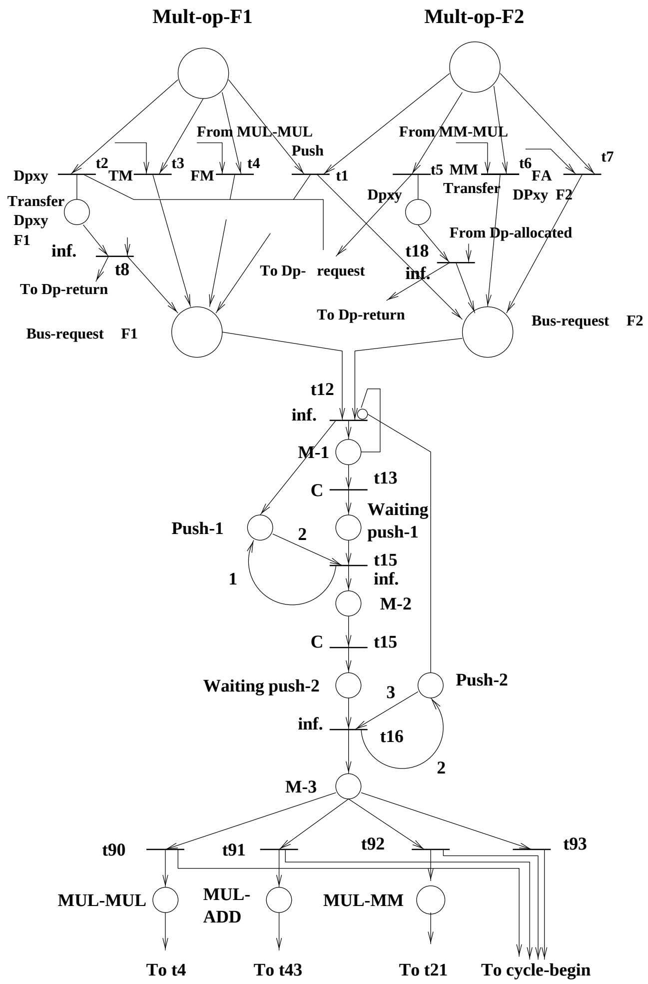


Figure 17 : A pipe-lined multiplier model

We sometimes use mixed models: they are queues with synchronisations, so they are mixed models between queues and Petri nets [HOU88].

Heuristic models : When modelling user behaviour or application profiles, there is not always an obvious mathematical model. An heuristic method may be used in order to learn what are adequate parameters of a model. **Neural net** models (cf. Figure 18 : A Neural Net) or **genetic algorithm** models may be used.

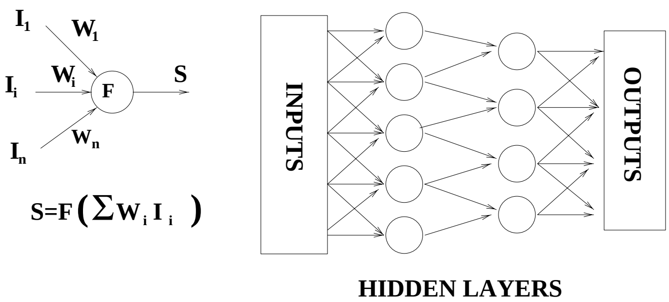


Figure 18 : A Neural Net

5.3. Performance Evaluation Methods

Mathematical solutions : When they are tractable, mathematical solutions allow an optimisation of performance criteria.

We mostly use solution of linear system in order to derive performance criteria from the solution of a Markov chain. When linear systems are too large we have to derive approximate solutions. They have to be validated by simulations or measurements. We use a tool whose first name was QNAP [POT84] and latest versions are called MODLINE.

Simulations : A simulation is an experiment. This technique lets us evaluate the performance of the model running on a computer. The performance criteria is estimated by a time average that is near enough the probability average, if the simulation is long enough.

We get results for an experiment and for a set of parameters values. These results, if the confidence interval is carefully derived, will let us validate an approximate mathematical solution, for the given set of parameters. We shall then expect the approximate solution to be valid for other sets of parameters [PRO89]. In communication networks, performance studies are made difficult by the problem of **rare events** [BEC93]. For example in ATM networks, loss rates should not be more than 10^{-9}. When running a simulation, in order such a rare event to happen, it should be necessary to run the simulation during several centuries.

Measurements : They are necessary to determine the model parameters. We work with a group in National Institute for Standards and Technology (Gaithersburg) where measurements are performed [FOU96]. Nowadays measurements are an important issue, because traffics are evolving very fast and classical source models are not valid anymore. For example no good model exists for multimedia traffic.

Aggregation techniques : Before solving a large complex system, it may be easier to part it into subsystems. Aggregation techniques are methods with two steps:

- aggregation step: the solution of each subsystem is derived independently from the other subsystems, or if there are relations between subsystems, they are simplified.
- disaggregation step: the global solution is derived from the solutions of the previous step.

Aggregation may be exact or approximate. In case it is approximate, the two steps may be used iteratively until convergence. We often used these techniques. There are 3 different types of applications:

- physical aggregation: subsystems are physically existing. They may be the cells of a wireless network, the coverage of a satellite, or a unit of a computer [BEC90]. On Figure 19 : Physical Aggregation - GSM system, GSM system is an example of physical aggregation: base station controller coverage is an aggregate of base transceiver coverage and switching centre coverage is an aggregate of base station controller coverage.

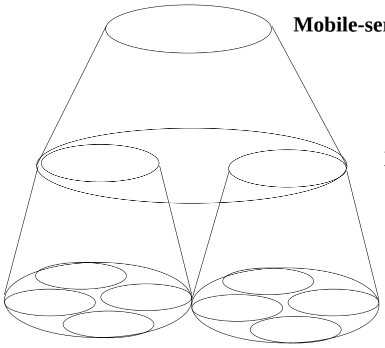


Figure 19 : Physical Aggregation - GSM system

- Markov chain aggregation: subsystems are subsets of the state space. [MOU98] presents an example of such an aggregation.
- time scale aggregation: subsystems correspond to different time scales. It is not possible to run simulations which compute the states of an ATM network at call level and at cell level simultaneously, because there are too many cells in a call. So, for example when evaluating the jitter, cell level studies are necessary and from the cell level studies may be derived information for a routing algorithm study at the call level. But when planning call reservations, if an operator wants to guarantee a QOS about jitter, aggregation methods will have to be used in order to evaluate jitter at call level.

The same technique will also have to be used for studies of Web servers where there are also different tiers.

6. SUMMARY OF THE RESULTS AND FUTURE WORKS

In this deliverable classifications of services and applications as well as a description of the application spaces are provided. They are needed in order to know what to model: which application or services and which type of user.

This deliverable provides a general presentation of the useful tools for modelling users and applications, as well as the presentation of the framework of our studies. It includes a large set of states of the art and pointers to research papers and standard organisations to help modellers in their researches of tools and techniques for modelling and studying performance. Other measurements with modelling are to be brought into focus, which will be done in the deliverable D1.2 and D1.3. Application models and user behaviour models will be integrated in the third deliverable of this work package D1.3, giving traffic profiles usable for network simulations.

Source and user behaviour modelling techniques are described. This description is followed by the proposal of a layered approach to user behaviour oriented network traffic modelling. A short presentation of the solution techniques, which includes methods and relevant characteristics for network performance studies, motivates the use of simulation.

A. SURVEY OF CODING STANDARDS

One interesting point in our modelling of applications is that we will be able to determine the data flows they generate on the network. As such, it is useful to include a survey of the existing coding standards, which generally have an impact on this output flows. Nowadays, the range of applications and distributed services is so wide that classifying all of them can be done in different ways. We are following a classification of services proposed by the ITU.

The growth of the use of multimedia applications has led to the birth of several coding standards for the transportation of audio and video files across a network. As each of these codes has its own formatting and consistency-checking policy, they have very different impacts on network traffic. As our goal is, among over things, to model the quantity of traffic generated by network applications, it should be useful as well to integrate in these models the impact of the use of particular codes.

Making an exhaustive survey of coding standards has never been truly explored, or so it seems, judging from the low number of documents on this subject, so readers should keep in mind that there may be omissions in the following classification.

Coding standards have been invented to answer to the needs of transmitting different kinds of data information, hence it should come at no surprise that these standards can be classified according to the kind of data they code.

A.1. TEXT VS. BINARY

Most of the files generated by applications and/or transmitted on a network are ever coded in ASCII or using binary code, so these two code types should remain the core of our simulation (but of course not the only ones).

Whenever someone refers to a file as a ‘’text file’’, he or she probably thinks of it as an ASCII file. ASCII code (American National Standard Code for Information Interchange) is effectively the most used text code nowadays. Using this code, every letter, number and punctuation symbol is coded on a 7 bits format. However, it is not complete as some symbols used mainly in programs (and not directly by users) are not coded in ASCII.

For this reason, files containing processor-specific information use a code called binary where characters are coded on 8 bits (a full byte), thus allowing a greater range of symbols to code. The full use of the 8 bits in a byte for coding useful information (as opposed to ASCII which leaves one bit unused) also means that this kind of coding is less resources-consuming, so for some kinds of text or data files (most graphics files, compressed files, ...), this code is definitely superior to ASCII.

There exists other codes (such as Unicode used in Windows 95), but they are not usually transmitted as such on networks (for example, FTP supports transfer for binary and ASCII files, but that is the end of it), so it limits their consequence in our model.

A.2. VIDEO CODES

Analog video can be represented in digital form. This enables us to process and transmit information using digital computers, processors and networks. However, the very high bit rate of digital video signals means that the information must be compressed before it can be processed and transmitted.

Many different techniques for compressing digital video have been developed in recent years. All of these exploit the spatial and temporal redundancy of a video sequence in order to achieve compression. Several international standards have been issued which provide a mean of encoding still and moving video information :

- The **JPEG standard** is designed to code still images
- The **H261 standard** is part of the **H320** videoconferencing standard and is optimised for two-way videoconferencing
- The **MPEG-1** and **MPEG-2** standards have been developed to support a wide range of digital video applications
- The **CCIR 601** standard provides a standard method of encoding television information in digital form.

Let's have a look at each specific code. Note that except for still image coding (such as JPEG), video coding standards do not only code video, but attached sound as well. For these standards, although they are generally referred to as *video codes,* the term *multimedia codes* is more accurate.

A.2.1. STILL VIDEO : THE JPEG STANDARD

ISO international standard 10918, ''Digital Compression and coding of continuous-tone still images'', is commonly known as the JPEG standard (for Joint Photographic Experts Group). The JPEG standard consists of two parts.

Part 1, ''Requirements and Guidelines'', describes the features of the standard and gives details of the coded bit stream. It defines four coding modes :

- **Sequential encoding** : each component is encoded in a single scan with the scanning order left-to-right and top-to-bottom. The encoding process is based on the DCT (Discrete Cosine Transform).
- **Progressive encoding** : each scan contains a partially coded version of the image. A ''rough'' image is quickly decoded and then this is built up using further scans.
- **Hierarchical coding** : each component is encoded at multiple resolutions. The image can be decoded at a low resolution without decompressing the full-resolution image.
- **Loss-less encoding** : it is based on a differential prediction system. This mode provides compression without any loss of quality, at the expense of a considerable reduction in compression efficiency.

The standard defines a baseline **CODEC** (encoder/decoder) which implements a minimum set of features :

- **The baseline CODE : DCT**
  - image is processed in 8x8 blocks
  - DCT is used to produce an 8x8 block of transform coefficients
  - algorithm used to perform the DCT is not specified by the JPEG standard
- **Quantisation :**
  - each DCT coefficient is quantised to reduce its precision

- the coefficients are divided by an integer (the quantiser step size)
- the result is rounded to the nearest integer
- the step size can be different for different coefficients in the block
- step size must be chosen so as to give a suitable level of compression

- **Zigzag ordering :**
  - the quantified coefficients are re-ordered in a zigzag scanning order, starting with the DC coefficient and ending with the highest-frequency AC coefficient.
  - this creates long runs of zero-valued coefficients
- **Encoding :**
  - the DC coefficient is encoded separately from the AC coefficients
  - each DC coefficient is encoded differentially from the DC coefficient in the previous block since there is usually a strong similarity between DC coefficients in adjacent blocks
  - in a given block, most of the AC coefficients are likely to be zero and these are converted into sets of run-length/value-symbols - this is a figure for the number of zero coefficients, followed by the next non-zero coefficient
  - the set of symbols are then encoded using Huffman coding
  - the baseline decoder carries out the reverse procedure to get back to a representation of the original image. The quality of the decoded image depends on the quantisation step chosen and the compression efficiency depends on the image content.

Part 2, ‘’Compliance Testing ‘’, specifies test procedures which can be used to verify whether a particular implementation conforms to the standard. Using an encoder/decoder based on the JPEG standard, it is possible to compress images by between 10 and 20 times without seriously reducing the quality of the decoded image.

A.2.2. CCIR 601

The CCIR 601 recommendation provides a method to code television information in digital form at a specified video rate of 167 Mbps. This means that the 4.7 Gigabytes capacity of a DVD can only contain roughly 4 minutes of video. Obviously this method is to be used in conjunction with a compression method, such as those described later.

A.2.3. MOVING VIDEO : THE MPEG STANDARDS

Video coding for broadcast applications is dealt with in the MPEG (Motion Picture Experts Group, a group of experts belonging to the ISO/IEC international organisation) standards.

- MPEG-1 is optimised for coding of video and associated audio for digital storage media such as CD-ROM.
- MPEG-2 enhances the techniques of MPEG-1 to support video coding for a range of video applications including broadcast digital television and High Definition Television (HDTV).
- MPEG-4, which is a brand new version released last year in version 1 (version 2 is due for middle-1999), is also shortly described

Readers should note that our goal is not to make an exhaustive presentation of the standard, but rather a summary of its features. Interested readers wanting more in-deep information should have a look at the MPEG's home page on http://cselt.it/mpeg, where all standards are detailed.

A.2.4. MPEG-1

MPEG-1 standard supports coding of video and associated audio at a bit rate of about 1.5 Mbps. MPEG-1 standard specifies the syntax of the coded bit stream and also describes a model of decoder.

Frames of video are coded as pictures with each frame being encoded in a progressive order. There are three main types of coded picture in MPEG-1 :

- **I-pictures** (intra-pictures) : are intra-frame encoded so do not use prediction. Blocks of pixels are transformed using the DCT, quantised, zigzag reordered and variable-length encoded. Coded blocks are grouped together in **macro-blocks**.

- **P-pictures** (forward predicted pictures) : are inter-frame encoded using motion prediction from the previous I or P-picture in the sequence. The difference is that the macro-block, together with the motion vector, is encoded and transmitted.
- **B-pictures** (bi-directionally predicted pictures) : are inter-frame encoded using interpolated motion prediction between the previous I or P-picture and the next I or P-picture in the sequence. B-pictures are not used as a reference for further predicted pictures.

I-pictures have the worst compression efficiency because they use only intra-frame compression. B-pictures have the best compression efficiency due to bi-directional motion estimation.

The three types of pictures are grouped together in GOPs (Groups Of Pictures), each one consisting of one I-picture followed by a number of P and B-pictures.

The compression achieved by MPEG-1 encoding is affected by several parameters. The most important of these is the quantiser scale factor. A high scale factor leads to high compression and poor quality, and vice versa.

A.2.5. MPEG-2

Television quality video information produces a higher bit rate which requires different techniques from those provided by MPEG-1. The MPEG-2 standard comes in three main parts : Systems, Video and Audio. MPEG-2 extends the functions provided by MPEG-1 to enable efficient coding of video and associated audio at a wide range of resolutions and bit rates.

The extra features include :

- features to support interlaced video as well as progressive (i.e. non-interlaced) video
- more chrominance sampling modes
- an alternative block scanning pattern to the zigzag scan
- four scalable coding modes :
    - Spatial Scalability : where each frame is encoded at a range of resolutions which can be built up to the full resolution

- Data Partitioning : where each block of 64 DCT coefficients is split into two bit streams. The first contains the low frequency coefficients and the second contains the high frequency coefficients
- Signal to Noise Ratio Scalability : where the picture is encoded in two layers, the lowest of which contains a coarse version of the picture
- Temporal Scalability : where the base layer is encoded at a lower frame rate

MPEG-2 describes a range of profiles and levels which provide encoding parameters for a range of applications. A profile specifies a particular set of coding features. Within each profile, one or more level specifies a subset of spatial and temporal resolutions which can be handled.

The profiles defined in the standard are as follows :

- Profile 1 : 4:2:0 sampling, I/P pictures only, no scalable coding
- Profile 2 : As above, plus B pictures
- Profile 3 : As above, plus SNR scalability
- Profile 4 : As above, plus spatial scalability
- Profile 5 : As above, plus 4:2:2 sampling

Each level puts an upper limit on the spatial and temporal resolution of the sequence as follows :

- Low Level : 352 x 288 luminance samples, 30 Hz
- Main Level : 720 x 576 luminance samples, 30 Hz
- High 1440 Level : 1440 x 1152 luminance samples, 60 Hz
- High 1920 Level : 1920 x 1152 luminance samples, 60 Hz

Only a limited number of profile/level combinations are recommended in the standard. These are summarised in the following table :

	Low	Main	High 1920	High 1440
Profile 1		X		
Profile 2	X	X	X	X
Profile 3	X	X		
Profile 4			X	
Profile 5		X	X	X

Particular profile/level combinations are designed to support particular classes of applications.

- Simple profile/main level :
    - suitable for conferencing
    - no B-pictures transmitted
    - low encoding and decoding delay
- Main profile/main level :
    - suitable for most digital television applications
    - majority of MPEG-2 encoders and decoders currently available support this coding

- The two high levels support HDTV applications.

It should be emphasised that the profiles and levels are only recommendations. Other combinations of coding parameters are possible.

A.2.6. FURTHER EVOLUTIONS IN MPEG FORMAT

So far, the MPEG standard has been a great success, and MPEG 1-2 are codes widely used in multimedia applications. This has prompted the expert group that birthed this concept to refine and expand the range of functionalities offered. This is why the MPEG-4 version is about to appear.

MPEG-4 is an ambitious project to develop a standard for interactive networked multimedia incorporating both natural (pixel/sample based) audio and video and synthetic (2D and 3D animation based on polygons, splines, etc.) audio and video.

MPEG-4's stated goals include:

- <u>Content-based Interactivity :</u>
    - Content-based manipulation and bit-stream editing : MPEG-1 and MPEG-2 were standards focused toward passive viewers. Games, editing tools, and other applications require ability to interact with objects within the audio/video stream such as background images, speakers, 3D sprites, etc.
    - Hybrid Natural and Synthetic Data Coding : Support for combining natural (sample/pixel based) audio and video "objects" with synthetic objects such as computer generated 3D objects.
    - Improved Temporal Random Access
- <u>Compression</u> :
    - Improved Coding Efficiency : Improved audio and video compression compared to previous international standards such as MPEG-1, MPEG-2, H.261, and H.263.
    - Coding of Multiple Concurrent Data Streams
- <u>Universal Access</u> :
    - Robustness in Error-Prone Environments : An Error-Prone Environment is a wireless (cellular telephone) communication channel or a packet switched network with packet loss.
    - Content-based scalability
- <u>Flexibility and Extensibility</u> :
    - Flexible/Programmable Behaviour
    - Downloadable Components/Decoders

MPEG-4 is a large project. One part of MPEG-4 is a new MPEG-4 video compression standard for natural (pixel/sample based) video. There are several other pieces such as the MSDL (MPEG-4 Systems/Syntactic Description Language) and SNHC (Synthetic Natural Hybrid Video Coding).

A.2.7. H260 FAMILY : H.261

The video coding component of a videoconferencing system is described in Recommendation H.261. The H.261 algorithm supports videoconferencing and video-telephony over Integrated Services Digital Networks. The user data rate is generally a multiple of 64Kbps (p * 64 Kbps, with p $\in$ [1 .. 30]).

- H.261 is designed for two-ways video communications over ISDN
- only two image resolutions are supported : Common Intermediate Format (CIF) and Quarter CIF (QCIF)
- in both cases, the chrominance components have half the horizontal and vertical resolution of the luminance components
- maximum picture rate is 29.97 frames per second - this can be reduced if required by dropping up to three frames between each pair of encoded and transmitted frames
- over a 64 or 128 Kbps ISDN connection, substantial data compression is required. QCIF is normally chosen in this case and the source frame rate is restricted to around 10 frames per second

The H.261 coding algorithm is based on DCT image compression and motion-compensated inter-frame prediction. The first frame in a H.261 sequence has no prediction reference since there is no previous frame to predict from.

Each frame is processed a **macro-block** at a time, where a macro-block consists of 6 blocs of 8x8 pixels : four luminance blocks and two chrominance blocks. Each macro-block is encoded using **intra-mode** encoding (i.e. DCT-based coding with no motion compensated prediction) or **inter-mode** encoding (motion compensated prediction + DCT coding).

H.261 is typically used to send data over a constant bit rate channel, such as ISDN (e.g. 128 Kbps). The encoder output bit rate varies depending on amount of movement in the scene. A **rate control** mechanism is required to map this varying bit rate onto the constant bit rate channel.

- the encoded bit stream is buffered and the buffer is emptied at the constant bit rate of the channel
- an increase in scene activity will result in the buffer filling up
- the quantisation step size in the encoder is increased which increases the compression factor and reduces the output bit rate
- if the buffer starts to empty, then the quantisation step size is reduced which reduces compression and increases the output bit rate
- the compression, and the quality, can vary considerably depending on the amount of motion in the scene : relatively ‘’static’’ scenes lead to low compression and high quality, whereas ‘’active’’ scenes lead to high compression and lower quality

ITU-T Recommendation H.261 is part of the H.320 group of standards. Recommendation H.320 describes the various components of a videoconferencing system. For more information about videoconferencing in general, and the H.320 Recommendation in particular, readers should refer themselves to the corresponding section.

A.2.8. H260 FAMILY : H.263

H.263 is a recent ITU-T standard and was designed for low bitrate communication, early drafts specified data rates less than 64 Kbits/s, however this limitation has now been removed. It is expected that the standard will be used for a wide range of bit rates, not just low bit rate applications. It is expected that H.263 will replace H.261 in many applications.

The coding algorithm of H.263 is similar to that used by H.261, however with some improvements and changes to improve performance and error recovery. The differences between the H.261 and H.263 coding algorithms are listed below. Half pixel precision is used for motion compensation whereas H.261 used full pixel precision and a loop filter. Some parts of the hierarchical structure of the data stream are now optional, so the codec can be configured for a lower data rate or better error recovery. There are now four optional negotiable options included to improve performance: Unrestricted Motion Vectors, Syntax-based arithmetic coding, Advance prediction, and forward and backward frame prediction similar to MPEG called P-B frames.

H.263 supports five resolutions. In addition to QCIF and CIF that were supported by H.261 there is SQCIF, 4CIF, and 16CIF. SQCIF is approximately half the resolution of QCIF. 4CIF and 16CIF are 4 and 16 times the resolution of CIF respectively. The support of 4CIF and 16CIF means the codec could then compete with other higher bit rate video coding standards such as the MPEG standards. The following table summarises the characteristics of the different picture formats.

Picture Format	Luminance Pixels	Luminance Lines	H.261 support	H.263 support	Uncompressed bit-rate (Mbit/s)			
					10 frames/s		30 frames/s	
					Grey	Color	Grey	Color
SQCIF	128	96		Yes	1.0	1.5	3.0	4.4
QCIF	176	144	Yes	Yes	2.0	3.0	6.1	9.1
CIF	352	288	Optional	Optional	8.1	12.2	24.3	36.5
4CIF	704	576		Optional	32.4	48.7	97.3	146.0
16CIF	1408	1152		Optional	129.8	194.6	389.3	583.9

A.3. AUDIO CODES

As for the coding of video, coding of audio files has generated a number of different methods with quite varied results. As always, we will concentrate on those which are used with the greatest frequency today (or that look promising enough to be in such a situation in the future).

A.3.1. THE MPEG AUDIO STANDARD

We have seen in the previous chapter that MPEG compression format codes video data as well as attached audio data. This is done using the MPEG Audio standard. This method is based on the PASC (Precision Adaptive Subband Coding) compression algorithm, based on the modelling of the human ear, thus allowing compression rates of at least 4, most often 6 and even more.

This norm is applied on high-quality audio files, with possible frequencies of 32 KHz, 44.1 KHz and 48 KHz for MPEG-1, and 24 KHz, 22.05 KHz and 16 KHz for MPEG-2. As said above, compression rates reach 6 without difficulty, which allows a stereo coding at 256 Kbps instead of 1.5 Mbps without loss of quality.

A.3.2. THE G.700S RECOMMENDATIONS

As for video coding, the ITU-T (ex-CCITT) has issued a number of recommendations concerning audio coding. They are all part of the G serial, and consist of the following ones :

- G.704 : synchronous frame structures used at 1544, 6312, 2048, 8488 and 44 736 Kbps hierarchical levels
- G.711 : pulse code modulation (PCM) of voice frequencies
- G.722 : 7 KHz audio-coding within 64 Kbps
- G.723 : dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbps
- G.726 : 40, 32, 24, 16 Kbps Adaptive Differential Pulse Code Modulation (ADPCM)
- G.727 : 5-, 4-, 3- and 2-bits sample embedded Adaptive Differential Pulse Code Modulation
- G.728 : coding of speech at 16 Kbps using low-delay code excited linear prediction
- G.729 : coding of speech at 8 Kbps using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)
- G.764 : voice packetisation - packetised voice protocols

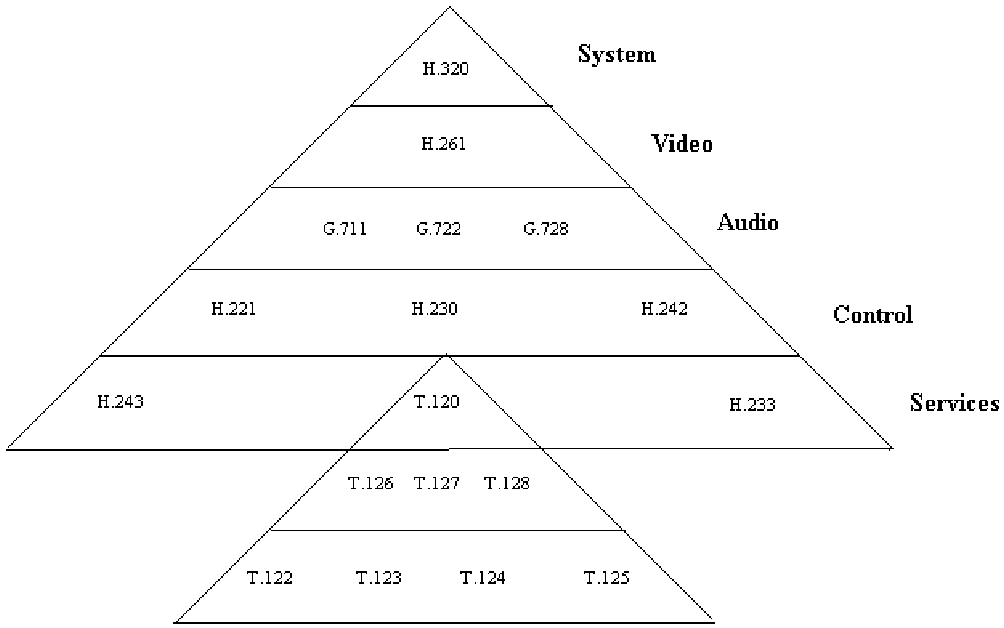
Some of these recommendations will be met again in the next chapter, where we will study the coding for videoconferencing.

A.4. VIDEOCONFERENCING CODING STANDARDS

Videoconferencing, with its mixing of audio and video data to be transmitted in real-time, poses quite a challenge in terms of compression. The problem is not without solutions however, as the ITU-T (International Telecommunications Union) has defined a number of recommendations, that have lead to the creation of the H.320 group of standards.

A.4.1. THE H.320 FAMILY

The recommendations of this family can be divided into 5 subgroups: system, audio, video, control, and services (with the T.120 protocols) with the following hierarchy :



We have already seen how the H.261 Video level works in the previous section, now let's concentrate on the remainder.

The Audio level comprises three recommendations of coding :

- G.711 for PCM modulation, with a ratio of 64 Kbps at 8 KHz
- G.722 at 7 KHz for up to 64 Kbps
- G.728 at 3 KHz for an output of 16 Kbps

The Control level comprises :

- H.221 : description of frames, and audio/video mixing
- H.230 : control and identification functions
- H.242 : for particular association functions regarding the 3 or 7 KHz audio channels

The Services level comprises :

- H.243 : videoconferencing in multi-point mode
- H.233 : security coding
- T.120 : protocols (more on these below)

A.4.2. THE T.120 FAMILY

The T.120 protocols are the core of the data transfer and control in the H.320 Recommendation. Using them, the clients can not only see each other, but exchange data files, share documents and participate in multi-user conferences.

The T.120 protocols are not interested in the audio and video aspects of videoconferencing, but rather in data in all its forms. They give a developer the possibility to move still images, to designate documents, to describe and manipulate objects, and to share applications from remote.

T.120 protocols do also add more controls to those provided in H.320, which allow session establishment, some more audio and video functions, and management of conferences. As a standard for conferencing, its action has two layers : services and infrastructure of communication.

The services layer includes 3 types of recommendations :

- T.126 for annotation and transfer from JPEG still images
- T.127 for binary files multi-point transfer
- T.128 for audio and video objects control

For reasons of fluidity and speed, as well as for synchronisation of the objects (a function H.320 cannot perform), the T.128 data is transferred independently from the T.120 proper data.

Eventually, the infrastructure layer includes 3 recommendations :

- T.124 for management of conferences (also called GCC for Generic Control Conference) : it allows to start/stop, join/quit, search for, divide/merge a conference
- T.122 and T.125, both multi-point communication services, allow data routing and the distribution of tasks between several sites and the interface of the physical layer
- T.123 allows the point-to-point connectivity between the various workstations, controls the data flows, and checks for errors

This package forms a complete, yet relatively open, base from which developers can build their own particular videoconferencing tools.

A.5. SECURITY CODING

As for now, we have surveyed many different kinds of coding standards : text and binary codes, compression codes for audio and video files, and codes for video-conferencing. All of these codes have an impact on the quantity of data generated by applications. There is another kind of coding that has been around for quite a time, namely the security coding.

There are many kinds of algorithms developed in this area. As said above, our goal is not to produce an exhaustive survey of the subject, but rather to pinpoint those methods that have an impact on traffic flows. As such, only those will be studied below.

A.5.1. IMPACT ON NETWORK TRAFFICS

Security coding can proceed in two approaches :

- **layer-to-layer approach** : in this case, a computer layer (usually, layer 3 or 4) receives an uncoded file from the above layers, encapsulate the file in a PDU, and code the whole frame before sending it to the other end. There, the corresponding layer of same level of the correspondent will decode the PDU before sending the file to the higher layers. That requires however that routers on the network are able to deal with completely coded frames (which is not the case on the Internet).
- **end-to-end approach** : in this case, the files are coded directly at the application layer by the user, and a coded file is handed out to the lower layers for delivery. This means that only the data payload of the frames is coded (contrary to the previous case, where all the frame was coded).

In the second case, cryptography has only indirect consequences on network traffic, and this only if the coding algorithm has an effect on the size of the data to be transmitted. This is the case for hashing functions or algorithms like RSA.

In the first case, this kind of coding implies an overhead in the frames, thus decreasing the useful load of data vehiculated. This kind of mechanism is implemented both (however differently) in IPv4 and IPv6.

In IPv4, cryptography is an option that is activated in the ‘’Options’’ field of the header (6th. 32-bits row), in Ipv6, it is included as an ‘’extra header’’ (as the Options field is discarded in this version) of 64 bits.

Another possible consequence, besides the added headers and the variation of the size of data, is the apparition of messages for exchange of session keys, which never happens under normal (i.e. without cryptography) circumstances.

It should be noted that, all in all, security doesn’t seem to have a major impact on traffic flows, so this kind of coding shouldn’t take too much of our time to analyse.

A.5.2. SINGLE-DIRECTION HASHING FUNCTIONS

A single-direction hashing function *H(M)* operates on a message *M* of arbitrary length. It gives as output a fixed length hashing code *h*.

Numerous functions take a variable-length input and give back a fixed-length output, but single-direction hashing functions have additional properties that make them popular :

- given *M*, it is easy to calculate *h*
- given *h*, it is difficult to find *M*
- given *M*, it is difficult to find another message *M’* such as *H(M) = H(M’)*

‘’Difficult’’ depends on the level of security specific to each situation, but the majority of existing applications define ‘’difficult’’ as ‘’needing 2^{64} or more operations to solve’’.

Current functions of this type include the MD4, MD5 and SHA (Secure Hash Algorithm).

From a network point of view, those algorithms are frequently used for authentication purposes.

A.5.3. SYMMETRICAL CODES (WITH PRIVATE KEYS)

An algorithm of coding with a secret key transforms a message *M* of arbitrary length in to a coded message $E_k(M) = C$ of same length using a key *k*. The reverse transformation ($D_k(M)$) uses the same key. Those algorithms verify the following characteristics :

- $D_k(E_k(M)) = M$
- given *M* and *k*, it is easy to calculate *C*
- given *C* and *k*, it is easy to calculate *M*
- given *M* and *C*, it is difficult to find *k*

Of course, in this case, difficulty is directly linked to the length of *k* (2^{56} for the DES algorithm and 2^{128} for the IDEA algorithm).

Those algorithms are used in networks for ‘’Encapsulating Security Payload’’ purposes (i.e. coding data), it is commonly used in the area of electronic commerce.

A.5.4. ASYMMETRICAL CODES (WITH PUBLIC/PRIVATE KEYS)

Contrary to the preceding case, those algorithms use two keys : one key k to code (called the public key) and one key k^{-1} to decode (called the private key).

Let's define :

$C = E_k(M)$

$M = D_{k^{-1}}(C)$

We have the following properties :

- $D_{k^{-1}}(E_k(M)) = M$
- given M and k, it is easy to find C
- given C and k^{-1} it is easy to find M
- given M and C, it is difficult to find k or k^{-1}
- given k, it is difficult to find k^{-1}
- given k^{-1}, it is difficult to find k

The two keys being ''independent'', the coding key can be widely known, this is why it has been christened the public key. The private key, in contrast, is only known to the entity decoding the message. The most common algorithm of this type is RSA (for the names of its authors : Rivest, Shamir and Adleman).

In networks, those algorithms are used mostly for coding transmissions between two or more people wishing to communicate in a secure way.

B. BIBLIOGRAPHY

[ABR98] P. Abry, D. Veitch, *Wavelet analysis of long-range dependent traffic*, IEEE Transactions on Information Theory 44 (1998), pp. 2-15. Available from http://www.serc.rmit.edu.au/~darryl/.

[ADD95] R.G. Addie, *Fractal Traffic: Measurements, Modelling and Performance Evaluatio*n, in Proceedings IEEE Infocom 1995, IEEE, 1995. Available from http://www.sci.usq.edu.au/staff/addie/papers.html.

[AKY97] I.F. Akyldiz, J. Liebeherr, I. Nikolaidis, *Multi-level rate-based flow control for ABR traffic*, Performance Evaluation 31 (1997), 107-131.

[BAR98a] P. Barford, A. Bestavros, A. Bradley, M. Crovella, *Changes in Web Client Access Patterns*, to appear in World Wide Web, Special Issue on Characterization and Performance Evaluation, 1999. Available from http://www.cs.bu.edu/faculty/crovella/papers.html.

[BAR98b] P. Barford, M. Crovella, *Generating Representative Web Workloads for Network and Server Performance Evaluation*, in Proceedings of Performance '98/ACM SIGMETRICS '98, pp. 151-160. Available from http://www.cs.bu.edu/faculty/crovella/papers.html.

[BEC86] M.Becker, G.Pion, « *Simulation of an Apollo Network in order to Evaluate Response-TimeSpeed-up of Parallel Algorithms* », *The Journal of Systems and Software*, no. 1--2, pp. 81--91, May 1986.

[BEC90] M.Becker, C.Dekoninck, J.P.Prost, B.Verrier, « *A Stochastic Petri Net Model for the FPS/264* », *Computer System Science and Engineering*, Vol.5, no. 2, April 90, pp. 65--72.

[BEC93] M.Becker, P.Douillet, *« Une méthode hiérarchique auto-régulée de simulations d'événements extrémement rares », Note aux Comptes-Rendus de l'Académie des Sciences*, 7-1-93, pp. 87--92.

[BEC98] M.Becker and A.-L.Beylot, « *Performance Comparison Criteria for ATM Switch Models* », 6th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, Bradford, July 1998, pp. 30/1--30/10.

[BER94] J. Beran, *Statistics for Long-Memory Processes*, Chapman & Hall, New York, 1994.

[BEY91] A.-L.Beylot, K.Waithayakul and M.Becker, « *Etude des performances d'un réseau d'inter-connexion de Clos dans l'environnement ATM* », *Réseaux et informatique répartie*, Vol. 1, no. 3, October 1991, pp. 275--295.

[BOW95] M. Bowman, B. Camargo, C. Dharap, M. Spasojevic, M. Satyanarayanan, *A Toolkit for Analyzing Client Access Patterns in the World Wide Web*, WWW'95 Conference, 1995. Available from http://www.transarc.com/afs/transarc.com/public/trg/papers.

[BOX70] G.E.P. Box, G.M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden-Day San Francisco, 1970.

[CAL90] M. Calzarossa, R. Marie, K.S. Trivedi, *System Performance with User Behaviour Graphs*, Performance Evaluation 11 (1990), pp. 155-164.

[CAT95] L.D. Catledge, J.E. Pitkow, *Characterizing browsing strategies in the World-Widw Web*, Computer Networks and ISDN Systems 26-6 (1995), pp. 1065-1073. Available from

http://www.igd.fhg.de/www/www95/papers/80/userpatterns/UserPatterns.Paper4.formatted.html.

[CHE97] C. Chen, Structuring and Visualizing the WWW by Generalized Similarity Analysis, in Proceedings of Hypertext '97. Available from http://www.brunel.ac.uk/~cssrccc2/papers/ht97.pdf.

[CHO98] B.D. Choi, G.U. Hwang, Y.W. Jung, H. Chung, *The periodic Markov modulated batch Bernoulli process and its application to MPEG video traffic*, Performance Evaluation 32 (1998), pp. 301-317.

[CON96] M. Conti, E. Gregori, A. Larson, *Study of the Impact of MPEG-1 Correlations on Video-Sources Statistical Multiplexing*, IEEE Journal on Selected Areas in Communications 14-7 (1996), pp. 1455-1470.

[CRO95] M.E. Crovella, A. Bestavros, *Explaining World Wide Web Traffic Self-Similarity*, Technical Report TR-95-015, Computer Science Department, Boston University, October 1995. Available from http://www.cs.bu.edu/faculty/crovella/papers.html.

[CRO97] M.E. Crovella, A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE/ACM Transactions on Networking 5-6 (1997), pp. 835-846. Available from

http://www.cs.bu.edu/faculty/crovella/papers.html.

[CRO99] M.E. Crovella, M. Taqqu, *Estimating the Heavy Tail Index from Scaling Properties*, to appear in Methodology and Computing in Applied Probability 1-1 (1999). Available from http://www.cs.bu.edu/faculty/crovella/papers.html.

[DAN98] T. Daniels, C. Blondia, *A discrete-time ATM traffic model with long range dependence characteristics*, Departments of Mathematics and Computer Science, University of Antwerp, March 1998. Available from http://win-www.uia.ac.be/u/pats/publications.html.

[DIA98] J.E. Diamond, A.S. Alfa, *Matrix analytical model of an ATM output buffer with self-similar traffic*, Performance Evaluation 31 (1998), pp. 201-210.

[DIL98] J. Dilley, R. Friedrich, T. Jin, J. Rolia, *Web server performance measurements and modelling techniques*, Performance Evaluation 33 (1998), pp. 5-26.

[ERR94] A. Erramilli, R.P. Singh, P. Pruthi, *Chaotic Maps as Models of Packet Traffic*, in Proceedings of the 14th ITC, 1994, pp. 329-338. Available from http://www.qmetrix.com/public.html.

[ERR95] A. Erramilli, P. Pruthi, W. Willinger, *Recent Developments in Fractal Traffic Modelling*, Proceedings of the ITC Regional Seminar, 1995. Available from http://www.qmetrix.com/public.html.

[FAN97] Y. Fan, N.D.Georganas, *Performance Analysis of ATM Switches with Self-Similar Input Traffic*, Systems and Computer Engineering J. 1 (1997). Available from http://www.mcrlab.uottawa.ca/publications/.

[FEL98A] A. Feldmann, A.C. Gilbert, W. Willinger, T.G. Kurtz, *The Changing Nature of Network Traffic*, Computer Communication Review 28-2 (1998), pp. 5-29.

[FEL98B] A. Feldmann, A.C. Gilbert, W. Willinger, *Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic*, Computer Communication Review 28-4 (1998), pp. 42-55.

[FLA92] P. Flandrin, *Wavelet Analysis and Synthesis of Fractional Brownian Motion*, IEEE Transactions on Information Theory 38-2 (1992), pp. 910-917,.

[FOU96] Y.Fouquet, R.Scheeman, D.Cypher and A.Mink. *« ATM Performance Measurement »*, In proceedings of TDP96, La Londe les Maures, France, June 1996, pp. 63--75.

[FRE98] M. Frey, S. Nguyen-Quang, A Gamma based Framework for Modelling Variable-Rate MPEG Video Sources: The GOP GBAR Model. Available from ftp://isdn.ncsl.nist.gov/pubs/papers/GOPGBARmodel.ps.Z.

[GAR94] M. Garrett, W. Willinger, *Analysis, Modelling and Generation of Self-Similar VBR Video Traffic*, in Proceedings SIGCOMM '94, 1994. Available from ftp://ftp.bellcore.com/pub/dvw/sigcomm94mwg.ps.

[HEY96a] D.P. Heyman, T.V. Lakshman, *Source Models for VBR Broadcast-Video Traffic*, IEEE/ACM Transactions on Networking 4-1 (1996), pp. 40-48.

[HEY96b] D.P. Heyman, T.V. Lakshman, *What Are the implications of long-Range Dependence for VBR-Video Traffic Engineering?*, IEEE/ACM Transactions on Networking 4-3 (1996), pp. 301-317.

[HEY97] D.P. Heyman, *The GBAR Source Model for VBR Videoconferences*, IEEE/ACM Transactions on Networking 5-4 (1997), pp. 554-560.

[HOS81] J.R.M. Hosking, *Modelling Persistence in Hydrological Time Series Using Fractional Differencing*, Water Resources Research 20-12 (1984), pp. 1898-1908.

[HOU88] P.Houeix and M.Becker, *« Modélisation d'une architecture parallèle pour le traitement d'images »*, *RAIRO Recherche Opérationnelle*, Vol. 22, no. 1, 1988, pp. 45-65.

[HUA95] C. Huang, M. Devetsikiotis, I. Lambadaris, A. Roger Kaye, *Modelling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach*, SIGCOMM'95, 1995. Available from http://www.acm.org/sigcomm/sigcomm95/papers/huang.html.

[HUE98] F. Huebner, D. Liu, J.M. Fernandez, *Queuing Performance Comparison for Traffic Models for Internet Traffic*. Available from

http://www.att.com/attlabs/archive/.

[ISM96] M.R.Ismail, I.Lambadaris, M.Devetsikiotis, A.R.Kaye, *Simulation and Modelling of Variable Bit Rate MPEG Video Transmission over ATM Networks*, International Journal of Communication Systems 9 (1996), pp. 283-297. Available from http://www.sce.carleton.ca/bbnlab/bnlpapers.shtml.

[IYE98] A.K. Iyengar, E.A. MacNair, M.S. Squillante, Li Zhang, *A General Methodology for Characterizing Access Patterns and Analyzing Web Server Performance*, MASCOTS '98, pp.167-174.

[JAG96] D.L. Jagerman, B. Melamed, W. Willinger, *Stochastic Modelling of Traffic Processes*, in Frontiers in Queuing:Models, Methods and Problems (J.H. Dshalalow, Ed.), CRC Press, 1996. Available from http://rutcor.rutgers.edu/~melamed/papers/my_papers.html.

[JEL96] P. R. Jelenkovic, A. A. Lazar and N. Semret, *Multiple Time Scales and Subexponentiality in MPEG Video Streams*. In Proceedings of the International IFIP-IEEE Conference on Broadband Communications, Montreal, Canada, April 23-25, 1996. Available from http://comet.columbia.edu/publications/conference.html.

[JEN96] A.K. Jena, A. Popescu., P. Pruthi, A. Erramilli, *Traffic Control in ATM Networks: Engineering Impacts of Realistic Traffic Processes*, in Proceedings of Nordisk Teletrafik Seminarium NTS - 13, Trondheim, Norway, August 1996. Available from http://www5.hk-r.se/fou/forskinfo.nsf/69756130e11c344cc1256483005b3e34/1e0a6a6ba9170c95c12565c900311e52?OpenDocument.

[KAD98] H.Kaddeche, A.-L.Beylot and M.Becker, *« Approche Analytique pour l'étude des performances de serveurs multimédias multidisques »*, *Recherche Opérationnelle/Operation Research*, Vol. 32, no. 3, 1998, pp. 227--251.

[KLE75] L. Kleinrock, Queuing Systems, Volume I, Wiley-Interscience New York, 1975.

[KOB92] H. Kobayashi, Q. Ren, *A Mathematical Theory for Transient Analysis of Communications Networks*, IEICE Transactions on Communications E75-B-12 (1992, pp. 1266-1276.

[KRI97] K.R. Krishnan, G. Meempat, *Long-range dependence in VBR video streams and ATM traffic engineering*, Performance Evaluation 30 (1997), pp. 45-56.

[KUH94] P.Kühn, *« Performance Modelling and traffic engineering for Broadband Communications Networks »*, Invited paper, Broadband Communications II, North Holland, 1994, pp. 245--274.

[KUM96] V. Kumar - MBONE, *Interactive Multimedia on the Internet*. New Riders Publishing, Indianapolis, USA. 1996.

[LAU95] W-C. Lau, A. Erramilli, J. Wang, W. Willinger, *Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and its Properties*, ICC '95.

[LAZ93] Lazar, A.A., Pacifici, G. and Pendarakis, D.E., *Modelling Video Sources for Real-Time Scheduling*. In Proceedings of the Globecom'93 , Houston TX, November 29 - December 2, 1993, pp. 835-839. Available from http://comet.columbia.edu/publications/conference.html .

[LAZ94] Lazar, A.A., Pacifici, G., Pendarakis, D.E., *Modelling Video Sources for Real-Time Scheduling*, ACM/Springer Verlag Multimedia Systems 1-5 (1994), pp. 253-266. Available from

http://www.seas.columbia.edu/columbia/departments/ee/faculty/lazar.html .

[LEL93] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, *On the Self-Similar Nature of Ethernet Traffic*, IEEE/ACM Transactions on Networking 2-1 (1994), pp. 1-15. Available from ftp://ftp.bellcore.com/pub/dvw/.

[LIE98] J. Liebeherr, D.E. Wrege, *Traffic characterization algorithms for VBR video in multimedia networks*, Multimedia Systems 6 (1998), pp. 271-283.

[LIK95] N. Likhanov, B. Tsybakov, N.D. Georganas, Analysis of an ATM buffer with Self-Similar ("Fractal") input Traffic, Proc. IEEE Infocom '95, Boston, pp. 985-992, 1995. Available from

http://www.mcrlab.uottawa.ca/publications/.

[MAN68] B.B. Mandelbrot, J.W. Van Ness, *Fractional Brownian Motions, Fractional Noises and Applications*, SIAM Review 10-4 (1968), pp. 422-437. Available from http://rutcor.rutgers.edu/~melamed/papers/qtes_papers.html.

[MCL91] D.L. McLaren, D.T. Nguyen, *A fractal based source model for ATM packet video*, in Proceedings 6th International Conference on Digital Processing of Signals in Communications, 1991, pp. 78-83.

[MEL92] B. Melamed, B. Sengupta, TES Modelling of Video Traffic, IEICE Transactions on Communications E75-B-12 (1992), pp. 1292 - 1300.

[MEL93] B. Melamed, *An Overview of TES Processes and Modelling Methodology*, in Performance Evaluation of Computer and Communication Systems (L. Donatiello and R. Nelson, Eds.), Springer-Verlag , 1993, pp. 359-393. Available from http://rutcor.rutgers.edu/~melamed/papers/my_papers.html.

[MOU98] M.Moulki, A.-L.Beylot, L.Truffet, and M.Becker, *« An aggregation technique to evaluate the performance of a two-stage buffered ATM Switch », Annals of Operation Research*, 79(1998), pp. 373--392.

[NEA95] T. M. Neame, R. G. Addie, F. Hubner and M. Zukerman, *Investigation of Traffic Models for High Speed Data Networks*, Proceedings, The Australian Telecommunication Networks and Applications Conference. Woolongong, 1995. Available from http://www.sci.usq.edu.au/staff/addie/papers.html.

[NEA97] T. Neame, R. Addie, F. Huebner, M. Zukerman, *Towards a Simple Traffic Model for High Speed Data Networks*, 1997. Available from http://www.sci.usq.edu.au/staff/addie/papers.html.

[PAX94] V. Paxson, *Empirically-Derived Analytic Models of Wide-Area TCP Connections*, IEEE/ACM Transactions on Networking, 2- 4, 1994. Available from http://ee.lbl.gov/nrg-papers.html.

[PAX95a] V. Paxson, *Fast Approximation of Self-Similar Network Traffic*, Technical Report, April 1995. Available from http://www-nrg.ee.lbl.gov/nrg-papers.html.

[PAX95b] V. Paxson, S. Floyd, *Wide-Area Traffic: The Failure of Poisson Modelling*, IEEE/ACM Transactions on Networking 3-3 (1995). Available from http://www-nrg.ee.lbl.gov/nrg-papers.html.

[PIR96] P. Pirolli, J. Pitkow, R. Rao, *Silk from a Sow's Ear: Extracting Usable Structures from the Web*, in Proceedings CHI'96. Available from http://www.acm.org/sigchi/chi96/proceedings/papers/Pirolli_2/pp2.html.

[PIT97] J. Pitkow, *In Search of Reliable Usage Data on the WWW*, GVU Technical Report GIT-GVU-97-13, Xerox Palo Alto Research Center, 1997. Available from http://www.gvu.gatech.edu/gvu/reports/1997/.

[POS98] E. Rehmi Post, *A Gentle Introduction to Wavelets*, available from http://i.fix.org/~rehmi/unpublished.html , 1998.

[POT84] D.Potier and M.Veran, *« QNAP2: a portable environment for queuing systems modelling »*, International Conference on Modelling and Performance Analysis Tools, May 1984.

[PRO89] J.-P.Prost, M.Becker, « Modelling Methodology for Performance Evaluation of Parallel Architectures: a case study, LCAP », International Journal of High Speed Computing, Vol. 1, no. 4, 1989, pp. 563--601.

[RAG85] S.V. Raghavan, R. Kalyanakrishnan, *On the Classification of Interactive Users based on User Behaviour Indices*, Performance Evaluation Review 13-2 (1985), pp. 40-48.

[RAG95] S.V. Raghavan, P.J. Joseph, *Workload Models for Multiwindow Distributed Environments*, in Quantitative Evaluation of Computing and Communication Systems, Springer-Verlag Heidelberg, 1995.

[REN98] W. Ren, K.-Y. Siu, H. Suzuki, G. Ramamurthy, *Performance of TCP in IP/ATM internetworks*, Computer Communications 21 (1998), pp. 1610-1623.

[ROB97] S. Robert, J.-Y. Le Boudec, *New models for pseudo self-similar traffic*, *Performance Evaluation* 30 (1997), pp. 57-68.

[ROS97] O. Rose, *Simple and efficient models for variable bit rate MPEG video traffic*, Performance Evaluation 30 (1997), pp. 69-85.

[RYU96] Ryu, B.K., Elwalid, A., *The Importance of the Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities*, Proceedings ACM SIGCOMM '96 Stanford University, CA, 1996.

[RYU98] B. K. Ryu, S. B. Lowen, *Point Process Models for Self-Similar Network Traffic*, with Applications, Stochastic Models, 1998. Available from http://engc.bu.edu/~lowen/pub_list.html .

[SAL87] G. Salton, C. Buckley, *Term Weighting Approaches in Automatic Text Retrieval*. Available from

http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell/TR87-881.

[SAL90] G. Salton, C. Buckley, *Approaches to Global Text Analysis*. Available from http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell/TR90-1113.

[SEB98] I. Sebüktekin, T.J. Bogovic, P. Krishnaswamy, *An empirical study of TCP/IP performance over ATM*, Computer Communications 21 (1998), pp. 1058-1069.

[SEN89] P. Sen, B. Maglaris, N.-E. Rikli, D. Anastassiou, *Models for Packet Switching of Variable Bit Rate Video Sources*, IEEE J. on Selected Areas in Communications 7-5 (1989), pp. 865-869.

[SHA95] C. M. Sharon, I. Lambadaris, M. Devetsikiotis and A.R. Kaye, *Accurate Modelling of H.261 VBR Video Sources for Packet Transmission Studies*. In Proc. IEEE Global Telecommunications Conference, Singapore, November 1995. Available from http://www.sce.carleton.ca/bbnlab/bnlpapers.shtml.

[SHA97] C.M.Sharon, I.Lambadaris, M.Devetsikiotis, A.R.Kaye, *Modelling and Control of VBR H.261 Video Transmission over Frame Relay Networks*, IEEE Transactions on Circuits and Systems for Video Technology 7-3 (1997).

[SHE98] Sheng Ma, *Network Traffic Modelling and Analysis*, Ph.D. Thesis, available from http://www.rpi.edu/~mas2/ , 1998.

[SLN94] M. Stoksik, R. Lane, D. Nguyen, *Accurate synthesis of fractional Brownian motion using wavelets*, Electronics Letters 30-5 (1994), pp. 383-384.

[TAR98] T. Taralp, M. Devetsikiotis, I. Lambadaris, and A. Bose, *Efficient Fractional Gaussian Noise Generation Using the Spatial Renewal Process*. In Proceedings of the International Conference on Communications, Atlanta, USA, June 1998. Available from

http://www.sce.carleton.ca/bbnlab/bnlpapers.shtml .

[TAU97] L. Tauscher, S. Greenberg, *How people revisit web pages: empirical fidings and implications for the design of history systems*, Int. J. Human-Computer Studies 47-1 (1997), pp. 97-137. Available from http://ijhcs.open.ac.uk/tauscher/tauscher-pdf.html.

[TSY97] B. Tsybakov, N.D. Georganas, *A Model of Self Similar Communications-Network Traffic*, Proc. Distributed Computer Communication Networks (DCCN) '97, Tel-Aviv, Israel, Nov. 1997, pp.212-217. Available from http://www.mcrlab.uottawa.ca/publications/.

[ULR98] R. Ulrich, U. Herzog, P. Kritzinger, *Modelling buffer utilization in cell-based networks*, Performance Evaluation 31 (1998), pp. 183-199.

[VLA98] N. Vlajic, H.C. Card, Categori, *Categorizing Web pages on the subject of neural networks*, Journal of Network and Computer Applications 21 (1998), pp. 91-105.

[YAN96] T.W. Yan, M. Jacobsen, H. Garcia-Molina, D. Umeshwar, *From User Access Patterns to Dynamic Hypertext Linking*, Computer Networks and ISDN Systems 28-11 (1996), p. 1007-. Available from http://www.imag.fr/Multimedia/www5cd/www293/overview.htm.

C. ACRONYMS

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ADPCM	Adaptive Differential Pulse Code Modulation
AR	Auto Regressive
ARFIMA	AutoRegressive Fractionally Integrated Moving Average
ARMA	Auto Regressive Moving Average
ARQ	Automatic Repeat reQuest
ASCII	American Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
BRI ISDN	Basic Rate Interface ISDN
CAC	Call Admission Control
CBR	Constant Bit Rate
CCIR	Comité Consultatif International des Radiocommunications
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CDV	Cell Delay Variation
CGI	Common Gateway Interface
CIF	Common Intermediate Format
CODEC	enCOder/DECoder
CS-ACELP	Conjugate-Structure Algebraic-Code-Excited Linear-Prediction
CSCW	Computer Supported Collaborative Works
DCT	Discrete Cosine Transform
DECT	Digital European Cordless Telecommunications
DVD	Digital VideoDisc
FBm	Fractional Brownian Motion
FGn	Fractional Gaussian Noise
FIFO	First In First Out
FPS	Frames Per Second
FTP	File Transfer Protocol
GBMA	Gamma processes with Beta distributions, then applying the Moving Average filter to it
GOP	Groups Of Pictures
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HDTV	High Definition Television
HTML	HyperText Markup Language
HTTP	*HyperText Transfer Protocol*
IDC	Index of Dispersion for Counts
IDI	Index of Dispersion for Intervals
IP	Internet Protocol
IP-M	IP Multicast
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISE	Infinite Server Effect
ITU-T	International Telecommunications Union-Telecommunications

JPEG	Joint Photographic Experts Group
LAN	Local Area Network
MA	Moving Average
MBONE	Multicast Backbone On the interNEt
MLM	Mailing List Managers
MMPP	Markov-Modulated Poisson Processes
MPEG	Motion Picture Experts Group
NFS	Network File System
NNTP	Network News Transfer Protocol (RFC 977)
PASC	Precision Adaptive Subband Coding (a compression algorithm)
PCM	Pulse Code Modulation
PDU	Protocol Data Unit
PLMN	Public Land Telephone Network
PMR	Peak-to-Mean Ratio
POTS	Plain Old Telephone System
PRI ISDN	Primary Rate Interface ISDN
PSTN	Public Switched Telephone Network
PTC	Poisson Traffic Comparison
PTM	Point-To-Multipoint
PTM-G	PTM Group Call
PTM-M	PTM Multicast
PTP	Point-To-Point
PTP-CLNS	PTP Connectionless Network Service
PTP-CONS	PTP Connection Orientated Network Service
QCIF	Quarter CIF
QoS	Quality of Service
RMD	Random Midpoint Displacement
SDU	Service Data Unit
SIC	Sensitive dependence on Initial Conditions
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPARC	Scalable Processor Architecture Reduced instructionset Computer
TES	Transform-expand-sample
UBE	User Behaviour Entropy
UBM	User Behaviour Mobility
UBR	Unspecified Bit Rate
URL	Uniform Resource Locator
UTP	Unshielded Twisted Pair
VBR	Variable Bit Rate
VOD	Video On Demand
WAN	Wide Area Network
WYSIWIS	What You See Is What I See
WWW	World Wide Web