ESPRIT PROJECT EP28425

Broadband Integrated Satellite Network Traffic Evaluations

Deliverable 2.1

Identification of Terrestrial Network Characteristics

Status / Version : **DELIVERABLE / FINAL**
Date : **March 31, 1999**
Distribution : **Public**
Code : **BISANTE/DEL21**
Author (s) : **J-Y. Chiaramella (Ed.), M. Becker, M.Marot**

Abstract : The goal of this deliverable is to provide accurate models of existing terrestrial networks, to be used in conjunction with the user behavior and application models defined in WP 1. In order to achieve this objective, a survey of the most widely used terrestrial network characteristics was done, with special attention given to those characteristics that should have a direct impact on user and applications behavior.
A second phase was to build the corresponding network models, with regard to the results of the survey done before, by selecting suitable modeling methods, and applying them to the task at hand.

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 ultimate goal of the BISANTE project is to provide network planners/developers with a software package allowing them to model the behaviour of their projected networks. For this, two kinds of models need to be established :

- first, of course, the working of the network's protocols need to be modeled, since they are the core of every network and determine almost all its characteristics

- second, the traffic flows sources must also be modeled, as no network simulation results can be deemed as accurate if traffic sources are not valid. In the case of communications networks, the traffic sources are the applications running on the computers connected to the networks, and, by extension, the users that manipulate them.

Modelization of user behaviors is to be done in WorkPackage 1. WorkPackage 2 is concerned with the modelization of the networks themselves. But it is important to note that the two models are not independent from each other, but are rather strongly linked. It should be clear that network performances have a major impact on the way the users act. For example, bad performances usually result in unhappy users that are far less likely to use the network (or at least certain applications). Thus QoS (Quality of Service) parameters have impact on users behavior, which means that network features that influence those parameters have to be identified and integrated in our models.

QoS is perceived by the user in two ways :

- the first thing of importance to the user is the way the application ends. Did the session end in failure or in success ? This is called the **failure rate**.

- the second factor is the amount of time passed before the session ends (either in failure or success). This is called the **delay rate**.

It is then paramount to identify network characteristics that have an impact on the two above-cited rates.

Now, there are so many different kinds of network that characteristics are likely to fluctuate wildly. This is why we will restrict ourselves in this document with the study of terrestrial networks characteristics (other documents in WorkPackage 2 will cover the other kinds of networks). Terrestrial networks alone are already a pretty much varied lot, encompassing several different kinds of technologies. The most important differences exist between **LANs** (Local Area Networks) and **WANs** (Wide Area Networks), as the difference in size implies numerous technical differences as outlined below :

- as WANs are greater sized than LANs, their cost is also far more expensive. In fact, if network planners try to use the same tools they use for LANs for setting up WANs (e.g. fiber optic lines), they will find the global cost completely prohibitive. This is why WANs offer generally lesser performances (in term of speed,

bandwidth, reliability, ...) as planners wanted to save costs by using less-reliable technology.

- WANs are also very difficult to maintain, due to their size and the number and diversity of users. To counter this, and also to make more efficient use of lesser bandwidth (compared to LANs), WANs protocols tend to offer more security and error-control mechanisms to users .

Now we have managed to break terrestrial networks into two categories, we need to select those networks that make up these categories :

- **LANs** : Three technologies dominate the field of local terrestrial network communications nowadays.
  + The first one is **Token Ring**, put up by IBM in the mid-70s, which is a 16 Mbps, reliable LAN.
  + Introduced later was **FDDI** (Fiber Distributed Data Interface) which was the first LAN technology to offer a speed of 100 Mbps.
  + Last was **Ethernet**, which is unarguably the most successful LAN to date.

- **WANs** : WAN technology has produced a very diverse range of networks and protocols, so we have restricted ourselves to those that are the most popular today.
  + **X.25** first appeared in the 60s as a reliable protocol for (mostly) audio transfer.
  + **Frame Relay** was created later as a less reliable, but quicker replacement for X.25.
  + However, all those technologies were used mostly for audio and data transfer, but had difficulties to adapt to other flows (video, multimedia, ...). This is why **ATM** at first, and then **the IP family of protocols**, came up with two specific (and vastly different) mechanisms to handle such kind of traffic.

This list covers all the most widely used networks in the terrestrial field, and all that have been characterized in the course of the BISANTE project. However, for the remainder of our study, it is important to determine which network types will continue to play important roles in the future, and which ones are currently still in use, but are progressively loosing their grip. As the process of building models is long and complicated, we have to pick up only the models that are truly interesting for the industrial world.

With this in mind, we have decided to discard some network technologies :

- Token Ring and X.25 : although still in use, both are old, slow, no more cost-effective technologies that are today more and more supplanted by the performances of newer network techniques.

- FDDI : due to its high cost, FDDI has been restricted to backbone use over the years. As such, it is still interesting, but is a stable technology with few innovations and improvements, and is also loosing its title of ‘’fastest network in existence’’ to Fast Ethernet (or even worse, Gigabit Ethernet).

- Frame Relay, which has supplanted X.25 in numerous countries (including the USA), now finds itself threatened by the development of ATM and, even more so, IP, which have proven to be far more flexible technologies.

As such, we will develop models for :

- Ethernet : most popular LAN today, and more so, still in evolution and development, as illustrates the release of two new versions in quick time : Fast Ethernet and Gigabit Ethernet.

- ATM : solid protocol, widely used, especially in the telecommunications area.

- IP (and associated protocols) : the most renowned protocol in use today, IP is the core of the extremely popular Internet, but is quickly finding new avenues of application as well.

Models were developed using various techniques. Although those techniques are well-known in the field of network modeling, we have nonetheless provided the readers with their review, including :

- A presentation of the **queue model**, according to Kleinrock's definition in [1] :
  + notion of servers and customers
  + arrival processes
  + service processes
  + Kendall's notation A/B/m/k/s

- The **mathematical tools and expressions** used in this context :
  + Markov chains, both discrete- and continuous-time
  + Birth-death processes
  + Little's formula
  + Theorems of Jackson, of Gordon and Newell
  + BCMP theorem

- A discussion about **computer simulations** : their strong points and their pitfalls :
  + Measurements, benchmarks and modelization
  + Various types of simulation (discrete, event-driven, …)
  + Problems linked to simulations (model validity, precision degree, aggregation methods, rare events …)

- A presentation of **LANs and WANs modeling** methodology, in accordance with the previous points.
  + CSMA / CD analysis and modeling (both simple and more complex models)
  + Notion of switching in WANs
  + Telecommunication systems modeling

2. INTRODUCTION

As the goal of the BISANTE project is to provide the industrial world with a mean to modelize the behavior of existing or planned networks with the highest possible accuracy, the partners involved have stated their intention to develop models both for the networks proper and also for the traffic sources (i.e. applications and network users). Modelization of networks is doubtless a critical part in the course of the project, for no traffic sources models can be validated if not run on correct network models. With respect to this possibility, it is then clear that WorkPackage 2 should provide the consortium with network models meant to interact with the models produced in WorkPackage 1.

This document is concerned specifically with Task 2.1 (part of WorkPackage 2), whose main objective is to produce models for terrestrial networks. Terrestrial networks have always been the most widely developed networks in use, both historically and nowadays, because they are the less expensive to install and to maintain, very flexible, and also because of their good performances, as terrestrial networks usually allow high-capacity transmissions with a relative low error rate. As a result of this relatively long lifespan and their good technical characteristics, the term ''terrestrial networks'' encompasses quite a few different types of networks, with significant differences between them. Perhaps the most significant differences exist between LANs and WANs (or Local Area Networks and Wide Area Networks), because the disparity in size of the two types of networks has resulted into very specific approaches in the conception of the protocols.

In order to achieve this goal, the following work plan has been adopted :

- Initially, we will have to make a survey of all existing terrestrial networks, select those which are the most widely developed and/or seem the most promising, and determine their technical characteristics, with particular interest given to those characteristics which should have an impact on the models developed in WorkPackage 1.

- Once this survey is done, we will begin the modelization phase in itself, by determining how the networks will be modeled, by selecting which ones we will be able to build successfully using the selected method in the allocated time, and by building the models themselves.

The present document will follow this work plan's structure closely. In a first part, we will identify the general network factors influencing user and application behaviors. Then, we will follow by a survey of existing terrestrial networks (LANs and WANs) and their characteristics (with special interest given to characteristics related to the factors identified before), before detailing the modelization method we have followed and drawing a final conclusion.

3. IDENTIFICATION OF NETWORK CHARACTERISTICS

As already said, terrestrial networks share some common characteristics, but not all of them are relevant to our project. Our first goal would be then to identify what kind of parameters are really interesting in our work.

3.1. FROM A NETWORK-ORIENTED POINT OF VIEW

One obvious way to proceed with this classification is to use the common performance criteria in terrestrial networks, and conduct a survey of those technologies accordingly.

The usual performance criteria along whom networks are judged are the following ones :

- **Queuing Delay :** the delay a PDU (Protocol Data Unit) has to wait in a queue in a network equipment (gateways, bridges, routers, repeaters, …).

- **Throughput :** usually measured either in bit per second, or in packet (or more precisely PDU) per second, represents the portion of input traffic that reaches its final destination and its evolution through time. It is sometimes interesting in certain cases to make the difference between the traffic going in one direction and the traffic flowing in the other one, instead of considering the whole aggregated traffic.

- **Utilization :** measures the percentage of bandwidth taken by the traffic, and its evolution through time. As in the previous case, it can be useful to make distinctions based on the flow direction.

- **Bit Error Rate :** can be measured either as a percentage (x % of bits are left as faulty because of network problems every unit of time, usually second), or as a number (x bits are left as faulty because of network problems every second).

- **Packet Loss Ratio :** very similar to the previous parameter, it measures the loss of packets (either in quantity or as a percentage) lost during transmission in a network, for various reasons (electronic defect on a cable, packet deemed to old and discarded, …).

- **Specific Criteria :** apart from the previous criteria that are common to all types of terrestrial networks, other characteristics exist that are specific to each network. For example, the Collision Ratio is very important for Ethernet networks. As they are too numerous to be detailed here, we will examine them in chapter 4 and 5 where the various kinds of networks are presented.

- Var**iation of the delay (jitter) :** For some traffics the delay has to be constant, for example for voice traffic variations of transmission delays may introduce bad results. For CBR traffic (Constant Bit Rate) an important criteria is the variation of transmission delay. For ATM cell traffic it is called CDV (Cell Delay Variation).

That method of classification would have been the perfect solution had we wanted to improve existing terrestrial network models. However, as said before, our main interest is not the terrestrial networks per se, but how their characteristics influence the upper layers (especially the users).

3.2. FROM AN UPPER LAYER-ORIENTED POINT OF VIEW

Another way to proceed with this classification, far more related to our goals in the project, is to take an user-oriented view of the subject. An user is mainly interested in calling for services and expect results from these calls. Therefore, the two parameters of concern for him are :

- is the service he called for accomplished successfully, or did the whole session end in failure ?

- how much time has passed before the user knows how the session ended (either in success or failure) ?

These two parameters are likely to have serious influence on the user's behavior and on the way he reacts to the network's feedback.

3.2.1. THE DELAY

In this part, we will consider the characteristics of networks that have impact on the delay between a user's call for a service and the end of the session.

- **Time delay related to connection-oriented networks** : some networks function on a connection-oriented mode, meaning that two entities must first agree on a connection before actually transmitting between themselves. This implies, from the application point of view, an additional delay corresponding to the time needed for the Connection Request and Connection Confirm messages to reach their destination.

- **Time delay between two nodes** : directly dependent on the speed of the network.

- **Time delay on a node** : a message circulating on a network can be forced to wait on a particular node before being forwarded (or sent if it is its originating node) for several reasons :

  - the node is waiting for an authorization to emit, this is typically the case for token networks, where only the possession of a token (usually a message of particular format) enables a node to ''speak''. In terrestrial networks, it is the case for Token Ring or FDDI networks.

  - the node is waiting for its turn to speak. On certain networks, to avoid collisions of messages, the protocol does require that a particular computer ''listen'' if another one is not already sending a message on the network before actually emitting. That is the case of all the different kinds of Ethernet which rely on the CSMA/CD

(Carrier Sense Multiple Access with Collision Detection, see below for details) protocol to avoid collisions.

- the node has several messages to send and the priority of this particular message is lower than the others'. Priority messaging is not widely used for performance reasons, but a number of networks, like Token Ring or IP networks (either the IPv4 or the IPv6 versions), have implemented this functionality (note that this is only an option in IP networks).

- the node has a buffer already loaded with other messages, so the message has to wait for its turn before being sent. This is perhaps the most basic reason for delay when transiting through a node, and can happen on practically every tools present on a network (routers, gateways, bridges, ...).

- **Error rate** : during a transfer a message can be corrupted, in this case, when the problem will be detected, an error message will be sent back, and the sending machine will have to resend the message. Of course, this result in an increase in delay before the session is actually completed. Note that this particular kind of delay is extremely dependent on the type of network, and particularly its politic of error-checking. If the network provides point-to-point error-checks, that means that the corruption will be detected by a transit node, so the loss in time is only double that delay to reach this particular transit node. On the other hand, if the error control is only done end-to-end, this means that errors are not detected until the message reaches its final destination, so in this case the delay is double the *total* time of transmission from end to end (this is the case for example in IP networks).

3.2.2. THE FAILURE RATE

In this part, we will identify what kind of factors have influence over the rate of failure in network sessions.

- **Error rate** : obviously, the more errors there are on a network, and the more delay it takes to spot them and remedy to them, the more likely the session will end in failure. Generally, an user's computer will try to repeat the transmission only for a fixed set of time, if this timer expired the whole session is declared a failure and the machine stop retransmitting the packet.

- **The error-managing policies** : the degree of quality of service in the transmissions as provided by the particular network is also a prime factor to reduce (or increase) failure rates. For example, that is typically the difference between ATM networks who offer very good quality of transmission (paid by a relative slow speed), and IP networks that offer very few in terms of guarantee of transmission (but offer far more attractive speeds).

- **Error rate related to connection-oriented networks** : if a particular protocol functions on a connection-oriented mode, there is a possibility that the connection requests will be refused, which can end the whole session in failure.

4. CHARACTERISTICS OF LANS

LANs, because of their relatively small size, usually enjoy the luxury of very high capacity. That allows a degree of liberty in design unfound in larger-size networks, which results in very specific characteristics.

4.1. ETHERNET

Ethernet is perhaps the most popular physical layer LAN technology in use today. It is so because Ethernet strikes a good balance between speed, cost and ease of installation. These strong points, combined with wide acceptance in the computer marketplace and the ability to support virtually all popular network protocols, make Ethernet an ideal networking technology for most computer users today.

There are three different kinds of Ethernet nowadays :

- the **''standard'' Ethernet** with a speed limit of 10 Mbps
- the new **Fast Ethernet**, build for networks that need higher transmission speeds, increasing the speed limit to 100 Mbps with only minimal changes to the existing cable structure.
- the **Gigabit Ethernet**, brand new, that supports up to 1Gbps transmission speeds. It is not widely used yet, being too fresh on the market, but, barring imponderables, it should become a standard quickly in the future.

Let's have a look at each of these subtypes.

4.1.1. THE ''VENERABLE'' ETHERNET

- The Ethernet standard was defined by the IEEE (Institute for Electrical and Electronic Engineers) as IEEE Standard 802.3. This standard defines rules for configuring an Ethernet as well as specifying how elements in an Ethernet network interact with one another.
- **Topologies** : Ethernet media are used in two general configurations : bus and star. Examples of bus topology include 10 BASE-2 and 10 BASE-5. 10 BASE-T and 10 BASE-FL Ethernet use a star topology. In all cases, the global length of the network cannot exceed 2500 meters.
- There are four types of Ethernet networks used, the main difference being the kind of cables they use for connecting devices. Those are :
  - 10 BASE-2 : ''Thin Ethernet'', officially called 10 Base-2, is a less expensive version of 10 Base-5 (Thick Ethernet) technology. It uses a lighter and thinner

coaxial cable and dispenses with the external transceivers used with 10 Base-5. Its main technical characteristics is that those cables are noise-resistant.

- 10 BASE-5 : ‘’Thick Ethernet’’, officially known as 10 Base-5, is the oldest form of Ethernet. It was originally developed in the late 1970's by Digital Equipment Corporation, IBM, and Xerox, and became an international standard in 1983. Its heavy shielded cables allow noise immunity. Likewise, it allows distances up to 500 meters (which is why such a configuration is very useful as a backbone technology for wiring together multiple locations within a building without the use of repeaters).

- 10 BASE-T : Ethernet was originally designed to operate over a heavy coaxial cable, and was later updated to also support a thinner, lighter, coaxial cable type. Both systems provided a network with excellent performance, but they utilized a bus topology which made changing a network a difficult proposition, and also left much to be desired in regard to reliability. Also, many buildings were already wired with twisted-pair wire which could support high speed networks. Installing a coaxial-based Ethernet into these buildings would mean they would have to be rewired. Therefore, a new network type known as 10 Base-T was introduced to increase reliability and allow the use of existing twisted-pair cable. This kind of cabling method is not without drawbacks however, for it doesn’t allow distances from the hub to the node superior at 100 meters, and its UTP cables are considerably more electrical noise-sensitive than the coaxial cables.

- 10 BASE-FL : 10 Base-FL is basically a version of Ethernet which runs over fiber optic cable. In physical topology, it is very similar to 10 Base-T. However, with the use of optic cables, it is possible to reach distances up to 2000 meters.

- **Collision Control** : Ethernet introduces a Collision Detection function. Ethernet is a shared media, so there are rules for sending packets to avoid conflicts and protect data integrity. Nodes on an Ethernet network send packets when they determine the network is not in use. It is possible that two nodes at different locations could try to send data at the same time. When both PCs are transferring a packet to the network at the same time, a collision will result. Minimizing collisions is a crucial element in the design and operation of networks, as collisions result in a lot of contention for network bandwidth, thus slowing network performance.

  This is done using the so-called CSMA/CD protocol (standing for Carrier Sense Multiple Access with Collision Detection). The way it works is that any device needing to transmit data to the network must first wait and listen to see if anyone else is transmitting. If the network is clear, then the device can transmit a packet to the network. On the other hand, if the network is not clear, then the device needing to transmit must wait until the transmission in progress ends before starting to send its data.

  It is possible in this protocol to have two devices with transmissions pending at the same time to see that the network is clear and both to begin transmitting. Naturally, this results in both transmissions being garbled, and is called a collision. All versions of Ethernet are designed to detect when this happens, and when it does the transmitting devices stop transmitting, wait a random amount of time, and try again. Since the delay is variable, it is

unlikely for both devices to attempt transmitting again at the same time (one will have to wait longer than the other).

In order for the CSMA/CD protocol to work properly, we have to ensure that the worst-case round-trip signal delay between any two points on the network is not so long that a device can finish transmitting before it can detect any collisions which may occur during the transmission. There is a small delay in the cabling, hubs, and NIC cards which has to be watched out for. This delay is called Propagation Delay and, if not watched out for, can cause a complete breakdown of the CSMA/CD protocol. This will result in a slow, unreliable network.

4.1.2. THE FAST ETHERNET

- Fast Ethernet technology (based on the IEEE standard 802.3u) is the high speed successor to 10 Base-T. This makes it a very attractive technology to use for upgrading existing 10 Mbps Ethernet networks, as it is based on a familiar technology. However, the high speed imposes limits on network design which must be carefully followed to ensure a successful implementation.

- Fast Ethernet cabling requirements vary with the specific type of Fast Ethernet used. There are three such variations in use :

    - 100 BASE-TX for use with level 5 UTP cable,

    - 100 BASE-FX for use with fiber-optic cable,

    - 100 BASE-T4 which utilizes an extra two wires for use with level 3 UTP cable.

  As for now, the 100 Base-TX standard has become the most popular, due to its close compatibility with the 10 Base-T Ethernet standard. In contrast, 100 Base-T4 products are not commonly available, so this document will not address the configuration of 100 Base-T4 networks in depth.

  Fast Ethernet networks are based on a star topology.

- **Basic Operation** : At the MAC (Media Access Control) layer, which controls who transmits data to the network and when they can do it, Fast Ethernet uses the same CSMA/CD protocol that 10 Mbps Ethernet uses, however just with a small variation. The Ethernet standards specify that the shortest transmission unit (packet) allowed on an Ethernet network is 512 bits. Therefore, the delay introduced by the network must be less than the time required to transmit 512 bits. In a traditional 10 Mbps Ethernet, this time is fairly long by computer standards, and can usually be discounted. When we take the network to 100 Mbps, however, this time is only 10% of what it is in a 10 Mbps network, and becomes far more important to network design.

- **Determining Propagation Delay** : As for standard Ethernet, every device or cable run that the Fast Ethernet signal must pass through between the two most distant nodes has a Propagation Delay associated with it. This is a measurement of how much the device or

cable delays the signal. The Propagation Delay is normally measured in a unit called a Bit Time. One Bit Time is defined as the duration of one data bit on the network, in this case 1/100,000,000 second. Since the CSMA/CD protocol requires that the first bit of any transmission reaches the most distant part of the network before the last bit of the transmission is sent, and since the shortest transmission allowed is 512 bits, the network has to guarantee that the absolute worst case delay is less than 512 Bit Times.

One other constraint must be accounted for in the design of a Fast Ethernet network - signal strength. Due to the high frequencies involved in 100 Mbps communications, the maximum distance of any particular cable run is limited to 100 Meters of Category 5 cable, or 2000 Meters of multimode fiber. These distances are the maximum achievable under ideal conditions. It is highly likely that propagation delay concerns will limit these distances considerably.

- **Larger-Scale Fast Ethernet Networks** : The previous sections may lead one to believe that Fast Ethernet networks are not able to be scaled to support a large number of users or cover long distances. In a network built around conventional hubs this is an accurate assessment. However, there are techniques which can be employed to increase the scalability of the network considerably.

  The first, and easiest, solution is to use only one repeater with the number of ports which need to be supported. Generally, this is accomplished by using stackable hubs. Stackable hubs are units which are interconnected with special "stack" ports and cables. All hubs in the same stack become part of one logical repeater, and the stack is counted as only one Class I or Class II repeater regardless of how many hubs are used to build it. For example, a stack consisting of ten hubs, each with twelve ports, behaves just like a single 120 port hub. Stackable hubs are an excellent solution for applications where all devices are within 330 feet of a centralized wiring location.

  The second solution is to use Fast Ethernet switches. Basically, a switch is a multi-port bridge. Each port on the switch is in its own collision domain, and each subnetwork is calculated separately according to the rules above.

  Sometimes a Fast Ethernet requires a long distance to be covered. An example would be interconnecting two buildings in a campus environment. The solution is to use Full Duplex technology and fiber optic cable. At each end, a Fast Ethernet switch is installed, and a fiber optic link is run from switch to switch. Since this link is run only from one point to another, and there are distinct circuits for transmit and receive, there is no risk of collision occurring. Therefore one basically "turns off" CSMA/CD and lets the switches transmit to each other at will. Propagation delay is not a factor in Full Duplex links.

  It is important to note that Full Duplex can not be used in a shared-media environment, such as is created by a hub where there are more than two devices present on one collision domain. Full Duplex can only be run from switch to switch, DTE to DTE, or DTE to switch.

4.1.3. THE GIGABIT ETHERNET

- A recent development in LAN technology is the new Gigabit Ethernet network. This is an even faster fiber-optics-based version of Ethernet, which runs at 1 billion bits per second. Gigabit Ethernet is still in the early stages of development, and at this time the standards are somewhat fluids. Once the technology has evolved in a true standard, it should become more widespread, but, as for now, very few people actually use this mean.

- Gigabit Ethernet is already well into the standard process. In July 1996, a task force was created for defining standard 802.3z, with the following goals in mind :

    - allow half- and full-duplex mode operations at 1000 Mbps rate

    - use the 802.3 frame format

    - use the CSMA/CD method with support for one repeater per collision domain

    - address backward compatibility with 10 BASE-T and 100 BASE-T technologies

- Several types of Gigabit Ethernet have been explored :

    - 1000 BASE-SX : used with lower-cost multimode fiber optic cables in horizontal and shorter backbone applications, over a distance of 260 meters

    - 1000 BASE-LX : is targeted at higher multimode building fiber-optic backbones and single-mode campus backbones, over a distance of 550 meters

    - 1000 BASE-CX : for use over copper cables, support interconnection of equipment clusters where the physical interface is short-haul copper. This standard uses the Fibre Channel-based 8B/10B coding as the serial line rate of 1.25 Gbps and runs over 150-Ohms balanced, shielded cables.

    - Another copper link standard is intended for use in horizontal copper cabling applications. In March 1997, a Project Authorization Request (PAR) was approved by the IEEE Standards Board, enabling the creation of a separate but related committee referred to as the 802.3ab task force. this new group is chartered with the development of a 1000 BASE-T physical layer standard providing 1 Gbps Ethernet signal transmission over four pairs of category 5 UTP cable, covering cabling distances of up to 100 meters or networks with a diameter of 200 meters. This standard will outline communications used for horizontal copper runs on a floor within a building using structured generic cabling, taking advantage of the existing UTP cable already deployed.

- To allow easy migration to higher speed without disrupting all existing networks, Gigabit Ethernet follows the same form, fit and function as its 10 Mbps- and 100 Mbps-forebears. More precisely, it uses the same 802.3 frame format, full-duplex operations, and flow-control methods. In half-duplex mode, Gigabit Ethernet uses the same CSMA/CD access method to resolve the contention between shared media.

  Similar to the case of Fast Ethernet (see above), the only difference with CSMA/CD between Gigabit and standard Ethernet is that it has been enhanced in order to maintain a

200-meter collision diameter at gigabit speeds. To resolve this issue, both the minimum CSMA/CD carrier time and the Ethernet slot time have been extended from their present value of 64 bytes to a new value of 512 bytes. (Note that the minimum packet length of 64 bytes has not been modified). Packets smaller than 512 bytes have an extra carrier extension, while longer packets are left unchanged. These changes, which can have impact on small packets traffic performance, have been offset by incorporating a new feature, called ‘’packet bursting’’, into the CSMA/CD algorithm. Packets bursting will allow servers and other devices to send bursts of small packets in order to fully utilize available bandwidth.

Devices that work in full-duplex mode (switches and buffered distributors) are not subject to the carrier extension, slot time extension, or packet bursting changes. Full-duplex devices will continue to use the regular Ethernet 96-bit interframe gap (IFG) and 64-bit minimum packet size.

4.2. TOKEN RING AND IEEE 802.5

- IEEE 802.5 Standard defines Token Ring, which is a 4 Mbps or 16 Mbps MAC level protocol (there is talk about adding a future 802.5 standard called Dedicated Token Ring, which will upgrade bandwidth up to 32 Mbps). It was originally developed by the IBM Corporation and introduced in the mid 1970’s.

- **Topologies** : IBM Token Ring and IEEE 802.5 are basically quite compatible, although the specifications differ in relative minor ways. One of these ways is the topology, with IBM’s Token Ring network specifying a star, whereas IEEE 802.5 does not specify any particular topology (although virtually all 802.5 implementations are also based on a star).

- **Characteristics :**

    - **Token Passing** :

Both IBM’s Token Ring and IEEE 802.5 are token-passing networks. Token-passing networks move a small frame, called a token, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it simply passes the token to the next end station. Each station can hold the token for a maximum period of time.

If a station possessing the token does have information to transmit, it seizes the token, alters one bit of the token (which turns the token into a start-of-frame sequence), appends the information it wishes to transmit, and finally sends this information to the next station of the ring. While the information frame is circling the ring, there is no token on the network (unless the ring supports early token release, which means : not very often), so other stations wishing to transmit must wait.

The information frame circulates the ring until it reaches the intended destination station, which copies the information for further processing. The information frame continues to circle the ring and is finally removed when it reaches the sending station. The sending

station can check the returning frame to see whether the frame was seen and subsequently copied by the destination.

- **Priority System** :

Token Ring networks use a sophisticated priority system that permits certain user-designated, high-priority stations to use the network more frequently. Token Ring frames have two fields that control priority : the priority field and the reservation field.

Only stations with a priority equal or higher than the priority value contained in a token can seize the token. Once the token is seized and changed into an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network. This s done by putting their priority value into the reservation field of the frame (note that this operation can only be done if the token wasn't already reserved by another node with a higher priority, i.e. if the current value in the reservation field is lower than the node's priority). When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

- **Fault Management Mechanisms** :

Token Ring networks employ several mechanisms for detecting and compensating for network faults. For example, one station in the Token Ring network is selected to be the Active Monitor. This station, which can potentially be any station on the network, acts as a centralized source of timing information for other ring stations and performs a variety of ring maintenance functions. One of these functions is the removal of continuously circulating frames on the ring. When a sending device fails, its frame may continue to circle the ring. This can prevent other stations from transmitting their own frames and essentially lock up the network. The active monitor can detect such frames, remove them from the ring, and generate a new token.

The IBM Token Ring network's star topology also contributes to overall network reliability. All end stations are attached to a device called MSAU (for MultiStation Access Unit). Since all information in a Token Ring network is seen by active MSAUs, these devices can be programmed to check for problems and selectively remove stations from the ring if necessary.

A Token Ring algorithm called ''beaconing'' detects and tries to repair certain network faults. Whenever a station detects a serious problem with the network (such as a cable break), it sends a beacon frame. The beacon frame defines a failure domain, which includes the station reporting the failure, its Nearest Active Upstream Neighbor (NAUN), and everything in between. Beaconing initiates a process called autoreconfiguration, where nodes within the failure domain automatically perform diagnostics in an attempt to reconfigure the network around the failed areas. Physically, the MSAU can accomplish this through electrical reconfiguration.

4.3. FDDI

- FDDI stands for Fiber Distributed Data Interface. FDDI is a 100 Mbps LAN technology which can run over fiber optic or copper cable. It is the oldest 100 Mbps network type commonly available, and is widely used as a backbone technology to interconnect several smaller Ethernet or Token Ring networks. It is also used whenever high reliability and/or high speed are required for a specific application.

- **Basic Topology** : FDDI uses three basic topologies : ring, star and tree (basically, a ‘’star of stars’’). These topologies can be combined to build large networks (up to 500 nodes) which makes better use of the advantages of each component, while limiting each topology’s drawbacks.

- There are four different types of FDDI networks, mostly because they use different kinds of cables. There are:

    - Multimode Fiber Optic Cable : fiber optic cable, usually with a core size of 62.5 microns. It allows distances up to 2000 meters.

    - Singlemode Fiber Optic Cable : fiber optic cable with a core size of 7 to 11 microns. It allows distances up to 10,000 meters.

    - Category 5 UTP : an unshielded copper cable, usually with eight wires. The wires are twisted together in pairs, and the cable is rated at frequencies up to 100 MHz. It allows distances up to 100 meters.

    - IBM Type 1 STP : a heavy, shielded copper cable. It consists of four wires, twisted in to two pairs. Each pair is covered with an individual shield, and an overall shield covers the entire cable. It allows distances up to 100 meters.

  Regardless of cable type, the maximum overall logical ring length of an FDDI network cannot exceed 200,000 meters. It is strongly recommended to keep the actual ring length below 100,000 meters to allow for situations where the primary ring is ‘’wrapped’’ around a break. Wrapping basically doubles the length of the ring. Also, as said earlier, there may be no more than 500 nodes on one ring.

- At the lowest layer, FDDI creates a network comprised of two rings interconnecting all the nodes on the network. Each ring transmits data in a direction opposite to the other one. These rings are logical in nature, and exist regardless of how the network is physically connected together.

  The reason for having two rings is **fault tolerance**. Most of the time, the primary ring carries the data and the secondary ring is idle. In the event of a break in the ring, the nodes nearest the break will loop the primary ring to the secondary ring, which bypasses the fault and results in an unbroken ring.

  Note that FDDI also offers the ability to use both rings for data transmission at the same time. This feature boosts the network speed to 200 Mbps. In the event of a fault, the secondary ring will revert to its previous function, and the overall network speed will drop to 100 Mbps.

FDDI uses a token passing protocol which is similar, but not identical to, Token Ring. In such an arrangement a special type of packet called a token is sent around the network. Any node which wishes to transmit data to the network first captures the token, sends a packet of data to the network, then it releases the token. Every station on the network will receive the transmission and repeat it. If a station receives a transmission addressed to it, it will mark the transmission as received and repeat it to the network. The transmission will travel around the ring until it is received by the station which originally sent it, which removes it from the ring. If a station does not receive its transmission back, it assumes that an error occurred somewhere.

- **Backbone Uses** : Most networks users do not require the high speed (and associated high cost) of a full-blown FDDI network. However, sometimes an organization will have such a large number of users that the aggregate bandwidth needed is far more than a 10 Mbps Ethernet or 16 Mbps Token Ring network can provide. One solution for this type of problem is to employ a combination of Switching and FDDI technologies to create a high speed backbone interconnecting a large number of small Ethernet or Token Ring networks.

  A network of this type allows taking advantage of the speed and reliability of FDDI where it is needed (at the servers) while protecting an existing investment in 10 Base-T technology. None of the workstations on the network need to be upgraded at all to take advantage of the increased aggregate bandwidth available. They are simply split into smaller groups, with each group having 10 Mbps of speed dedicated for its exclusive use. For example, if we have five groups, then there will be an aggregate of roughly 50 Mbps of bandwidth available where there was previously only 10 Mbps.

  Note that this type of solution will eliminate network slowness due to congestion, or too many nodes attempting to use 10 Mbps of bandwidth at once. If the application demands more than 10 Mbps at any workstation (for example, huge CAD files), then simply put an FDDI card in that workstation and connect it directly to one or both FDDI concentrators. This feature allows balancing cost with the actual need for speed and security of each station on the network.

5. CHARACTERISTICS OF WANS

WANs are networks that span a much larger area than the previously studied LANs. As such, their cost is far greater, which explains why their performances are far less outstanding than their smaller brethren. In order to make good use of these less-than-ideal characteristics, and to meet the unique requirements of a larger population of users, network designers have then created new protocols, that differ significantly in working from those we have seen in the previous LANs section.

5.1. X.25

- X.25 Packet Switched networks allow remote devices to communicate with each other across high speed digital links without the expense of individual leased lines. Packet Switching is a technique whereby the network routes individual packets of HDLC (High-level Data Link Control) protocol data between different destinations based on addressing within each packets.

- The connection-oriented protocol known as X.25 encompasses the first three layers of the OSI architecture. Most generally, the **architecture** associated with X.25 is as follows :

    - Physical Layer : includes several standards such as V.35, RS232 and X.21, concerned with electrical or signaling.

    - Data Link Layer : implementation of the ISO HDLC standard called Link Access Procedure Balanced (LAPB), provides an error-free link between two connected devices.

    - Network Layer : referred to as the X.25 Packet Layer Protocol (PLP) (only in the case of X.25, of course), primarily concerned with networking routing functions and the multiplexing of simultaneous logical connections over a single physical connection.

- Connections occur on **logical channels** of two types :

    - Switched Virtual Circuits (SVCs) : SVCs are connection-oriented, with call setup; a connection is established, data is transferred and then the connection is released. Each station on the network is given a unique address for use in connection.

    - Permanent Virtual Circuits (PVCs) : a PVC is similar to a leased line in that the connection is always present. The logical connection is established permanently by the Packet Switched Network administration. Therefore, data can always be sent, without any call setup.

- X.25 provides a virtual high quality digital network at low cost. Another useful feature is **speed matching** : because of the store-and-forward nature of Packet Switching, plus excellent flow control, users do not have to use the same speed. For example, a host connected at 56 Kbps can communicate with numerous remote sites connected with

cheaper 19.2 Kbps lines. X.25 is also well debugged and stable, so there are literally no data errors on modern X.25 networks.

- X.25 has some **drawbacks** :

  - There is an inherent delay caused by the store-and-forward mechanism. On most single networks the turn-around delay is about 0.6 seconds. This has no effect on large block transfers, but in flip-flop types of transmissions the delay can be very noticeable. Frame Relay (also called Fast Packet Switching, see below) does not store and forward, but simply switches to the destination part way through the frame, reducing the transmission delay considerably.

  - Another problem for these networks is a large requirement for buffering to support the store-and-forward data transfer. One of the reasons that Frame Relay is so cost-effective is that storage requirements are minimal.

5.2. FRAME RELAY

- Frame Relay is a fast packet switching technology which contains little functionality, and is thus potentially very fast, even if most functions are implemented in software. Frame Relay has been developed and implemented under two assumptions :

  - the underlying transmission infrastructure (i.e. cables) has a high quality, i.e. produces only very few errors (with a Bit Error Rate of 10^{-8} to 10^{-10})

  - the end-devices have a high degree of intelligence, i.e., if errors occur during transmission, the end-devices must recover from that error situation.

  Frame Relay is used mostly to route protocols such as IPX or TCP/IP. It can also be used to carry asynchronous traffic, SNA or even voice data. Its primary competitive feature is its low cost. In North America, it is fast taking on the role that X.25 has had in Europe : the most cost effective way to hook up multiple stations with high speed digital links.

- Frame Relay is a pure layer 2 protocol. Many of the functions that are typical of an OSI layer 2 protocol (such as they are implemented in the HDLC protocol) have not been implemented in Frame Relay.

- Frame Relay has the following characteristics :

  - **No flow control** : if the frame relay switch becomes congested due to high traffic load, frames that are marked as ‘discard eligible’ will be discarded. These are usually frames that exceed the CIR (Committed Information Rate) that has been agreed upon between the end-user and the frame relay service provider.

  - **No error recovery** : the frame relay protocol contains an error detection scheme (which consists of checking the FCS field). However, if a frame appears to have an error, it will be simply discarded and no further action will be taken from the frame

relay network. The missing of the frame and the following error recovery has to be performed by the end-devices (e.g. by TCP).

- **Very little congestion control** : the frame relay protocol specifies the optional usage of a BECN (Backward Explicit Congestion Notification) and a FECN (Forward Explicit Congestion Notification) bit. The BECN bit notifies the sender about a potential congestion situation at the frame really switch, and the FECN bit plays the same role for the receiver. However, the end-devices can choose to entirely ignore these 'markers'. It is important to notice that the FECN and BECN bits can only be used in conjunction with other user traffic. This is particularly crucial in the case of the BECN, which requires user data also to flow in the opposite direction to the sender's data.

- **Status polling** : the Frame Relay Customer Premises Equipment (CPE) polls the switch at set intervals to find out the status of the network and DLCI connections. A Link Integrity Verification (LIV) packet exchange takes place about every 10 seconds, which verifies that the connection is still good. It also provides information to the network that the CPE is active, and this status is reported at the other end. About every minute, a Full Status (FS) exchange occurs, which passes information on which DLCIs are configured and active. Until the first FS exchange has occurred, the CPE does not know which DLCIs are active, and so no data transfer can take place.

  There exist various standards for the Status Polling function. The oldest, the Link Management Interface (LMI), was a temporary standard adopted by manufacturers prior to the apparition of the official standard ANSI T1.617 Annex D, and it has since acquired a life of its own. A newer standard, Q.933, has also been approved, largely to accommodate Switched Virtual Circuits.

- As there are no guarantee of data integrity and flow control, Frame Relay is a much faster protocol than its close relative X.25 and as such minimizes network buffering. However, this same lack of supervision means that it is imperative to run an upper layer protocol above Frame Relay that is capable of recovering from errors, such as HDLC, IPX or TCP/IP. It is to be noticed that in practice, the ''reliable transmission infrastructure'' assumption, on which Frame Relay is based, is quite correct.

5.3. ATM

- The requirements of modern networking involve:

  - Handling multiple types of traffic (voice, video, data), all with individual characteristics that make very different demands (sometimes downright opposed to each other) of the communications channel

  - A fair and equitable way of charging for transport services, to provide the user with economically priced access, and the carrier with a profitable return on investment

- Reliability and flexibility of the communications links

- Ensuring accessibility to network capacity for both existing and future equipment and services with minimal disruption in existing operations

Asynchronous Transfer Mode (or ATM for short) is a very good answer to meet these demands.

- Let's examine the different types of traffic and their demands on a communication channel :

  - **Voice** :
    + its generation is asynchronous (a speaker may speak anytime)
    + its transmission must be synchronous (once the message starts, it must flow continuously as it is spoken)
    + the bandwidth required for a voice conversation in digital communication is relatively small and constant (64 Kbps)
    + the signals may contain a high degree of error and the information can still be retrieved correctly

  - **Video** :
    + the generation is synchronous (continuous)
    + its transmission is synchronous
    + the bandwidth required is variable and it could range from under 64 Kbps to several Mbps in the same session
    + humans require 25-30 images per second, but there is cases (where there are some motion recorded) where there is a tremendous amount of information to be sent in a awfully short time; some other times, only very small changes between consecutive screens need to be transmitted
    + error control should be tight, otherwise the wrong information on the monitor may trigger severe wrongful actions (security misinformation, wrong reaction of robots, ...)

  - **Data** :
    + its generation could be either asynchronous (text) or synchronous (telemetry)
    + its transmission in general can be asynchronous (data typically can wait patiently in buffers), so no special timing relationship between the transmitter and the receiver is required
    + the amount of bandwidth varies enormously from a few bits per second to billions of bits per second
    + the information is extremely error-sensitive, so extreme caution must be exercised in transmission and error control must be very tight

  The biggest problem in telecommunications nowadays is not only speed but that transmissions occur at statistically random intervals. Therefore, ideally, the data communication channel should be as flexible as possible to allow bursts to take place without the obligation of the user to purchase committed bandwidth to handle the peak.

  Traditionally, the 'bandwidth on demand' problem was solved with technologies like X.25 or the more recent Frame Relay. Both employ packet switching techniques that allow

variable-length data packets to be framed for error protection and then sent over links that are statistically shared (multiplexed) between various users. This way, billing can be done on a per-frame basis, rather than based on time of link utilization.

X.25 was conceived in an era (late ‘60s) where speeds were low and lines were of poor quality (analog). It’s therefore a very rigorous technology when it comes to error control, but X.25 has enough overhead to be useful only at relatively low link speeds (up to 64 Kbps).

Frame Relay was derived from X.25 to accommodate modern data networks. Most of X.25’s error control capabilities were removed and, instead, the speed was boosted to T1/E1 (1.544 Mbps/2.048 Mbps), with the possibility to be run at even faster rates (T3/SONET). Modern lines are digital, with bit error rates that can be brought under one in a billion (usually 10^{-8} to 10^{-10}). Also, the former terminals are discarded in favor of powerful PCs and workstations capable of running sophisticated error control software. Should a frame be corrupted in any way, a frame relay node may discard it without fear that the missing data might not be recovered. That’s why the frames are passed in a relay fashion, very fast, from switch to switch, with only three questions asked:

- is the routing information in the frame intact ?
- is the Data Link Connection Identifier (DLCI) on the list of known DLCIs ?
- is the node congested, and if so, is the frame eligible for discard ?

Should the answer to any of these questions be conductive to discarding the frame, that action is taken and no notification about it takes place. It’s a simple and efficient technology to carry data over clean lines.
Frame Relay frames have very little overhead (seven bytes, for hundred of data bytes). However, because frame lengths vary, their transit through the switch ports suffers variable delays. Therefore, mixing data, voice and video is not recommended. There are some solutions, especially in private networks; but typically it is agreed that frame relay (as well as X.25) is a technology good for data, especially LAN-to-LAN communications.

So if one wants to combine data, voice and video on the same links, one solution is to use fixed and relatively short packets. This way, the delays produced by each packet are going to be short and probably fixed; so, if voice and video traffic can be assured priority handling, they can be mixed with data without diminishing any reception quality. This is where ATM fits. ATM is a transmission technology that uses fixed-size packets called cells. A cell is a 53 bytes packet with 5 bytes of header/descriptor and 48 bytes of payload (voice, data, video).

- **Standards** exist that define switched (on demand) ATM service. The networking scheme for this type of service is known as B-ISDN (Broadband ISDN) and it works conceptually very similar to ISDN.

  In ISDN, signaling (requesting services from the network) is handled via a dedicated data path called the D channel. In B-ISDN, signaling is handled through a common Virtual Circuit (reserved for each user) called the 'metasignaling' channel. The standard that shows how to signal is called Q.93B and is an ITU-TSS standard.

Here is a list of the various standards related to ATM :

- Q.93B - TSS standard for signaling
- UNI 3.0 ILMI (Interim Link Management Interface) ATM Forum Signaling Standard for the User-to-Network Interface
- I.610 - TSS Management Specification (alarms)
- 1.555 - TSS - ATM/FRAME RELAY Internetworking
- RFC 1577 Internet - IP over ATM

- **Characteristics :** ATM (also known as 'Cell relay') does not protect data from errors (like Frame Relay). ATM relies on user equipment error control, and therefore works well on digital lines with low bit error rates. The cells for one user are transmitted over a Permanent or Switched Virtual Circuit Connection (PVC or SVC) just like in Frame Relay or X.25. However, while in those two technologies the circuit had to be built in its entirety at subscription time (PVCs) or at connection time (SVCs), in ATM, the paths over which various circuit connections are made are prebuilt. This saves processing time both at the user-to-network interface (UNI) and network-to-network interface (NNI). This also goes hand-to-hand with the carrier system of choice for ATM, namely SONET (Synchronous Optical Network), which defines user paths using its lines and sections of fiber.

  The cell header contains (among other things) a field of eight bits to define possible virtual paths at the UNI (256 physical destinations) and a field of sixteen bits to define up to 65,536 virtual circuits on each path. At the NNI (in the network -- between switches) the number of virtual paths is increased to 4,096 (twelve bits) because it is assumed that the network will carry many more than just one user. The two fields are called VPI (Virtual Path Identifier) and VCI (Virtual Circuit Identifier) respectively. A Virtual Connection (VC) will carry data, voice or video (not all simultaneously). Each type of traffic requires at least one unique VC.
  The VC passes through nodes and over links/trunks, and at each node port it has buffers allocated for transmit and receive. The buffers are identified by VCIs and VPIs unique to the trunk. The VC therefore has a number of attributes which describe its usage :

  - whether permanent or switched
  - VCI/VPIs assigned to it at various nodes and at the UNI interface
  - A 'sustainable cell rate' which basically states how many cells the user can send at any 'committed time' over that VC.

  The concepts are very similar to Frame Relay (or, for that matter, X.25). The 'sustainable cell rate' or SCR is called Committed Information Rate (CIR) in Frame Relay and Throughput Class in X.25. The VCI/VPI Cell Relay is what DLCI is in Frame Relay and LCN (Logical Channel Number) in X.25. Metering in ATM is done using a 'leaky bucket' or 'virtual schedule' algorithm, just like in Frame Relay. Every VC is given a timed buffer at every switched port. The sustainable cell rate (SCR) is the ratio between the Committed

Burst of Cells (Bc) and the time interval during which no more than Bc cells may be sent (Committed Time, Tc): SCR = Bc/Tc.

Typically, the customer may exceed Bc by an Excess Burst of cells (Be) during the same time (Tc), by using a second buffer of the same size as the first one. However, cells that exceed Bc are at risk: they are eligible for discard. There is a bit in the header called CLP (Cell Loss Priority) which, when set, indicates that the cell can be discarded by a congested node (similar to the Discard Eligibility bit of Frame Relay).

With virtual scheduling, a clock ticks at every node. The ticks act as a sort of pace counter, or metronome beat. Cells coming on or after the tick are not eligible for discard. Cells that come before the tick are arriving too soon, and so are eligible for discard.

In addition to these (VPI, VCI, CLP), the cell header contains a 'payload type' that describes what kind of information this cell carries - data or management (Operation And Maintenance - OAM) and whether the VC carrying it is congested or not, and a HEC or Header Error Control field which, with eight bits, provides enough redundancy to allow Forward Error Correction up to one bit. This makes the loss of cells due to errors less likely. The HEC is also used for synchronization: switches learn what a cell period is by continuously identifying good HECs.

Finally, there is a difference between the numbers of VPIs at UNI (User to Network Interface) and at NNI (Network to Network Interface). The UNI cell uses only eight bit and the NNI uses twelve bits for the VPI. At UNI the four-bit difference makes up a field called Generic Flow Control (GFC). It is 'generic' because each piece of equipment may use it as it pleases (although so far no definition has been given). For example, if all GFC bits are set to '0' that could indicate a ''no-congestion'' report, in which case the user can safely transmit, or that could mean the contrary, in which case the user is prohibited to emit due to the congestion.

- **Different Traffic** : Some cells will likely get lost due to noise or equipment failure, others due to congestion. Therefore, various types of traffic generators with their different requirements have to carefully prepare or 'adapt' their messages for travel over the ATM network. This is done in each case by a piece of software or firmware called AAL (ATM Adaptation Layer).

  The AAL has two stages:

  - A service (or traffic type) -dependent sublayer called Convergence Sublayer (CS); and

  - A service-independent Segmentation And Reassembly (SAR) sublayer

  The CS assures the necessary error control and sequencing as well as the sizing of information. The SAR then chops the CS message into the 48-byte payload packets and attaches them to the five-byte header. There are five types of adaptation layer services, designated AAL1, AAL 2, etc. At the transmit node AAL1 prepares voice traffic, AAL2 prepares video traffic, AAL3 and AAL5 prepare connection-oriented. AAL3 has been designed by a committee of the ITU-TSS (formerly CCITT), while AAL5 (also known as SEAL, for Simple and Efficient Adaptation Layer) is the creation of the ATM Forum (an

organization of users, manufacturers and carriers of ATM). AAL4 prepares connectionless data (SMDS or LAN-like) for cell relay switching. After the preparation stage, the message is delivered to the segmentation layer, where the cells are created and sent.

At the receive side the cells go through the reassembly layer and are passed to AAL1, 2, 3, 4, or 5 for the recreation of the original message. This message is then delivered to the video monitor, the voice receiver or the data process expecting it.

As said above, each AAL is intended for a specific purpose, therefore, their characteristics vary widely depending of the kind of data they code :

- AAL1 is intended for voice traffic. Since voice traffic is error tolerant, no error control (CRC) is required. However, what is important in the case of voice transmission is that cells are received in the exact sequence in which they were sent, and that they arrive at a constant rate. AAL1 assures sequence numbers. Also, one 48 byte cell may carry eight-bit voice samples from more than one source.

  Since voice is transmitted synchronously, without delay, it is possible that by the time the voice transmitter has sent a few samples, the cell must leave partially empty. This type of service is called 'Streaming Mode Service'. AAL1 is designed to handle this: it inserts sequence numbers in cells and identifies what portion of the cell carries voice and what portion carries nothing.

- With video, not only do we need synchronous and sequencing, but we also need error checking codes (CRCs). And, since a screen may have a lot of pixel information, many cells may have to be used to transmit the whole screen; so we need to know where the screen starts and where it ends. That's why the cells are labeled as 'the first' or 'intermediate one' or 'the last'. This way AAL2 can assure bandwidth-on-demand with a variable rate.

- Data transmission is of two kinds:
  + Connection-oriented: Before actually sending data, the calling side must first establish a 'circuit' or a 'connection' with the called node (just like in telephony)
  + Connectionless: A piece of data is 'thrown' in the network with a destination address in it and it arrives at the destination. This kind of service is also known as 'datagram' service and is like the letter delivery performed by the postal service.

- AAL3 and 5 are designed for connection-oriented service and AAL4 for datagrams. Like AAL2 for video, both data services require error checking (CRC), sequencing, and identification of the cells as part of the message. In addition, some sort of indication has to be given to the receiver about the total length of the message, so an appropriate buffer size can be reserved for the message.

- AAL3 is very similar to AAL2. The difference is in timing (AAL3 does not require synchronism between receiver and transmitter).

- AAL4, in addition, must identify each cell as belonging to one datagram. So each cell is given a 'Multiplex Identifier' (a ten-bit field) for this purpose.

- AAL5 does not bother to insert all this extraneous information into each cell. Instead, before the TCP or some other data message is chopped into cells, a 'trailer' is appended to that message, containing a 'length' indicator of two bytes (TCP segments can be 65,536 bytes long), a CRC error checking code for the whole message, and some bits signaling user-to-user (end-to-end) what this message is about (this is still under study and is to be used by each user equipment as it sees fit). Then this 'adapted' message is put through the chopper and the cells are sent.

The data sources (bridges, routers) or the video and voice sources (like PBXs) run their usual software (for example TCP/IP). A special interface sends the message to be adapted to a specialized box generally called an ATM CSU/DSU (Customer Service Unit/Digital Service Unit). This box contains the Adaptation Layer and the Segmentation and Reassembly Layer, and the fiber or wire or microwave interface to SONET, T1 or T3 lines.

The interface between the Router and the ATM CSU must be fast and must ensure data transfer protection (error and flow control). There are a number of solutions for this :

- DXI (Digital Transmit/Receive Interface) is based on the HDLC data communication protocol. It operates physically over a serial link (V.35, RS422 or HSSI). Whenever messages are to be sent, the link quality is tested; then the data is passed with an indication as to what Virtual Circuit/Path Identifier to use. The Circuit Identifier is mapped in the DSU to a specific type of AAL. There are three modes of operation for DXI:
  + Mode 1A is very similar to the PPP (Point-to-Point Protocol) and supports AAL5, with up to 1,023 connections and up to 9,232 octets of payload
  + Mode 1B supports AAL 3/4 and 5 and allows 1,023 VCIs, with messages of 9,232 octets for AAL5 and 9,224 octets for AAL3/4
  + Mode 2 provides for larger message sizes (65,535 octets) and more VCI/VPI connections (16,777,215 VCIs and 256 VPIs)

One other thing that the DXI sender (the router part) is setting is the CLP (Cell Loss Priority) bit that indicates whether the cells that will be generated by this message are eligible for discard or not.

- TAXI (Transparent Asynchronous Receiver/Transmitter Interface) is an FDDI (Fiber Distributed Data Interface) access protocol that is used to send cells over private fiber networks (100 MBPS).

- ATMR (ATM Ring) is a Token Ring (full duplex) interface for fiber or shielded wire. Since ATM switches are fast and have a large throughput, multiport switches can be used as LAN emulators with broadband capabilities (multiple connections can exist simultaneously).

Departmental LANs are hooked up through bridges and routers to a central hubbing switch on ports that talk to these routers via ATM DSUs using DXI or via TAXI or ATMR interfaces.

5.4. IP NETWORKS

Unarguably the most popular and used kind of network, IP-based networking is dominating the communication field nowadays and is still growing. As such, it is one of the most important terrestrial network model in our simulation.

5.4.1. INTERNET PROTOCOL (IP)

- Perhaps the most famous network protocol to date, IP has become the de facto standard for most networking issues, including WANs.

- Its **characteristics** are well known and defined :

    - **Fragmentation and Reassembly** :

    One of the IP's tasks is to break outgoing messages into chunks that fit within the predetermined size of a datagram and to reassemble a message that arrives in several datagrams into the original longer message. The size of data that can fit into an IP datagram depends on a number of factors, including the type of TCP/IP software in use. The specifications for IP provide a maximum packet size of 65, 535 bytes, but the typical datagram is much smaller than that (in many installations, a datagram is about a kilobyte or two in length). Thus, the need for breaking up a message is important and it has to be done properly to ensure the data gets through without corruption.

    IP handles the fragmentation automatically when it encounters a too large message; each part of the message is bundled into its own datagram, and the IP header is attached with details about the number of that datagram in the larger message. The datagrams that hold the message are then sent out over the network. Part of the process of reassembly begins when the first datagram is received by the destination. Remember first that there is no guarantee that the datagrams will arrive in the proper sequential order. So part of this reassembly job is to put them all back in the proper order and ensure that no one is missing. For this, IP is configured to enable a certain amount of time for all the message's datagrams to arrive, if there is still some missing after the preconfigured time has expired, all the datagrams received to that point are discarded and the transmission is declared a failure.

    The biggest problem with fragmentation and reassembly in IP is then obvious : the larger the message is, the more datagrams needed to hold it, and the less likely the message will be received properly. For this reason, many applications try to minimize the amount of fragmentation involved with messages.

    - **Handling Delivery Problems** :

    The IP layer has absolutely nothing to do with the actual transmission and reception of datagrams over a network, so IP can do nothing to physically ensure reliability of transmission. IP can't even verify that the contents of the datagram have been received without modification (in fact, IP does maintain a checksum but only for the IP header

itself). Other protocol layers (i.e. higher layers usually) have to deal with verifying the datagram's real contents.

All IP can do is to send error message once an error has been detected. The error-reporting system in IP is called ICMP, a protocol that must be included with every IP implementation. Any message generated by ICMP is treated like any other datagram by other layers of the TCP/IP stack and the network, as such each ICMP message has a header constructed exactly the same way as a standard datagram (although the content of the message is destined for the ICMP routines of the IP layer, of course).

- **IP Next Generation** : due to the limitations imposed by IPv4 32-bit addresses, a new version of IP called IP Next Generation (or more commonly IPv6) has been implemented. Though it is still a bit in the development phase, some characteristics can already be sorted out :

    - **Expanded Routing and Addressing Capabilities** : IPng increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy and a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses.

    - A new type of address called a "**anycast address**" is defined, to identify sets of nodes where a packet sent to an anycast address is delivered to one of the nodes. The use of anycast addresses in the IPng source route allows nodes to control the path which their traffic flows.

    - **Header Format Simplification** : Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to keep the bandwidth cost of the IPng header as low as possible despite the increased size of the addresses. Even though the IPng addresses are four time longer than the IPv4 addresses, the IPng header is only twice the size of the IPv4 header.

    - **Improved Support for Options** : Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

    - **Quality-of-Service Capabilities** : A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real- time" service.

    - **Authentication and Privacy Capabilities** : IPng includes the definition of extensions which provide support for authentication, data integrity, and confidentiality. This is included as a basic element of IPng and will be included in all implementations.

5.4.2. USER DATAGRAM PROTOCOL (UDP)

- UDP is a standard protocol with STD number 6. UDP is described by RFC 768 - User Datagram Protocol. Its status is recommended, but in practice every IP-network implementation which is not used exclusively for routing will include UDP.

- UDP is basically an application interface to IP. It adds no reliability, flow-control or error recovery to IP, and is connectionless. It simply serves as a ‘‘multiplexer/demultiplexer” for sending and receiving datagrams, using ports to direct the datagrams.

  UDP provides a mechanism for one application to send a datagram to another. The UDP layer can be regarded as being extremely thin and consequently has low overheads, but it requires the application to take responsibility for error recovery and so on.

- **Ports** : Applications sending datagrams to a host need to identify a target which is more specific than the IP address, since datagrams are normally directed to certain processes and not to the system as a whole. UDP provides this by using ports. A port is a 16-bit number which identifies which process on a host is associated with a datagram. There are two types of port :

  - **well-known** : Well-known ports belong to standard servers, for example TELNET uses port 23. Well-known port numbers range between 1 and 1023 (prior to 1992, the range between 256 and 1023 was used for UNIX-specific servers). Well-known port numbers are typically odd, because early systems using the port concept required an odd/even pair of ports for duplex operations. Most servers require only a single port. One exception is the BOOTP server which uses two: 67 and 68.

    The reason for well-known ports is to allow clients to be able to find servers without configuration information.

  - **ephemeral** : Clients do not need well-known port numbers because they initiate communication with servers and the port number they are using is contained in the UDP datagrams sent to the server. Each client process is allocated a port number as long as it needs it by the host it is running on. Ephemeral port numbers have values greater than 1023, normally in the range 1024 to 5000. A client can use any number allocated to it, as long as the combination of <transport protocol, IP address, port number> is unique.

    Note: TCP also uses port numbers with the same values. These ports are quite independent. Normally, a server will use either TCP or UDP, but there are exceptions. For example, Domain Name Servers (see Domain Name System (DNS)) use both UDP port 53 and TCP port 53.

Be aware that UDP and IP do not provide guaranteed delivery, flow-control or error recovery, so these must be provided by the application.

5.4.3. TRANSMISSION CONTROL PROTOCOL (TCP)

- TCP is a layer-four protocol defined in RFC 793. Its status is recommended for use with IP, but not mandatory (this explains why one can find UDP/IP networks). Nevertheless, TCP/IP are the most common combination around.

- TCP provides considerably more facilities for applications than UDP, notably error recovery, flow control and reliability. TCP is a connection-oriented protocol (unlike UDP which is connectionless). Most of the user application protocols, such as Telnet or FTP, use TCP.

- **Sockets** : Two processes communicate via TCP sockets. The socket model provides a process with a full-duplex byte stream connection to another process. The application need not concern itself with the management of this stream; these facilities are provided by TCP.

  TCP uses the same port principle as UDP (see above) to provide multiplexing. Like UDP, TCP uses well-known and ephemeral ports. Each side of a TCP connection has a socket which can be identified by the triple <TCP, IP address, port number>. This is also called a half-association. If two processes are communicating over TCP, they have a logical connection that is uniquely identifiable by the two sockets involved, that is by the combination <TCP, local IP address, local port, remote IP address, remote port>. Server processes are able to manage multiple conversations through a single port.

- **TCP Concept** : As noted above, the primary purpose of TCP is to provide reliable logical circuit or connection service between pairs of processes. It does not assume reliability from the lower-level protocols (such as IP) so TCP must guarantee this itself.

  TCP can be characterized by the following facilities it provides for the applications using it :

  - **Stream Data Transfer** : From the application's viewpoint, TCP transfers a contiguous stream of bytes through the internet. The application does not have to bother with chopping the data into basic blocks or datagrams. TCP does this by grouping the bytes in TCP segments, which are passed to IP for transmission to the destination. Also, TCP itself decides how to segment the data and it may forward the data at its own convenience.

    Sometimes, an application needs to be sure that all the data passed to TCP has actually been transmitted to the destination. For that reason, a push function is defined. It will push all remaining TCP segments still in storage to the destination host. The normal close connection function also pushes the data to the destination.

  - **Reliability** : TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. As the data is transmitted in blocks (TCP segments) only the sequence number of the first data byte in the segment is sent to the destination host.

    The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments.

- **Flow Control** : The receiving TCP, when sending an ACK back to the sender, also indicates to the sender the number of bytes it can receive beyond the last received TCP segment, without causing overrun and overflow in its internal buffers. This is sent in the ACK in the form of the highest sequence number it can receive without problems. This mechanism is also referred to as a window-mechanism and we will discuss it in more detail later in this chapter.

- **Multiplexing** : Is achieved through the use of ports, just as with UDP.

- **Logical Connections** : The reliability and flow control mechanisms described above require that TCP initializes and maintains certain status information for each ‘‘data stream”. The combination of this status, including sockets, sequence numbers and window sizes, is called a logical connection. Each connection is uniquely identified by the pair of sockets used by the sending and receiving processes.

- **Full Duplex** : TCP provides for concurrent data streams in both directions.

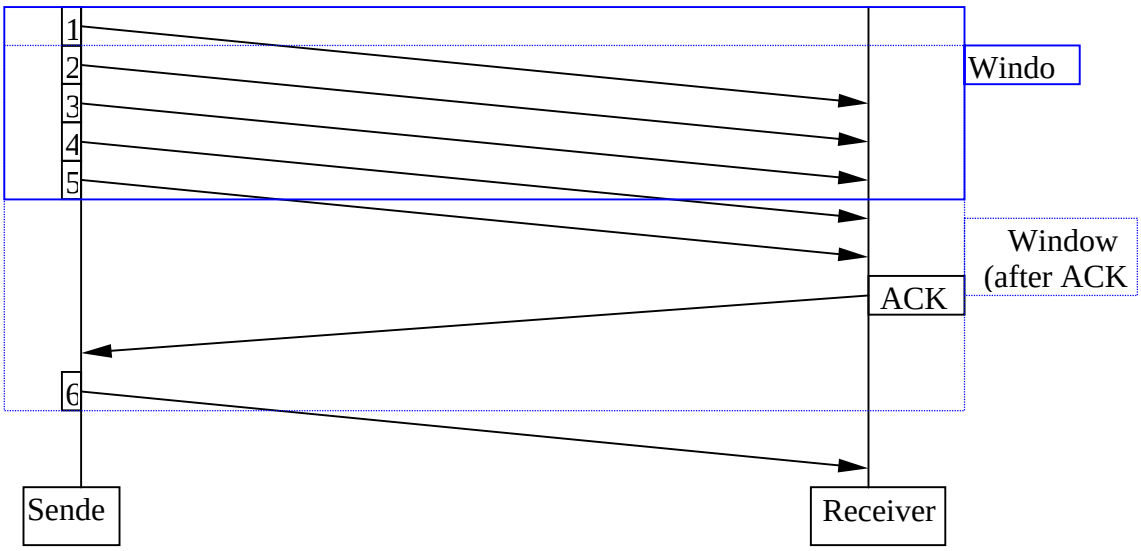
- **The Window Principle** : A simple transport protocol might use the following principle: send a packet and then wait for an acknowledgment from the receiver before sending the next packet. If the ACK is not received within a certain amount of time, retransmit the packet. While this mechanism ensures reliability, it only uses a part of the available network bandwidth.

  Consider now a protocol where the sender groups its packets to be transmitted and uses the following rules:

  - The sender may send all packets within the window without receiving an ACK, but must start a timeout timer for each of them.

  - The receiver must acknowledge each packet received, indicating the sequence number of the last well-received packet.

  - The sender slides the window on each ACK received.



At the moment the sender receives the ACK 1 (acknowledgment for packet 1), it may slide its window to exclude packet 1. At this point, the sender may also transmit packet 6.

Imagine some special cases: Packet 2 gets lost: the sender will not receive an ACK 2, so its window will remain in the position 1 (see figure above). In fact, as the receiver did not receive packet 2, it will acknowledge packets 3, 4 and 5 with an ACK 1, since packet 1 was the last one received ‘‘in sequence”. At the sender’s side, eventually a timeout will occur for packet 2 and it will be retransmitted. Note that reception of this packet by the receiver will generate an ACK 5, since it has now successfully received all packets 1 to 5, and the sender’s window will slide four positions upon receiving this ACK 5. Packet 2 did arrive, but the acknowledgment gets lost: the sender does not receive ACK 2, but will receive ACK 3. ACK 3 is an acknowledgment for all packets up to 3 (including packet 2) and the sender may now slide his window to packet 4.

This window mechanism ensures:

- Reliable transmission.
- Better use of the network bandwidth (better throughput).
- Flow-control, as the receiver may delay replying to a packet with an acknowledgment, knowing its free buffers available and the window-size of the communication.

- **The Window Principle Applied to TCP** : The above window principle is used in TCP, but with a few differences:

  As TCP provides a byte-stream connection, sequence numbers are assigned to each byte in the stream. TCP divides this contiguous byte stream into TCP segments to transmit them. The window principle is used at the byte level; that is, the segments sent and ACKs

received will carry byte-sequence numbers and the window size is expressed as a number of bytes, rather than a number of packets.

The window size is determined by the receiver, when the connection is established, and is variable during the data transfer. Each ACK message will include the window-size that the receiver is ready to deal with at that particular time.

Remember that TCP will block bytes into segments, and a TCP segment only carries the sequence number of the first byte in the segment.

- **Acknowledgments and Retransmissions** : TCP sends data in variable length segments. Sequence numbers are based on a byte count. Acknowledgments specify the sequence number of the next byte that the receiver expects to receive.

  Now suppose that a segment gets lost or corrupted. In this case, the receiver will acknowledge all further well-received segments with an acknowledgment referring to the first byte of the missing packet. The sender will stop transmitting when it has sent all the bytes in the window. Eventually, a timeout will occur and the missing segment will be retransmitted.

  Each TCP should then implement an algorithm to adapt the timeout values to be used for the round trip time of the segments. To do this, TCP records the time at which a segment was sent, and the time at which the ACK is received. A weighted average is calculated over several of these round trip times, to be used as a timeout value for the next segment(s) to be sent.

  This is an important feature, since delays may be variable on an internet, depending on multiple factors, such as the load of an intermediate low-speed network or the saturation of an intermediate IP gateway.

- **Establishing a TCP Connection** : Before any data can be transferred, a connection has to be established between the two processes. One of the processes (usually the server) issues a passive OPEN call, the other an active OPEN call. The passive OPEN call remains dormant until another process tries to connect to it by an active OPEN.

  On the network, three TCP segments are exchanged : Connection Request, Connection Acknowledged and Connection Confirmed. This whole process is known as three-way handshake. Note that the exchanged TCP segments include the initial sequence numbers from both sides, to be used on subsequent data transfers.

  Closing the connection is done implicitly by sending a TCP segment with the FIN bit (no more data) set. As the connection is full-duplex (that is, we have two independent data streams, one in each direction), the FIN segment only closes the data transfer in one direction. The other process will now send the remaining data it still has to transmit and also ends with a TCP segment where the FIN bit is set. The connection is deleted (status information on both sides) once the data stream is closed in both directions.

5.4.4. IP OVER ATM

ATM-based networks are of increasing interest for both local and wide area applications. There are already some products available to build your physical ATM network. The ATM architecture is new and therefore different from the standard LAN architectures. For this reason, changes are required so that
traditional LAN products will work in the ATM environment. In the case of TCP/IP the main change required is in the network interface to provide support for ATM.

There are several approaches already available, two of which are important to the transport of TCP/IP traffic. They are described in Classical IP over ATM and ATM LAN Emulation, but first we have to get a look at the problem of address resolution in this context.

5.4.4.1.ADDRESS RESOLUTION (ATMARP AND INATMARP)

- The address resolution in an ATM logical IP subnet is done by the ATM Address Resolution Protocol (ATMARP) based on RFC 826 and the Inverse ATM Address Resolution Protocol (InATMARP) based on RFC 1293. ATMARP is the same protocol as the ARP protocol with extensions needed to support ARP in a unicast server ATM environment. InATMARP is the same protocol as the original InARP protocol but applied to ATM networks. Use of these protocols differs depending on whether PVCs or SVCs are used. Both ATMARP and InATMARP are defined in RFC 1577, which is a proposed standard with a state of elective. The encapsulation of ATMARP and InATMARP requests/replies is described in Classical IP over ATM (below).

- **InATMARP** : The ARP protocol is used to resolve a host's hardware address for a known IP address. The InATMARP protocol is used to resolve a host's IP address for a known hardware address. In a switched environment you first establish a VC (Virtual Connection) of either PVC (Permanent Virtual Connection) or SVC (Switched Virtual Connection) in order to communicate with another station. Therefore you know the exact hardware address of the partner by administration but the IP address is unknown. InATMARP provides dynamic address resolution. InARP uses the same frame format as the standard ARP but defines two new operation codes:

    - InARP request=8

    - InARP reply=9

  Basic InATMARP operates essentially the same as ARP with the exception that InATMARP does not broadcast requests. This is because the hardware address of the destination station is already known. A requesting station simply formats a request by inserting its source hardware and IP address and the known target hardware address. It then zero fills the target protocol address field and sends it directly to the target station. For every InATMARP request, the receiving station formats a reply using the source address from the request as the target address of the reply. Both sides update their ARP tables. The hardware type value for ATM is 19 decimal and the EtherType field is set to 0x806, which indicates ARP according to RFC 1700.

- **Address Resolution in a PVC Environment :** In a PVC environment each station uses the InATMARP protocol to determine the IP addresses of all other connected stations. The resolution is done for those PVCs which are configured for LLC/SNAP encapsulation. It is the responsibility of each IP station supporting PVCs to revalidate ARP table entries as part of the aging process.

- **Address Resolution in an SVC Environment** : SVCs require support for ATMARP in the non-broadcast environment of ATM. To meet this need, a single ATMARP server must be located within the Logical IP Subnetwork (LIS) (see The Logical IP Subnetwork (LIS)). This server has authoritative responsibility for resolving the ATMARP requests of all IP members within the LIS. For an explanation of ATM terms please refer to Classical IP over ATM.

The server itself does not actively establish connections. It depends on the clients in the LIS to initiate the ATMARP registration procedure. An individual client connects to the ATMARP server using a point-to-point VC. The server, upon the completion of an ATM call/connection of a new VC specifying LLC/SNAP encapsulation, will transmit an InATMARP request to determine the IP address of the client. The InATMARP reply from the client contains the information necessary for the ATMARP server to build its ATMARP table cache. This table consists of :

- IP address

- ATM address

- Timestamp

- Associated VC

This information is used to generate replies to the ATMARP requests it receives.

Note: The ATMARP server mechanism requires that each client be administratively configured with the ATM address of the ATMARP server.

- **ARP table add/update algorithm :**

  - If the ATMARP server receives a new IP address in an InATMARP reply the IP address is added to the ATMARP table.

  - If the InATMARP IP address duplicates a table entry IP address and the InATMARP ATM address does not match the table entry ATM address and there is an open VC associated with that table entry, the InATMARP information is discarded and no modifications to the table are made.

  - When the server receives an ATMARP request over a VC, where the source IP and ATM address match the association already in the ATMARP table and the ATM address matches that associated with the VC, the server updates the timeout on the source ATMARP table entry. For example, if the client is sending ATMARP requests to the server over the same VC that it used to register its ATMARP entry,

the server notes that the client is still "alive" and updates the timeout on the client's ATMARP table entry.

- When the server receives an ARP_REQUEST over a VC, it examines the source information. If there is no IP address associated with the VC over which the ATMARP request was received and if the source IP address is not associated with any other connection, then the server adds this station to its ATMARP table. This is not the normal way because, as mentioned above, it is the responsibility of the client to register at the ATMARP server.

- **ATMARP table aging :**

  ATMARP table entries are valid:

  - In clients for a maximum time of 15 minutes

  - In servers for a minimum time of 20 minutes

  Prior to aging an ATMARP table entry, the ATMARP server generates an InARP_REQUEST on any open VC associated with that entry and decides what to do according to the following rules:

  - If an InARP_REPLY is received, that table entry is updated and not deleted.

  - If there is no open VC associated with the table entry, the entry is deleted.

  Therefore, if the client does not maintain an open VC to the server, the client must refresh its ATMARP information with the server at least once every 20 minutes. This is done by opening a VC to the server and exchanging the initial InATMARP packets.

  The client handles the table updates according to the following:

  - When an ATMARP table entry ages, the ATMARP client invalidates this table entry.

  - If there is no open VC associated with the invalidated entry, that entry is deleted.

  - In the case of an invalidated entry and an open VC, the ATMARP client revalidates the entry prior to transmitting any non-address resolution traffic on that VC. There are two possibilities:
    + In the case of a PVC, the client validates the entry by transmitting an InARP_REQUEST and updating the entry on receipt of an InARP_REPLY.
    + In the case of an SVC, the client validates the entry by transmitting an ARP_REQUEST to the ATMARP server and updating the entry on receipt of an ARP_REPLY.

  - If a VC with an associated invalidated ATMARP table entry is closed, that table entry is removed.

As mentioned above, every ATM IP client which uses SVCs must know its ATMARP server's ATM address for the particular LIS. This address must be named at every client during customization. There is at present no "well-known" ATMARP server address defined.

5.4.4.2.CLASSICAL IP OVER ATM

- The definitions for implementations of classical IP over ATM (Asynchronous Transfer Mode) are described in RFC 1577 which is a proposed standard with a status of elective according to RFC 1720 (STD 1). This RFC considers only the application of ATM as a direct replacement for the "wires", local LAN segments connecting IP end-stations ("members") and routers operating in the "classical" LAN-based paradigm. Issues raised by MAC level bridging and LAN emulation are not covered.

  Initial deployment of ATM provides a LAN segment replacement for:

  - Ethernets, token-rings or FDDI networks
  - Local-area backbones between existing (non-ATM) LANs
  - Dedicated circuits of Frame Relay PVCs between IP routers

  This RFC also describes extensions to the ARP protocol (RFC 826) in order to work over ATM. This is discussed separately in Address Resolution (ATMARP and InATMARP).

- As said above, the only way for a higher layer protocol to communicate across an ATM network is over the **ATM AAL**, as its function is to perform the mapping of PDUs into the information field of the ATM cell and vice versa. There are four different AAL types defined, AAL1, AAL2, AAL3/4 and AAL5. These AALs offer different services for higher layer protocols (for more on these, see above in section 4.3). Here are the characteristics of AAL5 which is used for TCP/IP:

  - Message mode and streaming mode
  - Assured delivery
  - Non-assured delivery (used by TCP/IP)
  - Blocking and segmentation of data
  - Multipoint operation

  AAL5 provides the same functions as a LAN at the MAC (Medium Access Control) layer. The AAL type is known by the VC endpoints via the cell setup mechanism and is not carried in the ATM cell header. For PVCs the AAL type is administratively configured at the endpoints when the Connection (circuit) is set up. For SVCs, the AAL type is communicated along the VC path via Q.93B as part of call setup establishment and the endpoints use the signaled information for configuration. ATM switches generally do not

care about the AAL type of VCs. The AAL5 format specifies a packet format with a maximum size of 64KB - 1 byte of user data. The "primitives" which the higher layer protocol has to use in order to interface with the AAL layer (at the AAL service access point - SAP) are rigorously defined. When a high-layer protocol sends data, that data is processed first by the adaptation layer, then by the ATM layer and then the physical layer takes over to send the data to the ATM network. The cells are transported by the network and then received on the other side first by the physical layer, then processed by the ATM layer and then by the receiving AAL. When all this is complete, the information (data) is passed to the receiving higher layer protocol. The total function performed by the ATM network has been the non-assured transport (it might have lost some) of information from one side to the other. Looked at from a traditional data processing viewpoint all the ATM network has done is to replace a physical link connection with another kind of physical connection - all the "higher layer" network functions must still be performed (for example IEEE 802.2).

- **The logical IP subnetwork (LIS)** : The term LIS was introduced to map the logical IP structure to the ATM network. In the LIS scenario, each separate administrative entity configures its hosts and routers within a closed logical IP subnetwork (same IP network/subnet number and address mask). Each LIS operates and communicates independently of other LISs on the same ATM network. Hosts that are connected to an ATM network communicate directly to other hosts within the same LIS. This implies that all members of a LIS are able to communicate via ATM with all other members in the same LIS (VC topology is fully meshed). Communication to hosts outside of the local LIS is provided via an IP router. This router is an ATM endpoint attached to the ATM network that is configured as a member of one or more LISs. This configuration may result in a number of separate LISs operating over the same ATM network. Hosts of differing IP subnets must communicate via an intermediate IP router even through it may be possible to open a direct VC between the two IP members over the ATM network.

5.4.4.3.ATM LAN EMULATION

- Another approach to provide a migration path to a native ATM network is ATM LAN emulation. ATM LAN emulation is still under construction by ATM Forum working groups. There is no ATM Forum implementation agreement available covering virtual LANs over ATM but there are some basic agreements on the different proposals made to the ATM Forum.

  The concept of ATM LAN emulation is to construct a system such that the workstation application software ‘‘thinks”it is a member of a real shared-medium LAN, such as a token-ring for example. This method maximizes the reuse of existing LAN software and significantly reduces the cost of migration to ATM. In PC LAN environments for example the LAN emulation layer could be implemented under the NDIS/ODI-type interface. With such an implementation all the higher layer protocols, such as IP, IPX, NetBIOS and SNA for example, could be run over ATM networks without any change.

- **LAN Emulation Layer** (Workstation Software) : Each workstation that performs the LE function needs to have software to provide the LE service. This software is called the LAN emulation layer (LE layer). It provides the interface to existing protocol support (such as

IP, IPX, IEEE 802.2 LLC, NetBIOS, etc.) and emulates the functions of a real shared-medium LAN. This means that no changes are needed to existing LAN application software to use ATM services. The LE layer interfaces to the ATM network through a hardware ATM adapter.

The primary function of the LE layer is to transfer encapsulated LAN frames (arriving from higher layers) to their destination either directly (over a ‘‘direct VC”) or through the LE server. This is done by using AAL5 services provided by ATM.

Each LE layer has one or more LAN addresses as well as an ATM address.

A separate instance (logical copy or LE client) of the LE layer is needed in each workstation for each different LAN or type of LAN to be supported. For example, if both token-ring and Ethernet LAN types are to be emulated, then you need two LE layers. In fact they will probably just be different threads within the same copy of the same code but they are logically separate LE layers. Separate LE layers would also be used if one workstation needed to be part of two different emulated token-ring LANs. Each separate LE layer needs a different MAC address but can share the same physical ATM connection (adapter).

- **LAN Emulation Server** : The basic function of the LE server is to provide directory, multicast and address resolution services to the LE layers in the workstations. It also provides a connectionless data transfer service to the LE layers in the workstations if needed.

  Each emulated LAN must have an LE server. It would be possible to have multiple LE servers sharing the same hardware and code (via multithreading) but the LE servers are logically separate entities. As for the LE layers, an emulated token-ring LAN cannot have members that are emulating an Ethernet LAN. Thus an instance of an LE server is dedicated to a single type of LAN emulation. The LE server may be physically internal to the ATM network or provided in an external device, but logically it is always an external function which simply uses the services provided by ATM to do its job.

- **Default VCs** : A default VC is a connection between an LE layer in a workstation and the LE server. These connections may be permanent or switched. All LE control messages are carried between the LE layer and the LE server on the default VC. Encapsulated data frames may also be sent on the default VC.
  The presence of the LE server and the default VCs is necessary for the LE function to be performed.

- **Direct VCs** : Direct VCs are connections between LE layers in the end systems. They are always switched and set up on demand. If the ATM network does not support switched connections then you cannot have direct VCs and all the data must be sent through the LE server on default VCs. If there is no direct VC available for any reason then data transfer must take place through the LE server (there is no other way).

  Direct VCs are set up on request by an LE layer (the server cannot set them up as there is no third party call setup function in ATM). The ATM address of a destination LE layer is provided to a requesting LE layer by the LE server. Direct VCs stay in place until one of the partner LE layers decides to end the connection (because there is no more data).

- **Initialization** : During initialization the LE layer (workstation) establishes the default VC with the LE server. It also discovers its own ATM address - this is needed if it is to later set up direct VCs.

- **Registration** : In this phase the LE layer (workstation) registers its MAC addresses with the LE server. Other things like filtering requirements (optional) may be provided.

- **Management and Resolution** : This is the method used by ATM end stations to set up direct VCs with other end stations (LE layers). This function includes mechanisms for learning the ATM address of a target station, mapping the MAC address to an ATM address, storing the mapping in a table and managing the table.

  For the server this function provides the means for supporting the use of direct VCs by end stations. This includes a mechanism for mapping the MAC address of an end system to its ATM address, storing the information and providing it to a requesting end station.

  This structure maintains full LAN function and can support most higher layer LAN protocols. Reliance on the server for data transfer is minimized by using switched VCs for the transport of most bulk data.

6. MODELIZATION

In the previous sections, we have make a survey of the state of the art concerning terrestrial network and their characteristics. Now, we have enough information to move to the next phase, which is to build the actual models needed by the other parts of the project.

A first comment on the survey done before is that there are definitely many kinds of networks with vastly different features in existence. Also, not all technologies hold the same interest, some are new and promising ones, but others decline both in popularity and use.

As such, we are dedicated to provide future clients with models covering the range of terrestrial networks. But technology evolves, and it is likely that some of the networks studied in the previous part will have fallen more or less completely into disuse at the end of the BISANTE project. Therefore, it is a complete loss of time for us to modelize them. Those ‘’dead-end’’ technologies encompass :

- Token Ring
- X.25

Both types of networks that have been implemented more than 20 years ago, for use in less-than-ideal conditions (slow speed, unreliable cables with an important bit error rate, PCs with few processing ressources, ...) that differed vastly of what is found nowadays. As such, they propose extensive error control mechanisms that result in losses of bandwidth, but were necessary at this time. Now that technical conditions have evolved, such possibilities are seen by network users and planners only as drawbacks because of the limitations they impose on network performances. For these reasons, no models will be provided for those two kinds of networks.

FDDI is in a very similar position, as it is a now stagnant technology. Nonetheless, it still had an interest for many years, as it was the quickest network type around, and, as such, had many uses as a network dedicated to bandwidth-consuming applications, or (more often) as a backbone for other kinds of network (the typical case being that of a FDDI network operating as a backbone for several Ethernet subnetworks). But the apparition of Fast Ethernet and later Gigabit Ethernet, networks with far more interesting features and similar or even better speed, seems to have sent FDDI definitely on a losing curve.

Frame Relay, due to its qualities and impressing results, was considered as a candidate to modelization in the BISANTE project, but was finally discarded due to time constraints. Terrestrial network modelization plays an important role in our project, as it will serve to validate the user behavior and application models produced in WorkPackage 1 (how can we plan to validate our traffic generator models if we run them on unreliable network models). It is then of utmost importance to be sure that our models are reliable. As we have only 6 months, we have decided to discard Frame Relay, the least attractive network left, and concentrate on the following networks:

- Ethernet : the most widespread terrestrial LAN network in existence today, and still undergoing changes to make it more attractive

- ATM : this widely-developed and used protocol is the most successful one today for transmitting multimedia data

- IP (and associated protocols) : certainly the most popular WAN network, achieving amazing results because of its simple yet efficient design

Now we know what we are going to modelize, we need to determine the best way to build those models. In the following sections, we will examine the different methods we have at our disposal, and choose among those the one that is the most adapted to our needs.

6.1. QUEUING MODELING

As said before, our goal in this chapter is to detail all the various techniques used today to modelize telecommunications networks. Our first point of interest in this optic will then be the concept of queues which is one of the most basic one in this field (in order to represent buffers, processing times, …). Of course, we do not plan to make an exhaustive survey of the matter in the course of the BISANTE project, but only a general presentation. Interested readers should be referred to Kleinrock's work in [1].

6.1.1. BASICS OF QUEUE THEORY

We will first begin with short-hand queuing, which in turn will allow us to introduce a number of notions that will be recurrent all along this chapter. This particular queuing theory, more detailed in [2], offers methods to characterize queue behaviors from a low-detailed point-of-view.

Queues can be referred to as *A/B/m*, *A* and *B* being respectively the inter-arrival time distribution and service time distribution, and m being an integer indicating the number of servers for the queue. *A* and *B* take most often one of the following values (although others are used sometimes):

- Deterministic : deterministic arrivals and/or services have constant values (as such, they are modeled using stochastic processes)

- Markovian : arrivals and services have exponentially-distributed inter-arrival times

- General : indicates that the stochastic process is arbitrary

Such *A/B/m* models imply that there is no limit to the number of customers in the system. This is totally unrealistic of course, but as a general approximation it can be admitted in a number of cases. There are other cases, however, where a greater degree of realism is required. In those cases, the basic queue model is expanded to form the *A/B/m/k/s* model, where the first three parameters refer to the same values as before, *k* represents the queue length, and s the population of customers.

Although these models are generic, a number of special instances are of particular interest in network modeling (such as *M/M/1, GI/GI/1,* …) that will be described in a later part.

The arrival process is defined when the interarrival law is known. In the following the interarrival rate will be called λ and the square of the variation coefficient of interarrivals will be called C_a^2

6.1.1.1.ARRIVAL PROCESS

In the last paragraph, we presented a number of inter-arrival times distribution, which will now be detailed.

- **Random Arrivals :**

As stated earlier, Markovian processes are exponentially distributed inter-arrival times distributions, which have given rise to completely random arrival patterns.

If we consider a short time period δt and λ the average arrival time, on the interval $(t, t + \delta t)$ we can define the following probabilities :

- Probability that no customer arrives : $P[N = 0] = 1 - \lambda * \delta t + o(\delta t)$
- Probability that one customer arrives : $P[N = 1] = \lambda * \delta t + o(\delta t)$
- Probability that more than one customer arrive : $P[N > 1] = o(\delta t)$

These three equations describe a completely random arrival process resulting in exponentially distributed inter-arrival process, more commonly known as Markovian process, the probability density function for the exponential distribution being $f(t) = \lambda e^{-\lambda t}$.

Hence, customers arriving at an average rate λ in a completely random pattern have inter-arrival times with a negative exponential distribution, and the number of customers arriving in a given interval has a Poisson distribution. One of the interesting properties of this situation is that the choice of the time origin is completely without importance. This is known as the memoryless property.

- **Batch Arrivals and Burstiness :**

In the previous point, we have described how varies the interval between reception of packets, but nothing has been said about the arrival duration proper, which is directly dependent on the size of the packets. If the size of the packets is fixed, then the problem is fairly straightforward, but if it is not the case, as it is true for a number of types of network (see preceding chapters), then we need to study the effects of varying arrival lengths as well as varying arrival times. This is called batch arrival processing.

As usual, random batch sizes can be modeled using a variety of probability distributions. However, not all the various discrete random variables that may be used in this case hold the same interest for network traffic modeling.

- Geometric distribution : using the geometric distribution with parameter *q*, the probability of having *k* data units (being bits, bytes or even packets of fixed length) in a batch is :

$$P\,[X = k] = q^{k} * (1 - q)$$

- Bernouilli trials : in this case the message length is given by the number of successes in n trials (k successes for $n - k$ failures), which gives us a probability of :

$$P\,[X = k] = q^{k} * (1 - q)^{n-k}$$

- Binomial distribution : it is very similar to the previous one, except that we integrate the fact that the k successes and n – k failures can have occurred in a number of sequences :

$$P\,[X = k] = C_{n,k} * q^{k} * (1 - q)^{n-k}$$

- Poisson distribution : if we allow n to increase without limit, and we require the mean batch length to remain a constant *nq*,, then we obtain a Poisson distribution with parameter *nq* (which can be useful to modelize sources for which there is no known limit to message's size) :

$$P\,[X = k] = (nq)k / k! * e^{-nq}$$

Burstiness is a relatively new notion in study of communication network traffic. It introduces a new parameter, the traffic burstiness, referred to as *B*, which is obtained numerically using the ratio of the mean and peak arrival rate. We now consider a node collecting single data units from a variety of sources, grouping them into batches and sending them on a single communication channel.

- For geometrically-distributed batch sizes with a mean $q / (1 - q)$:

$$B = 1 - U * q / (1 - q)$$

- For binomially-distributed batches :

$$B = 1 - U * q$$

- For Poisson batches (meaning that the number of sources is infinite) :

$$B = 1 - U$$

6.1.1.2.SERVICE MECHANISM

Service mechanism encompasses several aspects :

- Service discipline : in which order clients are served on a server (usually first come first served)

- Number of servers

- Server availability : a server in a single server queue may not be always available since a client is using the service, and subsequent calls must wait.

The number of servers and server availability are two aspects that we will investigate in some details. The server in a single-server queue is not always available, since at any given time a customer may be utilizing that service and subsequent arrivals are required to wait. Hence we can use the same probability distributions to model this service mechanism than we used for arrival process modeling.

One particular service mechanism that results in a distribution well used in telecommunications is worth investigating. Let's consider a service facility, such as telephone exchange. We consider a messaging service that it able to handle calls at an average rate of μ, the service time being of course random. The mean duration of a call is then $1 / \mu$, and the service time distribution is :

$$b(x) = \mu * e^{-\mu x}$$

Let us assume, as it is often the case on networks, that these calls proceed into several stages (connection and transmission for example), and let us call r the number of different stages. The service time distribution in each stage becomes :

$$b_{1,2}(x) = r * \mu * e^{-r\mu x}$$

as the mean duration on a particular stage is $1 / r\mu$.

Using Laplace transforms and the fact that the distribution of a random variable, whose value is determined by the sum of r different random variables, is given by the sum of the convolution of each variable, it is easy to determine that

$$b(x) = r\mu * (r\mu)^{r-1} e^{-nx} / (r-1)!$$

which is the expression known as the r-stage Erlang distribution, usually referred to as *Er*, which has numerous applications in the field of queue modeling.

6.2. MATHEMATICAL SOLUTIONS

Let us remind the main mathematical results that may be used in order to study network performance.

Mathematical solutions may be obtained when there is an imbedded Markov chain (see below) in the stochastic process describing the network state.

Usual results concern network models whose operation is simple:

- traffic is consistent: (a customer goes through the network, but he does not generates son customers, so for example a message must not generate ACKs)
- there maybe classes of customers, for each class the distribution of the route inside the network is given
- service times (which model either transmission delay on a link or treatment delay in a node) have to follow a Cox law: they can be decomposed into a number of stages of exponential services μ_n (on fig.1 the interstage transition probabilities are b_i).

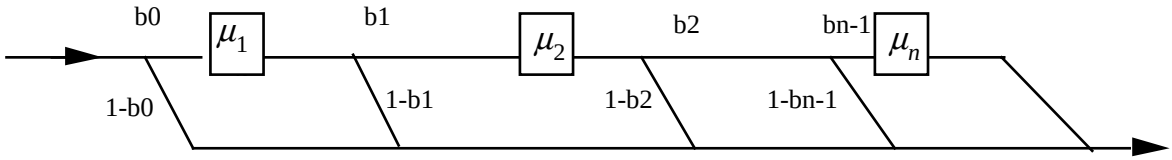


Fig.1: a Cox law

Let us remind that what is called an exponential law is the distribution of a random variable whose probability density function is:

$$f(x) = \mu \cdot e^{-\mu x}$$

The random variable is then memoryless.

6.2.1. LITTLE'S LAW

The most important performance result is valid for any stationary system, it relates the mean number of customers in the system L, the mean response time R and the throughput λ:

$$\mathbf{L = R\,\lambda}$$

6.2.2. CHANG LAVENBERG RESULT

The second very important result relates the mean number of visit to queue i : e_i , the utilization rate U_i of the server of queue i (in case queue i has only one server) and the mean service time S_i :

Chang and Lavenberg proved that for any stationary network the ratios U_i / e_i S_i are the same for all the one server queues:

$$\mathbf{U_1 / e_1 S_1 = U_i / e_i S_i = U_j / e_j S_j}$$

6.2.3. DEFINITION OF A DISCRETE TIME MARKOV CHAIN

Let E be the state space
States are numbered $\{1,2\ldots k\ldots\}$

Let $\{X_n\}, n \in N$ be a chain taking its value in E.

It is a discrete Markov chain if: "the probability for the chain to be in state j at time t = n +1 is known when state i_n at time t=n is known".

$$P(X_{n+1} = j / X_1 = i_1 \wedge X_2 = i_2 \wedge \ldots X_n = i_n) = P(X_{n+1} = j / X_n = i_n)$$

In fact for homogenous chains, which often will be the case, this transition probability will not depend on n but depends only of i and j states.

$$p_{ij} = P(X_{n+1} = j / X_n = i)$$

In this definition the time is discrete and may be the number of the present event.

From these conditional probabilities, may be derived the studies of the convergence of the chain to a stationary state and the eventual values of the stationary state probabilities.

6.2.4. DEFINITION OF A CONTINUOUS TIME MARKOV CHAIN

The definition is very much the same but the time is not discrete anymore.

$(X_t, t \in R)$ is a continuous time Markov chain if for any sequence of times :

$\forall (t_1, t_2 \ldots, t_n) \in R^n$ with $t_1 < t_2 < \ldots < t_n$

and for any state of the state space:

$\forall (i_1, i_2 \ldots, i_n) \in R^n$

$$P(X_{t_n} = j / X_t = i, \ldots X_{t_{n-1}} = i_{n-1}) = P(X_{t_n} = j / X_{t_{n-1}} = i_{n-1})$$

and for an homogenous Markov chain:

$$P(X(s+t) = j / X(s) = i) = P(X(t) = j / X(0) = i)$$

6.2.5. BIRTH AND DEATH PROCESS

It is a Markov process, for which state space is a set of integer numbers, the set of numbers of a population (see fig.2) and the only eventual events are a birth (arrival): state goes from *n* to *n+1* or a death (departure): state goes from *n* to *n-1*

- Probability of a birth between time t and time $t + dt$ is assumed to be $\lambda(n)dt$
- Probability of a death between time t and time $t + dt$ is assumed to be $\mu(n)dt$

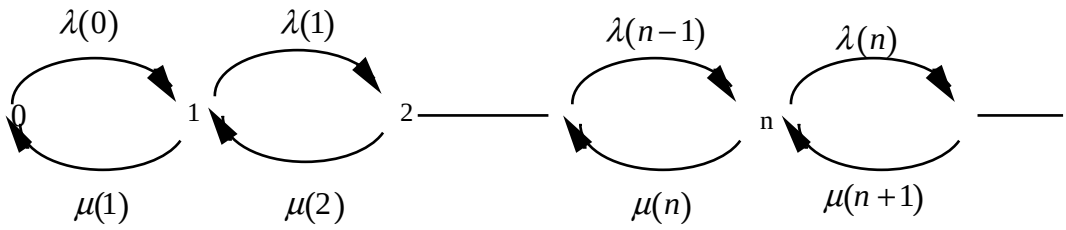


Fig. 2: Birth and death process

The stationary limit distribution is then :

$$P(n) = \frac{\lambda(n-1)\lambda(n-2)\cdots\lambda(0)}{\mu(n)\mu(n-1)\cdots\mu(1)} P(0)$$

if and only if the sum over every eventual states converges to value 1, which leads to the computation of *P(0).*

This basic model may be applied to solve different usual queue models. For example, for an *m* exponential server queues (see fig.3)

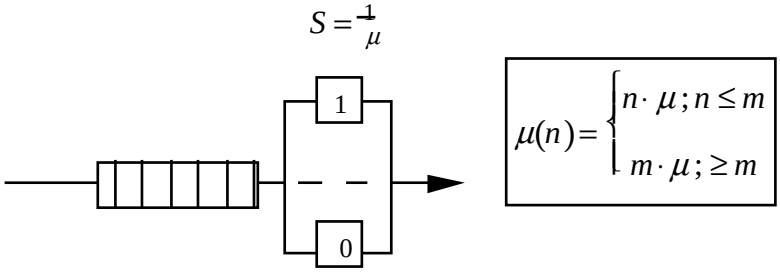


Fig.3 : an m exponential server queue

For a delay:

$$\mu(n) = n \cdot \mu \, , n \succ 0$$

Let us give a few performance results for the most usual queues.

6.2.6. *M/M/1* QUEUE

In case of Poisson arrivals, one exponential FIFO server, with infinite queues, the stability condition is $\lambda < \mu$
or,

$$\rho = \frac{\lambda}{\mu} = \lambda \cdot S < 1$$

The stationary probabilities are

$$P(n) = \rho^n \cdot (1-\rho)$$
$$P(0) = 1-\rho$$

$$\boxed{U = \rho}$$
$$\boxed{L = \frac{\rho}{1-\rho}}$$

$$\boxed{R = \frac{1}{\mu - \lambda}}$$

6.2.7. *M/M/1/N* QUEUE

In case the capacity of the queue is N and arrivals are rejected if the queue is full ,

$$\lambda(n) = \begin{cases} \lambda, \; si \; n < N \\ 0, \; si \; n \geq N \end{cases}$$

$$\mu(n) = \mu$$

leads to:

$$P(n) = \begin{cases} \frac{1-\rho}{1-\rho^{N+1}} \cdot \rho^n & si \; \rho \neq 1 \\ \frac{1}{N+1} & si \; \rho = 1 \end{cases}$$

Reject probability :

$$P(N) = \begin{cases} \frac{1-\rho}{1-\rho^{N+1}} \cdot \rho^N & si \; \rho \neq 1 \\ \frac{1}{N+1} & si \; \rho = 1 \end{cases}$$

$$U = 1 - P(0) = \rho \cdot [1 - P(N)]$$

$$\Lambda = \frac{U}{S} = \lambda \cdot [1 - P(N)]$$

$$L = \frac{\rho}{1-\rho}[1 - (N+1) \cdot P(N)] \qquad \text{if} \; \rho \neq 1$$
$$L = \frac{N}{2} \qquad \text{if} \; \rho = 1$$

$$R = \frac{L}{[1 - P(N)] \cdot \lambda} = \frac{L \cdot S}{U} = \frac{S}{1-\rho} - N \frac{\rho^N}{1-\rho^N} S$$

6.2.8. *M/M/1/N/N* QUEUE

This queue is very useful for local area network models (see Fig.4):There are N stations with N users for which the thinking time is exponential of mean $Z = 1/\lambda$ and one server with exponential service of mean $S = 1/\mu$.

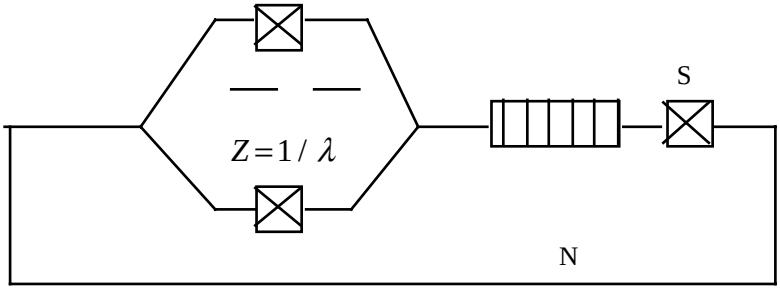


Fig.4 : local area network

The classical birth and death model with:

$$\lambda(n) = (N - n) \cdot \lambda$$
$$\mu(n) = \mu$$

leads to the performance results:

$$\cdot z = \frac{Z}{S} = \frac{\mu}{\lambda}$$

$$\cdot P(n) = P(0) \cdot \frac{N!}{(N-n)!} \cdot \frac{1}{z^n}$$

$$\cdot P(0) = \frac{\frac{z^N}{N!}}{1 + \sum_{1}^{N} \frac{z^n}{n!}} = D(N, z)$$

$$\cdot U = 1 - P(0)$$

$$\cdot R = \frac{N}{\Lambda} - Z = \frac{N \cdot S}{U} - Z$$

$$\cdot L = U \cdot \frac{R}{S}$$

Saturation (N very large and U = 1) leads to $R = N.S - Z$

6.2.9. *M/G/1* QUEUE

For a general service queue, if C is the squared variation coefficient of the service, stability is true when the traffic rate is less than 1

Then the performance results are :

$$U = \rho$$

$$L = \rho\left[1 + \frac{\rho}{1-\rho}\left(\frac{1+C^2}{2}\right)\right]$$

$$\frac{R}{S} = 1 + \frac{\rho}{1-\rho}\left(\frac{1+C^2}{2}\right)$$

6.2.10. JACKSON 'S THEOREM

It is the simplest theorem that may be used for an open network of n queues, each of them having one FIFO exponential server and an infinite capacity queue.(see Fig.5) The Poisson arrival rate $\lambda(K)$ may depend on the total number K of customers in the network and each service rate $\mu_i(k_i)$ may depend on the number of customers k_i in its own queue.

$$k_1 + k_2 + \ldots k_n = K$$

When each queue is stable (the condition is the same as it is for *M/M/1*), the network is stable and the performance results are for each queue the *M/M/1* results.

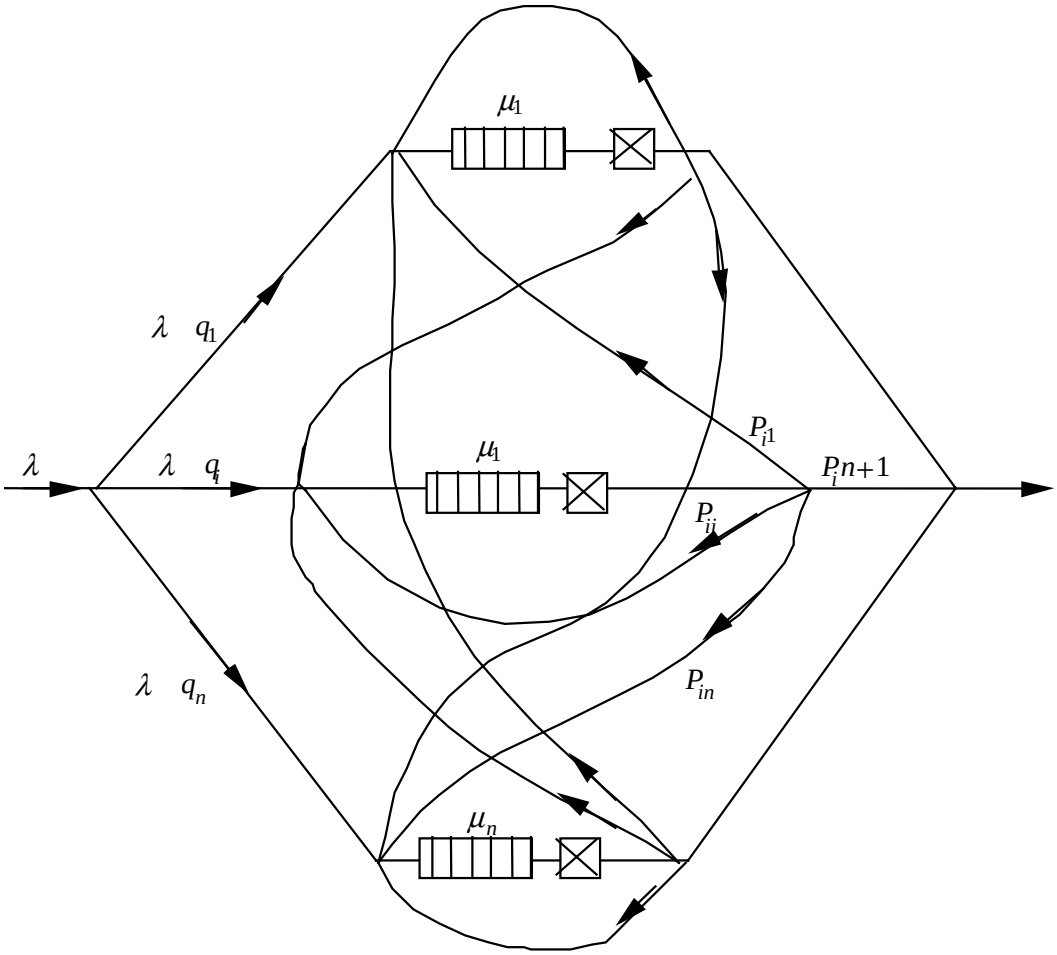


Fig.5 :Jackson network

Let us give the performance results when the arrival and service rates are constant:

$$\boxed{U_i = \frac{\lambda e_i}{\mu_i}}$$

$$L_i = \frac{\lambda e_i}{\mu_i - \lambda e_i}$$

$$\boxed{R_i = \frac{1}{\mu_i - \lambda_i} = \frac{1}{\mu_i - \lambda e_i}}$$

The global response time is:

$$R = \sum_{i=0}^{n} e_i \, R_i$$

6.2.11. GORDON AND NEWELL THEOREM

There is also a theorem in the case when the network is closed and the total numbers of customers is constant (see Fig.6). The hypothesis for the queue operations are the same as those of Jackson theorem, but each queue does not need to have an infinite capacity, it is enough to have a capacity that is as much as the total number of customers in the network: K.

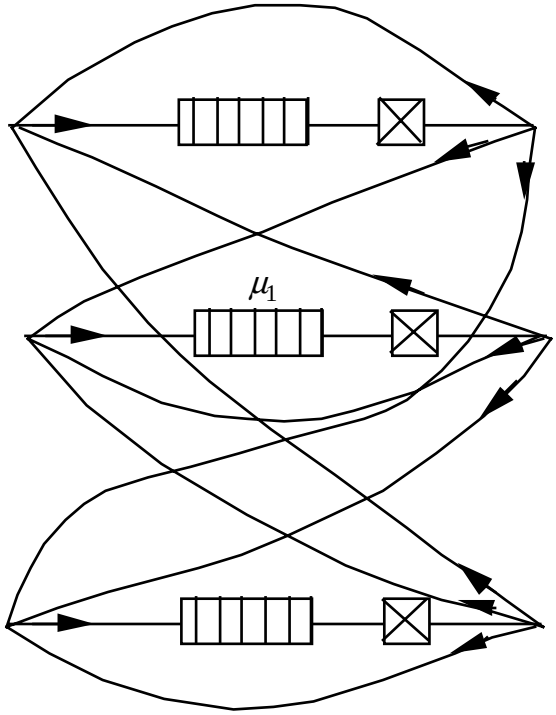


Fig.6 : Gordon and Newell network

The results are not as simple as Jackson results, because the sum of the probability over every possible states is not a sum from 0 to infinity, but it is more complex. There are several algorithms that let evaluate performance criteria. They are implemented in packages such as MODLINE.

6.2.12. B/C/M/P THEOREM

This theorem is more general than Jackson and Gordon and Newell (cf [3]).

It may be applied for multiclass networks, for which the route for each class may be different, and for which each of the queues belongs is of one of the following types:

- Type 1: One exponential server, FIFO and infinite capacity (Jackson and Gordon and Newell cases)

- Type 2: Processor Sharing, COX service law

- Type 3: Last Come First Served, preemptive service, COX service law

- Type 4: Delay , COX service law (this will model the transmission on multiplexed links or the think time in stations of a LAN).

The performance formula are not simple but they may be derived from packages such as MODLINE.

Please note that no theorem exists in the case of priority classes. For FIFO, the service has to be exponential.

6.3. APPROXIMATION METHODS

The results that were given before show that there is a very limited set of queueing networks for which an exact solution is known:

- when scheduling is FIFO in a queue, its server has to be exponential,
- queues have to have an infinite capacity
- no blocking is allowed
- routing has to be according to a given probability
- no synchronization is allowed
- no priority classes
- no fork no join

Moreover, even when there is an exact solution (we call it product form solution), its computation is not simple and may lead to computing errors.

So in order to be able to derive performance criteria, for real networks or for real future networks, it is necessary to use an approximate solutions. There are many types of approximate solutions. Of course none of them is perfect and each of them has to be validated by measurements or by simulations.{reference A1?????}

In the following we shall only say a few words about some of them.

- The 3 first ones are decomposition into a set of solved queues.
- The 4 following ones are examples of Aggregation Techniques
- The 8th is mean value analysis
- The 9th is diffusion approximation

We could also mention heavy traffic approximation, which lead to worst case studies, and large deviation studies, and many other approximations. However, we shall only present 9 examples. Readers should be aware that this list is non-exhaustive, also we will not detail them into many details, as it is not our point. We shall discuss later on the choice that was made to use simulation and not to use approximate analytical methods.

6.3.1. DECOMPOSITION INTO A SET OF SOLVED QUEUES.

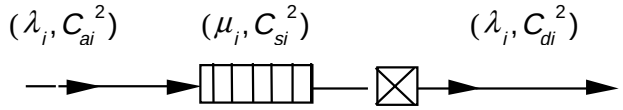
The most common approximation is a **decomposition of the network into M/GI/1 queues.** {references A2 et A3?????}. It means that the arrival process in each queue is approximated by a Poisson process. For each queue the service is assumed to be general but services are independent. The parameters of each queue have to be chosen properly.

In case of bursty traffic an alternative is **decomposition of the network into IBP/D/1 queues.** {reference A4?????}. It means that the arrival process in each queue is approximated by an IBP process whose parameters are chosen properly..

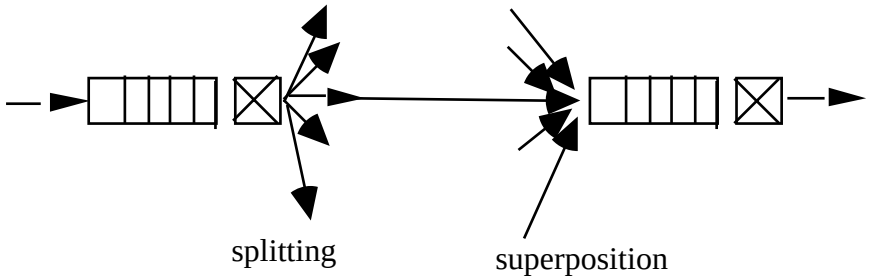
Another approximation is a **decomposition of the network into G/GI/1 queues.** {reference A5?????}. For each queue the service is assumed to be general but services are independent.

The two first moments of the arrival process in each queue are computed.from the splitting and the superposition of the preceding queues output process, by using Marshall formula that computes the square of variation coefficient of the output process from a queue C_{di}^2, as a function of the square of the variation coefficient of the arrival process C_{ai}^2, of the square of the variation coefficient of the service process C_{si}^2 and of the traffic rate ρ_i.

$$C_{di}^2 = C_{ai}^2 + \rho_i^2 (C_{si}^2 - C_{ai}^2)$$



Then flows have to be splitted and superposed. In each case the traffic characteristics have to be computed.



Then performance criteria for each queue may be derived from approximations such as Kingman's formula.

$$L_i = \rho_i \left(1 + \frac{\rho_i (C_{ai}^2 + C_{si}^2)}{2(1-\rho_i)}\right)$$

6.3.2. AGGREGATION METHODS

In order to study it, a complex system can be parted in subsystems and two steps can be performed :

Step 1 : aggregation. To study the subsystems independently from each other by simplifying as much as possible the relations between a subsystem and the others.

Step 2 : desaggregation. To study the total system by replacing the subsystems by their solution deduced from the preliminary step.

There are cases when this aggregation method is exact. It is the case for example for of queuing networks for which the BCMP theorem is valid, if the subset of files which are incorporated are well chosen.

There are cases when the method is not exact, but allows in a reasonable time to obtain an approximate solution. It is necessary in this case to validate the approximate solution by comparing for given values of the parameters the solution obtained by aggregation with the exact solution.

Sometimes the method of approximate aggregation is iterative. One repeatedly carries out the two steps until convergence of the parameters characterizing the relations between subsystems.

This method is applied in four different ways :

- **Aggregation of physical systems :** There are many cases when the physical system includes natural aggregates. For example the cell of a radio-mobile system or the macro-cell is a natural aggregate. A cell can be studied only by simplifying as much as possible the relations between the cell and the rest of the system. The flow of the calls coming from the mobiles non-present in the cell, and the calls towards the mobiles non-present in the cell will be simplified. The movements of the mobiles will be studied. It will then be necessary to validate these simplifications by studying the total system.

- **Aggregation of Markov chains :** States of the Markov chain are aggregated. This can be particularly useful when the Markov chain has too many states for a direct solution to be possible. The aggregation can be exact if for each of the elements of an aggregate, the sums of the transition probabilities towards the elements of an other aggregate are equal. It can also be approximate.

- **Aggregation of systems corresponding to different time scales:** When a system is studied there are often several possible time scales corresponding to different levels of details. In general models corresponding to different levels cannot be simultaneously studied, because if a study at a given level had to take into account the inferior level, it would have too many events of the inferior level to take into account. An aggregation method can do it possible to use the results obtained on a level of study to treat another level. For example, in the ATM networks the call level model, can not be solved simultaneously with the cell level model. The time scales are very different. In a call there

are too many cells. If some call routing or some CAC algorithm are studied, the levels of interest are call level or burst level. If cell buffers are to be dimensioned or jitter is to be studied, the level of interest is the cell level.

- **Aggregation of queues in a queueing network** : In a B/C/M/P network, if a subset of queues is chosen properly (if there is no problem in defining the customer transitions in the aggregated network), this subset of queues may be replaced by a single composite queue whose service is obtained by closing the subsystem {reference A6-7 ?????}and computing the output process for every possible choice of the population inside the closed subset

Let us show a small example:about the following network:

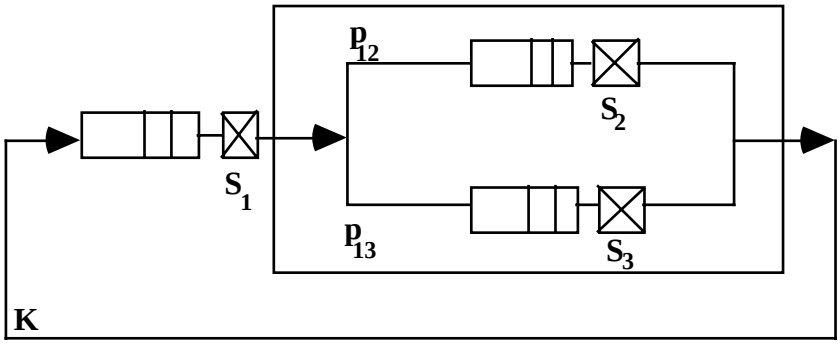


Figure 1 : initial network

The subset of queue 2 and queue 3 may be aggregated and closed:

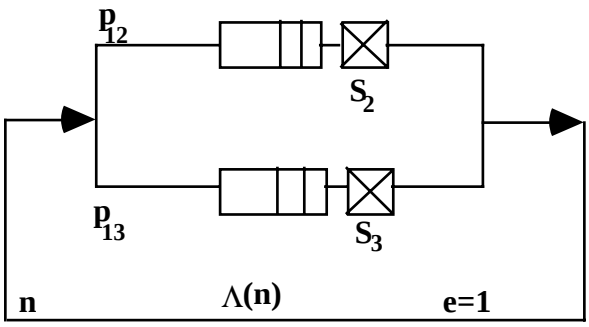


Figure 2: closed subset

and it may be replaced by a composite queue with service rate μ(n):

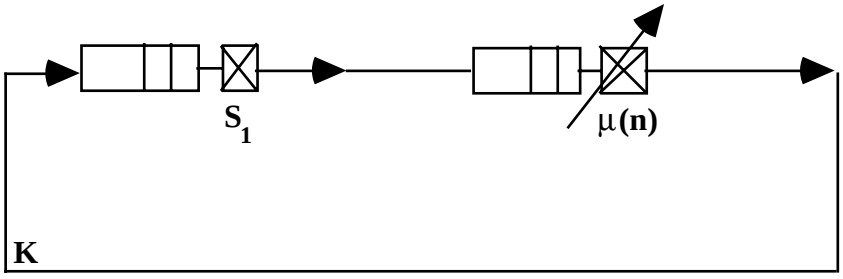


Figure 3: subset is replaced by a composite queue

and μ(n) is equal to the global throughput Λ(n) of the closed subset in Figure 2

This result that is exact for BCMP networks. It may also be used iteratively in order to derive the characteristics of a composite queue that is an approximation of the subsystem.

Another way of computing the approximation queue was introduced by R. Marie{reference A8-9?????}It was a fixed point method computing the characteristics of the composite equivalent queue.

6.3.3. MEAN VALUE ANALYSIS

They are derived from exact results that are due to a theorem giving the mean values for response time for a customer as a function of the unfinished work at his arrival instant. {reference 10?????} This leads to exact results for BCMP networks. {reference 11-12?????}. Let us use the same notations as in part 6.2, but each criteria will be a function of the number of customers inside the closed network. Let us only give the results for a FIFO monoclass exponential queue: at an arrival instant in a K customer network, the unfinished work is the same as the average work in a K-1 customer network.

$$R_i(K) = S_i(1 + L_i(K-1))$$

Then Little Formula applied to the whole network let us derive the global throughput::

$$\Lambda(K) = \frac{K}{\sum_i e_i R_i(K)}$$

and Little Formulas applied to each queue lead to:

$$L_i(K) = \Lambda(K)\, e_i R_i(K)$$

The 3 preceding formulas constitute a loop that derives the performance criteria of a network with K customers from the performance criteria of a network with K-1 customers. Since the case of 1 customer is easy to solve, the performance criteria may be derived for any value of K.

Those formulas may also be applied iteratively in order to compute approximate results in non BCMP networks.

For example in the case of a network with many customers, it may be assumed that the response time does not depend much on the number of customers inside the network. Then loop K with the 3 formulas may be computed until a fixed point is reached.

6.3.4. DIFFUSION APPROXIMATIONS

{reference A14?????} Several processes such as the number of customers in the queue or the unfinished work, which are discrete, may be replaced by continuous processes which are solutions of differential equations such as:

$$\frac{\partial f}{\partial t} = -\frac{\partial}{\partial x} m(x,t)\, f + \frac{1}{2}\frac{\partial^2}{\partial x^2}\sigma^2(x,t)\, f$$

The limit conditions, corresponding for example to a number of customers that is negative may be chosen in several ways: (mirror type or jump from 0 type).

After having solved the continuous process, it is necessary to discretize it.

Diffusion approximations are especially interesting when dealing with the tail distribution. They lead to large deviation techniques.

6.3.5. METHODS USED BY THE CONSORTIUM

The consortium intend to design a tool that could be used by non specialists and so that does not need to use complex analytical approximations and complex validations of them. The consortium choose to realise simulation models. But it will be explained later on, that an aggregation technique will be used inside the simulation tool. It is a mix between time aggregation, and model aggregation. But this aggregation will be used for the experiments on computers which are called simulations. Let us present first computer simulations in the next section.

6.4. COMPUTER SIMULATIONS

Performance studies of data-processing and telecommunication networks constitute one of the key stages in their deployment and the optimization of their use. Indeed, the used resources are generally expensive and the quality of service requirements are increasingly constraining. When wanting to implement a network, it is necessary to check that the system will satisfy the constraints imposed by its users. These constraints are expected performance which can be seen both from the user (rate of loss of information, mean answer time, call blocking rate) and from the resource (mean packet number in a router, utilization ratio of different resources). When studying the performance of a complex system, several techniques may be used.

6.4.1. MEASUREMENTS

When a system exists, measurements may be performed to obtain the performance of the system under given conditions. Let us note, and we shall reconsider these problems thereafter, that measurements are experiments and that, because of the complexity of given systems, it is necessary, as in physical sciences, to have an idea of the quality of the results of the presented measurements. The major problem with measurements is that it is generally difficult even impossible to test a sufficient number of configurations to optimize the operation of the system. Moreover, if the system does not exist, measurements are impossible In addition, it is often necessary to check that the measuring instruments do not disturb the operation of the system itself.

6.4.2. BENCHMARKS

This technique is very widespread in the field of the parallel machines. They can also be used to compare several network hardware. Usual configurations of the system are studied and measurements are performed. Generally, the problems arising are similar to those arising when measuring.

6.4.3. MODELING

The objective of modeling is initially to build a model of the operation of the studied system. These models use a mathematical formalism. The formalisms most usually used in the field of network performance are queues, stochastic Petri nets, Markov chains. The load is not constant, the models which are set up are generally probabilistic models which are more realistic than deterministic models.

The next step consists in studying the performance of the model. Exact or approached mathematical solutions or computer simulations are used. The mathematical solutions consist in writing relations between the characteristics of the models. The wanted performance criteria, according to the various characteristics of the model, are obtained. This type of method unfortunately imposes often strict constraints which in problems of large size are seldom right.

The computer simulation consists in reproducing the operation of the model during a given time length and estimating the performance criteria during this time. Modeling makes possible to vary easily the parameters of the system without having to disturb the operation of the network. Let us note that these steps are sometimes complementary.

Indeed, the measuring instruments can be used to provide the input parameters of the models and to validate the results obtained by analyzing the models. The objective of this document is to give a state of the art of the various types of simulations and problems inherent in the computer simulation.

6.4.4. COMPUTER SIMULATIONS

The objective of a computer simulation is to reproduce the operation of a model of the system which is studied. During this simulation, performance criteria are estimated. Simulation will consist in studying a given number of stochastic processes which will characterize the system. These process will be primarily of two types:

- processes related to the state of the various resources. For example the packet number in a queue of a router according to time $L(t)$ will be observed

- processes related to the various entities treated by the network (packages, cells...). In those cases, one will observe for example the response time of each package crossing the network R_i being the response time of the client *i*

The performance criteria that one wants to determine are mean sizes in a stationary state of the system, i.e. criteria of the form $L = \lim_{t\to\infty} E[L(t)]$ or more generally $Z = \lim_{t\to\infty} E[f(L(t))]$. It makes then possible to determine quantities such as the variance of the response time. In the case of the criteria of performances related to the treated entities, one tries to estimate in the same way criteria of the type: $Z = \lim_{n\to\infty} E[f(R_n)]$, where *n* is the event number. The estimate which will then be obtained from simulation will be:

- the event related average for the performance criteria related to the treated entities. I.e. if *n* packets go through the network during simulation, the estimator will be:

$$Z_n = \frac{1}{n}\sum_{i=1}^{n} f(R_i)$$

- the temporal mean for the criteria of performances related to the use of the resources:

$$Z_T = \frac{1}{T}\int_{t=0}^{T} f(L(t))dt$$

Let us note that this integral notation is general but that generally one will integrate simply stepwise functions

6.4.5. DIFFERENT TYPES OF SIMULATIONS

- **Discrete-event simulation :**

The type of simulation most usually used and which is included in the majority of the simulation tools is discrete-events simulation. This type of solution corresponds well to the types of problems which involve problems of discrete event systems: the events correspond to displacements of the entities treated by the studied system (end of processing, end of emission, arrival of a new package to be emitted...).

In this type of simulation, a table of the next event which may happen with their time are managed. Simulated time is of course different from the time necessary to carry out the simulator. To update this table, the event which has the smallest time is extracted. The

processing of this event generally results in the creation of new events which are then inserted in the table. Several data structures can be used such as chained lists or tree structures for example.

A simulation duration is generally fixed (in terms of simulated time), simulation stops with the time of the first event higher than the simulation duration.

- **Roulette Simulation :**

Roulette simulation is a particular case of discrete-events simulation. One is implemented when all the distributions (inter-arrivals, services) are exponential. Under these very particular conditions, the dates of the next events are not needed to be stored but only the state running of the system. Events are numbered, whatever their duration is . The memoryless property of exponential laws is used.

At each step, the next event which occurs is randomly drawn according to the current state of the system. The state of the system is modified and the simulation goes on. The analysis of the simulation results then consist in making averages compared to the events and no more temporal averages, which, in this precise case is similar. The essential interest of this method is that there are very few data to store; nevertheless its application field is somewhat restricted.

- **Instant-driven Simulation :**

The last type of simulation consists in looking at the state of the system when time is incremented of a unit step value and looking at all what has happened between the two successive instants. These methods are rather inadequate and are not often used in the most usual network simulators.

- **Fluid Simulation :**

A lot of queuing systems are studied with fluid approximations. In this case, the input and output processes are replaced with their means. With the approximation by the diffusion processes these means can vary according to the two first moments which are parameters of the model. These two first moments define a normal (or gaussian) law, which is so chosen because the random variables which are to be approximated are sums of independent random variables and the distribution of such sums tends asymptotically towards a normal law.

This method can be applied for example in order to obtain the customer number $N(t)$ and the virtual waiting time $W(t)$ at the instant t. The number of customers $N(t)$ is the difference between two processes tending towards a normal law, and hence a process with a normal distribution, by assuming their independence. This assumption is a hot practical problem because the output process is generally dependent on the input process. But if the number of customers in the system is not equal to zero, independence criteria are verified. That is why the approximation by a diffusion processes is essentially used for saturated systems or heavily loaded systems.

6.4.6. PROBLEMS INHERENT TO SIMULATION

The computer simulation is undoubtedly the technique most generally used when one wants to study the performances of a model. The interest comes from the fact that one can make simulations in almost all the cases which is not the case with mathematical methods whose fields of applications are unfortunately rather reduced. The simplicity of implementation should not however hide a given number of points which it should particularly be taken care of, when a simulation is run. These problems can be classified in two subsets:

- problems involved in the model: As an indication, one can cite the problems of validity of the model or the level of study. If the model is not correct, even if the simulator is good, the results will not be valid. If the model is too detailed, simulation will not be able to make it possible to simulate in a reasonable time the studied system.

- problems involved in simulation: In this category, one can classify the quality of the generators of pseudo-random numbers, the rounding errors, the establishment of the stationarity and the validity of the results of the simulation. We shall present in this document only the last cited point. The problems involved in the generators of pseudo-random numbers or the rounding errors are rather negligible if the traditional generators are used and if one takes the trouble to code information with a sufficient precision. The establishment of the stationarity is in itself a very complicated problem. The method most commonly used consists in collecting information only at the end of one predetermined duration. It is however difficult to choose in an automatic way this duration. As an indication, one may say that it is necessary that each variable from which it possible to derive an evaluation of the performance criteria, is modified at least "a given" number of times. On the other hand, it seems absolutely essential to us, to determine the confidence which one can allot to the results presented.

Each parameter should then be explored in detail :

- **Validity of the model :**

The first problem which is met comes from the quality of the developed model. Indeed, in the modeling phase, the essential characteristics of the system which is studied may be represented but all cannot be represented because too much complicated models would be obtained ... and then simulations could not be run any more in reasonable times. Let us note moreover that the smoothness of the model and the essential aspects which are represented in the model are strongly dependent on the performance criteria which are to be estimated. Consequently, the model will be a simplified vision of reality. Even if we are able then to run the most perfect simulation, the result cannot be perfectly concordant with reality. This phenomenon is very important and it is partly why, when a simulation is made, only a rough estimation of the studied performance criteria is needed.

The only way of validating a model consists in performing measurements on the real system and checking the validity of the results. Obviously, this solution is possible only if the system is already set up. In the contrary case, results obtained from models or from prototypes can make it possible to partially validate the results presented.

If the simulation is validated (the traditional techniques to do it are indicated in the following paragraph) and the simulation results and the measurements do not coincide, it is then necessary to modify the model to take into account some factors which were neglected initially.

- **Level of the study :**

This point is according to us particularly critical during the use of sophisticated simulation tools for which one piles up models which are extracted from libraries of models.

This problem appeared in a particularly crucial way in ATM networks. The level of study is related to a particular scale of time. The various difficulties which arise in the networks are related to these time scales: for example when the studied problem concerns a call acceptance policy, the time scale of interest is the one of the calls and it is not necessary to take into account the binary error rates on optical fiber. When the problem concerns cell loss rates, calls cannot be simulated: a simulation of an ATM switch during a simulated time of a few seconds generally takes approximately one day; a phone call lasts on average 3 minutes.

This difficulty will arise in an increasingly crucial way when multimedia networks are simulated; it is not reasonable to want to obtain performance criteria of application level by using sophisticated propagation models in satellite systems. A solution consists in running several simulations, in each one of them, one can as well as possible take one or two levels of study.

This problem is often hidden by the commercial simulation software. Indeed, modeling consists in designing hierarchical models, each one of these models represents for example a layer of protocol quickly making simulation impossible in a reasonable time.

- **Confidence intervals :**

The determination of the confidence intervals is one of the keys-points in computer simulations. It can even be said that the objective of a computer simulation is less to provide a value of the sought performance criteria than to give an interval in which the required value is likely to be found.

A computer simulation is an experiment and, as in physical sciences, it is necessary to know the confidence which can be given to the presented results.

6.4.7. SIMULATION METHODS

Several techniques are then possible to determine the confidence intervals:

- "batch mean" method
- " replication " method
- method of the renewal points

These techniques are implemented in given tools for simulation. The theoretical results on the confidence intervals show that they are in general proportional to the standard deviation of the

estimator. The proportion depends on the method employed, but the main problem comes from the fact that, generally, the variance of the estimator cannot be computed but can only be estimated. Obviously, it is necessary that the simulated processes are strictly and asymptotically stationary of the second order so that the estimator converges towards the performance criterion and that it admits a finite variance.

- **The « batch mean » method :**

This technique consists in cutting out the simulation in several blocks with identical duration. The performance criteria are then estimated on each block and their arithmetic mean gives the estimate of the performance criterion.

Le Gall showed that under such conditions the variance of the estimator was inversely proportional to the duration of the simulation. Consequently, the variance of the estimator can be estimated over the duration of a block by using the empirical variance and from it an estimator of the variance of the estimator over the duration of simulation is then derived.

Let us note that the confidence interval is inversely proportional to the square root of the duration of simulation.
The interest of this method is also that it makes it possible to reduce the transient period since only the results obtained at the beginning of the first block are rejected. On the other hand, in the formula of the empirical variance, one makes the assumption of independence between the various blocks of simulation what is of course only one approximation and imposes blocks of rather significant duration.

- **Replication method :**

In this technique, one does not run a simulation but N simulations of identical duration. These simulations are run for example by changing the initial conditions and/or by changing seeds of the generator of pseudo-random numbers. The results are then supposed to be independent from each other. One uses the central limit theorem then to affirm that the arithmetic mean tends towards a normal variable; this method leads then to the same result as previously but by using different arguments.

It is theoretically always possible to run a great number of independent simulations, and to consider the average of the results, but *practically it is quite unsafe to affirm that two simulations are independent*. In addition, it is in general very expensive to simulate for each block a transient period. It is undoubtedly the principal problem arising from this type of method.

One can still reduce the confidence interval by making assumptions on the estimator of one of the blocks of the simulation. While supposing for example that they are normally distributed, one can replace the coefficient obtained using the central limit theorem by a coefficient of Student to (n-1) degrees of freedom. We do not recommend this kind of methods because they are based on not easily verifiable assumptions.

- **The renewal point method :**

This method, particularly rigorous, was developed by Crane, Iglehart and Fishman. A renewal point is an instant T such as the evolution of the system starting from T is

independent of the preceding evolution of the system. The principle then consists in analyzing the occupation periods between two renewal points and using the fact that there is independence between these various periods. The performance criteria are then estimated over the period between two renewal points, multiplied by the duration of this period and divided by the average duration between two renewal points.

These quantities can simply be estimated by an arithmetic mean of the obtained results.

The disadvantage of this method is to lengthen simulation especially if the periods of occupation are very long. In addition, it is not always as simple as in the case of a file on Poissonnian arrivals to choose the renewal points of the system.

- **Rare events :**

Often arises during simulations the problem, of the rare events (described in more details in [4]). In ATM networks the loss rates should not exceed 10^{-9}. A loss thus requires the generation of about 10^{9} events, which considerably lengthens the simulation duration, which can then be very large (until several centuries!). Several solutions appear to run simulations with rare events.

Speed-up techniques are based on the principle of the variance reduction. As earlier indicated, the estimate of the confidence interval for a given estimator is derived from the variance of this estimator. The smaller this one is, the weaker the number of events necessary to obtain a good confidence interval is. The idea of the speed-up techniques is thus to decrease the variance by introducing bias or by sampling only over the interesting data.
Importance sampling and RESTART are two techniques based on some statistical properties which tend to reduce the number of non-rare event in comparison with a simulation without speed-up, and thus to speed-up the simulation.

- **Importance sampling :**It is a speed-up technique developed at the origin for the calculation of the integrals by the Monte-Carlo method. It can be used in any simulation using a calculation of an integral or can be adapted for some other cases. The principle is the following one.

  Let us assume that the probability of a rare event is estimated by :

$$\gamma = \int_{-\infty}^{+\infty} 1_{\{x \in A\}} p(x)dx = E_p\left[1_{\{x \in A\}}\right],$$

  where 1 is the index function. The standard estimation method of this estimator is the average:

$$\hat{\gamma}_n = \frac{1}{N}\sum_{n=1}^{N} 1_{\{x \in A\}}(X_n)$$

  The importance sampling method consists in rewriting this integral :

$$\gamma = \int_{-\infty}^{+\infty} 1_{\{x \in A\}} p(x)dx = \int_{-\infty}^{+\infty} 1_{\{x \in A\}} \frac{p(x)}{p'(x)} p'(x)dx = E_{p'}\left[1_{\{X \in A\}} L(X)\right],$$

where $L(X)=p(x)/p'(x)$ is the likelihood ratio and p' is another "well chosen" probability density. The estimator to be calculated is then

$$\hat{\gamma}_n(p') = \frac{1}{N} \sum_{n=1}^{N} 1_{\{x \in A\}}(Y_n) L(Y_n),$$

where Y_n are generated according to the law p' . This sampling procedure is sometimes called change of measure. As any non-zero density can be appropriate for a change of measure, the question is to know which one is optimal, i.e. which one minimizes the variance of the new estimator $\hat{\gamma}_n(p')$. Let

$$\forall x \in A, p'(x) = p^*(x) = \frac{p(x)}{\gamma}$$

and $p^*(x)$ always equal to zero except for the values in A, then

$$\hat{\gamma}_n(p') = \frac{1}{N} \sum_{n=1}^{N} 1_{\{x \in A\}}(Y_n) L(Y_n) = \frac{1}{N} \sum_{n=1}^{N} 1_{\{x \in A\}}(Y_n) \gamma = \frac{N}{N} \gamma = \gamma.$$

This estimator is then equal to γ with a probability equal to 1 and a variance equal to zero. That is to say that when γ is known, it is easy to estimate its value by simulation ! Naturally, a rough idea of the value of γ can be known, which permits to chose judiciously a "proper function" p', with a rather small variance for the new estimator.

- **The RESTART method :** The RESTART method is based on the following idea described in [5]. Being given a rare event A of which one wants to estimate the probability by simulation, an event C is defined which is necessary to the event A and such as $1 >> P(C) >> P(A)$. The probability for A to happen is then : $P(A) = P(A/C) \times P(C)$. Usually, the estimation of P(C) is better than the one of P(A/C) because the estimated value of P(C) is obtained from the entire simulation, whereas the one of P(A/C) is only estimated from a small part of the simulation during which the event C arises. If it is possible to force the event C to occur more often during the simulation, then the confidence of P(A/C), and consequently the confidence of P(A), is better.

  We assume that a system can be modeled by a function S(t) ; that the state A corresponds to a time interval during which S(t) is greater than a given threshold L ; that an other intermediary threshold T smaller than L can be defined, allowing to define the event C as a time interval during which S(t) is greater than T ; and lastly that C=[B ;D].

  The RESTART method consists in restarting the simulation. When an event B occurs, the state of the system is recorded, and, when the event D occurs, this recorded state is restored and the C interval is then simulated again. This repetition is done R times, given R successive trials for which S(t) is greater than the intermediary threshold. The R repetitions being achieved, the simulation is continued as usually, with S(t) which can be

less or equal to the intermediary threshold until a new event B occurs. Then, the process of the R repetitions is started again. This method permits to estimate P(C) and P(A/C).

- **The Becker and Douillet 's method :** This method is based on the work in [6], and uses the same principle that the preceding one, but is applied in a hierarchical way. Let be give a rare event z of which the probability is to be estimated, by conditioning z by y and x and by assuming that y implies z and x implies y:

$$P(z)=P(z/y).P(y/x).P(x).$$

If S is a system whose probabilities $P(e_j)$ of the various possible states e_j spread out in various states of rarity, the choice of these intermediaries can be automated in the following way. A frequency of threshold f_L is fixed, as well as a size N largely exceeding $1/ f_L$ but leading nevertheless to a realizable simulation, called level 0. The subsystem S1 of the states e_j whose actual frequencies $f(e_j)$ are lower than f_L, or who are immediate predecessors of such states is considered. Then : $f(e_j)>f_L$ for all e_j not being in S1 and $f(S1) \geq f_L$; these events are thus acceptable estimators.

New simulations are then run (the level n°1), concerning only the sub-system S1. In the same way, S2 is found, and so on until that the conditional probability of the state of interest is greater than f_L. To obtain the wanted confidence interval, the global size of the simulation has only to be increased.

Hybrid methods can also be used. These techniques consist in including in simulations the mathematical models and their resolutions. For example, a complex system can be broken up into subsystems analytically soluble, the global system being then simulated in substituting the subsystems by the resolutions of their mathematical models (cf paragraph on aggregation methods). In a simulation of a queue network including a sub-network with N queues of priority FAFS (First Arrived First Served), in input of which the traffic is poissonian and of which the servers have exponential service times, the probability of the number of customers in each queue of the sub-network at the stationary state is known and given by the Jackson theorem. This sub-network can then be replaced by its mathematical model in the simulation.

The drawback of these methods is that they are very specific to given systems : with the Jackson theorem can only be modeled "Jackson networks".

6.5. LOCAL AREA NETWORKS

In this chapter, we are going to concentrate on the traffic analysis specific to Local Area Networks, as they were described in previous sections. As said before, the principal feature of this kind of network is their relatively short geographical span and the usually higher throughput of their lines. LANs can be differentiated, in terms of performances evaluation, by their topology but the major distinction between them is undoubtedly the used MAC protocol. Of course, higher layer protocols are not without impact on performance parameters, but we are not planning to deal with them in this document.

Methods for simulating LANs vary widely in terms of complexity, from very simple models that monitor global variables using a simple set of parameters, to very complex models that

are able to monitor the evolution of individual packets. As such, we will first begin by introducing a set of very common techniques for LAN simulation, before moving to more MAC-protocol specific notions.

Readers should note that, as said in the beginning of this part, we plan only to modelize the Ethernet LAN. Therefore we will only go into details for this protocol.

6.5.1. EFFECTIVE CHANNEL LENGTH

One way to predict performances of a MAC protocol implementing mechanisms such as CSMA/CD and token passing is to monitor what Stalling in calls the effective channel length, *a*.

Let us define the following parameters :

- *D* is the data transmission rate (in bit/s)
- *P* : end-to-end propagation delay
- *L* : packet length (in bits)

Then we have $a = D * P / L$

We are now assuming that both the propagation delay and packet length are constant. The value of a then specifies an upper bound for the channel utilization of a LAN, stating that no more than one packet is to be found on the network at any time (something which should be useful to modelize CSMA / CD-based protocols). Further, the fact that propagation delay is constant means that no matter the physical location of the originating node, the propagation time of a packet remains the same. Also, we will operate under the additional assumption that transmission is perfect, i.e. there is no error and as soon as the transmission is complete, another waiting message can be sent.

The degree of utilization of the network is then given by :

$$U = throughput / D$$
$$= 1 / (1 + a)$$

Hence the utilization is inversely proportional to the effective channel length (a result that is relatively intuitive).

It is obvious then that small values of a parameter are desirable for good performances in single-packet networks. This is further enhanced by the fact that CSMA / CD protocols will likely add overheads (for example, time allocated for resolving collisions, which is very likely to happen under heavy traffic conditions), which might turn our upper bound condition as over-conservative and resulting in a loss of bandwidth.

6.5.2. SIMPLE MODELS FOR CSMA / CD

As said in the conclusion of the previous section, the assumptions that we have made in order to investigate on of the interesting performance of LAN networks could be relaxed in order to give more credibility to the models.

First, CSMA / CD introduce overheads that were neglected previously. We assume that we place our Ethernet network under heavy traffic conditions. For this, time is split into equal length slots, the length being determined as twice the end-to-end propagation delay in the network (as seen in previous chapters, this is a common situation in real random-access networks such as Ethernet). This is the maximum time required to detect a collision.

The throughput S is given by the following equation :

$$S = \textit{packet transmission time / cycle time}$$
$$= 1 / (1 + a / N)$$

Let us determine the values of packet transmission time and cycle time. This first extremely simple to spot that transmission (in case it is successful) takes $1 / 2a$ slots. Cycle time includes the possibility that collisions can have occurred (which is very likely if we monitor huge networks). All nodes can emit with probability p in each slot time. We can send split cycle into two different intervals :

- a successful transmission, with transmission interval $T = 1 / 2a$
- a contention interval, where either no one emits or collisions occur

We can then modify equation into

$$S = 1 / (a + a / N)$$

The average length of a contention interval is given by the formula (A stands for the probability that any one node emits, while all other N –1 nodes are silent; and Pr [C = i] is the probability of having i contention slots, due to either collisions or no transmissions, followed

$$\overline{C} = \sum_{i=1}^{\infty} i - \Pr[C = i]$$

by a successful transmission) :

$$= (1 – A) / A$$

hence we have

$$S = \frac{1}{1 + 2a(1 - A) / A}$$

Using equations, we can successfully calculate maximum throughput for CSMA / CD, provided that we assign values to the parameters a, N and p.

6.5.3. CSMA / CD ANALYSIS

In the previous section, we have monitored the performances of CSMA / CD under heavy traffic conditions. Now, we plan to develop analyses that will allow us to perform more detailed investigations of the mechanisms that affect CSMA / CD performances.

The first task will be to determine the throughput of a basic random access method for which collision detection is implicit, under the assumption that propagation time is negligible. A similar case is explored by Tanenbaum [7] in the well-known ALOHA system. A successful reception of a message is acknowledged, meaning that no acknowledgment within a certain time-out period implicitly points out to a collision. The colliding packets are then retransmitted at a latter interval chosen randomly.

We assume that traffic is generated according to Poisson distribution with average rate λ.

The total traffic on the channel consists of packets attempting transmission for the first time, and packets trying to be retransmitted, this time without collision. Let the average rate of total traffic be γ packets/s. If the packet length is *x*, then we have the throughput S and offered load *G* given by the equations :

$$S = \lambda x$$
$$G = \gamma x$$

The probability that k packets are transmitted in *2x* seconds is :

$$P = \frac{(2G)^k}{k!} e^{-2G}$$

The throughput is equivalent to the combined probability that exactly one packet is transmitted in *x* seconds, and that no packets are transmitted in *2x* seconds. The former probability is simply *G*, and the latter is calculated using equation above. This gives us :

$$S = Ge^{-2G}$$

This is for what has been referred to as “Pure ALOHA” in the literature. Another option exists called “Slotted ALOHA”, in which transmission of a packet is restricted to occur on a slot boundary (the length of slot being in this case *x* seconds). The throughput is then for slotted ALOHA :

$$S = Ge^{-G}$$

The traffic delay is the time elapsed between the first transmission attempt and the successful receipt acknowledgment. Let us call *a* the end-to-end propagation delay, and *r* the random delay before retransmission if a collision occurs. Then the time taken by an unsuccessful transmission attempt is given by *1 + 2a + r*, whereas a successful attempt costs only *1 + a*. This is further enhanced by the fact that they can be several retransmissions attempts, let *k* be that number.

The mean delay can then be given by the following equation :

$$E[D] = E[k](1 + 2a + \overline{r}) + 1 + a$$

We can simplify this expression by noting that in pure ALOHA the probability that a message does not suffer from a collision is e^{-2G}, thus :

$$E[k] = \frac{1 - e^{-2G}}{e^{-2G}}$$

And equation can be simplified as follows :

$$E[D] = e^{2G} + a(2e^{2G} - 1) + \bar{r}(e^{2G} - 1)$$

These results may seem strange, but they are in fact quite logical. The offered load consists of new messages as well as old messages being retransmitted. When the offered load exceeds that which gives the maximum throughput, the utilization becomes increasingly small. This causes a surge in collisions as the offered load increases, which in turn causes the mean delay to increase as well. Meanwhile, of course, throughput is reduced.

This means however that this system exhibits a form of instability, for there are two possible values for delay at any time. In order to determine the effective value for a given time, it is also necessary to know whether the offered load is high or low. It is then tempting to say that networks subject to important traffic should not use random-access-based protocols.

CSMA / CD, however, was proposed as an improvement of those kinds of methods, by forcing each node to *sense* if the channel is already busy before sending packets. The details of this mechanism were already given in section , so we are not going to present them again. As there exists a variety of different ways to implement this mechanism, we will restrict ourselves to the most common variant, known as non-persistent method. Please note that the following analysis follows closely the one provided by Vo-Dai in [8].

As usual, the throughput is given by the ratio between transmission time and cycle time. Transmission time is given by equation , and cycle time can be easily derived. If we call n the number of unsuccessful transmissions, we have :

$$S = \frac{T}{E[n+1] - E[I] + E[n] - E[B] + T}$$

We said earlier that transmissions were restricted to slot boundaries (in the case of slotted ALOHA), which in turn implies that the collision window is equal to the end-to-end propagation delay *A*. We also assumed that traffic followed Poisson distribution law. Hence the probability of a collision *p* :

$$p = \int_0^1 p(t)dt$$

The probability that there is no collision *q* is the corollary of *p* :

$$q = 1 - p = e^{-\gamma A}$$

And the number of unsuccessful transmissions is then given by :

$$E[n] = e^{\gamma A} - 1$$

We know that idle periods are periods in which no transmissions occur. If the total traffic is γ and follows a Poisson distribution, the inter-arrival times are :

$$E[I] = \frac{1}{\gamma}$$

A busy period can be described as a period in which collisions occur between messages. The length of a contention interval *X* is the time for the transmitting node to detect the collision. Of course, this means that the detection must occur before the collision window has passed. Once collision has been detected, the jamming signal of duration σ is transmitted and the busy period is completed with the two end-to-end propagation delays required for this jamming signal to reach all existing stations to inform them of the collision. Thus :

$$E[B] = E[X] + 2A + \sigma$$

The expected length of the contention interval *E[X]* can be determined by spotting the fact that *X* is a random variable with distribution function *F(x*), which is further defined s the probability function that there is a collision in the interval *(0, x)*, given that there is more than one arrival in the interval *(0, A).* Thus :

$$F(x) = \frac{\sum_{k=1}^{\infty} \left(\frac{x}{A}\right)^k \frac{(\gamma A)^k}{k!} e^{-\gamma A}}{1 - e^{-\gamma A}} = \frac{e^{\gamma x} - 1}{e^{\gamma A} - 1}$$

We have to consider that the previous argument applies when the length of a contention interval is smaller than the collision window. The complete distribution function is then given by :

$$F(x) = \begin{cases} 0 & , for\ x < 0 \\ \dfrac{e^{\gamma x} - 1}{e^{\gamma A} - 1} & , for\ 0 \le x \le A \\ 1 & , for\ A < x \end{cases}$$

Which gives us also the density function, and the mean

$$f(x) = \begin{cases} \dfrac{\gamma e^{\gamma x}}{e^{\gamma A} - 1} & , for\ 0 \le x \le A \\ 0 & , otherwise \end{cases}$$

$$E[X] = \int_0^1 x \frac{\lambda e^{\lambda x}}{e^{\lambda A} - 1} dx$$

We now have all the elements to go back to equation , which gives us :

$$S = \frac{Ge^{-aG}}{aG + e^{-aG}(1+G) + G(1-e^{-aG})(2a+\sigma)}$$

A last parameter worth considering may be the mean delay of traffic in CSMA / CD networks. For this, we need to determine the expected number of retransmissions for a message. This given by p_f / p_s, where p_f is the probability of failure in transmission, and p_s the probability of success in transmission. Hence, the mean delay is

$$\overline{D} = (\frac{G}{S} - 1)\overline{d_r} + 1 + a$$

We will then examine the mean retransmission delay, and the mean delay caused by a collision. The value of the former is given by the following formula :

$$\overline{d_r} = S\bar{r} + (1-S)[p_c \overline{d_c} + (1-p_c)\bar{r}]$$

The value of the latter is directly linked to the retransmission algorithm chosen once a collision has been detected. Here we will assume that messages are retransmitted as if the channel has been sensed busy such that :

$$\overline{d_c} = \bar{r}$$

6.6. WIDE AREA NETWORKS

Now we have discussed a good part of the techniques used to modelize LANs, we will present in the present section techniques that are used when larger networks, namely WANs, are studied. The greatest difference between those two kinds of techniques is the introduction of the notions of switching and routing in WANs, which hold determinant impact on their performances.

6.6.1. INTERCONNECTED LANS

In practice, a number of WANs (including the famous Internet) can be described as a collection of LANs interconnected with each other. In a previous chapter, we describe how such multiple-segment networks could be modeled by using discrete-time Markov chain analysis. This technique can be applied to interconnected LANs without much difficulty.

Suppose that we deal only with interconnected CSMA / CD segments, as is often the case in real networks. The conditional probabilities for the slot analysis can be denoted as follows :

- *SI(s)* : probability to be in an idle state
- *S(s)* : probability to have a successful transmission from a node
- *Ss(s)* : probability to have a successful transmission from a switch

given that the state of the system at the beginning of the slot was *s*.

Results in terms of performances are very dependent on the CSMA / CD algorithm implemented. For example, between the various algorithms 1-persistent, p-persistent, and optimal-ALOHA.

For this latter algorithm, we obtain (where *k* is the number of busy users, and *p* the probability that a transmission is attempted):

$$\begin{cases} SI(S) = 1 - kp(1-p)^{k-1} \\ S(s) = \dfrac{i}{k} kp(1-p)^{k-1} \\ Ss(s) = \dfrac{u(j)}{k} kp(1-p)^{k-1} \end{cases}$$

Of course, one can found various combinations of LANs using different implementations of the CSMA / CD method interconnected with each other, and even LANs that do not use the CSMA / CD system at all, but rather a token-passing protocol such as Token Ring or FDDI.

Also, we have done a certain number of assumptions concerning the switches, by considering that switching process takes place instantaneously and that the switch can deal with all the queues connected to it simultaneously. In real networks, it is likely that we will encounter an overhead due to the processing time of these tasks.

6.6.2. SWITCHES, BRIDGES AND ROUTERS

Switches, bridges, routers and repeaters serve as interconnection means between various networks. Repeaters should normally not introduce any overhead, so they won't be deal with in this document (in reality, this is not completely true, but overhead is usually negligible compared to other sources of delay. If it is not the case, it can only indicate a major problem in the network). The other three kinds of devices make decisions about the forwarding of packets, which makes them prime candidates for monitoring traffic flows. The connection of these links in a non-uniform pattern are known as mesh topology networks, and analysis of traffic on such topologies can be forbidding, depending on the complexity of the topology itself.

We have already presented the queuing/buffering mechanisms related to a node in a previous section. As the task of defining the switching policies of all different mesh topologies is a task that is vastly beyond our goals, we will illustrate instead the analysis technique on a particular example, using the node analysis we developed earlier.

We concentrate on the topology depicted in Figure 7 below.

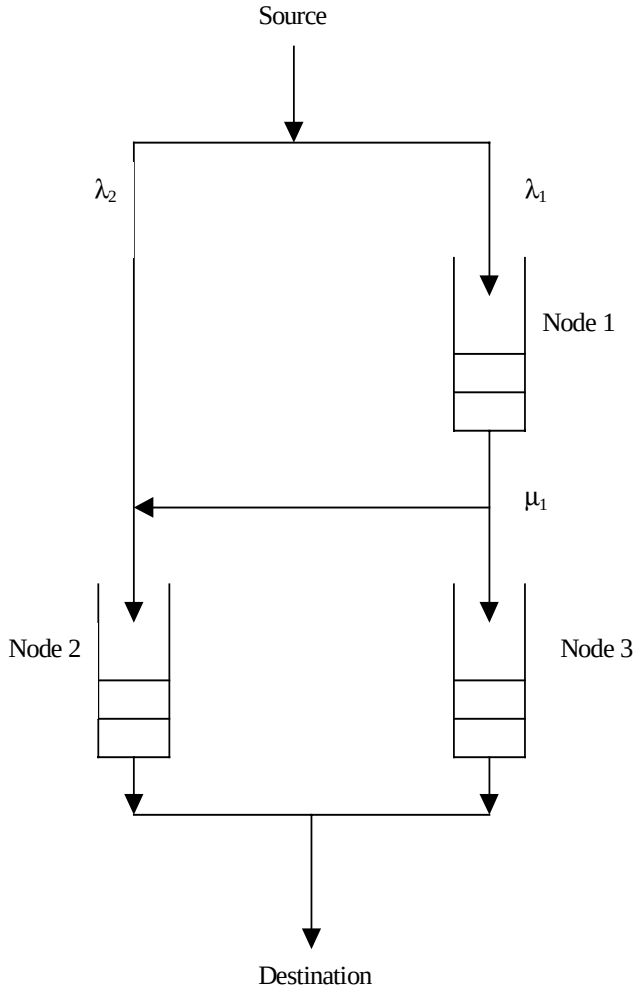


Figure 7 : Sample Two-node Network

The dynamics of this network can be described with three integers that represent the number of messages stored on node 1, 2 and 3 respectively. Thus the state *(i, j, k)* indicates that there are *i* packets stored in the input queue of node 1, *j* packets in node 2 and *k* packets in node 3. We will design as λ*x* and μ*x* the packet arrival and departure rate corresponding to node *x*. (One can remark that those rates are not independent from each other, as the departure rate of one node can be the arrival rate of another. However, as shown in, this can be a legitimate claim in certain circumstances).

The analysis is very similar to the one we encountered during the presentation of the Markov chain sequences. The only "difference" is that we specified the Markov chains as models for systems with one state variable, and thus a one-dimensioned associated transition diagram. Here, we manipulate three different variables, and thus a three-dimensioned transition diagram (see below). However, this isn't much in terms of added complexity.

The balance equation for state *(i, j, k)* is :

$$(\lambda_1+\lambda_2+\lambda_3+\mu_1+\mu_2+\mu_3)P(i,j,k)=\mu_1P(i+1,j,k)+\mu_2P(i,j+1,k)+\mu_3P(i,j,k+1)$$
$$+\lambda_1P(i-1,j,k)+\lambda_2P(i,j-1,k)+\lambda_3P(i,j,k-1)$$

which has been resolved in :

$$P(i,j,k)=\left(\frac{\lambda_1}{\mu_1}\right)^i\left(\frac{\lambda_2}{\mu_2}\right)^j\left(\frac{\lambda_3}{\mu_3}\right)^k P(0,0,0)$$

and

$$P(0,0,0)=(1-\frac{\lambda_1}{\mu_1})(1-\frac{\lambda_2}{\mu_2})(1-\frac{\lambda_3}{\mu_3})$$

Hence, if one can obtain/estimate the average arrival and departure rate of a particular network, then the average statistics for this network can be computed.

6.6.3. SWITCHING

Switching the path for data from one output channel to another is one of the most basic and important techniques in WANs. As such, it has enormous impact on traffic statistics, which justifies the fact that we spend time in this section to study the phenomenon in greater details.

We will restrain ourselves in this chapter to the description of traditional circuit-switched networks, a switch being represented very simply as shown in Figure 8, as a box with *N* input and *N* output channels. Other kinds of switches (such as time-division switches), or techniques (such as fast packet switching), will be either presented later on (for the former case), or will not be presented due to their complexity (for the latter, interested readers are strongly advised to look at [9] and [10] for an exhaustive presentation).

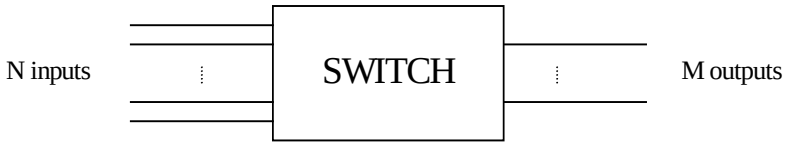


Figure 8 : Switch with different number of input/output channels

6.6.3.1. OUTPUT BUFFERING

When incoming packets are switched immediately, each output requires some buffering mechanism in order to store the incoming traffic from more than one source, as depicted below.

If we plan to form a Markov type model of this system, a number of parameters have to be defined :

- each buffer is able to store *N* packets
- a_x is the possibility that x packets arrive
- the state of the system is defined by the number of packets in a particular buffer

The balance equations can then be written as follows :

$$p_i = \begin{cases} \frac{1}{a_0}(1-a_0-a_1)p_0 & , for\ i=1 \\ \frac{1}{a_0}[(1-a_1)p_{i-1} - \sum_{j=2}^{n+1} a_j p_{i-j+1} & , for\ 1<i<N \\ \frac{1}{a_0}\sum_{j=1}^{N-1}\sum_{k=1}^{j} a_{N-k+1}\, p_j & , for\ i=N \end{cases}$$

It is assumed that arrivals are random and memoryless, then the transition probabilities can be modeled using the binomial distribution, which gives us (*p* being the probability that a packet is generated in a slot) :

$$a_i = \binom{N}{i}(p/N)^i(\frac{1-p}{N})^{N-i}$$

Probability of a packet loss is then given by :

$$p_L = 1-\frac{\rho}{p}$$

The mean queuing function is given has been resolved in,

$$\overline{Q} = \frac{N-1}{N} * \frac{p^2}{2(1-p)}$$

6.6.3.2.INPUT BUFFERING

In the previous section, we have placed the buffer at the output channels. Of course, associating them with the input channels is also possible. In this configuration, the switch operates by inspecting the destination address of the head packet in each queue, and by determining the corresponding output channel.

It is relatively safe to assume that the speed of each input and the processing speed of the switch itself will be equal. In this case, the switch imposes a maximum throughput for the system. Karol et al [10]. Have described in a method to calculate this maximum throughput. If we call ρ the output for queue *i*, we obtain :

$$\rho = \sum_{i=1}^{N} \binom{N}{k} \left(\frac{p}{N}\right)^{k} \left(\frac{1-p}{N}\right)^{n-k}$$

As expected, this shows that the throughput of the switch is reduced when the number of input/output pairs increases.

Throughput can be improved in a variety of ways. First, it could be possible for a switch to know the length of each input buffer, select the longest and give priority to the processing of packets in this particular queue. Another less simple, but perhaps more efficient solution is traffic smoothing (as described in), in which the buffer collects a fixed number *b* of packets, before accessing the switch. All *b* packets are then switched at the same time. Such a method effectively requires a *Nb * Nb* matrix of cross points, and it implies in terms of space division that its physical size will be considerably increased.

In this latter case, we obtain :

$$\rho = \frac{1}{bp}[b - \sum_{k=0}^{b-1} (b-k) \binom{Nb}{k} \left(\frac{p}{N}\right)^{k} \left(\frac{1-p}{N}\right)^{Nb-k}]$$

$$L = \frac{1}{bp} \sum_{k=b+1}^{Nb} (k-b) \binom{Nb}{k} \left(\frac{p}{N}\right)^{k} \left(\frac{1-p}{N}\right)^{Nb-k}$$

This allows us to conclude that, at least under certain conditions, the smoothing scheme is the input queuing method that provides the best throughput. This only seems natural, as this technique increases the size of the switch, and thus reduces the blocking probability and the associated packet loss.

6.6.3.3.SHARED MEMORY

A third method for switch buffering is shared memory. This case is similar to output buffering in that all new arrivals are immediately presented to the switch crosspoints. If, in any particular slot, more than one arrival is destined to the same output port, then one packet is transferred to that port while the remainder enter a pool of memory that is shared by all the ports. This memory is fed back to the input of the switch so that during the next slot both the new arrivals at the input port and the previously blocked packets contend for access to the outputs. Shared memory is a method of switching that offers the throughput performance of output buffering while reducing the buffer space needed.

We will follow the configuration for shared memory described in Figure 9 (though other configurations are known to exist). The size of the switch is then $(N + 1)^2$. If the shared memory is large enough, all the traffic will eventually arrive at the required destination port and the saturation throughput is the same as far output queuing. Even though the memory space has to be finite, the fact that it is shared means that it will be more efficiently used that dedicated memory spaces of output buffering.

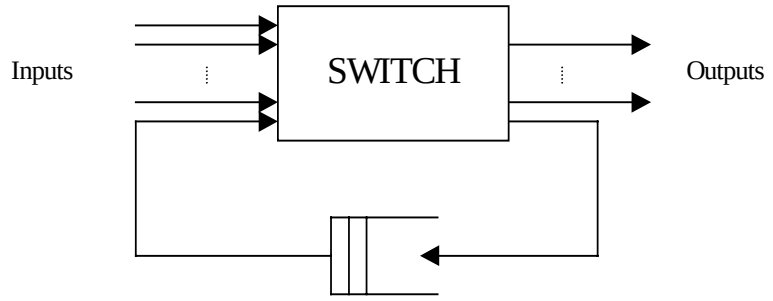


Figure 9 : Switch with shared input and output memory

A simple analytic problem for shared buffering is available if it is assumed that the buffer space is infinite. In this case the number of packets in the system destined for output i at the end of slot *m* (noted) may be noted in terms of the new arrivals A^i_m, as

$$Q^i_m = \max[0, Q^i_{m-1} + A^i_m - 1]$$

Which implies for the mean queuing delay :

$$\overline{Q} = \sum_{i=1}^{N} Q^i$$

6.6.4. TELECOMMUNICATION SYSTEMS

Now that we have detailed the various aspects of switch modeling, it is time to find a practical subject where they can be put to use. As such, telecommunication networks consist of different types of switches interconnected with each other in a variety of configurations.

Two main types of switches exist : space division switches and time division switches (which use a combination of time division multiplexing and slot interchanging). Another additional kind of switches may be frequency division multiplexing switches, but from a traffic analysis point of view they can be described as space division switches.

The goal of teletraffic engineering is to analyze the systems based on those two kinds of switches. As such, it is then useful to list those different kinds of systems.

6.6.4.1. BLOCKED CALL SYSTEMS

Until now, we have always considered that for a switch, the number of output channels was equal to the number of input channels. In reality, this is not always the case. As output channels are not busy all the time, it is then possible (and economical) to install on a switch less output channels than input ones. This is known as a blocked call system.

Such systems can be modeled using a standard *M / M / m / m* queue model, which depicts an infinite source model. This may seems unrealistic, but it is in fact a good approximation when one has to modelize cases where the number of input sources is very large (such as in telephony for example).

Another model, more accurate, is the *M / M / m / m / s* queue model. The probability that a call is blocked is given by :

$$p_m = \frac{\left(\frac{\lambda}{\mu}\right)^m \frac{1}{m!}}{\sum_{k=0}^{m}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!}}$$

which is known as Erlang's B formula (or loss formula).

Furthermore, we have :

$$p_n = \left(\frac{\lambda}{\mu}\right)^n \binom{s}{n} p_0$$

$$p_0 = \frac{1}{\sum_{n=0}^{m}\left(\frac{\lambda}{\mu}\right)^n \binom{s}{n}}$$

Please note that these three equations are the results commonly known as Engset distribution.

One should note that switches with reduced number of outputs are cascaded into switching stages. A typical example of this situation being the three-stage switch (commonly known as TST) depicted below.

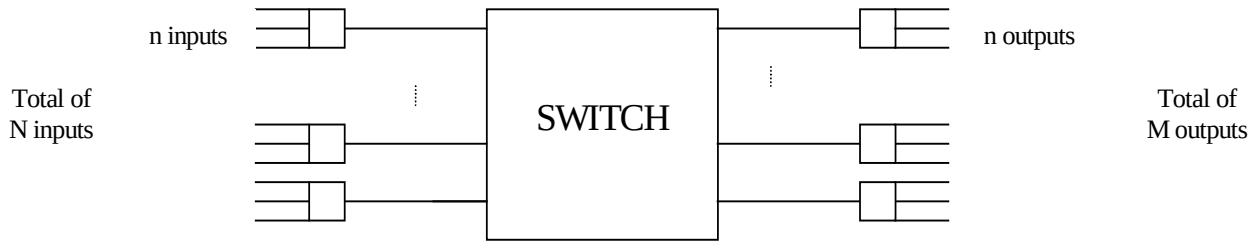


Figure 10 : three-stage TST switch

6.6.4.2.QUEUE CALL SYSTEMS

A possible refinement of the above system is to place all blocked calls in a queue. This kind of system is then referred to as queued call system.

A possible way to modelize this queue call system is to use a single-stage switch with infinite number of Poisson-type sources, infinite size queue and N channels at the output. In fact, this model is very similar to the *M / M / m / k* queue model.

This means that the birth and death formulas are given by :

$$\lambda_n = \lambda \quad , \forall n$$
$$\mu_n = \begin{cases} n\mu & , for\ n < N \\ N\mu & , for\ n \geq N \end{cases}$$

which means that we have :

$$p_n = \begin{cases} \dfrac{\lambda^n}{\mu^n}\dfrac{1}{n!}p_0 & , for\ n < N \\ \dfrac{\lambda^n}{\mu^n}\dfrac{1}{N!N^{n-N}}p_0 & , for\ n \geq N \end{cases}$$

If we look at p_Q, which is the probability that a new call must join the queue because all trunks are busy, we have:

$$p_Q = \sum_{n=N}^{\infty} p_n = \frac{\dfrac{\lambda^N}{\mu^N}\dfrac{1}{N!}\dfrac{N}{(N-\lambda/\mu)}}{\displaystyle\sum_{n=0}^{N-1}\frac{\lambda^n}{\mu^n}\frac{1}{n!} + \frac{\lambda^N}{\mu^N}\frac{1}{N!}\frac{N}{(N-\lambda/\mu)}}$$

which is known as Erlang's C formula.

6.6.4.3.CIRCUIT SWITCHING

Calls on a telecommunication network are facilitated by means of circuit and packet switching mechanisms. A simple model for a switched two-node network. The link between node A and B is divided into n trunks. If a packet is sent by an user through node A to node B, it transits through one of these trunks, unless all are occupied, in which case it joins a queue in node A.

The circuit switching control signals (including connection request and release messages) are assumed here to be transmitted in-band. We define the following variables :

- T : total time for a call setup
- t_r : transmission time for the send request from a user to node A
- w : queuing time in node A
- t_c : transmission time for the connect message from node A to node B
- t_a : transmission time for the answer message from node B to node A
- t_s : transmission time for the start-to-send message from node A to the user

We have :

$$T = t_r + w + t_c + t_a + t_s$$

If the queue system is a purely blocking system, then the waiting time is zero, and the probability that a call is lost may become an important performance indicator. On the other hand, if the switch has queuing facilities, then the waiting time can be calculated using the method described in the previous section (for example, the probability for a message to be queued is given by Erlang's C formula $E_{2,N}(\lambda/\mu)$).

The actual waiting time can then be calculated by :

$$\overline{w} = \frac{1}{\lambda}\sum_{n=N+1}^{\infty}(n-N)\frac{\lambda^n}{\mu^n}\frac{1}{N!\,N^{n-N}}p_0 = \frac{1/\mu}{(N-\lambda/\mu)}E_{2,N}(\lambda/\mu)$$

As may have been expected, the useful capacity of the link is greater when the control signals are relatively small. Such comparisons may become important when particular signals are required to carry an unusually large amount of overhead information, as is sometimes the case for connection request signals that carry lists of addresses for routing purposes.

6.6.4.4.PACKET SWITCHING

The alternative for establishing end-to-end connections on networks is to send the data in chunks (or packets) that may travel along different paths, and thus reach their final destination at various times, and not necessarily in the same order as they were emitted. This additional difficulty makes acknowledgment messages almost mandatory.

We carry the analysis of this scenario on the same two-nodes network as in the previous section. The communication session begins with the user sending its message, or a portion of it, in a fixed-size packet to node A. As before, the packet must wait in the switching node A until a trunk becomes available. After the message is received by B, this latter node sends back to the user an acknowledgment message.

In this case, if we call t_h the transmission time of the header :

$$T = t_m + t_h + w_m + w_a + t_a$$

The waiting times for both data and acknowledgment message can be modelized using the *M/M/1* queue model. We have then :

$$\overline{w_{m,a}} = \frac{\lambda/\mu_{m,a}^2}{(1-\lambda/\mu_{m,a})}$$

$\mu_{m,a}$ being the mean departure rates for messages (either data or acknowledgments).

It is then easy to figure that when the message lengths are large, the efficiency of circuit switching is better than that of packet switching because the influence of overhead signaling becomes less significant. However, the reverse stands true when packets are small, as

overhead is then a major liability for circuit switching. This explains why in the classical debate “circuit-switching vs. Packet-switching”, the usual answer is to state that the choice should be made according to the applications (i.e. the traffic load and type) run on that network.

7. CONCLUSION

In this document, we have provided a complete survey of the characteristics of terrestrial networks (at least for the most interesting ones, both technically and commercially). As said earlier, one of the most interesting prospect in the BISANTE project is the modelization of user behaviors done in WorkPackage 1. As such, network features that should have impact on users have been give particular attention in this document. They include parameters having an impact on :

- **Delay**
- **Error rate**

which are the two main performance criteria of a network that users perceive.

We have first studied the **Local Area Networks** :

- **Token Ring**,
- **FDDI**, and
- **Ethernet**

and then the **Wide Area Networks** :

- **X.25**,
- **Frame Relay**,
- **ATM**
- and **IP**

Although sorting out the various characteristics of the terrestrial networks was an important part of our goal in the course of Task 2.1, it wasn't the only one, as this characterization phase paved the way for the modeling of those networks. Due to technical and time constraints, we choose to modelize only a few of those networks namely :

- **Ethernet,**
- **ATM**
- and **IP**

and put the others aside due to their age and / or their low performances. Once the choice of the networks was done, we provided a survey of the various modeling techniques that are used in such cases, including :

- **Mathematical Models** (Markov chains, queue models, …)

- **Computer-assisted Simulations** (event- or time-driven simulations, performance criteria and measurements, …)

and a discussion on how they could be applied to the modelization of the three networks we are interested in.

The actual building and implementation of these models is beyond the scope of this document, and will be examined in another document as part of WorkPackage 3. This is why no actual software is given nor described in the present deliverable that deals only with the characterization and modeling of terrestrial networks.

8. BIBLIOGRAPHY

[1] Kleinrock, L., *Queueing Systems, Volume I : Theory*, New York : John Wiley & Sons, 1975.

[2] Kleinrock, L., *Queueing Systems, Volume II : Computer Systems Applications*, New York : John Wiley & Sons, 1975.

[3] Bunday, Brian D., *Basic Queueing Theory*, London : Edward Arnold Ltd., 1986

[4] *B/C/M/P Theorem*, Acta Informatica 7, 35-60, 1976

[5] Ph. Heidelberger , *Fast Simulation of Rare Events in Queuing and Reliability Models* , , ACM Transactions on Modeling and Computer Simulation, Vol. 5, NO 1, pp 43-85, Jan. 1995.

[6] M. Villen and J. Villen, *« RESTART : A method for accelerating rare events simulations »*. 13th International Teletraffic Congress, Copenhague, Acts of Queuing Performance and Control in ATM. North Holland, Jun. 1991.

The same paper can be found from the web site : http://www.tid.es/presencia/publicaciones/comsid/esp/articulos/vol24/restart/restart.html ,entitled *« Método RESTART para acelerar simulaciones de sucesos infrecuentes »*.

[7] M. Becker, P.-L. Douillet, *« Une méthode hiérarchique, auto-régulée, de simulation d'événements extrêmement rares »*. C.R. Académie des Sciences, Paris, t.316, Série I, p.87-92, 1993.

[8] Tanenbaum, A. S., *Computer Networks*, second edition, Englewood Cliffs, NJ : Prentice-Hall, 1989

[9] Vo-Dai, T., *Throughput delay Analysis of the Non-Slotted and Non-Persistent CSMA/CD Protocol*, North-Holland : Local Computer Networks, 1982

[10] Karol, M. J., M. G. Hluchyj, and S. P. Morgan, *Input Versus Output Queuing on a Space-Division Packet Switch*, IEEE Transactions on Communications, Vol. COM-35, No. 12, pp. 1347-1356, Dec. 1987

[11] Hluchyj, M. G., and M. J. Karol, *Queueing in High Performance Packet Switching*, IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, pp. 1587-1597, Dec. 1988

[12] M.L. Santamaria and R. Puigjaner, Banyan ATM Switch: Grade of Service under Unbalanced Load; in Computer Networks, Architecture and Applications (C-13), S.V.Raghavan, G.V.Buchmann and G.Pujolle eds. Elsevier Science Publishers B.V. (North Holland) (1993).

[13] E. Gelenbe and G. Pujolle, The Behaviour of a Single Queue in a General Queueing Network, *Acta Informatica,* **7,**123-136.

[14] A.-L. Beylot and M. Becker Performance Analysis of Multipath ATM Switches under Correlated and Uncorrelated IBP Traffic Patterns, IFIP Conference : Performance of Information and Communications Systems, PICS'98, Lund, Sweden, 25-28 mai 1998.

[15] P. J. Kuehn, Approximate Analysis of General Queueing Networks by Decomposition, *IEEE Trans. on Comm.,* Vol. 27, NO1, pp. 113-126, 1979.

[16] P.J. Courtois, Decomposability Queueing and Computer Science Applications, ACM Monograph series, Academic Press, N.Y. (1977)

[17] C.H.Sauer and K.M.Chandy, Computer Systems Performance Modeling (Prentice Hall,Englewood Cliffs, NJ, 1981).

[18] R. Marie, An Approximate Analyical Method for General Queueing Networks,*IEEE Trans. on Software Engineering,* SE-5, NO5, pp. 530-538, Sept.79.

[19] B. Baynat and Y.Dallery, A Unified View of Product-Forn Approximation Techniques for General Closed Queueing Networks, *Performance Evaluation* 18 (1993) pp.205-224.

[20] K. Sevcik and I. Mitrani, The Distribution of Queueing Network States at Input and Output Instants. Proc. Int. Symp..Performance of Computer Systems. Vienna (1979)

[21] M. Reiser and S.S. Lavenberg, Mean Value Analysis of Closed Multichain Queueing Networks,, *J.A.C.M.,* Vol 27, NO 2, pp. 313-323, 1980.

[22] E. Gelenbe and G.Pujolle, Introduction to Queueing Networks, Wiley, (1998).

[23] J.Labetoulle and G.Pujolle, A Study of Flows through Virtual Circuits Computer Networks, *Computer Networks,* **5,** 119-126.

[24] E. Gelenbe, On Approximate Computer Systems Models, *J. ACM,* **22,** 261-269.