Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics

Performance Evaluation, Modelling and Optimisation of Call Processing in Broadband Networks

Sándor Székely

Ph.D. Dissertation

Advisors:

Sándor Molnár, Ph.D.
Béla Frajka, Ph.D.

High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary
2004

Sándor Székely
High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
mailto: szekely@ttt-atm.ttt.bme.hu
sandor.szekely@siemens.com

Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Média Informatikai Tanszék

Szélessávú hálózatokbeli hívások jelzésrendszerének teljesítményelemzése, modellezése és optimalizálása

Székely Sándor

Doktori Értekezés

Tudományos vezetők:

Dr. Molnár Sándor
Dr. Frajka Béla

Nagysebességű Hálózatok Laboratóriuma
Távközlési és Média Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Budapest
2004

KIVONAT

A jelen dolgozatban megfogalmazott eredmények röviden a következőképpen foglalhatóak össze:

Elsőként az ATM jelzés-forgalom sajátos tulajdonságait fogalmaztam meg pont-pont összeköttetésekre, amelyeknek az eddigiekben talán mindössze 20%-a volt ismert. Közben fényt derítettem a korábbi vizsgálati módszerek hiányosságaira, kidolgoztam új elemzési módszereket, majd megalkottam egy, az ATM jelzés-forgalmat leíró új csomópont-modellt, amely ezen új tulajdonságokon alapszik. Ezen modell bonyolultsága miatt később egy egyszerűsített modellt is készítettem analitikus vizsgálatok céljából. Vizsgálataimat rengeteg mérési eredménnyel támasztottam alá. Vizsgálatom végső célja a jelzés-forgalom leírása volt bonyolult hálózati topológiák esetében.

Továbbá, a már megalkotott ATM jelzés-modellt illesztettem AAL2-es típusú kapcsolókra, melyek az UMTS hálózatok gerincét alkotják, majd kidolgoztam két optimalizáló algoritmust, melyek egy adott kapacítású AAL2 processzor erőforrásait osztják el úgy, hogy közben a hívás-felépítési időket minimalizálják, akár priorításos akár priorítás-mentes jelzési üzenetek esetén.

Végül bemutattam egy olyan új mechanizmust, amellyel kiegészülve a jelenleg szabványosított jelzési protokollok képesek a hálózatban blokkolt szélessávú hívásokat a hálózat szélén sorbaállítani, majd egy megfelelő idő múlva újra beengedni a hálózatba, feljavítva ezen hívások blokkolási valószínűségét. Fenti tény igazolására analitikus és szimulációs vizsgálatokat végeztem.

ABSTRACT

The results of the current work can be briefly described as follows:

First of all, I have formulated the intrinsic properties of ATM signalling for point-to-point connections. Out of these properties a maximum of 20% was probably known before. During my investigations I have pointed out the deficiencies of previous research methods, I have introduced new evaluation methods and performance metrics, then I have constructed a new node-model for ATM signalling based on these intrinsic properties. Due to the complexity of this model, later I have constructed a simplified model as well for analytical studies. My investigations have been validated with a huge amount of measurement results. The final objective of my investigation was to obtain a proper description of the broadband signalling traffic in arbitrarily complex network topology.

Furthermore, I have adjusted this ATM signalling model to fit the characteristics of AAL2 switches, which form the basis of UMTS networks. Then, I have constructed two optimisation algorithms that (re)allocate the resources of an AAL2 processor with fixed capacity in order to minimise the call establishment times, regardless of FIFO or priority queueing of signalling messages, respectively.

Finally, I have presented an enhancement to the current signalling standards. Based on this mechanism the signalling protocol can insert the previously blocked wide-band calls into an additional queue at the access nodes of the network. Later these calls will be automatically retransmitted, thus reducing the blocking probability of the wide-band calls. My results have been validated by analytical and simulation studies.

To my friend

Acknowledgements

A number of persons have directly or indirectly contributed to this work. To all these people I express my deepest gratitude.

First of all, I would like to express my deepest appreciation to Tamás Henk, head of the High Speed Networks Laboratory, for his continuous support since 1995. He was the first person who believed that this research subject is suitable for me and never gave up encouraging me, that this work should come to a mature state.

I would like to express my sincerest gratitude and appreciation to my main advisor, Sándor Molnár for his support in the last 3 years. The current form of this thesis has been reached due to his helpful suggestions, inspiration and critical comments. I am especially grateful to my second advisor, Béla Frajka, for his encouraging support in the first years of my Ph.D. studies (1995-1999).

My special thanks go to Harry Perros, Arne Nilsson, János Sztrik and to the anonymous reviewers for their precious comments and reviews of some of my publications.

I had the opportunity to spend several months as a visiting researcher at the technical universities in Karlsruhe, Stuttgart and Munich. I would like to acknowledge Georg Carle, Gerhard Krüger, Paul J. Kühn and Jörg Eberspächer for their support and supervision during these periods.

I am especially grateful to my colleague, István Moldován for the co-design and implementation of a new simulator extensively used in this work, called ACCEPT ©, and for the numerous revisions of this software during all these years.

Very special thanks to my friends and colleagues, Gábor Szűcs, Csaba Simon, Ferenc Fábián, Krisztián Németh and their students for the invaluable assistance while configuring the testing environments, preparing the measurements, running the tests and processing the results.

I am also thankful to my colleagues, István Cselényi, Csaba Antal, Krisztof Szarkowicz, István Szabó and Gábor Fodor for the long and fruitful discussions about designing solutions in ATM and UMTS networks and about queueing in broadband networks, respectively.

I am very thankful to my family. Throughout the years of my study, they provided me a stable background and there is no doubt that their love and patience made it possible for me to carry out my research.

Last but not least, special thanks are due to Géza Gordos, former head of the department of Telecommunications and Telematics at BUTE and to Miklós Boda, head of the R&D department at Ericsson Hungary, for their continuous support during my Ph.D. studies, to Peter Braun, former head of the System Integration Team at Siemens AG, Munich, for his support in the last 3 years I was with this company, and to the Hungarian Ministry of Education for a four years scholarship between 1995 and 1999.

Budapest,
November 3, 2003 Sándor Székely

Table of Contents

List of Figures

List of Tables

CHAPTER 1

1 General Introduction

The beginning of the 1990's brought new technologies in the telecommunication networks. Asynchronous Transfer Mode (ATM) has been chosen as the transmission technology of broadband networks [Pri95]. The standardisation of ATM is practically finished today, the implementations are ready and the ATM equipments are largely introduced by users and service providers. Last years brought however many addendums to early versions of the protocols, or even some new versions, e.g., the ATM signalling protocol of the user-network interface UNI4.0 (developed in 1996) was updated to SIG4.1 version in 2002.

1.1 The place of the ATM Technology today

The ATM has been widely recognised today as the key switching and multiplexing technology in bringing true multimedia and high bandwidth integrated services networking into reality. Indeed, ATM has the potential to provide voice, data, text and high quality video services due to its flexibility in terms of network resource (e.g. bandwidth) allocation for, and the control of the so called virtual channel connections, VCCs. In order to utilise the bearer- and call control potentials of its VCC, the need for a more efficient and advanced signalling protocol - in many respects different from those standardised today – was claimed by [Min91], but it was never reconsidered by the standardisation institutes. Some disadvantages of the early standard signalling protocols for ATM (e.g., monolithic view for message definitions, need of supplementary services for multiparty and multimedia calls, difference between the user-network interface (UNI) and network-node interface (NNI) protocols, etc.) have been identified in many papers, see e.g., [Dow96], [Kes92]. Furthermore, existing protocols still lack the flexibility and modularity to support easy introduction of new features and require new versions (or Addendums) of the protocol to be defined each time [C-2].

At the end of the 1990's the Internet Protocol (IP) became the fastest growing network layer protocol that is applicable over any data link layer. The convergence of the ATM and IP technologies became reality at the beginning of the year 2000, but a couple of months later the telecommunication market entered the deepest recession ever seen. After 3 years since that time are there now the first signs visible for investments in new technologies. In the current marketplace the service providers must improve network management to reduce their operation costs. While in the last years the ATM network service providers offered only permanent virtual circuit connections to the customers, *today there is an increasing interest to offer switched virtual circuit connections to end users.* This later solution is based on the use of *signalling protocols*, and does not need hardware extension (additional cost) of existing equipment. A good survey of signalling protocols for ATM networks can be found in [Sti95], [Onv97] and [Blac98].

1.2 Dynamic Call Establishment in ATM Networks

Early ATM implementations employed Permanent Virtual Connections (PVCs) between communicating endpoints. It is nearly impossible to support the growing number of users with the current method of cross-connecting using PVCs, because, e.g. for n users $n(n\text{-}1)/2$ virtual paths should be established manually (eventually automated by writing script-files) by the operator. Switched Virtual Connections (SVCs) controlled by high-speed signalling protocols are now widely available allowing endpoint connections to be set up and disconnected dynamically (advantage). The signalling function of an ATM switch includes establishing, maintaining, and clearing virtual circuits (VCs). These switch operations must conform to the industry standards for network signalling. Most ATM inter-working protocols rely implicitly on the use of signalling for establishing SVCs. The ability to transfer data through an ATM network depends on the rate at which SVC connections can be set up through an ATM switch and on the time it takes for an SVC connection setup request to be propagated through a network (connection time). In many cases, the amount of time required to establish an SVC connection may be considerably greater than the time required to transfer the data over the connection. There are many factors that may influence the number of switched connections that a network can accept and the rate at which they can be accepted. Both of these performance criteria are influenced by the signalling performance of the User-Network Interface (UNI).

1.3 Complexity and performance of ATM Signalling

With the increased capability and complexity of ATM networks, call processing and especially signalling procedures become more and more complex which causes a performance degradation of the signalling networks. Moreover, the size of the signalling messages is larger and larger. Therefore the existing models for SS7 networks, e.g. [Baf93], do not represent correctly the behaviour of the signalling processing in broadband networks. *There is a need for new signalling models.* The basic *performance metrics* for ATM signalling are described in [ATMF00], while *performance benchmarking* of ATM signalling software is presented [Nie97], showing test taxonomy and some results obtained on switches from four vendors in a number of hardware, software and network configurations. These results are based on a subset of tests described in [Kaus97] and [Bat96]. Most of these test results are focused on the call establishment time as a global parameter, but they do not investigate the effect of the specific components of this parameter. Therefore they cannot answer many questions regarding performance degradation starting at a certain call arrival rate. Moreover, the performance of tearing down connections is not studied at all. It has some early results on multiple host connections, but a significant work still remained. In addition, a 'call generator software' used for ATM signalling performance evaluation is presented in [Kaus97]. Their goal was to establish a tool set for an ATM benchmarking system that is reliable, portable and easily expanded. Further experiments were also designed to address the issue of long call establishment times for point-to-multipoint connections, but the causes for this degradation in performance has not been proven yet [Nie97].

Design principles and performance analyses of a reliable transmission of layer 3 ATM signalling protocols via a new transport protocol (Service Specific Connection Oriented Protocol, SSCOP) are in focus in [Hen95]. This protocol provides error correction and guaranteed message delivery (of layer 3 messages) via selective retransmission of erroneous protocol data units. The SSCOP design is compared with other similar protocols using analytical and simulation results. An other analytic model for studying the delay performance of SSCOP is described in [Kan94], analysing this protocol with and without high priority given to management messages. We have also a contribution to this subject, namely the investigation of introducing a forward error correction (FEC) mechanisms in the signalling protocol stack instead of retransmissions (see [J-3]), but in this dissertation we will primarily focus on the measured performance of the SSCOP protocol, showing its influence on the delay performance of layer 3 messages (see Section 4.2).

Although the importance of the signalling performance of ATM networks has been recognised as a potential bottleneck in [Kim96] and [Man97], very few papers addressed the congestion situation in switches due to signalling message flows. Furthermore, the evaluation in [Kol98] is based on estimations and external inputs and expresses the necessity to have real measurements of the equipment that is going to be installed for Broadband-Video-on-Demand (B-VoD).

In [Gel97] it is argued that congestion can occur at ATM nodes due to request messages, involving path selection, routing and call establishment. In [C-4] it is shown that a batch arrival of signalling messages introduces long delays in the connection establishment time, and that the effect of the number of open calls on the connection time is negligible. In [Wu98], a queueing model of a node is given, aiming to minimise the call setup time by avoiding the potential bottleneck process via optimal capacity allocation. Some scalability issues related to a Broadband-Intelligent Network signalling system in the *ACTS* project *INSIGNIA* are reported in [Kar98] and [Schw97].

A global *methodology for testing* the signalling performance of an ATM system (which can be a single ATM switch, a part of it, or a network of switches) is first presented in [ATMF00]. Since the results of any test case may be significantly affected by the values chosen, all input parameter values for a test case must be stated along with the test results. This requirement ensures that the test results are reproducible. The signalling performance of an ATM switch is determined by the speed it can process signalling traffic. In a switch this is limited by its architecture and its call processing capacity. During my Ph.D studies I have investigated the architectures and features of existing signalling protocols and I have collected and processed industrial and academic papers related to this field. Based on these studies *I have concluded that only a few research studies are focused on practical experiences with broadband signalling networks, and even those, they do not investigate the basic components of the call establishment and release times or even worse, they deliver some ambiguous results due to some systematic errors, that were not observed* (see e.g., [Pil99], [Mau01]). In general, the delivered results are okay, but some conclusions are misleading. Furthermore, there is no consensus in using the right terminology to the performance metrics. Therefore, the first part of my dissertation focuses on detailed measurements and analysis of these networks.

1.4 Outline of the dissertation

My dissertation focuses on call processing performance evaluation, optimisation and enhancement of signalling procedures in broadband networks with a special attention to ATM, Voice over DSL (VoDSL) and third generation of mobile networks (UTRAN).

The main tasks addressed in this dissertation are the followings:

- to describe quantitatively the relationship between the main components of call processing delays in ATM networks based on a series of measurements carried out on isolated switches manufactured by different vendors;
- to develop and analyse a new method that reveals the hidden complexity of signalling burst arrivals;
- to develop and analyse new models based on the above measurements which more appropriately describe the message latencies in ATM and UMTS signalling nodes; then to use these models for designing large broadband networks;
- to develop and analyse new algorithms which lead to optimised distribution of processor resources, thus improving the performance of broadband signalling networks;
- to define mechanisms that facilitate protocol extensions, especially when introducing queueing of blocked wide-band calls in order to reduce the call blocking probability of these calls in the network.

The main focus is on ATM networks, but similar problems may appear in the UMTS Terrestrial Radio Access Networks, where ATM AAL2 has been selected as the transmission technology (see [Ene99], [C-7]). Moreover, the evolution of the Digital Subscriber Line technologies (xDSL), intelligent networks, IP-over-ATM, Voice over ATM (AAL1, AAL2), Voice over IP (H.323, SIP, MGCP), and WWW applications (RSVP) argue the introduction of the signalling capabilities of ATM networks, and therefore the need of real performance measurements of these networks [Kol98], [Mer00].

Chapter 4 focuses on the performance of call processing based on *measurements* of isolated signalling switches. The aim is to identify the main components of call processing in broadband networks and to describe quantitatively the effect of different call profiles. Point-to-point (p-to-p) single connections, p-to-p multiple connections and point-to-multipoint (p-to-mp) calls are observed and studied. Based on these results, I have identified a *set of intrinsic properties of ATM signalling* that was not discussed in the literature before. Furthermore, a new method ("population-diagram") of evaluating the results makes it easier to analyse the signalling performance of switches at their limits (e.g., overload, burst), and to eliminate the eventual bias or systematic errors of the tester or tested equipment. Finally, a guideline for a correct evaluation of signalling performance is attached to the end of Chapter 4.

Chapter 5 deals with the construction and performance analysis of a new generic call processing model based on these measurement results from Chapter 4. The analysis is extended to network level in Chapter 6, case studies for 10-node cascaded, 4-node fully meshed, 7-node, 30-node and 35-node arbitrary networks are investigated by simulation.

This new call model originally developed for ATM can be used for some other new technologies where signalling is used, e.g., as the AAL2 was adopted as standard for UMTS, we have re-designed and adapted our model to AAL2 signalling. Chapter 7 presents two algorithms to optimise the performance of an AAL2 signalling node in UMTS Terrestrial Radio Access Networks.

As already mentioned, current ATM standards need every time extensions to support new features of ATM signalling. Chapter 8 describes such an enhancement, and focuses on queueing analysis of wide-band blocked calls at the access point of broadband networks.

A summary of the dissertation and some current and possible applications of my theses are described in Chapter 9. A list of terminology is to be found in Chapter 10, followed by a list of abbreviations. Two Appendices are attached to my dissertation, the first one gives an example of capturing and decoding a SETUP message, the second one gives a short introduction to the simulation tool, called ACCEPT ©, developed by the author and his colleague, I. Moldován, and used throughout the dissertation. Finally, the related and cited publications are listed, complemented by a list of recommendations of the standardisation institutes. My own publications are shown at the end of this work.

CHAPTER 2

2 Background: An Overview of ATM Signalling

It is almost impossible to give a detailed presentation of ATM Signalling in a couple of pages. However, I will try to present briefly those parts of the signalling architecture, protocols and standards which are directly involved in my dissertation. Fore more details, I would recommend three books written in this subject [Onv97], [Blac98] and [Per01].

ATM, as a networking technology, has been gaining increasing popularity. It is widely perceived to be the underlying technology for high speed networks of the future. Its strengths are that it is highly scalable, in terms of access speeds (from as low as 1.5 Mbps to as high as 1.2 Gbps and more), in terms of geography and topology (LANs, WANs) and in terms of application traffic (voice, video and data). It also has its drawbacks, the primary one being its complexity and overheads, from the physical level (e.g. when SONET framing is used) to the ATM level (e.g. traffic and congestions control) to higher level functions (e.g. signalling).

One characteristic of ATM networks is that it is connection oriented. So before two end systems can communicate they need to establish a connection between them. Unlike circuit switched networks (e.g. telephone networks), the connection between two end points does not consume a fixed bandwidth. Instead, bandwidth is allocated statistically, so a large number of connections can share the bandwidth of individual links in the network. Since these connections are not dedicated bandwidth channels, they are referred to as *Virtual Channel Connections (VCC)*.

A VCC is the unit of connection over which user data transfer actually occurs. VCCs on a link are identified by a combination of two numbers: a 16-bit *virtual channel identifier* (VCI) and a *virtual path identifier* (VPI) which is 8-bits at the UNI and 12-bits at the NNI. Each VPI value is associated with 2^{16} VCI values. The network elements switch the cells on this connection based on the VPI and VCI value.

VCCs between two endpoints can be established in one of two ways:

- By provisioning:
  These VCCs are called *permanent virtual channel connections (PVC)*. They are established by configuring each network element along the path with the required information to establish the end to end VCC
- By signalling:
  These VCCs are called *switched virtual channel connections (SVC)*. They are established on demand by the communicating end systems, using dynamic protocol message exchange.

Since ATM networks are expected to be geographically large and topologically complex (i.e. they will have a large number of network elements with complex interconnections), it is expected that most communication over ATM networks will take place using SVCs because managing PVCs in a large network can be a nightmare. Hence, the signalling protocols needed to establish SVCs assume an important role in any complete discussion of ATM technology deployment.

Signalling is the process by which ATM users and the network exchange the control of information, request the use of network resources, or negotiate for the use of circuit parameters. The VPI/VCI pair and requested bandwidth are allocated as a result of a successful signalling exchange.

2.1 ATM Signalling Architecture

The signalling module in any system consists of the following pieces:

- *Call Control (CC):* This layer manages the resources of the system, decides whether to establish outgoing calls and whether to accept incoming calls, etc. On the user equipment, this is referred to as Host Call Control (HCC) and may perform additional functions such as routing incoming calls to the appropriate applications. On the network equipment, this is referred to as Switch Call Control (SCC) and performs additional functions such as routing calls across ATM links, call admission control and managing the switching fabric. The call control uses the services of the signalling protocol layer.

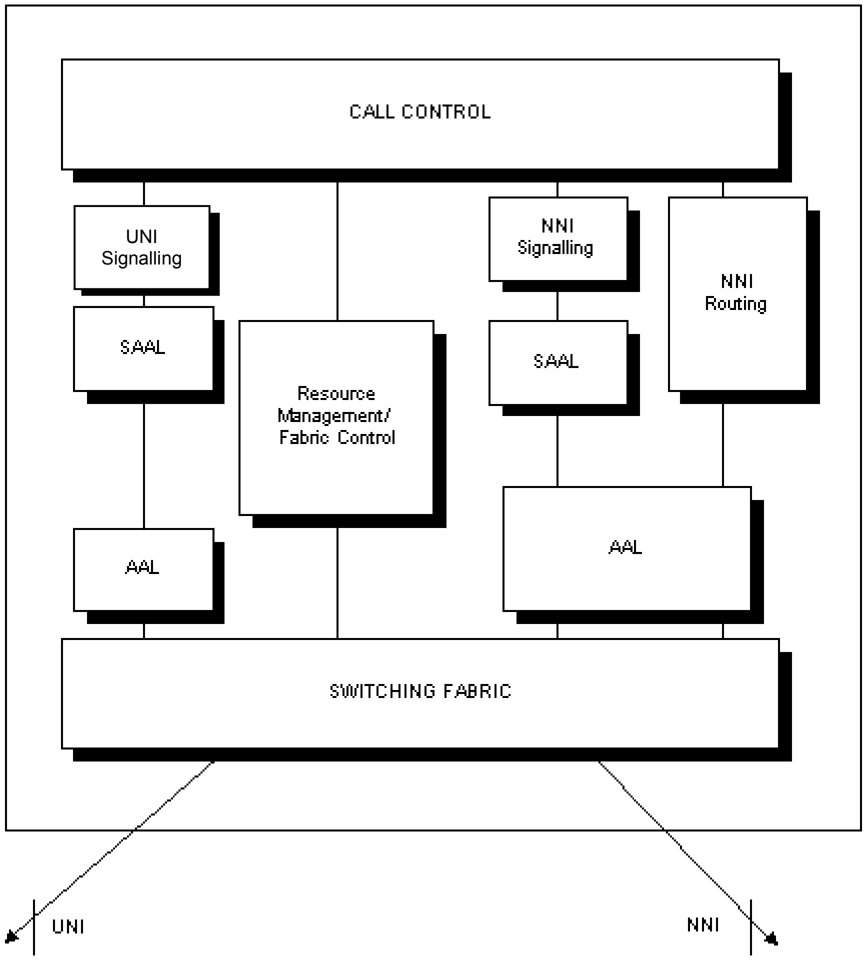


Figure 2.1 Protocol stack of an ATM switch

- *Signalling Layer 3:* This layer implements the signalling protocol necessary to establish end-to-end connections. There is one such (logical) module for each ATM link in a system. Coordination among multiple ATM links is done by call control. A signalling protocol module communicates with its peer entity sitting across an ATM link, and a chain of such modules are involved in setting up an end-to-end connection. The signalling protocol module uses the services of the SAAL layer.

- *Signalling AAL (SAAL) Layer 2:* This layer fits above the ATM layer and provides adaptation between the signalling protocol PDUs and the ATM cells. It has the following sub-parts:
    - *Segmentation and Reassembly (SAR):* This layer sits right above the ATM layer, and segments its SDUs into 48-byte ATM cell payloads on transmission, and reassembles ATM cells into SDUs on reception.
    - *Common Part Convergence Sublayer (CPCS):* This layer sits above the SAR layer. It aligns its SDUs on 48 byte boundaries (by adding padding), adds a length field and a CRC check on transmission. On reception, it verifies the length field and the CRC and strips the padding, if any.
    - *Service Specific Connection Oriented Protocol (SSCOP):* This layer sits above the CPCS layer. It implements a reliable data link protocol to provide robust transmission of signalling protocol PDUs between peer signalling entities.
    - *Service Specific Coordination Function (SSCF):* This layer sits above the SSCOP layer. It provides a mapping between the SSCOP capabilities and the needs of the signalling protocol module, using the SSCF. There is a SSCF defined for the UNI and an SSCF defined for the NNI, since the needs of the signalling protocol module are different at the two interfaces.

Typically, the SAR and CPCS layers are implemented in the hardware (in the SAR device). The functions of the ATM and PHY layers are also in hardware. The SSCOP and SSCF layers are usually in software, and are often referred to as SAAL. The signalling protocol module and other higher layers are also implemented in software. An example of the protocol stack on a switch is shown in Figure 2.1. The signalling protocol stack at the UNI side comprises of the following layers (see Figure 2.2):

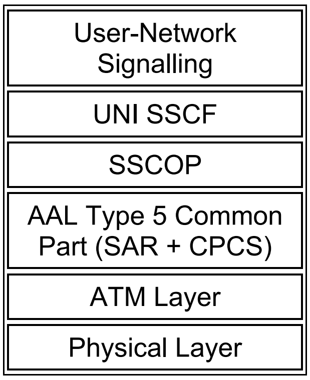


Figure 2.2 ATM signalling protocol stack at the UNI

The UNI signalling protocols are responsible for ATM call and connection control, including call establishment, call clearing, status enquiry, and point-to-multipoint control. The SSCOP layer (described in ITU-T Recommendation Q.2110) provides reliable transmission of signalling protocol messages between communicating signalling entities e.g. between user equipment and the local exchange. Signalling protocol messages are carried over a virtual channel connection on an ATM link. This VCC is usually provisioned e.g. it is a well known PVC (VPI=0, VCI=5). However, it can also be established by another signalling protocol, which is quite appropriately known as meta-signalling. This is not a commonly used procedure (*I will not use it during my investigations*). Once a signalling VCC is known at both ends of the communicating link, SAAL establishes a reliable data link over this VCC and sends signalling protocol messages reliably using sequence numbers, acknowledgements and retransmissions to ensure error-free, in sequence delivery. Furthermore, the SSCOP is a peer-to-peer protocol which provides in addition the following functions: flow control, connection control, error reporting to layer management, connection maintenance in the prolonged absence of data transfer (keep-alive), and status reporting. It uses a sliding window (or credit based) scheme for flow control which some consider to be inappropriate for ATM networks - a rate based flow control mechanism may be more appropriate. The effect of the credit window on the layer 3 signalling performance will be in our focus in Section 4.2.

SSCOP was initially intended to become a general transport layer for ATM networks, with different SSCF layers defined to map its services for different applications. SSCFs for UNI and NNI signalling have already been defined. However, recent advances in TCP optimisations for operations over high speed networks have resulted in TCP being seen as the transport layer for ATM (among other) networks. So there is less interest in using SSCOP for user data transfer. This is ironic considering that SSCOP was designed to operate over high bandwidth-delay product links, but if it is used only for signalling, then that will be a low bandwidth environment (less than 1 Mbps). In general, SSCOP borrows from previous work in high delay-bandwidth environments like satellite transmissions. Conversely, SSCOP can be used in non-ATM networks as well, as long as they have the same operating assumptions e.g. sequence integrity of frames, etc. SSCOP is not suited for operations over connectionless networks. However, given the lack of interest in using SSCOP as a general transport layer, new work items may not see the light of the day within the standardisation bodies.

The functions performed in the AAL depend upon the higher layer requirements. In short, the AAL supports all of the functions required to map information between the ATM network, and the non-ATM application that may be using it. Different adaptation layers exist to carry traffic as diverse as packet-based or isochronous (T1 or E1) over the ATM backbone. AALs are standardized in the ITU-T I.363.x series of Recommendations. The two most commonly implemented are AAL1 (per I.363.1, standardised in1993), which supports isochronous transmission — circuit emulation, for example — and AAL5 (per I.363.5, standardised in 1995), which supports carrying packet data and signalling. The AAL2, that fully satisfied the requirements for Voice-Over-ATM, had its beginnings at the May 1996 meeting of Study Group 13 in Geneva and resulted in arguably the most rapid and stable development of any Recommendation within the ITU-T, AAL2 was completed in the record time of 9 months (per I.363.2).

AAL2 goes beyond AAL1 by defining a structure that includes functions supporting higher layer requirements neither considered or possible within the structure of AAL1. AAL2 provides for the bandwidth-efficient transmission of low-rate, short, and variable packets in delay sensitive applications. It enables support for both Variable-Bit-Rate (VBR) and Constant-Bit-Rate (CBR) applications within an ATM network. VBR services enable statistical multiplexing for the higher layer requirements demanded by voice applications, such as compression, silence detection/ suppression, and idle channel removal. In addition, AAL2 enables multiple user channels on a single ATM virtual circuit and varying traffic conditions for each individual user, or channel. The structure of AAL2 also provides for the packing of short length packets into one (or more) ATM cells, and the mechanisms to recover from transmission errors. In contrast to AAL1, which has a fixed payload, AAL2 offers a variable payload within cells and across cells. This functionality provides a dramatic improvement in bandwidth efficiency over either structured or unstructured circuit emulation using AAL1. AAL2 channels are established over an ATM layer Permanent Virtual Circuit (PVC), Soft Permanent Virtual Circuit (SPVC) or Switched Virtual Circuit (SVC). In this latter case a new AAL2 Signalling is defined (see [Ene99]), whose performance optimisation is the subject in Chapter 7.

2.2 Signalling Layer 3

2.2.1 Point-to-point Connections

This is a basic connection service that is provided in all signalling protocols such as ATM Forum UNI 3.0/3.1, ITU-T Recommendation Q.2931, etc. To establish a new connection, the calling party at an ATM endpoint sends out a *SETUP* message. This message contains the following information:

- *ATM address of the called party:* This is either a public E.164 address (ISDN number) or a private ATM end system address. The latter is a 20-byte number, hierarchically structured and could be either an individual or a group address. Where sub-addressing is needed (e.g. to support connections across multiple network types), another ATM address can be specified as a sub-address.

- *Quality of Service (QoS)* for the connection: This may be specified as discrete classes of service or as specific values for parameters such as cell loss ratio (CLR), cell transfer delay (CTD), etc. These determine how the ATM cells of this connection will be handled in the network e.g. voice and video traffic is tolerant of small losses but requires strict delay and jitter bounds whereas data traffic is very sensitive to loss although it may be tolerant of delays.

- *ATM Transfer Capability:* This indicates the bearer class (or service category) of the traffic that is going to be carried by this connection. This can be Constant Bit Rate (CBR), Variable Bit Rate - real time (rt-VBR) or non-real time (nrt-VBR), Available Bit Rate (ABR) or Best Effort (UBR) service. This also specifies whether the desired service is for a virtual channel connection or a virtual path connection. Services like circuit emulation and PCM voice would use CBR services as it closely resembles traditional voice trunks. Data services like file transfer are more "elastic" and are likely to use UBR or ABR services.

- *ATM traffic descriptor:* This indicates the cell rates necessary to allocate the required bandwidth for this connection such as peak cell rates (PCR), sustainable cell rates (SCR), burst sizes (MBS), etc. It also specifies other traffic management options such as cell tagging and frame discard.

- *End-to-end parameters:* These are optional and specify parameters of significance only to the communicating end systems such as MTU size for data transfer, application protocols to use at each end, etc. These are not meaningful to the network providing the connection and are simply transported transparently by it.

- *Additional parameters:* These may be specified to indicate additional parameters such as calling party ATM address, transit network selection, endpoint reference for leaves in a point-to-multipoint connection, etc.

The flow of messages for basic call establishment and release is described below (see Figure 2.3):

i. The calling user initiates the call establishment process by sending a *SETUP* message on the assigned signalling channel. The *SETUP* message contains all the information required by the network to process the call, such as the traffic profile, QoS parameters and the destination address. Apart from the mandatory information, optional information that characterises in more detail the call, may be carried in the *SETUP* message.

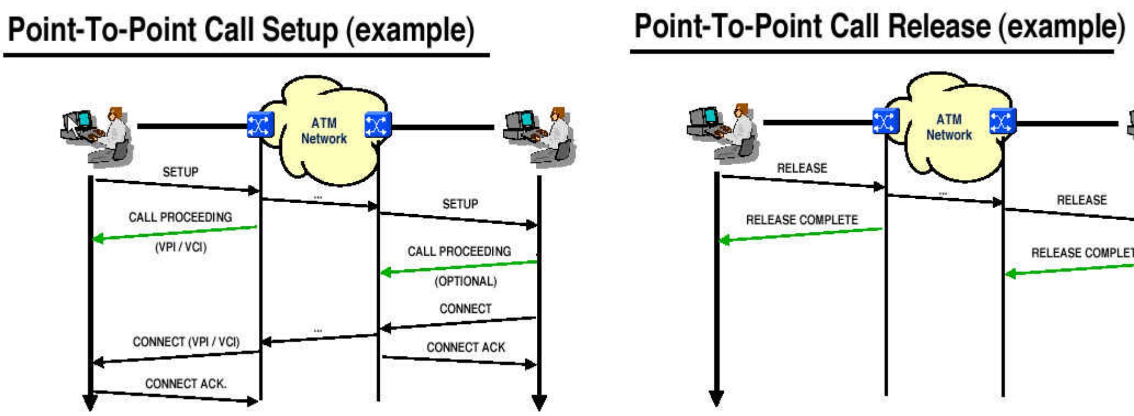


Figure 2.3 Successful call establishment and release

ii. The network node checks if the called address is contained in the topology database and selects a route to the called user. Then, it invokes the Call Admission Control (CAC), which determines if the call can be accepted. If the network can satisfy the user requirements it reserves the requested resources and replies to the calling user with the *CALL_PROCEEDING* message. The *CALL_PROCEEDING* provides to the calling user the allocated VPI/VCI identifiers and indicates that the call request is forwarded to the other end.

iii. As the *SETUP* traverses the network (private NNI or public NNI) each node allocates the required resources, selects a path from the routing tables and forwards it to the called user.

iv. When the called user receives the *SETUP* checks if sufficient information has been received and compatibility requirements have been satisfied and if decides to accept the connection responds with the *CONNECT* message. Optionally it can respond to the network either with the *CALL_PROCEEDING* or *ALERTING* message asking for more time to process the *SETUP* message.

v. A called user indicates acceptance of an incoming call by sending the *CONNECT* message to the network. When the network gets the *CONNECT* message responds with *CONNECT_ACK* to the called user and propagates the *CONNECT* back to the calling user.

vi. The calling user upon reception of *CONNECT* message is informed that the called user accepted the call and responds with the *CONNECT_ACK* message. At this point the connection between the two end-users is established successfully and both users may start transferring data.

The call release process is normally initiated by either user or in some exceptional cases by the network (see Figure 2.3):

i. An endpoint initiates the call clearing procedure by sending the *RELEASE* message to the network. The network de-allocates the reserved resources associated with the call, responds with *RELEASE COMPLETE* and forwards the release indication to the other endpoint. The *RELEASE COMPLETE* message has a local significance only and it does not imply an acknowledgment of call clearing from the remote user.

ii. The destination endpoint upon reception of the *RELEASE* message frees any resources associated with the call and responds with the *RELEASE COMPLETE*.

2.2.2 Point-to-multipoint Connections

This capability allows a unidirectional point-to-multipoint connection to be set up between a "root" user and multiple "leaf" users. Only the root can transmit on the connection. Connection parameters are established when the first leaf is joined to the connection. Additional leaves must use the same connection parameters. The connection may be setup entirely by the root or it may be initiated by the leaves. The latter capability, known as Leaf Initiated Join (LIJ), has been discontinued since the newest version of ATM Forum’s UNI4.1 [SIG 4.1].

This capability is provided in basic call/connection control in ATM Forum UNI 3.0/3.1/4.0/4.1 and ITU-T Recommendation Q.2971. The Q.2971 signalling protocol specifies the procedures for the establishment, maintenance and clearing of point-to-multipoint virtual channel calls/connections by means of Digital subscriber signalling system 2 (DSS2) at the B-ISDN user network interface. The procedures are defined in terms of messages exchanged. Q.2971 uses the same message capabilities as Q.2931. However, in addition, it also supports point-to-multipoint unidirectional switched channel connections. A point-to-multipoint virtual channel connection is a collection of associated ATM virtual channel links connecting 2 or more endpoints. Q.2971 supports only unidirectional transport from the root to the leaves.

The following are the additional messages (not used in Q.2931) used with ATM point-to-multipoint call and connection control:

- ADD PARTY, adds a party to an existing connection
- ADD PARTY ACKNOWLEDGE, acknowledges a successful ADD PARTY
- PARTY ALERTING
- ADD PARTY REJECT, indicates an unsuccessful ADD PARTY
- DROP PARTY, drops a party from an existing point-to-multipoint connection
- DROP PARTY ACKNOWLEDGE, acknowledges a successful DROP PARTY.

As it is out of the scope of my dissertation to evaluate the performance of the point-to-multipoint calls, these are not presented here in more details. The first performance results we have in this subject is to be found in [C-8]. Some other measurement results of p-to-mp calls can be found in [Nie97], [Far01] and [Mau01], but none of them reached a mature level yet, the results are not general enough. It is still some more work to be done here.

2.3 Standardisation of signalling protocols

2.3.1 UNI Signalling

The first aspect of signalling that was subject to standardization was UNI signalling. The work for defining a UNI signalling protocol for ATM networks was initially done in the ITU-TSS (formerly CCITT). The work was largely based on the existing UNI signalling protocol for narrowband ISDN (N-ISDN), as specified in Recommendation Q.931, and also referred to as Digital Service Signalling System No.1 (DSS1). The new protocol for broadband ISDN (B-ISDN) UNI signalling was designated as Digital Service Signalling System No. 2 (DSS2). The first specification for DSS2 was intended to describe the basic connection control for point-to-point connections in ATM networks and was designated Q.93B. Later, when the specification became more well defined, it was renamed to Q.2931. This was published in 1995. While the ITU was working on Q.2931, the ATM Forum took upon itself to accelerate the development of a signalling specification. Hence, in September 1993, it came out with the ATM UNI 3.0 specification, which described signalling based on a subset of Q.2931. It also added extensions for point-to-multipoint connections. These extensions were later incorporated by the ITU in a new recommendation Q.2971.

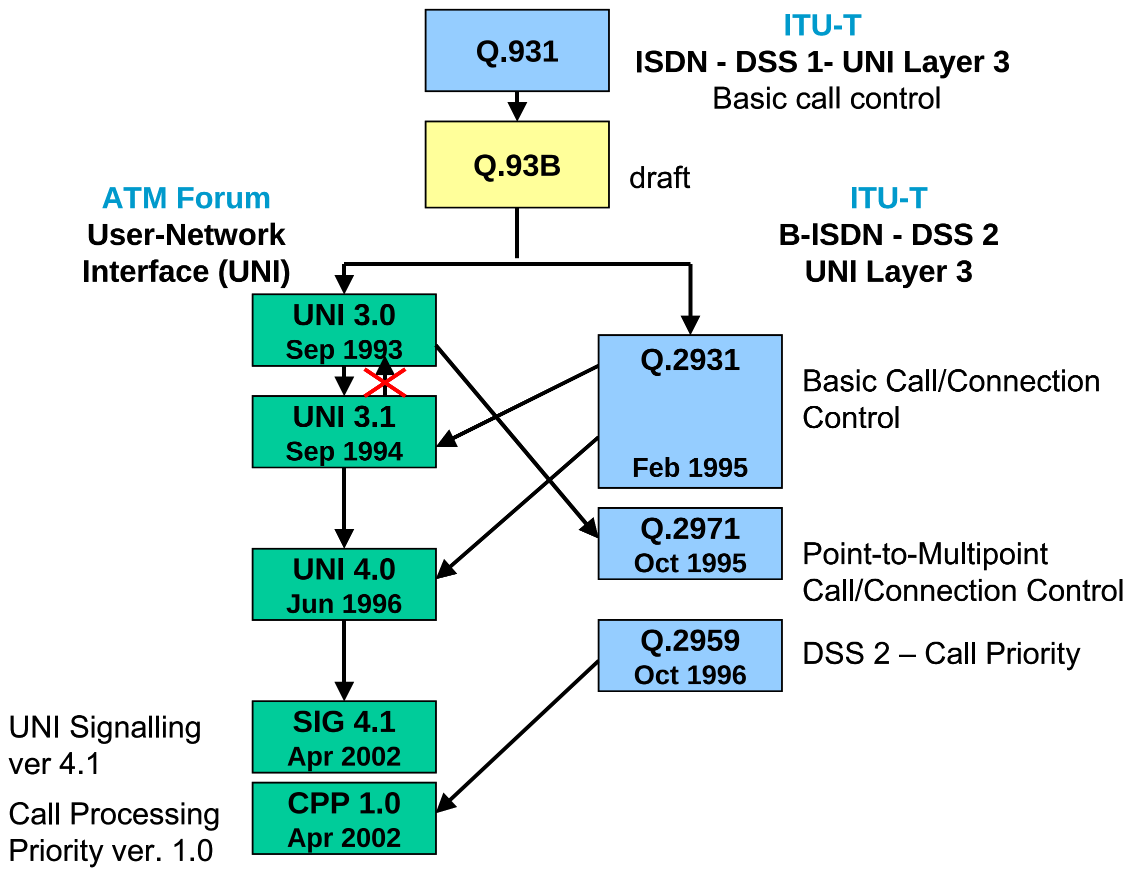


Figure 2.4 History of UNI Signalling Protocols

Since the ATM Forum wanted to stay aligned with the ITU in its signalling specifications (see Figure 2.4), it introduced another specification, ATM UNI 3.1, in September 1994, which changed a few aspects of UNI 3.0, to align it closer with Q.2931. Unfortunately, this made UNI 3.1 incompatible with UNI 3.0, without adding new capabilities. Since then the ATM Forum worked on UNI 4.0, which was approved in June 1996. This is largely based on Q.2931, with a few exceptions, the most notable being that overlap sending and receiving procedures are not supported for interworking with N-ISDN. In addition, UNI 4.0 adds many new capabilities such as leaf initiated join (LIJ), anycast, proxy signalling, etc.

Meanwhile, the ITU's strategy is to produce new specifications for new capabilities (as opposed to modifying or updating existing specifications). Hence, while Q.2931 is now frozen, work continues on other recommendations that add new capabilities such as Q.2961 (additional traffic parameters for new services), Q.2962 (parameter negotiation during call setup), Q.2963 (parameter modification after call setup), Q.2964 (look ahead feature), etc. In fact, a whole series of recommendations are to be defined in the Q.29xx series that relate to all aspects of DSS2.

The ATM Forum has decided at this point to slow work on further changes to the UNI signalling protocols. There were vendors who wished to add new capabilities such as multipoint connections, closed user groups, etc. but these changes did not happen soon. Quite recently, in April 2002, an update to ATM UNI 4.0 was approved, the so called ATM SIG 4.1. This document specifies some corrections to UNI 4.0, some additional optional capabilities, and the deletion of the optional Leaf Initiated Join capability. This and some additional current activities (working on call priority procedures, e.g. CPP 1.0) within both the ATM Forum and ITU-T shows the continuous interest in expanding the introduction of ATM signalling to the market.
In the followings, I will make a short introduction to the basic signalling recommendations: ITU-T Q.2931 and UNI 4.0/ SIG 4.1, because these protocols and the related signalling messages are extensively tested and analysed in Chapter 4.

2.3.1.1 ITU Q.2931 Signalling

This is the ITU version of signalling. The Q.2931 signalling protocol specifies the procedures for the establishment, maintenance and clearing of network connections at the B-ISDN user network interface. The procedures are defined in terms of messages exchanged. The basic capabilities supported by Q.2931 Signalling are as follows:

- Switched virtual channel connections
- Connections with symmetric or asymmetric bandwidth requirements
- Single-connection (point-to-point) calls
- Basic signalling functions via protocol messages, information elements and procedures
- Request and indication of signalling parameters
- VCI negotiation
- Out-of-band signalling for all signalling messages
- Error recovery
- Public UNI addressing formats for unique identification of ATM endpoints
- End-to-end compatibility parameter identification
- Signalling interworking with N-ISDN and provision of N-ISDN services
- Forward compatibility.

The message types for Q.2931 are the same as in UNI 3.0/3.1, with the exception of the point-to-multipoint messages which are not supported. A signalling message uses the Q.931 message format. It is made up of a message header and a variable number of Information Elements (IEs). The message header is shown in the following diagram (see Figure 2.5):

The *Protocol discriminator* distinguishes the messages for user-network call control from other messages. The *Call reference* is a unique number for every ATM connection which serves to link all signalling messages relating to the same connection. It identifies the call at the local user network interface to which the particular message applies.

Bit								
8	7	6	5	4	3	2	1	Octet
Protocol discriminator (9 for Q.2931 messages)								1
0	0	0	0	Length of call reference value				2
Flag	Call reference value							3
Call reference value (continued)								4
Call reference value (continued)								5
Message type								6
Message type (continued)								7
Message length								8
Message length (continued)								9
Variable length Information Elements as required								etc.
IE								
…								

Figure 2.5 ATM signalling header structure and Information Elements

The call reference is comprised of the call reference value and the call reference flag. The call reference flag indicates who allocated the call reference value (user or network). According to the *Message type* the message may be of the following types (see Table 2.1).

Table 2.1 Q.2931 message types

Call Establishment Messages	**Call Information Messages**	**Call Clearing Messages**	**Miscellaneous Messages**
ALERTING	USER INFORMATION	DISCONNECT	SEGMENT
CALL PROCEEDING	SUSPEND REJECT	RESTART	FACILITY
PROGRESS	RESUME REJECT	**RELEASE**	REGISTER
SETUP	HOLD	**RELEASE COMPLETE**	NOTIFY
CONNECT ACKNOWLEDGE	SUSPEND ACKNOWLEDGE	RESTART ACKNOWLEDGE	**STATUS ENQUIRY**
SETUP ACKNOWLEDGE	RESUME		CONGESTION CONTROL
CONNECT	HOLD ACKNOWLEDGE		INFORMATION
	SUSPEND		**STATUS**
	RESUME ACKNOWLEDGE		
	HOLD REJECT		
	RETRIEVE		
	RETRIEVE ACKNOWLEDGE		
	RETRIEVE REJECT		

Out of these, all messages they appear in bold in Table 2.1 have been in focus of our investigations in Chapter 4. The *Information Elements* that may occur in the Q.2931 signalling messages are listed in Table 2.2 (M=Mandatory, the rest is optional). Their effect on the latency of messages when they pass the switch has been evaluated in Section 4.1.

Table 2.2 Information Elements of the Q.2931 messages

Called party number (M)	ATM traffic descriptor (M)
Called party sub-address	Connection identifier (M)
Transit network selection	OAM traffic descriptor
Restart indicator	Quality of Service parameter (M)
Narrow-band low layer compatibility	Broadband bearer capability (M)
Narrow-band high layer compatibility	Broadband Low Layer Information
Broadband locking shift	Broadband High Layer Information
Broadband non-locking shift	End-to-end transit delay
Broadband sending complete	Notification indicator
Broadband repeat indicator	Call state
Calling party number	Progress indicator
Calling party sub-address	Narrow-band bearer capability
ATM adaptation layer parameters	Cause

2.3.1.2 ATM Forum UNI Signalling (UNI 4.0 and SIG 4.1)

UNI 4.0 provides the signalling procedures for dynamically establishing, maintaining and clearing ATM connections at the ATM User-Network Interface. UNI 4.0 applies both to Public UNI (the interface between endpoint equipment and a public network) and private UNI (the interface between endpoint equipment and a private network).

The following new features are available within the UNI 4.0 signalling protocol:

- Enhanced ATM traffic descriptor
- Available bit rate capability
- Individual QoS parameters
- Leaf initiated join (LIJ)
- Narrowband ISDN over ATM
- AnyCast capability
- New information elements
- New VPI/VCI options
- Proxy signalling capability
- Virtual UNIs
- Supplementary services such as direct dialing in, multiple subscriber number, etc.calling line identification presentation, calling line identification restriction, connected line identification presentation, connected line identification rest, user-to-user signalling
- Error handling for instruction indicators
- Using setup for adding parties
- Both NSAP and ATM end-system addresses
- Network can support leaves that do not support P-MP.

The message types for UNI 4.0 are the same as in Q.2931, with the exception of the *SETUP ACKNOWLEDGE* and *INFORMATION* messages which are not supported. The following are new signalling messages specific to UNI 4.0: *LEAF SETUP REQUEST* and *LEAF SETUP FAILURE*, but they are removed from the updated version SIG 4.1.

The ATM SIG 4.1 document (approved in April 2002) is an update to ATM UNI 4.0. This document specifies some corrections to UNI 4.0, some additional optional capabilities, and the deletion of the optional Leaf Initiated Join capability. It incorporates and supersedes: the ATM UNI 4.0, the Addendum to UNI for ABR parameter negotiation and the Addressing Addendum to ATM UNI. The SIG 4.1 is backward compatible with UNI 4.0. For this reason, this document does not specify a new protocol version (i.e. the UNI Signalling Version, as indicated by the ILMI procedures, is still 4.0). The SIG 4.1 specification also incorporates material from ITU-T Recommendations published since the release of UNI Signalling 4.0.

From our prospective, there is no difference in the content of the 8 analysed messages (see Table 2.1) when carrying out signalling measurements at the UNI interface, regardless of they belong to Q.2931, UNI4.0 or SIG 4.1.

2.3.2 Private and Public NNI Signalling

This is part of ATM Forum P-NNI 1.0/1.1 specification which specifies both a dynamic routing protocol and a signalling protocol for call establishment and release. Note that the signalling and routing protocols are typically supported by distinct entities in the protocol stack. The signalling protocol is based on an extension of UNI signalling. It uses the same SAAL as UNI signalling and extensions to the layer 3 signalling protocol to support P-NNI specific elements that are used to route the call e.g. Designated Transit Lists (*DTL*), Crankback, etc. P-NNI 1.0 signalling supports all UNI 3.1 features and some UNI 4.0 feature, e.g. ABR services, parameterised QoS, etc. It does not support some services such as LIJ.

When a call is initiated at a UNI, the ingress node into the network translates the *UNI SETUP* message to the *P-NNI SETUP* message by modifying some information elements and adding a new *DTL IE* which specifies a partial source route for that call through the network. Succeeding switches process the *DTL* to progress the call and generate new *DTL*s when required by the P-NNI hierarchy and routing protocols. If the call routing fails, the switch at which the call fails may generate crankback information so that the call may be re-routed through an alternate path, if possible. The route chosen for a particular call attempts to accommodate the connection parameters such as cell rates, QoS parameters, etc. so that the route will not fail within the network. The P-NNI routing protocol distributes such information dynamically among the switches so that each node has a partial topography of the network, using which it can generate partial source routes that conform to the traffic and QoS requirements of the connection.

On the NNI front, the ATM Forum initially decided to define just a signalling protocol (that would use static routes) since the task of defining a routing protocol was rather complex. This was initially designated P-NNI Phase 0 signalling and later renamed to Interim Inter-Switch Signalling Protocol (IISP). It was published in December,1994, as shown in Figure 2.6. Since then, the ATM Forum has completed work on the P-NNI (Phase 1) routing and signalling protocols in April 1996. The new version 1.1 of the PNNI specification is comprised of PNNI version 1.0 (April 1996), its Addendum to Soft PVC MIB (September 1996), another Addendum for ABR parameter negotiation (January 1997), PNNI v1.0 Errata and PICS (May 1997), PNNI Soft PVC Addendum Version 1.0 (July 1999), any additional technical or editorial corrections or updates. With other words, the PNNI v1.1 Signalling is based on the UNI Signalling 4.1 specification and supports all capabilities defined in UNI Signalling 4.1, except proxy signalling. PNNI 1.1 Signalling also adds new features which pertain to the use of PNNI Routing for dynamic call setup.

The ATM Inter-Network Interface protocol version 1.1 (AINI 1.1) is a document for use between ATM networks. The scope of the specification is limited to signalling that is the procedures for dynamically establishing, maintaining, and clearing ATM connections between ATM networks. The procedures are defined in terms of messages and information elements used to characterize the ATM connection. The AINI 1.1 protocol specification is based on ATM Forum PNNI Signalling version 1.1. This new version 1.1 of the AINI specification is comprised of the ATM Inter-Network Interface (AINI 1.0) Specification (July 1999) and the updates to the original text to have it reference PNNI 1.1 (and so benefit from the updates contained in that specification). One major difference between AINI 1.1 and AINI 1.0 is the discontinued specification of AINI / B-ISUP interworking. The networks on either side of the AINI may be running any protocol internally. However, the goal in defining this AINI 1.1 protocol is to facilitate interworking of one network running PNNI internally with another network running B-ISUP internally. The protocol also supports interworking of two networks running PNNI internally. The networks involved can either be ATM service provider (ASP) networks or private networks.

The first ATM Forum specifications that addressed the signalling interoperability at the public NNI interface were based on the ITU-T SS#7 (Q.7xx) and B-ISUP documents (Q.276x) completed in 1995 as B-ICI v2.0, and updated as B-ICI v2.1 in 1996 (see Figure 2.6).

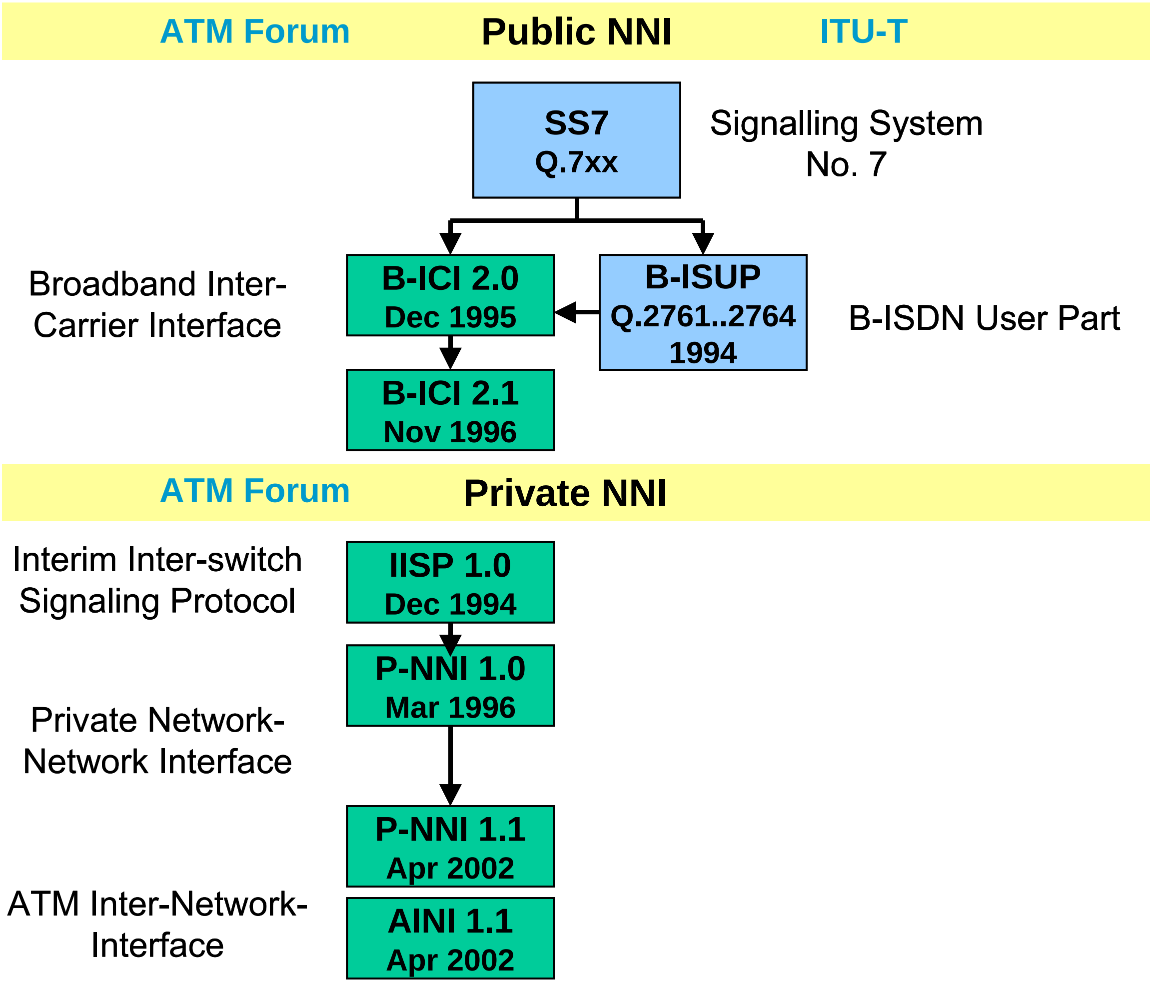


Figure 2.6 History of NNI Signalling Protocols

2.4 ATM Signalling message length distribution

In general, the *SETUP* messages have a *'default'* length (3 cells) when only the mandatory Information Elements (IE) are set, and they are *'complex'* (4, 5 or 6 cells, respectively) when, in addition to the mandatory IEs, some other optional IEs are also added.

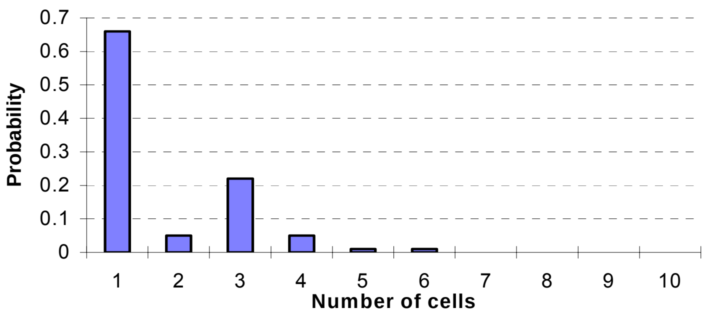


Figure 2.7 Probability distribution of signalling message length (units are given in ATM cells)

The probability distribution of message length from different signalling scenarios carried out during our tests is presented in Figure 2.7. The majority of messages have a default length of one cell (e.g., *CONNECT, RELEASE, RELEASE COMPLETE,* etc.), but a *CONNECT* message might be of 2 cells long if e.g., the *Cause IE* is longer. A *SETUP* message is typically 3 cells long, but its maximum length can reach even 6 cells, if all the optional *IEs* are selected. Most of our measurements were done using the *'default' SETUP* messages. However, we show that using *'complex' SETUP* messages the call establishment time increases considerably (e.g., see Section 4.1.4).

CHAPTER 3

3 Signalling measurements

3.1 The rationale behind signalling performance measurements

The term benchmark generally refers to a set of experiments which can be used to evaluate the performance of a piece of equipment against a standard and against of other manufacturer's equipment. The tests comprising the benchmark must be specific enough to stress the abilities of the equipment under test in various ways, but general enough to be applicable to equipment from a range of manufacturers.

The rationale for each test in our benchmark is equally important because it provides a basis for interpretation of the results and integration with the results of other tests.

A subtle aspect of the design and use of benchmarks is that the results of individual tests are often less meaningful than those from a set of tests. The reason for this is that each test generally probes a specific aspect of a complex design, and the individual aspects of a complex design can only be understood *in relation to one another*. Therefore, I have collected the results of these aspects into a group of individual statements, and these only all-together represent *a value*. These statements are *general to all investigated ATM switches*, independent of the manufacturers, production year and software release. Due to the fact that the total number of investigated switches by our team and the other research groups is above ten (which is the same order as the number of currently available types of ATM switches), I believe that my statements are general to all ATM switches as well.

Before going into details of evaluating the performance of the ATM signalling, let's have a look on some parallel works, which also investigated this subject by doing measurements (see [Bat96], [Kaus97], [Nie97], [Nov99], [Pil99], [Mau01], [Far01]). Generally, their main deficiency is that either they have carried out only a small subset of tests, or they have focused only on the measurement of the call establishment time, which is the must important metric – of course, but alone is not enough to derive general conclusions. All in all, they represent a very good basis to compare to our own results. By accident, it happened that some of them have tested the same equipment as we have done, which helped us to validate our results, but more importantly they have covered also another set of ATM switches. There are also some scenarios that we were not able to test (e.g. PNNI hierarchy), but used the results of other's measurements.

3.1.1 The complexity of the problem

To get a feeling about the complexity of the problem, let's have a look at the must important performance measure for signalling. The *call establishment time* may depend on many factors (see some of them in Table 3.1).

Many of these factors were addressed by the above mentioned papers, but not necessary proven. Not all these results were available at the time we have carried out our own measurements. From my point of view, it was important what cannot be neglected here, and secondly, what is not general to all ATM switches. As many of these factors have an influence of order of tens of milliseconds, we will neglect the effect of all those parameters which have the order of tens or hundreds of microseconds (e.g., the impact of the physical layer propagation delay on layer 3 message latencies).

So, we have focused our investigation on the parameters that have a strong influence on the signalling performance of an ATM switch. This performance was evaluated by measuring not only the call establishment time, but also the release time, release latency, throughput, destination response time, and more importantly: each message delay throughout the switch and the delay a message was acknowledged by the switch on the same interface. Thus we have covered some aspects which were not addressed by others (e.g., complexity of the *SETUP* messages, ratio between message delays, impact of the release messages on call establishment time of new calls, etc.).

Table 3.1 An overview of factors that may influence the signalling performance

Factors that may influence the call establishment time	**Investigated by**	**Order of**
• capacity of the signalling processor (HW)	[Kaus97], [Nie97]	nx1..100 [ms]
• software implementations (SW versions)	[Nie97]	nx10 [ms]
• signalling architecture (on-board/off-board)	[Kol98], [Nie97]	nx10..100 [ms]
• complexity of SETUP messages (kind of Info. Elements)	n.a.[1]	-
• call arrival rate	[Nie97]	nx1..100 [ms]
• size of burst arrivals (simultaneous calls)	[Pil99]	nx10..100 [ms]
• number of active calls	[Pil99], [Mau01]	nx1 [ms]
• WAN environment (PNNI hierarchy, routing)	[Nie97], [Far01]	nx10 [ms]
• length of the signalling path (number of switches in the path)	[Nov99], [Mau01]	nx10…100 [ms]
• speed of the physical interface	[Cyp94]	nx0.01 [ms]
• delay introduced by lower layers	[Cyp94]	nx0.1 [ms]
• parallel load, multiple hosts	[Nie97], [Mau01]	nx1 [ms]
• transmission errors, OAM messages (F4, F5)	n.a.	-
• link monitor functions (within SSCOP)	n.a.	-

3.1.2 Weaknesses of others' work

My results are in many cases in accordance with the results obtained on other switches by [Kaus97], [Nie97], [Nov99], [Pil99], [Far01] and [Mau01]. However, there are some divergences as well, and I have explained whenever necessary, why my conclusions differ from those of others. All these papers are independent of each other and unfortunately there is *no consensus between them in using the same terminology for the performance metrics or in evaluating the results*, e.g.,

• [Kaus97] and [Nie97] provides only results on call establishment times, no investigation of the message delays, usage of a software based call generator on PCs does not offer the possibility to measure delays throughout the switch, furthermore this generator supports maximum 50 calls per second, its '*SVC setup time*' corresponds to (→) the standard '*call establishment time'* (see [ATMF00]);

• [Nov99] derives the call establishment times from 'ping' (ICMP) messages on a 7-node international network, i.e,
'*call set-up time*' → difference between round-trip-time (RTT) of first and second 'ping' packet when there is no SVC but the cache of the address resolution protocol (ARP) server is still populated (not empty);

• [Pil99] is the only one who looks at the constituents of the call establishment time as well (using the same *HP BSTS 75000* tester as we did), but its definition for this metric includes the acknowledgement to the incoming *CONNECT* message as well, which differs from the standard definition [ATMF00], i.e.,
'*call setup delay*' → call establishment time + *CONNECT-to-CONNECT ACK* delay at source. The disadvantage of this type of definition will be seen later by burst measurements (see Figure 4.9);
'*call processing time*' → destination response time on *SETUP* + *CONNECT-to-CONNECT ACK* delay at switch;

[1] n.a.= not addressed

- [Mau01] uses another special software for generating calls (max. up to 20 calls/sec), and measures three different vendors switches in cascade providing no details of isolated switch measurements, and its '*setup time*' → call establishment time;
- [Far01] focuses on PNNI signalling only, observing differences in the average time spent with PNNI signalling during a UBR and a CBR call setup, respectively, its '*time spent with PNNI signalling*' → call establishment time;
- None of these works quantified the impact of the test environment (e.g., tester, call generator, flow control) on the measured results;
- We have got also a misleading conclusion in one of our previous papers, e.g., in [C-4] we concluded that the number of active calls has an influence on the call establishment time, which idea I do not share anymore today (see Section 4.1.3), furthermore that time we used a different notation for the call establishment time, i.e., '*Round Trip Time delay*' → call establishment time.

Therefore, I aimed *to obtain a consensus* between all these previous results and discussions, which is far more than a simple engineering task. As a first rule, I have adapted all previous definitions to the newly adopted [ATMF00] recommendation. Those definitions missing from this recommendation, but needed for a better characterisation of signalling performance in large network topologies will be introduced in Section 4.1 and Section 6.3.

3.1.3 Investigated ATM switches

Let me summarise a short list of the currently available ATM switches on the telecommunication market with signalling capabilities: Alcatel 7670, Alcatel (formerly Newbridge) MSX36170, Cisco BPX8620, Cisco (formerly LightStream) LS2020, Ericsson AXD301, Ericsson (formerly General DataCom) GDC APEX DV2, Lucent (formerly Ascend) CBX500, Marconi (formerly Fore) ASX200/1000, Nortel Passport7400, Siemens (formerly Seabridge) XpressPass140 and Siemens XP190, etc. Basically, there are no more than 10-20 competitive products on the world market.

Out of these, four ATM switches were used in our measurements: *GDC APEX DV2* (1995), *Fore ASX200BX* (1997), *Newbridge MSX36170* (1999) and *Seabridge XP140* (2001). This is a good representative sample, covering a range of products from 1995 to 2001. My results are presented in Section 4.1 and they are in many cases in accordance to those obtained by other research groups (see [Kaus97], [Nie97], [Nov99]) on five other switches. Divergences are also explained, whenever necessary. Moreover, recent measurement results of 2, 3 and 4 cascaded ATM switches of different types (see [Mau01], [Far01]) confirm my results.

My aim was in fact not to decide which ATM switch is faster or more suitable to a certain application, but a detailed analysis of all components of the signalling process in ATM nodes, in order to extract some generalised conclusions (i.e., set of properties). Therefore, I have conducted frame-level (layer 2 and layer 3) measurements of signalling flows, which was a very difficult and time consuming task, the evaluation of these switches has taken more than four years, basically because we had access to these switches at different times. The two newest switches ([MSX99], [XP01]) we have studied were not even manufactured at the time we have carried out our measurements on the first two switches in 1997/98 ([GDC95], [Fore97]).

3.2 The description of the testing environment

We have carried out a set of signalling performance measurements on *p-to-p single-* and *p-to-p multi-connection* (simultaneous) ATM calls. Our *p-to-mp* measurement results are not general enough yet to be included into this dissertation (but they were already published in [C-8]). As it can be concluded from Table 3.1, the main interest is on the performance of layer 3 call processing in ATM switches. The basic test configuration that we have used is the following: one call generator and one receiver (emulated by the same tester) connected to one isolated ATM signalling node. In this simple case we have set two user-network interfaces (UNI) with STM-1 (155Mbps) optical interfaces (see Figure 3.1a). For cascaded switch measurements we have connected 2, 3 and 4 switches is cascade, respectively (see Figure 3.1b). This kind of environment allows us to obtain all details (timestamps for layer 2 and layer 3 messages on both interfaces, statistics on throughput, unsuccessful calls, etc.) The other research groups either used the same testers as we did (e.g., [Pil99]), generated calls by ‘ping’s from PCs with ATM network interface cards (e.g., [Nov99]) or used special signalling softwares installed on PCs (e.g., [Nie97], [Mau01]). Using such special software on PCs offers the possibility of testing much larger network (advantage), but does not provide any detail on message delays throughout the switches (as the clock on the PCs is usually not synchronised).

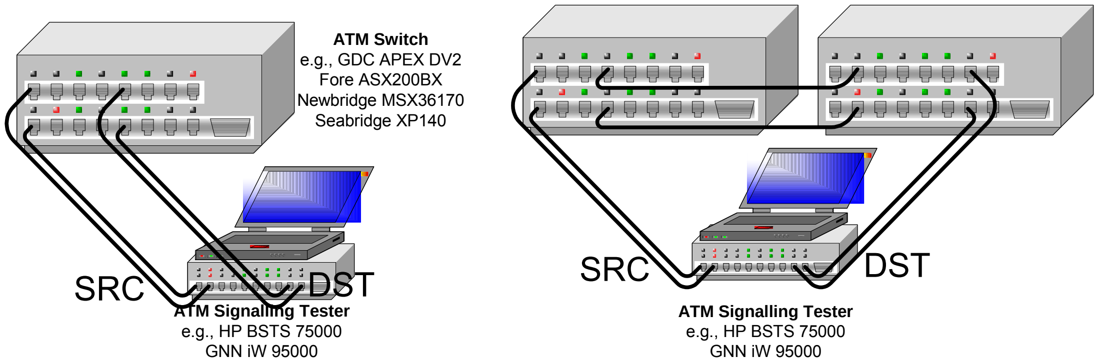


Figure 3.1 a) Isolated ATM switch test; b) Testing cascaded ATM switches

During our measurements we made some simple assumptions and configured the testing environment as such:

- we assured only successful p-to-p and p-to-mp calls (no errors during call establishment);
- we have assigned very low bandwidth requirements for calls, therefore calls were never rejected due to unavailable bandwidth on links (the user plane was not a bottleneck);
- the input pattern of the *SETUP* messages was set to constant rate or burst arrival (generation of Poisson arrival was not possible with none of our testers: *HP BSTS75000* [HP96] and later, a *GNN iW95000* [GNN00]. However, as shown by the measured results in Figure 3.2, the offered call arrival rate differs from the accepted call arrival rate (CAR), this latter varied from constant rate to on-off type starting at a certain rate due to a systematic error of the *HP75000* tester, which was caused by the link layer (SAAL layer) flow control mechanism (i.e., credit window) explained later in Section 4.2.1;
- we have found that at an offered call arrival rate of 4-to-5 calls/sec (using the HP75000), the accepted call arrival rate is becoming “nearly” poissonian with an average of 4 calls/sec (average ≈ standard deviation), as illustrated in Figure 3.2*b*.

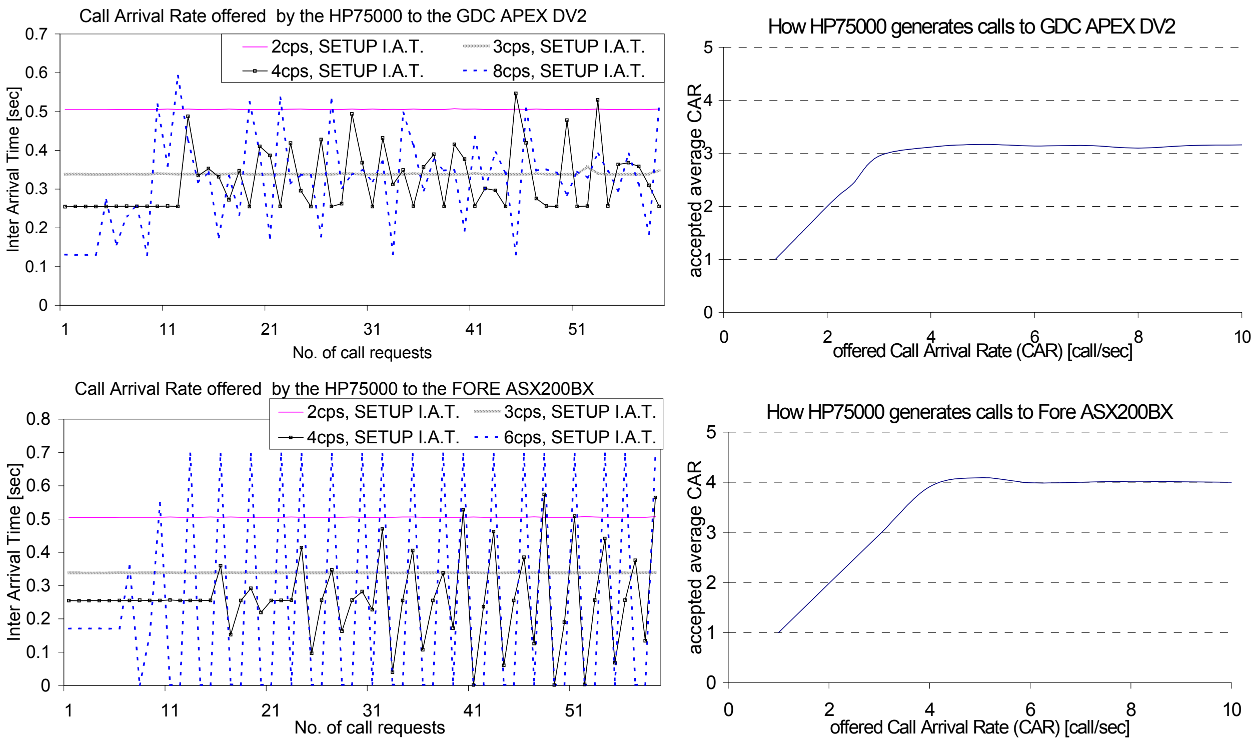


Figure 3.2 Deviations between the generated and offered call arrival rates by the HP75000 tester (see left: the inter arrival times (IAT) and right: the average of accepted call arrival rate seen on the interface)

Unfortunately, we were not aware of the problem with the *HP BSTS75000* at the time of doing the measurements (it happened in 1997/98), just a couple of years later when I have repeatedly analysed the results (in 2000), therefore we had to remove the results of this tester for call arrival rates over 4 calls/sec (the measurements of the *GDC APEX DV2* were not affected, only those of the *Fore ASX200BX* for higher rates). Instead, later (in 2001) we repeated some of these measurements with the *GNN iW95000* and additionally obtained the message delays versus the call arrival rates up to 200 calls/sec for two other switches ([MSX99], [XP01]).

3.3 Performance metrics

Most of the call performance metrics I am using throughout this dissertation are depicted in Figure 3.3 below.

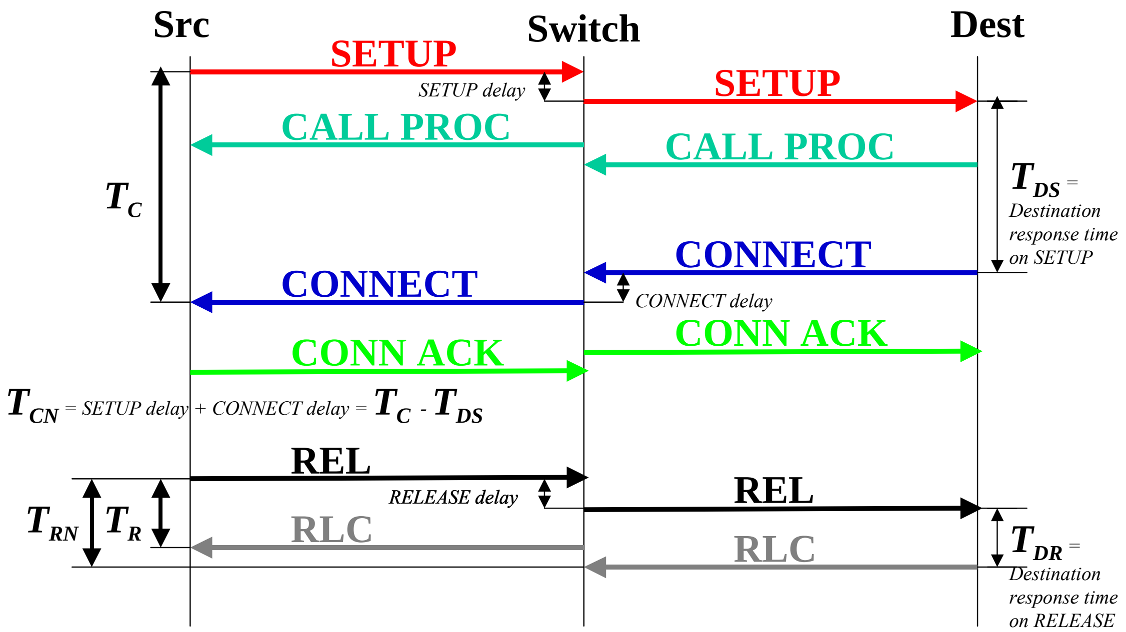


Figure 3.3 Definition of performance parameters for call establishment and release

A couple of *definitions* are given here in order to help understanding the subsequent statements, the rest is given in the *'Terminology'* Section in Chapter 10 of this dissertation. These definitions below are also to be found in [ATMF00].

1. The *call establishment time* is the amount of time that it takes for a signalling system (e.g., ATM) to establish a switched virtual connection between network components.
$$T_C = t_S\,(Y) - t_S\,(X), \tag{1}$$

where $t_S\,(.)$ is the timestamp of an outgoing or incoming message at the source; *X=SETUP, ERQ; Y=CONNECT, ECF* are signalling messages defined in [Q2931], [UNI40], resp. [AALQ99]. This is arguably the most fundamental signalling performance metric. An example of call establishment is shown in Figure 3.3.

2. The *call release time* is the amount of time taken to release a connection over one network element:
$$T_R = t_S\,(Y) - t_S\,(X), \tag{2}$$

where *X=REL,Y= RLC* are signalling messages described in [Q2931], [UNI40] and [AALQ99].

3. The *call establishment latency* is the difference between the call establishment time and the response time of the destination to a setup message (see Figure 3.3):
$$T_{CN} = T_C - T_{DS}, \tag{3}$$

where T_{DS} is the destination response time on call setup. I have found another definition for T_{CN} by [Jain97], where a so called 'MIMO latency' is involved (MIMO = Message in Message out, but it is not used here).

4. The *relative call throughput* (γ_R) of a switch (network) is the number of successful calls over the number of generated calls. By definition, the throughput is $\gamma = \lambda \cdot (1 - P_B)$, where λ is the arrival rate of calls, P_B is the blocking probability of calls. The probability of successful calls (relative throughput) is: $\gamma_R = \gamma / \lambda = 1 - P_B$.

CHAPTER 4

4 Intrinsic Properties of Point-to-Point Call Processing in ATM networks

I have analysed the signalling measurement results obtained on the four aforementioned commercial ATM switches. In this chapter I have extracted a set of intrinsic properties, which are *switch invariant features* of the ATM signalling flows and thus *general to a wide range of ATM switches.* We call it *intrinsic properties* of ATM switches those properties that are not specific to SS#7 signalling, but only to ATM signalling. We have carried out both steady state and transient state signalling measurements, the results are grouped into two sections according to these two sets.

4.1 Intrinsic properties of p-to-p single calls in ATM switches (steady state measurements)

In this section I have proven that the standard performance measures (e.g., call establishment time, release time) defined by the standardisation institutes (e.g. [ATMF00]) are not enough to properly characterise the performance of a broadband signalling network. I have shown that the impact of the destination response time, release latency, call throughput of switches, size of the routing table and complexity of call profiles (i.e., kind of information elements in *SETUP* messages) have to be considered as well. Measurements are taken after the steady-state flow of the signalling traffic has been reached. These properties are grouped into 7 independent statements, a brief overview of these statements is shown in the Table 4.1.

Table 4.1 Overview of the Section 4.1

4.1.1	The dominancy of layer 3 processing of messages + new performance measures defined
4.1.2	Differences between message processing delays
4.1.3	Dependency of the T_C on routing table, bandwidth allocation and number of active connections
4.1.4	Dependency of message processing times on different call profiles (signalling overhead)
4.1.5	The impact of the call release phase on the T_C
4.1.6	The impact of the signalling overload on the T_C and T_{RN}
4.1.7	Estimation of the T_{CN} for cascaded switches

Signalling protocol implementations in switches are primarily done in software. There are two important reasons for this choice. First, signalling protocols are quite complex with many messages, parameters and procedures. Second, signalling protocols are updated often requiring a certain amount of flexibility for upgrading field implementations. While these are two good reasons for implementing signalling protocols in software, the price paid is performance. Even with the latest processors, signalling protocol implementations are rarely capable of handling over 1000 calls/sec (status 2003).

4.1.1 The dominancy of layer 3 processing of messages

Based on measurements and analysis I have concluded that the dependency of call establishment time T_C on layer 3 (e.g., UNI4.0) processing is at least one order of magnitude higher than its dependency on lower layer processing.

Proof: In this first analysis the configuration in Figure 3.1*a* was used. 100 *SETUP* messages (containing the mandatory Information Elements only) were sent at a rate of 1 call/sec. Calls were released after a mean holding time of 10 seconds. All layer 3 UNI messages related to one call have been captured (inclusive time-stamps) at both sides of the switch. In addition, layer 2 messages were also recorded (e.g., *POLL*, *STAT*). Moreover, I have calculated the maximum transmission delay due to the ATM and physical layers (STM-1 interface) in the case of a *SETUP* message, which is the largest message.

The layer 3 processing is software implementation, layer 2 is partially implemented in software (SSCOP+SSCF, see Figure 2.2) and partially in hardware (SAR+CPCS), while layer 1 processing is implemented only in hardware. Accordingly, measurement results on five switches and calculation results for physical layer delays are given here to prove my statement (see Table 4.2).

Table 4.2 Average delays at UNI interface

	FORE ASX200BX [msec]	Seabridge XP140 [msec]	Newbridge MSX36170 [msec]	GDC APEX DV2 [msec]	Ericsson AXD311 see [Mau01] [msec]
Call establishment time	15	40	23	510	25
*Setup-to-Call Proceeding delay**	5.5	22	12.4	72.3	10.3
*Release-to-Release Complete delay**	3.3	10	8	49.1	4
*Connect-to-Connect Acknowledge delay**	2.4	19	11	68.4	3.8
Poll-to-Stat response time of the switch (layer 2)	1.5	3	2.5	12	n.a.
Poll-to-Stat response time of the end system (layer 2)	0.8	0.8	0.8	0.8	n.a.
(ATM+physical) layers transmission delay at 155Mbps (calc.)	0.011	0.011	0.011	0.011	0.011

**network side*

The delay for transmitting 3 ATM cells (i.e., a *SETUP* message) at 155 Mbps interface will consist of the propagation delay between the user equipment and the switch (practically null in our case) + the delay an ATM cell must wait to access a free slot on the physical medium (max. 53 octets) + the time needed to transmit the given PDU at the interface $\leq \dfrac{(0+53+3\cdot 53)\cdot 8bits}{155Mbps} \approx 11\mu s$. For an E3 interface with 34 Mbps transmission rate this delay will be of 50 microseconds. Finally, for an E1 interface (2 Mbps) we will have 850 microseconds. Some similar calculations on the message propagation delays, where the delay components of the SAR and SSCOP layers are decomposed as well, can be found in [Cyp94].

Unfortunately, the tester allocates the same time-stamp for all PDUs belonging to one message travelling up or down the UNI, SSCOP, and AAL5 layers. Therefore, I had to use an indirect way to prove my statement. I have compared the latencies between *SETUP* and *CALL PROCEEDING*, *CONNECT* and *CONNECT ACKNOWLEDGE*, respectively *RELEASE* and *RELEASE COMPLETE* messages (peer-to-peer communication between UNI layers, see Figure 3.3) to the latency between *POLL* and *STAT* messages in both direction (response time of the switch and end system, respectively). I have observed that the *POLL-to-STAT* delays represent less than 10% of the call establishment time for each investigated switch. It is quite obvious, that a layer 3 message suffers only a short delay to pass the layer 2 (while getting a sequence number and a few additional bytes will be added to its tail, see Figure 2.7), except one case when the layer 3 messages used up the credit window, which is discussed later in Section 4.2.1. Furthermore, layer 2 messages are not direct constituents of the call establishment time, they act periodically (e.g., T=0.75 sec), independently of layer 3 to ‘keep alive’ the peer-to-peer layer 2 communication and to exercise flow control. Usually, they do not introduce any bias in the layer 3 measurements, except the case when ending up the credit window. Therefore, as a conclusion of the statement in Section 4.1.1, in the followings (throughout the steady state measurements) I have investigated only the impact of layer 3 processing on the signalling performance of ATM switches.

New definitions

Next, I have stated that the call release time T_R *does not* provide a measurement of the time taken to tear down a connection over a call-path from end-to-end in ATM networks. Therefore, *I have introduced new performance measures* to properly characterise the signalling behaviour throughout the network:

Def 4.1 The *call release latency* is the amount of time taken for a *RELEASE* message to travel along the path from end-to-end, followed by an acknowledgement of the destination (see Figure 3.3):

$$T_{RN} = t_D\,(RLC) - t_S\,(REL), \qquad (4)$$

where $t_S\,(.)$ and $t_D\,(.)$ are the timestamp of an incoming or outgoing message at the source and at the destination, respectively.

Def 4.2 The *overall handling time* is the sum of the call establishment time and call release latency:

$$T_H = T_C + T_{RN}\,. \qquad (5)$$

The Def 4.1 does not exist in the standards (see [ATMF00]). A similar definition to Def 4.2 appears under the name of *Call Cycle Time* (see Terminology in Chapter 10), but its second term contains T_R instead of T_{RN} .

4.1.2 Differences between message processing delays

I have found that the following relationship between the minimum message delays is valid for all (tested) ATM switches regardless of the processing capacity: the CONNECT and RELEASE delays are (25...35)% of the SETUP[2] delay, respectively.

Proof: In the following tables a couple of examples are given for two different switches (GDC APEX DV2 and Fore ASX200BX) and at two different call arrival rates (1 call/sec and 2 calls/sec, respectively) in order to show the ratio between the minimum delays of *SETUP*, *CONNECT* and *RELEASE* messages, respectively (see Table 4.3*a* to Table 4.3*d*). Note that the first ten and the last ten values are excluded from the measurements, i.e. only steady-state measurements are considered. All values (except the ratios and correlation coefficients) are given in seconds. For the ratios between the minimum values please check the grey marked fields in tables.

Table 4.3a Delays for GDC APEX DV2 at offered call arrival rate of 1 call/sec (accepted rate is 0.99 call/sec)

	Setup Delay	Connect Delay	Call estab time	Release Delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.3546	0.1022	0.4602	0.1097	0.0807	0.1119	0.2882	0.3095
MAX	0.3896	0.1603	0.5362	0.1513	0.1188	0.1535		
MIN	0.3124	0.0727	0.4165	0.0791	0.0639	0.0812	0.2326	0.2533
STDEV	0.018027	0.020315	0.027546	0.019371	0.012758	0.019378		
Corr. coef. To T_C	0.7046	0.7608						
Corr. coef. To T_{CN}	0.7047	0.7608						

[2] It represents a SETUP message containing the mandatory Information Elements only, with a length of 3 ATM cells.

Table 4.3b Delays for GDC APEX DV2 at offered call arrival rate of 2 call/sec (accepted rate is 1.98 call/sec)

	Setup Delay	Connect Delay	Call estab time	Release Delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.3808	0.1113	0.4958	0.1346	0.1063	0.1368	0.2923	0.3536
MAX	0.4448	0.2005	0.6487	0.1781	0.1463	0.1802		
MIN	0.3407	0.0785	0.4492	0.0987	0.0767	0.1009	0.2303	0.2898
STDEV	0.020711	0.0223	0.033944	0.017698	0.016911	0.017696		
Corr. coef. To T_C	0.7328	0.8978						
Corr. coef. To T_{CN}	0.7401	0.8937						

Table 4.3c Delays for Fore ASX200BX at offered call arrival rate of 1 call/sec (accepted rate is 0.99 call/sec)

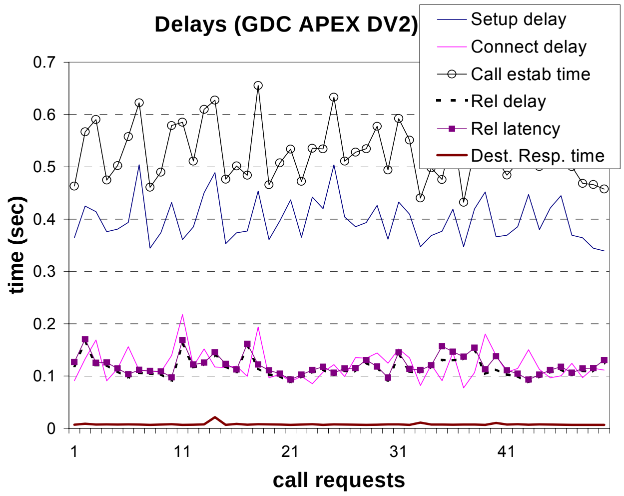
	Setup Delay	Connect Delay	Call estab time	Release Delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.0054	0.0017	0.0108	0.0018	0.0034	0.0043	0.3159	0.3408
MAX	0.0062	0.0023	0.0120	0.0021	0.0037	0.0057		
MIN	0.0049	0.0015	0.0093	0.0017	0.0033	0.0041	0.3155	0.3583
STDEV	0.000208	0.000184	0.000497	6.76E-05	6.9E-05	0.000314		
Corr. coef. To T_C	0.5656	0.6857						
Corr. coef. To T_{CN}	0.7793	0.7101						

Table 4.3d Delays for Fore ASX200BX at offered call arrival rate of 2 call/sec (accepted rate is 1.98 call/sec)

	Setup delay	Connect delay	Call estab time	Release Delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.0054	0.0017	0.0108	0.0019	0.0034	0.0043	0.3085	0.3487
MAX	0.0060	0.0022	0.0120	0.0025	0.0037	0.0055		
MIN	0.0052	0.0016	0.0105	0.0018	0.0033	0.0041	0.2988	0.3448
STDEV	0.000177	0.000126	0.000294	0.000119	8.9E-05	0.000262		
Corr. coef. To T_C	0.6614	0.4042						
Corr. coef. To T_{CN}	0.8545	0.5399						

In addition, the results show that the call establishment time and call establishment latency have a strong linear relationship with the *SETUP* delay and the *CONNECT* delay, respectively (see the correlation coefficients in Table 4.3*a*-*d*). By definition, the correlation coefficient is the ratio between the covariance of two stochastic series divided by the product of their standard deviations (e.g., see [Gel80]).

Similar results can be found in Figure 4.1, which correspond to Table 4.3*a* and Table 4.3*c*, respectively. In addition, the destination response time on *SETUP* message (T_{DS}) is plotted as well, in order to show that many times this parameter cannot be neglected. For example, while it is less than 3% of the *SETUP* delay in the case of a GDC APEX DV2 switch (thus negligible), the same response time is higher than the *SETUP* delay in the case of a FORE ASX200BX switch. Similar conclusions shall be drawn for the *RELEASE* delay versus the destination response time on *RELEASE* message (T_{DR}).



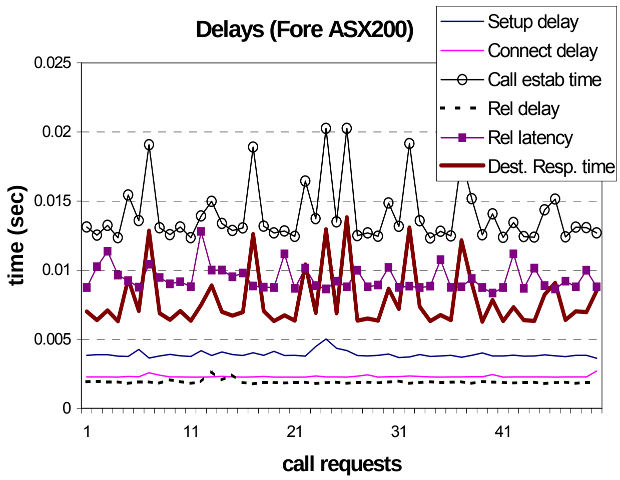


Figure 4.1 Message delays, call establishment times and release latencies at 1 call/sec load case a) GDC APEX DV2 switch; case b) FORE ASX200BX switch

Exceptions

The statement in Section 4.1.2 is valid not only for the *minimum* values, but for the *average* values as well, however not always, only for $\lambda_C < 0.9\ \mu_C$. Once the call arrival rate (λ_C) gets closer to or exceeds the service rate (μ_C) of the switch (e.g., for an accepted average call arrival rate of 2.92 calls/sec, which is higher than the service rate of the GDC APEX DV2 switch, i.e. 2.5 calls/sec), then the ratio between the message delays will suffer drastical changes (see Table 4.3*e*). Important change is first of all suffered by the *CONNECT* delay which leads to an increase of the ratio *CONNECT* delay / *SETUP* delay from 25% to 250%!

Table 4.3e Delays for GDC APEX DV2 at offered call arrival rate of 6 call/sec (accepted rate is 2.92 call/sec)

	Setup delay	Connect delay	Call estab time	Release Delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.4716	1.1973	1.6738	0.2301	0.1502	0.2347	2.5389	0.4879
MAX	0.7293	1.3052	1.8683	0.4086	0.3073	0.4112		
MIN	0.3828	0.9771	1.3939	0.1229	0.0907	0.1250	2.5526	0.3211
STDEV	0.074513	0.086273	0.108205	0.067192	0.057123	0.068093		
Corr. coef. To T_C	0.5086	0.8394						
Corr.coef. to T_{CN}	0.5127	0.8375						

More details to this surprising increase of the *CONNECT* delay can be found in Figure 4.2 below on the left side (GDC). The offered call arrival rate is increased consecutively (2, 3 and 4 calls/sec, respectively). It can be seen that once the upper limits for the GDC APEX DV2 are reached (arrival rate over 2.5 calls/sec), the *CONNECT* delay will drastically increase. In the second case (see right side) the upper limits for the Fore ASX200BX are far not reached by the given call arrival rates (2, 3 and 4 calls/sec, respectively), therefore the statement in Section 4.1.2 applies. However, it can be observed that at 4 calls/sec the average *SETUP* delay will remain still constant, but its standard deviation will increase. This is due to the already discussed problem of the HP 75000 tester, that at this rate it cannot keep the constant rate of generating calls (due to the window control mechanism of the SSCOP layer). With a GNN iW95000 tester we obtained the same picture for 100 calls/sec as for 2 calls/sec here.

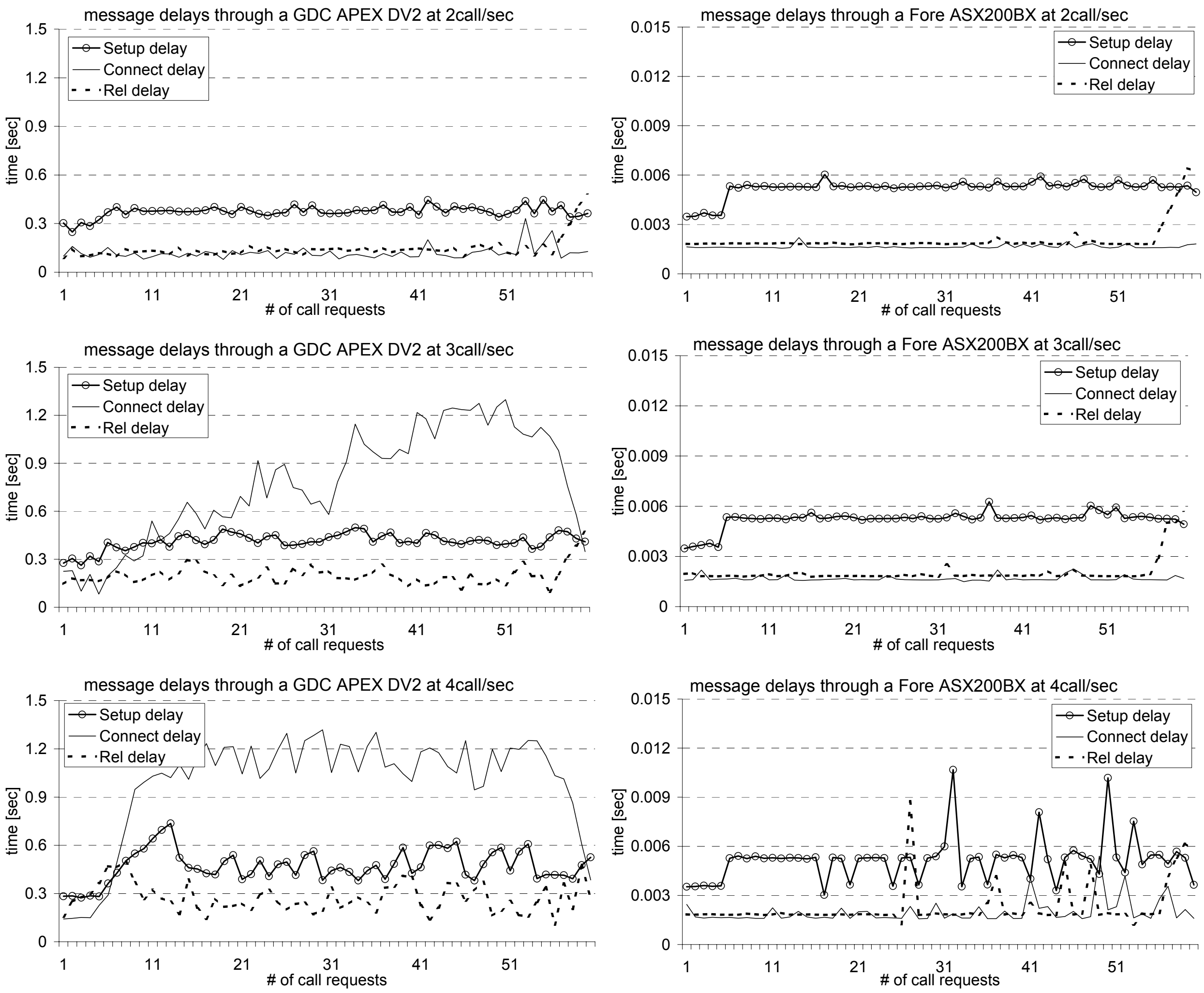


Figure 4.2 Message delays at 2, 3 and 4 calls/sec offered rate (left) GDC APEX DV2; (right) Fore ASX200BX

Note that the first five and last five samples in each chart should be ignored, as they represent the transient phase when calls are established but not yet released, or in the last phase no more calls are generated, but all active ones released in a burst, respectively.

Comparison to others' results

In Table 4.3*f* another example is presented. This time we obtained the results with a GNN iW95000 on a *Seabridge XP140* switch at 1 call/sec arrival rate. The statement in Section 4.1.2 is not easy to be compared to others' results as they did not investigate this subject, but obtained only a part of the necessary measurement results (see e.g., some measured values for a FORE ASX100 switch by [Pil99] in Table 4.3*g*).

Table 4.3f Delays for Seabridge XP140 at offered call arrival rate of 1 call/sec

	Setup Delay	Connect delay	Call estab time	Release delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.026	0.009	0.039	0.010	0.018	0.016	0.3462	0.3846
MAX	0.037	0.015	0.055	0.15	0.027	0.031		
MIN	0.024	0.008	0.035	0.009	0.016	0.015	0.3333	0.375
STDEV	0.00119	0.00048	0.00223	0.00063	0.00098	0.00091		
Corr. coef. to T_C	0.5086	0.5264						
Corr. coef. to T_{CN}	0.5127	0.5325						

Table 4.3g Delays for Fore ASX100 at offered call arrival rate of 1 call/sec, see Table 1 in [Pil99]

	Setup delay (calc.)	Connect delay	Call estab time	Release delay	Release time	Release latency	Ratio CONNECT/ SETUP	Ratio RELEASE/ SETUP
AVERAGE	0.006	0.00487	0.01640	n.a.	0.00447	n.a.	0.81	n.a.
MAX	n.a.	0.02246	0.06375	n.a.	0.02319	n.a.		
MIN	0.00578	0.00368	0.01322	n.a.	0.00388	n.a.	0.63	n.a.
STDEV	0.004	0.00214	0.00653	n.a.	0.00138	n.a.		
Corr. coef. to T_C	n.a.	0.4151						

I have collected into Table 4.3*g* all the measured results obtained by [Pil99] for a *Fore ASX100* switch (where $T_C = T - PR_{cl}$). It can be observed, that the ratio *CONNECT/SETUP* delays is out of the range that I have defined, but we should also see that the results are not so accurate, i.e., the standard deviation is the same order as the values for the *SETUP* and *CONNECT* delays, respectively. Moreover, we do not know exactly, if the length of the *SETUP* message was also 3 cells long as in our case or just 2 cells. At least it is interesting, why did they need to calculate the value of the *SETUP* delay from other measured parameters, once they could measure the *CONNECT* delay, which is measured between the same measurement points (e.g., see Figure 3.3).

4.1.3 Dependency of the T_C on routing table, bandwidth allocation and number of active connections

I have shown that the call establishment time T_C depends linearly on the size of the routing table, but it is independent of the size of the allocated bandwidth. Moreover, I have shown that the number of active calls in the switch has no influence on the call establishment time of a new call.

Routing complexity:

We have investigated the effect of routing complexity on the signalling performance. For this purpose we artificially inserted new entries in the routing table of the switches. The entry corresponding to the destination address was always at the end of the list. The results are shown in Table 4.4.

Table 4.4 Effect of number of routing entries on the call establishment times and message delays

Number of Routing Entries	Call Establishment Time (increased by)		SETUP delay (increased by)		CONNECT delay (increased by)	
0	460 msec (0%)	10.8 msec (0%)	355 msec (0%)	5.4 msec (0%)	102 msec (0%)	1.7 msec (0%)
10	515 msec (12%)	11.34 msec (5%)	407 msec (15%)	5.62 msec (4%)	104 msec (2%)	1.99 msec (17%)
20	543 msec (18%)	11.61 msec (7.5%)	432 msec (22%)	5.78 msec (7%)	106 msec (4%)	2.09 msec (23%)
30	570 msec (24%)	11.88 msec (10%)	457 msec (29%)	5.94 msec (10%)	108 msec (6%)	2.21 msec (30%)
	APEX DV2	ASX200	APEX DV2	ASX200	APEX DV2	ASX200

We obtained similar results on two other switches (*Seabridge XP140* and *Newbridge MSX36170*), but for the sake of simplicity the results are not shown here. Similarly, [Nie97] stated that "by hop-by-hop routing the delay is experienced on each hop towards the destination", however they did not carry out special test cases to show it implicitly.

Bandwidth allocation:

We have investigated the effect of the allocated bandwidth for calls on the signalling delays and call establishment time. Therefore, we have generated CBR calls with a Peak Cell Rate (PCR) of 0.1, 1, 5 and 10

Mbps, respectively at an arrival rate of 1 call/sec. Note that, when running the tests with higher bandwidth sizes (e.g., 10Mbps), we paid attention to avoid call rejections due to overloaded links (STM-1 link). We did not observe any changes in the measured parameters by increasing the bandwidth, as shown in Table 4.5. Thus, we concluded that the bandwidth allocation has no influence on the call establishment time.

Table 4.5 Effect of the allocated bandwidth on the call establishment times and message delays

Bandwidth [Mbps]	Call Establishment Time increased by		SETUP delay increased by		CONNECT delay increased by	
0.1	0%	0%	0%	0%	0%	0%
1	0%	0%	0%	0%	0%	0%
5	0%	0%	0%	0%	0%	0%
10	0%	0%	0%	0%	0%	0%
	APEX DV2	ASX200	APEX DV2	ASX200	APEX DV2	ASX200

Simultaneous calls:

This is a very simple but controversy subject. We did a wrong statement in 1998 in [C-4], namely that "the RTT delay of a call establishment increases linearly with the number of open connections". But we have disregarded that the called party was emulated by a port of the same GDC APEX DV2 switch! Later, we have repeated this measurement with an independent tester (HP 75000) and published the correct results in [C-9] (see Figure 4.3 (left) below). An other example is shown in Figure 4.3 (right), carried out by [Mau01] on two cascaded switches. They claim that "the average SETUP time is neither affected significantly by the call interval time nor by the increase of active calls". However, 10 lines above (!!!) in the same paper they mentioned that "as the number of active calls is growing the SETUP time is gradually increased". Wrong statement, because the increase was due to the fact that the call arrival rate exceeded the service rate of one of the 2 cascaded switches causing waiting times in the buffers and thus an increase in the call establishment time. Finally, the subject was investigated by [Pil99] as well, who stated that "there was no difference in call setup delay with increase in the number of unreleased calls". As a conclusion, I believe that the this statement is also correct.

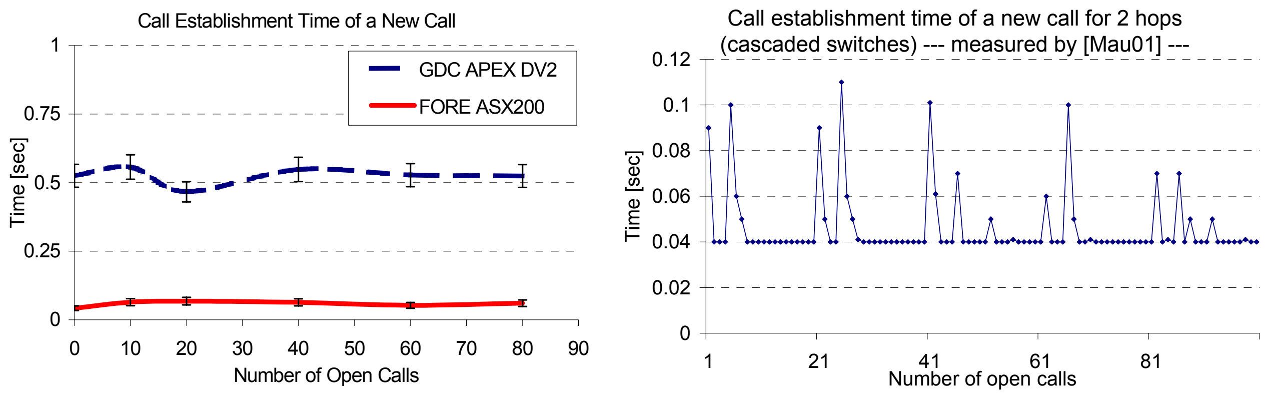


Figure 4.3 Effect of simultaneous active calls on establishment time of new calls

4.1.4 Dependency of message processing times on different call profiles (signalling overhead)

I have given a quantitative measure, how the mean delays of SETUP, CONNECT and RELEASE messages (and therefore the call establishment time T_C) depend on different call profiles, when processed at an ATM signalling node:

- The mean *SETUP delay* and the T_C depend on the call profiles as follows:

Table 4.6 Effect of IEs on SETUP delay and call establishment time

Adding to the *default SETUP message* (containing only the mandatory IEs)	**Mean SETUP delay**	**Call establishment time**
AAL (1, ¾ or 5) Parameters IE	1-2% decrease!	0% increase
P-to-MP connection IE (Bearer Capability)	0% increase	0% increase
Called Party Sub-address IE	10-32% increase	10-22% increase
Calling Party Address IE	12-14% increase	6-10% increase
Higher Layer IE	38-60% increase	28–40% increase

This variation of the mean *SETUP delay* is due to the time taken to process call profiles of different complexity (i.e. kind of information elements). There is a large scale from simple voice call to complex multimedia calls. Some examples are given in Table 4.6 and Table 4.7, respectively. The results are obtained at 1 call/sec load. In general, the increase in the call establishment time due to the complexity of the *SETUP* message can be described as follows:

$$T_C^{CCP} = (1+s)\cdot T_C^{default} \text{, where } s \text{ is the } \textit{signalling overhead},\ 0 < s < 1. \tag{6}$$

I will try to prove this above statement by listing the average of the measured values of the parameters shown in Table 4.6 in the case of two switches: GDC APEX DV2 and Fore ASX200BX (see Table 4.7). In fact, I have analysed the effect of many other Information Element than shown here (see e.g., Table 2.2), and similarly, many of them have caused an increase in the *SETUP* delay and call establishment time, respectively. Adding more IE at the same time (obtaining logically correct combinations), I reached an upper limit for the call establishment time of a double value, which is described in equation (6). No other papers investigated this subject (impact of IEs on T_C).

Table 4.7 Effect of the Information Elements on the call establishment times and message delays (average values)

Parameter (IE)	Call Establishment Time [msec]		SETUP delay [msec]		CONNECT delay [msec]		RELEASE delay [msec]	
only mandatory IE	487	14.0	364	3.89	112	1.7	102	1.8
AAL (1, 3/4 or 5) Parameters IE	485	14.0	360	3.82	111	1.7	103	1.7
P-to-MP conn. IE (Bearer Cap.)	488	14.1	365	3.88	112	1.7	103	1.8
Called Party Sub-address IE	536	14.9	408	3.87	112	1.7	102	1.6
Calling Party Address IE	593	15.4	480	3.82	113	1.7	104	1.8
Higher Layer IE	623	28.3	502	6.1	115	1.8	103	1.8
	APEX DV2	ASX200	APEX DV2	ASX200	APEX DV2	ASX200	APEX DV2	ASX200

- I have found that in contrast to the mean *SETUP delay*, the mean *CONNECT delay* and mean *RELEASE delay do not increase* when the parameters (*IEs*) from Table 4.6 are added to the default *SETUP* message. Furthermore, the call release latency of complex calls is:

$$T_{RN}^{CCP} = T_{RN}^{default} \quad \forall s,\ 0 < s < 1. \tag{7}$$

This statement seems to be obvious, since there is no change in the content of the *CONNECT* and *RELEASE* messages by this test case, but in fact different jobs are to be done by the processor for these different call profiles before propagating a *CONNECT* or a *RELEASE* message, respectively (see values in Table 4.7). This statement is therefore important, because it highlights an important aspect of ATM signalling (*not all message delays through a switch are affected by a change in the call profile*).

- I have shown that *the mean call establishment time does not depend on the type of call*.

$$\text{e.g.: } \overline{T}_C\big|_{VBR} \approx \overline{T}_C\big|_{CBR} \approx \overline{T}_C\big|_{ABR} \approx \overline{T}_C\big|_{UBR}, \tag{8}$$

where $\overline{T}$ denotes the average time.

As a proof, please take Table 4.7, the case when adding AAL1, AAL3/4 or AAL5 parameter IEs. In addition, two mandatory IEs might suffer some changes as well, e.g., the ATM Traffic Descriptor IE to indicate the PCR, SCR and MBS of the UBR, CBR, VBR or ABR calls, and the QoS parameter IE to limit the value of

CDVT, respectively. However different jobs are to be done by the processor for all these different call types, I have found very little changes in the message delays and call establishment time, respectively (therefore the approximation formula in equation (8)).

Early papers in the literature expected that the call establishment time of VBR calls will be larger than that of CBR calls (see [Gel97], [Wu98]), but our measured results contradict to these statements. A recent paper, [Far01] has shown again by measurements that the T_C of UBR calls is only a bit shorter than that of CBR calls in a 4-node PNNI hierarchical network, which is in line with the equation (8).

4.1.5 The impact of the call release phase on the T_C

I have investigated the mean call establishment latency T_{CN} of simple calls (i.e. default SETUP) in two cases: call establishment followed by call release vs. call establishment without release phase and I have found the following properties:

- In case of releasing the calls after a given holding time, the mean of the measured call establishment latency is (15...20)% longer compared to the case when calls are established but not released. This property is independent of the call arrival rate.

None of the previously cited papers (e.g., [Nie97], [Pil99], [Mau01]) has investigated this important case. Due to the fact that the call establishment latency (T_{CN}) is the sum of the *SETUP* and *CONNECT* delays, let me plot these parameters for two switches (GDC APEX DV2 and Fore ASX200BX) at a call arrival rate of 2 calls/sec (see Figure 4.4). With intention, I have formulated this property for the *call establishment latency* instead of the call establishment time, as the destination response time on the *SETUP* message might deteriorate the relationship using different types of terminals. On the left side of Figure 4.4 the message delays are shown, when calls are established subsequently, but they are not released. On the right side of the same Figure 4.4 three message delays are shown, because after 5 calls were already establishments, we started to release the first call, then establishing a new one, releasing the second one, etc, always keeping 5 calls active. It can be immediately observed, that the only change is in the *SETUP* delay. However, this change is significant (>50%), and has an impact on the call establishment latency and the call establishment time (>15-20%), respectively. Thus, all the papers (e.g., [Kaus97], [Nie97], [Mau01]) who have simply ignored the impact of the release phase on the call establishment time (which is THE only realistic scenario) have committed an error in their conclusions. Therefore, I have given further examples for the call establishment time and release latency in Figure 4.5 on all four switches we have investigated. The significant difference (15-20%) in the call establishment time for unreleased calls compared to the case when keeping 5 calls active is shown in all four cases. In addition, the call throughput of the switches is shown as well, which starts to deteriorate once the call arrival rate exceeds the call service rate of the switches (overload).

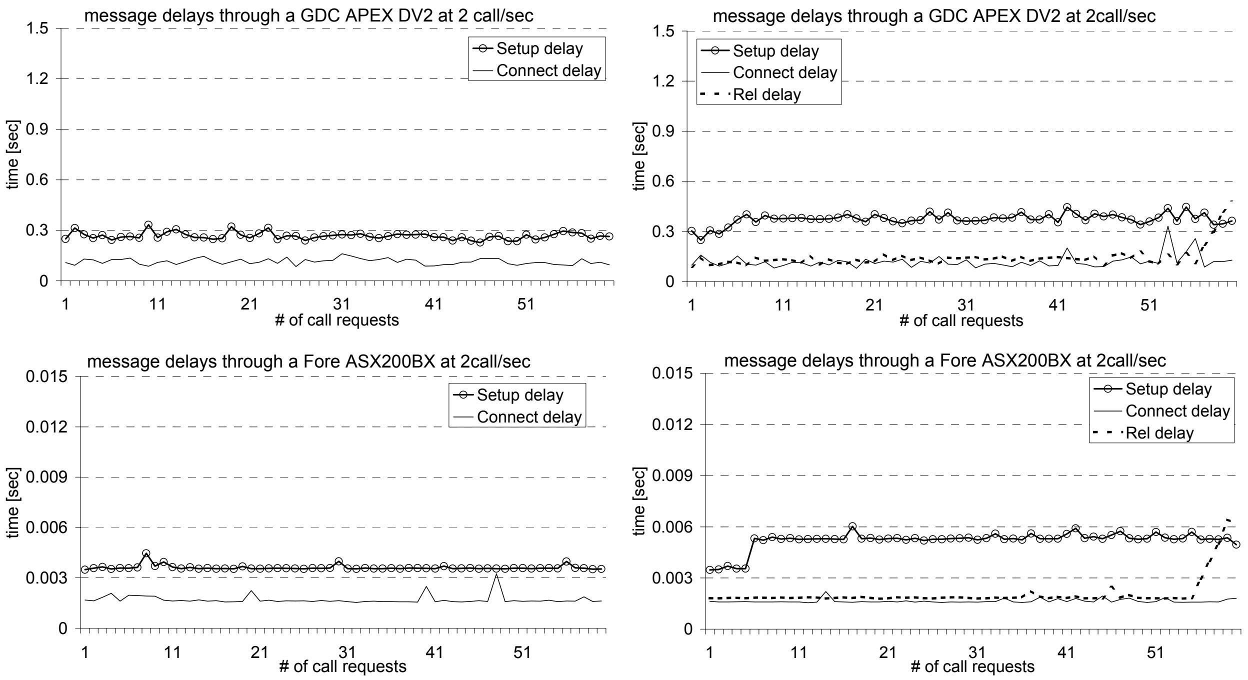


Figure 4.4 (left) Call setup without releasing calls; (right) Call setup followed by call release, keeping 5 calls active

Another important property can be concluded from the same set of measurements:

- In case of releasing the calls after a given holding time, the call intensity threshold where the call establishment latency starts to increase dramatically is (65...70)% that of the case when calls are established but not released. This threshold is also related to the point where rejected calls start to appear.

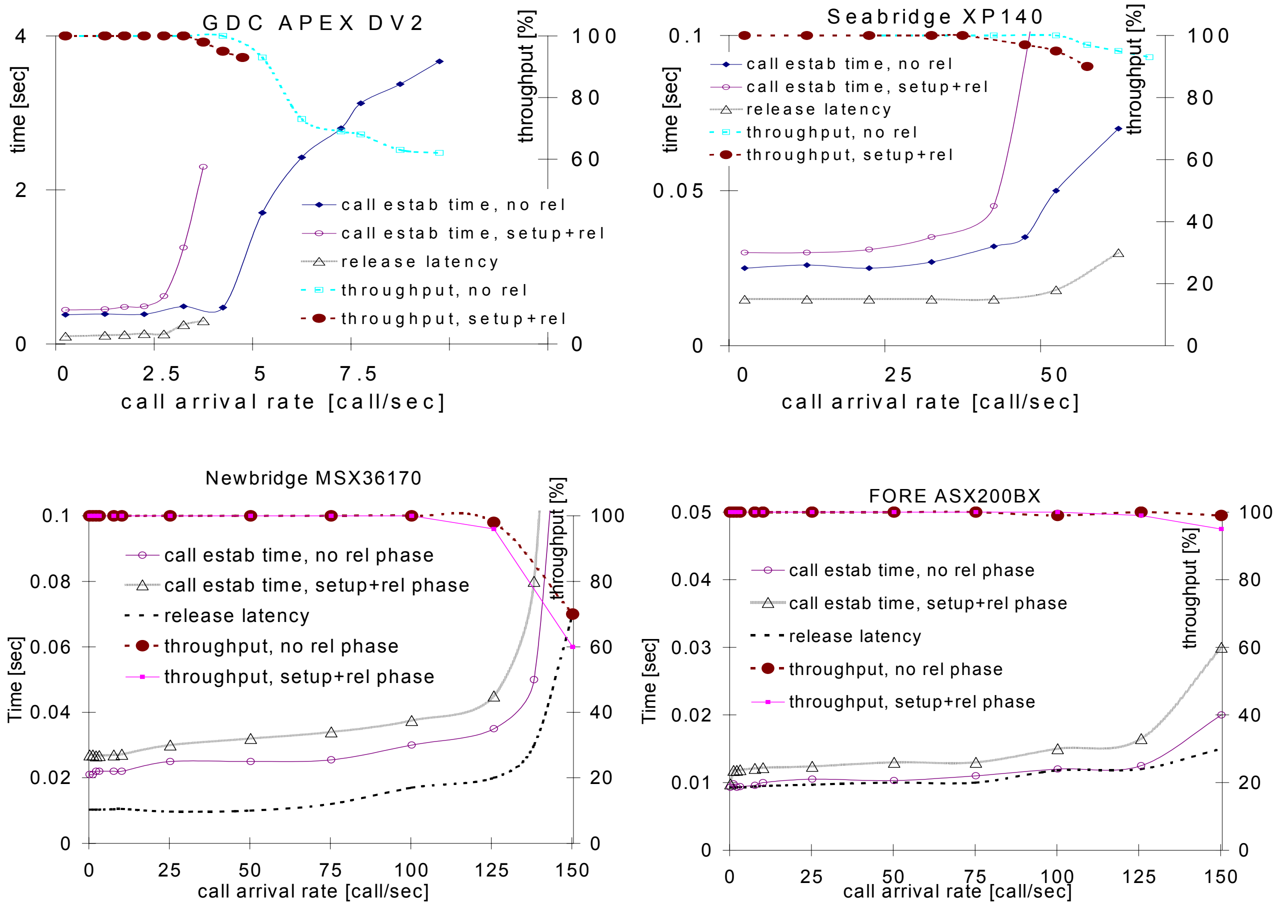


Figure 4.5 Call establishment time, release latency and throughput vs. call intensity
a) GDC APEX DV2; b) Seabridge XP140 c) Newbridge MSX36170; d) Fore ASX200BX

All these measurement were obtained with the GNN iW95000 (generating up to 200 calls/sec), because the other tester (HP75000) could not generate calls over a rate of 4 calls/sec, as already mentioned. None of the other research groups reached a call arrival rate of over 50 calls/sec in their tests. E.g., [Kaus97] and [Nie97] obtained a maximum call arrival rate of 50 calls/sec with their call generator, [Mau01] stopped by 20 calls/sec using another software based call generator, while [Pil99] did not investigate this subject (was using an HP75000).

- I have shown that the duration of calls has no influence on the call establishment time, except one case, when the holding time is infinitely long (then there is no release phase).

I have repeated the measurement many times, while the number of active calls was gradually increased from 0, 10, 20, etc. until 100 unreleased calls. At an arrival rate of 1 call/sec that means an average call holding time of 10, 20, …, 100 seconds. The results were similar in all cases, except the case when I have not released the calls at all (for this case, see first statement of this section). In [Kaus97] and [Nie97] it is expected that calls of zero (or non-zero) duration give very similar results to those not tearing down switched virtual connections at all. As seen also in Figure 4.5, our results contradict to their statement.

4.1.6 The impact of the signalling overload on the T_C and T_{RN}

In case of a light call overload in the switch (P_B<0.1, i.e., less than 10% of the calls are rejected), the call establishment time is dramatically increased, but the slope of the call release latency remains for this range still unchanged (see Figure 4.5). Further increasing the call intensity leads to a dramatical increase of the release latency as well.

This subject was not investigated by others. I do not intend to present other results here, than shown in Figure 4.5. This is again a specific feature of ATM signalling, and it may happen in two cases: a) when there is a priority mechanism applied or b) when the *SETUP* and *RELEASE* messages visit (partly) different paths inside the signalling processor (distributed call processing). The 4 switches we have studied belong to this latter case. The call rejections are due to the buffer overflows in the processor. I do not intend to quantify the slope of the throughput curve in the overload region, as this is strongly dependent on implementation specific call limitation mechanisms. However, this parameter (throughput) will be also an important criteria to tune the new call processing model presented in Section 5.1.

For the next statement the test configuration in Figure 3.1b was used.

4.1.7 Estimation of the T_{CN} for cascaded switches

I have shown that the call establishment latency of calls with the same call profile that go through 'r' similar cascaded switches (within one PNNI peer group) satisfies the following inequality:

$$r \cdot \overline{T}_{CN}\Big|_{1switch} \geq \overline{T}_{CN}\Big|_{'r'switches},\; r = 1,2,...,k\,. \tag{9}$$

Between cascaded switches we set the PNNI interface, which means in fact the same set of signalling messages as at the UNI. In a pilot-test for MOL Telekom Hungary in 1998 we had the opportunity to connect a maximum of four Fore ASX200BX switches in cascade and measure the message delays throughout the switches. Calls were generated at an arrival rate of 1 call/sec. We obtained the message delays for 1, 2, 3 and 4 cascaded switches, respectively. In 2002, I had the possibility to repeat these tests in the Siemens System Test Laboratory in Munich with four Seabridge XP140 switches in cascade. I have defined only one PNNI peer group (no hierarchy), i.e. we did not have delays due to routing between different hierarchical levels. Our results are summarised in Table 4.8 (average values only). The results confirm the equation (9) for the parameter T_{CN}, e.g., 4x35msec>133msec for the XP140, and 4x7.5msec>28msec for the ASX200. The formula can be used for T_C as well, however equation (9) is stronger, as it does not contain the destination response time on *SETUP* message (which shall not be multiplied by '*r*'). This behaviour is due to message overlapping in the cascaded switches. Equation (9) can be used to obtain the upper limit for the average latency of a homogeneous (all nodes are of the same type) cascaded network, once the message delays for one switch are known.

Table 4.8 Measurement of cascaded Fore ASX200BX and Seabridge XP140 switches, respectively (average)

Cascade	*Sum of SETUP delays* [msec]		*Sum of CONNECT delays* [msec]		*Call establishment latency* [msec]		*Call establishment time* [msec]	
	XP140	ASX200	XP140	ASX200	XP140	ASX200	XP140	ASX200
1 switch	26	5.4	9	1.9	35	7.5	39	12
2 switches	51	11	16	3.6	68	14.6	74	19
3 switches	75	16.1	24	5.1	101	21	106	26
4 switches	100	21	32	6.8	133	28	140	34

The formula given in equation (9) can be generalised to a heterogeneous (not all nodes are of the same type) cascaded network (within one PNNI peer group), as follows:

$$\sum_{i=1}^{P} n_i \cdot \overline{T}_{CN}^{\,\text{type}\,i} \geq \overline{T}_{CN}\Big|_{'r'switches}, \quad r = \sum_{i=1}^{P} n_i, \tag{10}$$

where P is the number of different types of ATM switches in the network and n_i is the number of switches of one type. This generalisation is based on the simple fact (as previously shown in homogeneous case) that the message delays are additive when crossing multiple hops, but due to overlapping, the sum of the delays of individual switches will be higher than or equal to the end-to-end delay measured in the cascade.

I did not have the opportunity yet to prove this generalisation by direct measurements, however I have found one example in [Mau01], which confirms my statement (see below).

Case study:

Measurement results presented by [Mau01] on 3 different types of cascaded switches confirm the equation (10). In this case, we have: $n_1=n_2=n_3=1$, $r=P=3$.

Table 4.9 Measurement of cascaded switches: Seabridge XP140, Newbridge MSX36170, Ericsson AXD311 (average)

Cascade	Type	*Call establishment latency* [msec]	*Call establishment time* [msec]
1 switch	XP140	39	42
1 switch	MSX36170	33	35
1 switch	AXD311	22	25
2 switches	XP140 + MSX36170	71	73.1
3 switches	XP140 + MSX36170 + AXD311	85	87.8

The property in Section 4.1.7 cannot be applied to the case when in the network more PNNI hierarchical levels are built (see e.g., [Nie97] for 2 levels) or secondly, when between network nodes instead of PNNI the public NNI (B-ISUP) protocol is applied (see e.g., [Pil99]). In this latter case, [Pil99] states that "the call setup delay with two switches is higher than the sum of the setup delays measured previously at two individual switches (Fore ASX100 and Fore ASX200, respectively)", but the "NNI introduces a significant delay in call setup involving multiple switches". As not sufficient measurement results are available, I could not build up a similar model for public NNI solution as I did for the private PNNI (see Section 5.1).

Investigating a 2-level PNNI group, [Nie97] pointed that "the change in slope at hop '*n*' is due to the additional route computation experienced upon entering a new peer group". We did not have the opportunity to carry out similar tests in PNNI environment, but I have included those results from [Nie97] in the new call model in Section 5.1.

An approximation formula instead of these inequalities (9) and (10) is developed later in Section 5.2.4.

4.2 Intrinsic properties of p-to-p simultaneous calls in ATM switches (transient state measurements)

Performance evaluation of calls with constant arrival rate in steady-state (see Section 4.1) does not give a complete characterisation of the exact behaviour of ATM switches. I expected different behaviour when generating p-to-p multi-connection (or simultaneous) calls, or at a network node failure automatically followed by a re-establishment of previously existed connections, which lead to burst arrival of signalling messages, and thus to potential overload in the signalling processor.

4.2.1 Introduction

Motivation

Due to a high number of ambiguous published results (see e.g., [Mau01], [Pil99]) regarding burst measurements of signalling messages or overload situations, I have decided to carry out a detailed analyses of these measurements.

Discussions

E.g., [Mau01] states that “the occurred delay peaks are probably due to the launching of new switching tasks” (see Figure 4.6*a*). [Pil99] concludes that “when the number of simultaneous calls is small, it is the ATM user causing the increase in delay because the switch can still handle the connection requests, however, when the number of simultaneous calls is large the ASX200 switch directs the delay to the ATM user” (see Figure 4.6*b*). Two conclusions, that are not easy to understand. For comparison, I have placed our measured results for the same Fore ASX200BX switch into the same chart (the measurements represent the average time calculated for a number of ‘*n*’ calls in a burst).

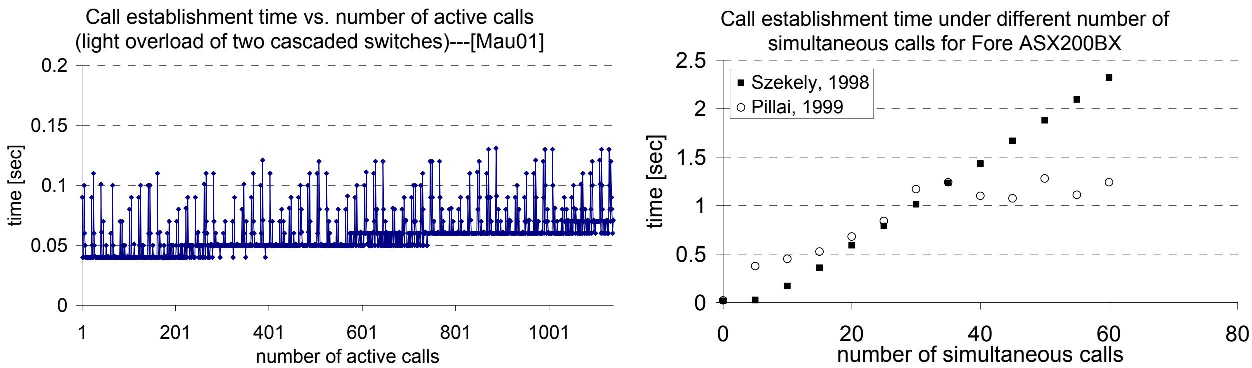


Figure 4.6 Two examples of measurement results which lead to questionable conclusions a) [Mau01]; b) [Pil99]

Open questions

My decision was driven also by the fact, that our first results regarding burst measurements left a number of open questions. E.g., how is that possible, that for steady-state measurements (see Section 4.1) for the same switch (Fore ASX200BX) we obtained call establishment times of order of tens of milliseconds, while by burst measurements we obtain rapidly increasing call establishment times of the order of seconds!?! One thing from the beginning was clear: *analysing only the call establishment times will never answer our questions*. Therefore, I decided to represent each message delay in the burst, which is depicted in Figure 4.7.

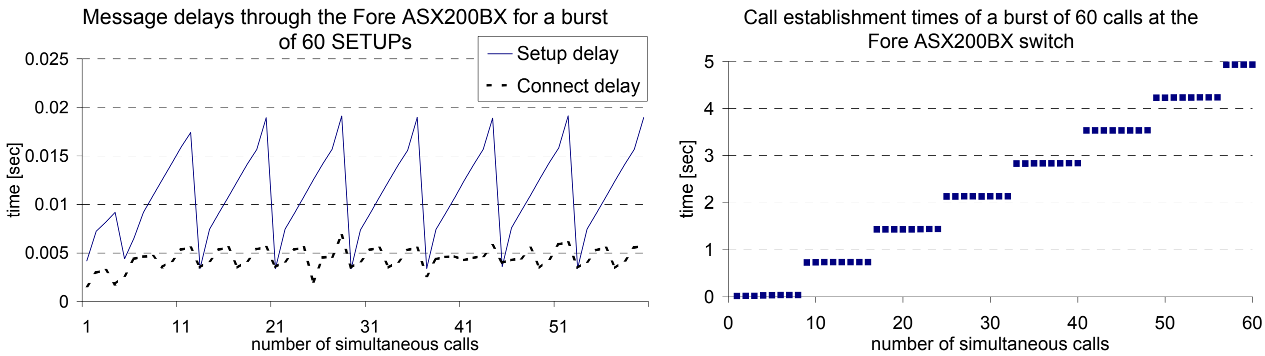


Figure 4.7 Analyses of message delays at the Fore ASX200BX for a burst of 60 calls

This kind of representation brought a not surprising result, that the message delay components are still kept in the range of tens of milliseconds, but somehow the call establishment time is step-wisely jumping to order of seconds. But who introduces these delays? Secondly, how does this "see-saw" function appear in the curve of the *SETUP* delay, once we expected a "nearly" linearly increasing curve for the whole burst? And thirdly, why is this behaviour independent of the length of the burst? (Note: a similar see-saw function appears in Figure 4.6*a* as well. Do we have the same problem in both cases?)

Conclusions

The traditional way of representing the flow of signalling messages (using arrows between two nodes, as in Figure 3.3) or looking at the call establishment times and message delays (as in Figure 4.7) does not help us here finding the "hidden" system faults, therefore *I have looked for new representation methods*.

4.2.2 A new evaluation method: the 'population-diagram'

I have introduced a new method ('population-diagram') of representing the flow of signalling messages, which reveals the detailed structure of the signalling burst mechanism.

The main idea is:

- to represent all the (interesting) layer 3 messages of the system related to the observed burst in *one* diagram split into many time axis on the horizontal scale and representing the links instead of nodes in the vertical scale;
- to represent one signalling message with a *plot* at the link (interface between two nodes) instead of an *arrow* between two nodes;
- to create "*m*" time axis at each interface, representing each (interesting) type of message in order of appearance at separate time axis, where *m* is the number of message types at that interface;
- to split the diagram into (n+1) parts if the messages visit *n* nodes during a call setup;
- if the calls are generated from both sides of the communication path, then split the "*m*" axes into *"2xm"* sub-axes, to be able to distinguish between the two directions (tx; rx) when a certain message is propagated through the link;
- to represent the layer 2 messages (*POLL, STAT, USTAT*) in the same diagram, any time it will be necessary to avoid confusions.

The lower segment covers the signalling events at the link between the source and network node, while the upper part shows the messages at the link between the node and destination, sorted by the time they have arrived. It is very important that the messages are arranged in a consecutive order defined by standards (e.g., [Q2931]) from the bottom to the top (e.g., *SETUP*→ *CALL_PROC*→ *CONNECT*→ *CONN_ACK*→ *STATUS_ENQ*→ *STATUS*→ *REL*→ *REL_COMP*). In fact, it is not necessary to represent all messages, but it is recommended to use at least m=3 (e.g., Figure 4.8).

This diagram provides a better global overview than the "arrow-type" diagram used previously in the literature (see e.g., Figure 3.3). To show the advantages and usefulness of using this type of representation, let me guide you throughout three case studies. However, the 'population-diagram' is not limited to analysing the ATM signalling flow only, but can be used to any type of signalling protocol analysis (I am using it in my everyday work at Siemens AG, at system testing activities of IP-based next generation networks, e.g., analysing message flows of MGCP, H.323 and SIP protocols). For further details, please check [T-2].

4.2.2.1 Case study 1: Burst arrival at a FORE ASX200BX switch

As a first example, I have represented the same scenario as in Figure 4.7 with (1 source; 1 node; 1 destination) when 40 *SETUP* messages are generated in a burst, see Figure 4.8 (m=3; n=1).

The first observation we can make in this example is the presence of long 'silent' periods of approx. 630 milliseconds and short burst activities of approx. 70 milliseconds.

Secondly, we can observe a long gap of *CONNECT ACKNOWLEDGE* messages at the lower interface in the first half of the burst.

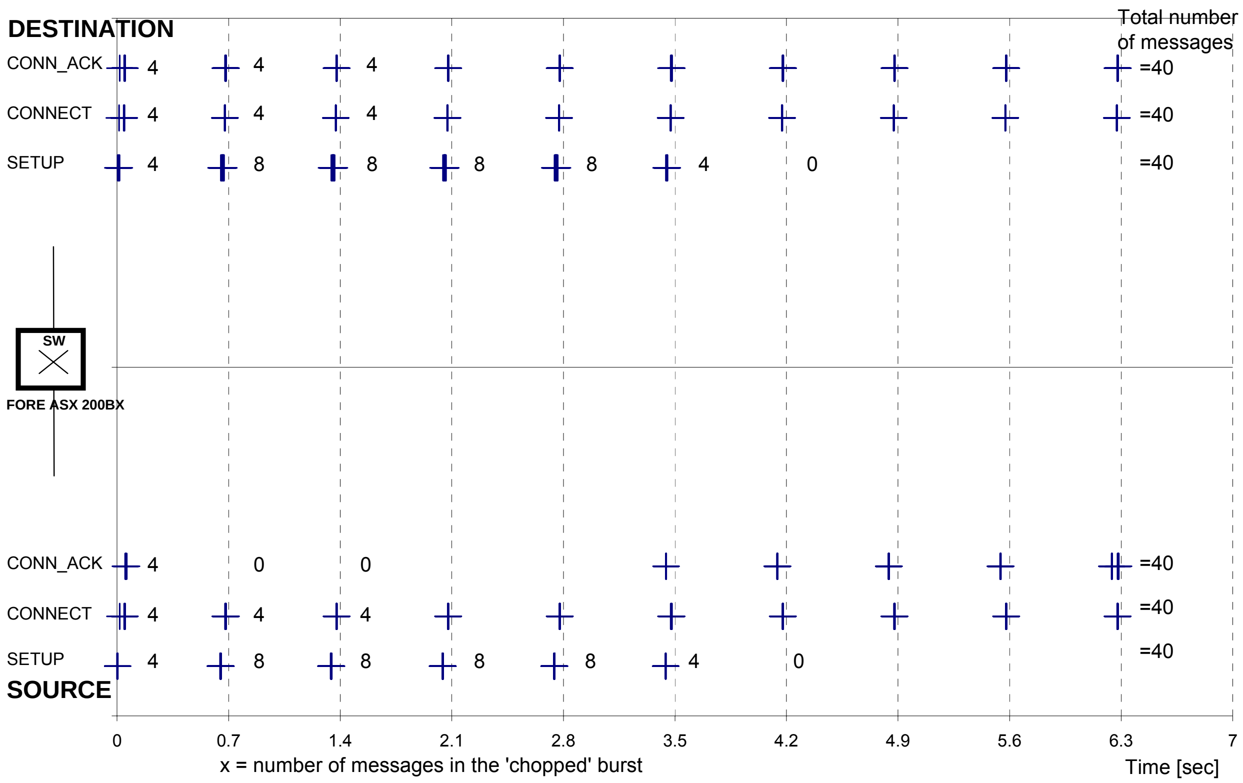


Figure 4.8 Representing in 'population-diagram' all messages belonging to a burst of 40 calls at the Fore ASX200BX

Thirdly, we can observe that the 40 *SETUP* messages originally offered in one burst, are "chopped" here in many small bursts. But it is still unclear, why it happens so? Let's make one further step.

The x axis representing the time scale can be compressed in order "to make the things visible", e.g., the "active" periods of 70 msec are shown, followed by "silent" periods of 630 msec, which are compressed. The same results as in Figure 4.8 are now plotted in the modified version of this diagram (see Figure 4.9), where we compress the silent periods, thus making visible the details of the active periods.

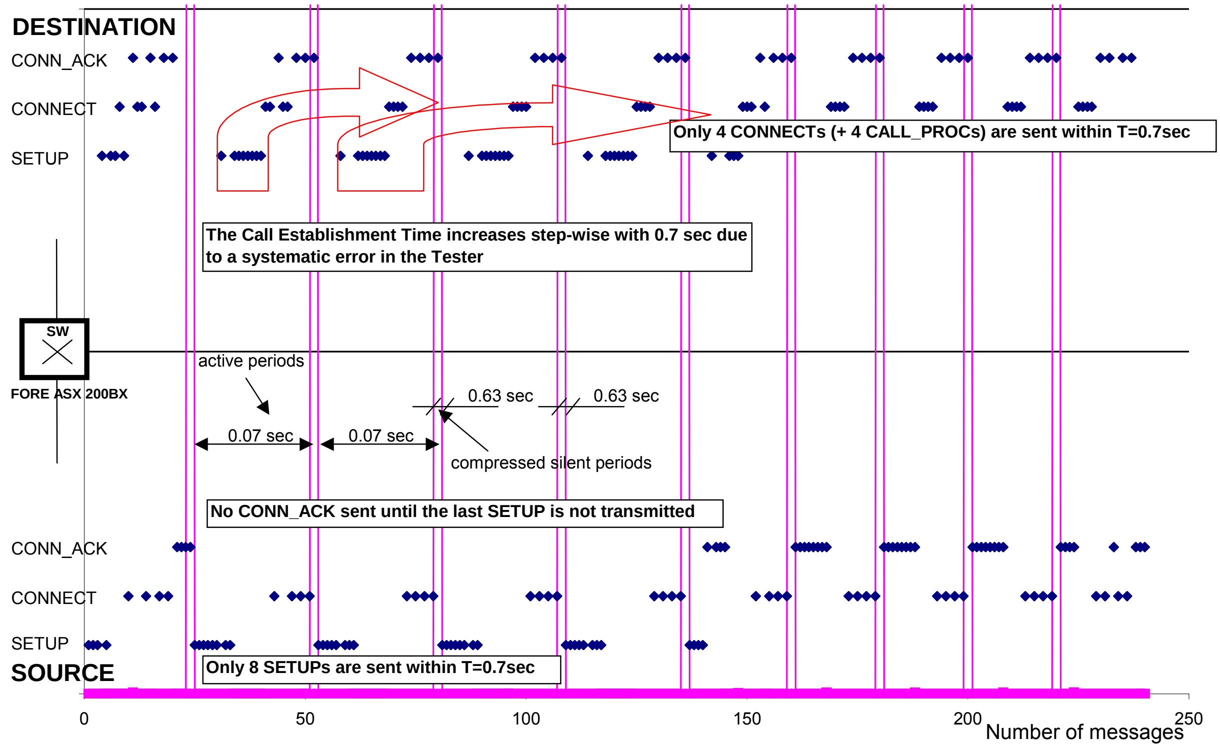


Figure 4.9 Modified "population-diagram" of signalling messages (compressed time scale)

Discussions

At the first look we still cannot answer the first question, namely what causes the appearance of "silent" periods, but at least we can solve the second and third questions. We can observe that the gap in the *CONNECT ACK* axis disappears first when the last *SETUP* message of the original burst reaches the switch. That means, the generator has buffered originally all 40 *SETUP* messages in one burst and passed down to the SSCOP layer to be transmitted. When the first *CONNECT* message arrives back to the source (please, exclude the first T_b = 0.7 second period from the Figure 4.9, which is a transient phase), this message will found a buffer filled with *SETUP* messages at layer 3, therefore the acknowledgement message (*CONNECT ACK*) will be placed to the end of this buffer and first propagated to the switch when all *SETUP* messages already left the buffer. We can answer our third question as well. We can count now the number of *SETUP* messages in one "chopped" burst, there are 8 of them at the lower interface and again 8 at the upper interface, i.e., all 8 *SETUP* messages passed through the switch in the same "active" period. This "chopped" burst causes then a *linear increase* in the *SETUP* delays, but the buffer of the switch will be emptied again in the subsequent "silent" period, thus in the next "active" and "silent" periods the same "see-saw" form will be repeated (check Figure 4.7*a*). So, we have the answer for the "see-saw" form, but a next question arrives immediately, because only 4 instead of 8 *CONNECT* messages can be found in the same "active" period. Why only 4? What happens with the other 4? They will be answered by the destination in the next "active" period (as shown by the big arrow in Figure 4.9), which means a jump of 0.7 second in the call establishment time of these calls. Thus the call establishment time suffers further jumps in the coming T_b = 0.7 second periods (check Figure 4.7*b*). So, we have answered that question due to the jump in the call establishment time as well. Let's come back to the 4 "missing" *CONNECT* messages in the same "active" period. Before the destination user replies to an incoming *SETUP* message with a *CONNECT*, first he sends a *CALL PROCEEDING* message (see Figure 3.3). Thus we have a sum of 8 messages again (*CALL PROCEEDING* + *CONNECT*) from the destination to the switch, the same number (8) again. Let's put these two phenomena together: transmission of *SETUP* messages blocked after 8 messages were sent out from the source to the switch + transmission of *CONNECT* messages blocked after 8 messages were sent out from the destination to the switch (opposite direction). This event repeats himself periodically, with T_b = 0.7 second. But who can cause such an event? The answer is: *the flow control mechanism of the link layer* (SSCOP). Well, we could guess the answer, but how can we prove that? We need to represent the layer 2 messages (*POLL* and *STAT*) as well in the same Figure 4.8. Let's do that in Figure 4.10, because our 'population-diagram' enables us to add these messages to our original data.

Further questions

We did not represent the *CALL PROCEEDING* messages in Figure 4.9. Are there 4 or 8 *CALL PROCEEDING* messages propagated back from the destination to the switch in the first half as reply to the 8 *SETUP* messages? How do we know, that there are only 4? Secondly, there are 12 messages (8 *SETUP* + 4 *CONN ACK*) at the upper link sent from the switch to the destination in one active period. How is that possible once we supposed a window size of 8?

Answer

There are definitely only 4 *CALL PROCEEDING* messages per active period in the first half, otherwise we would receive later 8 *CONNECT* messages in the second half instead of receiving still 4 of them. Secondly, the credit window in the opposite direction is larger than 8 (controlled by the emulated end-users), therefore it was possible to send even 12 messages in one active period.

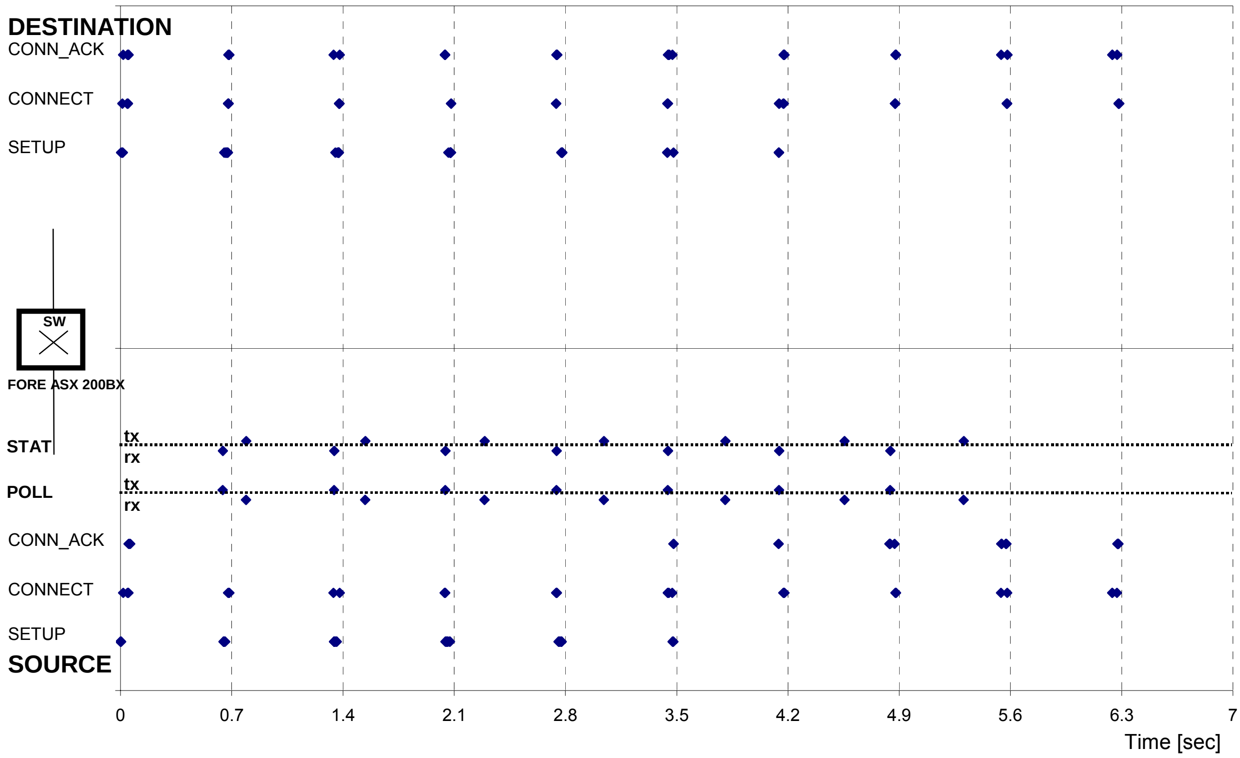


Figure 4.10 Dependence of layer 3 messages from layer 2 messages in burst mode

In Figure 4.10 we can observe pairs of layer 2 (*POLL*, *STAT*) messages in both direction at the same interface (for simplicity, we have represented the layer 2 messages only for the lower link in Figure 4.10). The inter-arrival time between *POLL* messages for one direction is 0.7 second. As shown, the pair of (*POLL, STAT*) from the source to the switch arrives first, followed immediately by an "active" period of layer 3 messages.

Conclusions

Our 'population-diagram' helped us to see the way the SAAL flow control can impose a bias on the transmission of messages. E.g., the 9^{th} *SETUP* message cannot be transmitted by the source (tester) in the same "active" period, because it has used up the credit window. This is the exception already mentioned in Section 4.1.1. Note that, the latency between *POLL* and *STAT* messages was measured to be 1.5 milliseconds for a Fore ASX200BX switch (see Table 3.3). If the tester would send a *POLL* message immediately after the credit window was used up, then the imposed bias of 1.5 msec could be neglected (appearing only once for 8 calls with call establishment times of 15 msec, i.e. approx. 0.1%). But why does it wait for 0.7 seconds here? After an extensive study of the ITU-T recommendation of the SSCOP layer (Q.2110), I came to the conclusion that, the tester's SSCOP layer should have been sent a new *POLL* immediately after the credit window was used up, secondly the flow control mechanism of the Fore ASX200BX switch (by default) has been set to keep the credit window fixed and reduced. The *STAT* message coming from the switch (responding to a status request *POLL*) contains information regarding the reception status of data PDUs (i.e., any layer 3 message) and *credit information* for the peer transmitter. Generally, during a connection, the window size can change dynamically based on local requirements, but here is being kept constant by the switch. Otherwise, if the window is increased sufficiently, an entire burst of calls can be established without the tester or the switch having to block once. This will completely remove the above effect (however the window cannot be extended indefinitely).

4.2.2.2 A need to find a new definition for the signalling burst

Once we have answered all the open questions related to the "ambiguous" results presented by [Pil99] and [Mau01] with the help of our 'population-diagram', let's make one more thing clear, and namely, how can we make a difference between an offered signalling burst and a "chopped" burst? To answer this dilemma, I have introduced a new definition.

Def 4.3 I have given a new definition of the signalling burst arrival at a network node, which "looks" at the node that receives a message, instead of looking at the source. This is in contrast to the classical way of defining a burst arrival, which captures the idea of clustering in time. My definition is based on the observation of a state variable:

It is considered a signalling burst arrival, if at the time a CONNECT message arriving to a node founds more than 2 SETUP messages in the buffer, i.e.:

$$n\left[SETUP_{in}(i)\right] - n\left[SETUP_{out}(i)\right]\Big|_{\Delta t} > 2 \qquad (11)$$

where the i=1,…,r is a node, $n[X(i)]$ is the number of elements of type X at node i, Δt can be arbitrarily chosen.

This definition is needed, since e.g., we have generated 40-100 *SETUP* messages in a "classical" burst at the source (with the *HP75000*), but never happened that more than 8 *SETUP* messages reached the *FORE ASX200BX* switch in a "chopped" burst within a period of 0.7 sec (see Figure 4.9), moreover this "chopped" burst left the switch before the first *CONNECT* message arrived back to the switch. Its effect has influenced the *SETUP* delay only ("see-saw" form). In such a case, according to Def 1.3 we cannot speak about "signalling burst arrival".

The *HP BSTS 75000* generator we have used can generate almost 500 *SETUP* messages/sec, e.g., 100 messages in 200 milliseconds in a burst, to a *GDC APEX DV2* (see Figure 4.12), but it is not able to offer more than 8 *SETUP* messages in a burst to a *FORE ASX200BX* switch, which is using a credit window of size 8 at the SSCOP layer, and therefore would need more frequent *POLL-STAT* exchange of layer 2 messages. This *limitation of the tester* has not been discovered by [Pil99], thus driving to ambiguous conclusions. As such a systematic error appeared during the burst-measurements of the *FORE ASX200BX* switch, *we does not consider those results for the call establishment times as relevant for the case studies of simultaneous call arrivals*.

In the followings we use the term "burst-arrival" according to the definition Def 1.3. Let's have a look at two other examples to show further benefits of the 'population-diagram'.

4.2.2.3 Case study 2: Burst arrival at a GDC APEX DV2 switch

Let's generate a burst of 10 and another burst of 100 calls to a *GDC APEX DV2* switch. The 'population-diagrams' related to these bursts are depicted in Figure 4.11 and Figure 4.12, respectively.

Analysing a burst of 10 messages

The first observation in Figure 4.11, is that the burst of 10 *SETUP* messages will be shaped, once it has passed the switch. A more detailed analyses of this burst process will be described in Section 4.2.3. Secondly, we can observe that all 10 calls are established successfully within 2.2 seconds (i.e., the call establishment burst time, see Terminology, Chapter 10), which is an average of 220 milliseconds for each call. But this is only the half of that we obtained for steady state measurements! Yes, but we do not have *RELEASE* and *RELEASE COMPLETE* messages here. So, this is in fact another confirmation of property in Section 4.1.5.

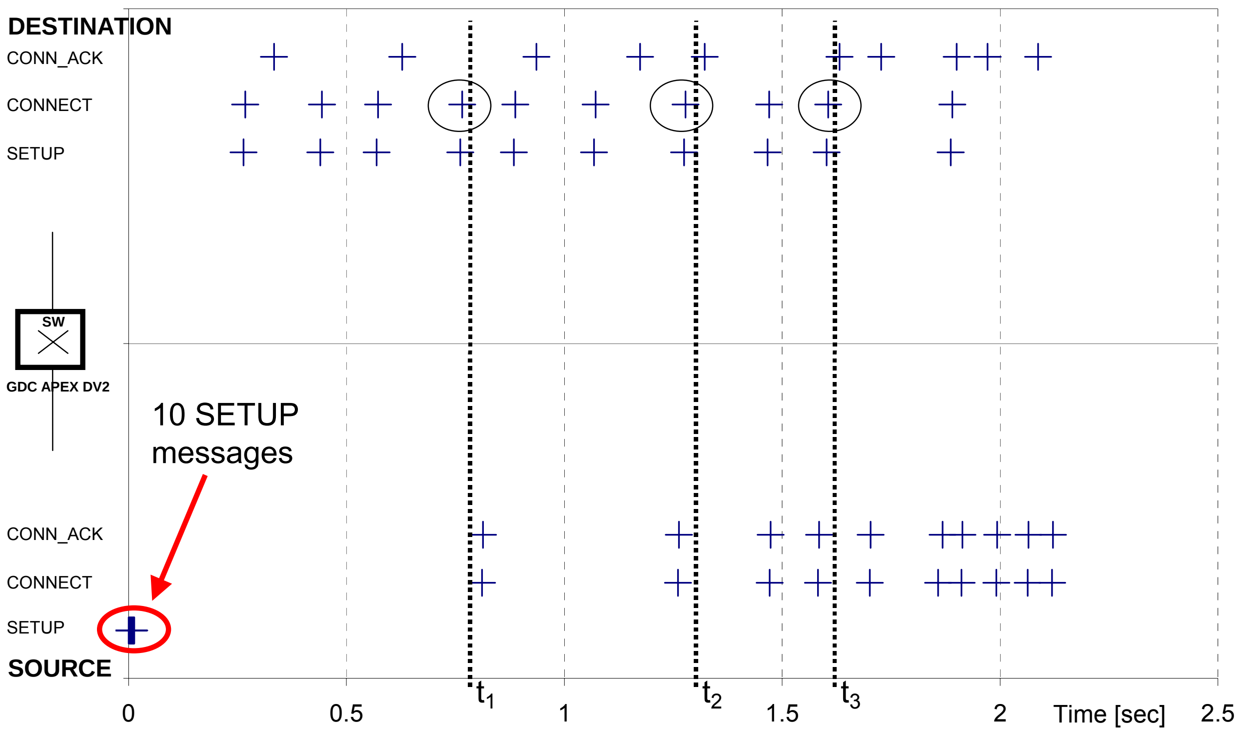


Figure 4.11 Burst of 10 calls at the GDC APEX DV2

We can check in Figure 4.11 if we have a burst arrival of signalling messages here, according to the equation (11) in Def 1.3. Let's select three different moments t_1, t_2 and t_3, respectively, when a *CONNECT* message enters the switch.

- at t_1: $n[SETUP_{in}(1)] - n[SETUP_{out}(1)]\big|_{t1} = 10 - 4 = 6 > 2$;
- at t_2: $n[SETUP_{in}(1)] - n[SETUP_{out}(1)]\big|_{t2-t1} = 6 - 3 = 3 > 2$;
- at t_3: $n[SETUP_{in}(1)] - n[SETUP_{out}(1)]\big|_{t3-t2} = 3 - 2 = 1 < 2$.

As shown, at time t_1 and t_2 the conditions for a burst arrival are satisfied, but at t_3 not anymore.

Analysing a burst of 100 messages

Figure 4.12 shows one more advantage of the 'population-diagram', namely that it helps us easily finding if buffer overflow occurs in the switch or to see the eventually retransmitted *SETUP* messages. Secondly, *it shows that an arrival of a CONNECT message back to the source does not necessarily mean a successful call establishment*!

It can be observed that after t = 15 seconds, no more *CONNECT ACKNOWLEDGE* messages are sent by the source, however still many *CONNECT* messages are arriving back to the source after this time. Therefore, I have extra illustrated in Figure 4.12*a* the internal notifications sent to the layer management of the source (captured by the tester), reporting unexpected messages in the NULL state (U0) of calls. These are reporting that many *CALL PROCEEDING* and *CONNECT* messages were received too late, when the T303 and T310 timers already expired, and therefore these calls turned back to state NULL (for more details, please check the state transition diagram in Figure 8.3).

As a result, the throughput of calls (γ_R) will drop down. E.g., for 100 calls sent in a burst we obtain a throughput of 26% in Figure 4.12. Another 33% of calls were not considered successful due to timer expiry. This burst size is far over the limit of the *GDC APEX DV2* switch. We can observe that only 57 *SETUP* messages passed the switch out of 100, which demonstrates the buffer overflow in the switch. Previously, as shown in Figure 4.11, for 10 calls in a burst we still obtained a throughput of 100%.

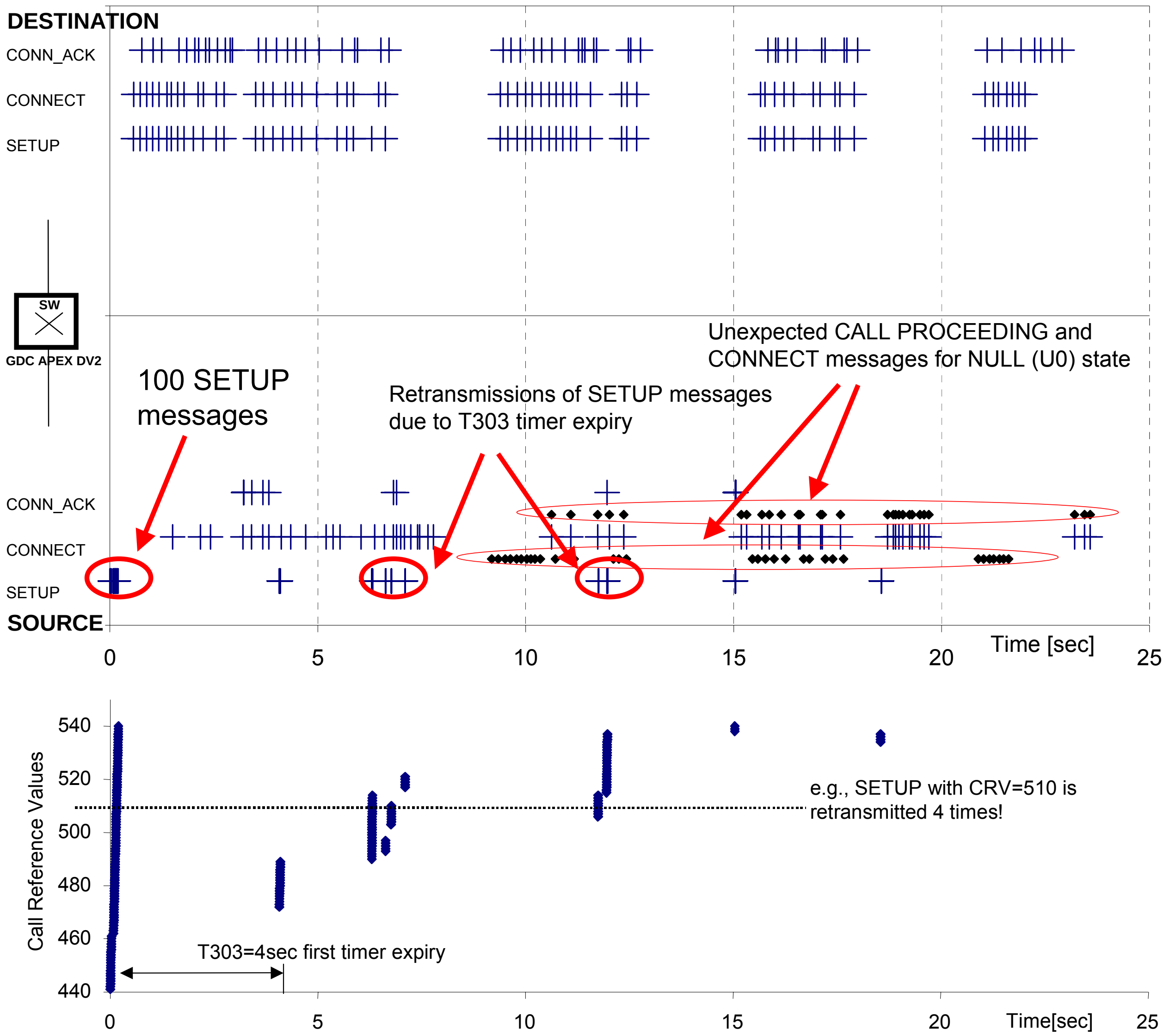


Figure 4.12 a) Burst of 100 calls at the GDC APEX DV2 (up); b) Retransmissions of SETUP messages (down)

This time, instead of using a "time-compression" technique as in Figure 4.9, we will represent the burst activity directly under the 'population-diagram'. Thus we can take a closer look to the retransmitted *SETUP* messages. Figure 4.12*b* shows that originally 100 *SETUP* messages were generated in a burst, but due to retransmissions we end up in 200 messages, while some of them were retransmitted even four times (see e.g., the *SETUP* message with the Call Reference Value CRV=510)! Opposite to the case study 1, here we obtained an offered average *SETUP* inter arrival time of 2 milliseconds with the same *HP BSTS 75000* tester, because the transmission was not blocked by the flow control mechanism of the *GDC APEX DV2* switch.

4.2.2.4 Analysing a burst of 1000 messages generated by a GNN iW95000

Finally, let us show that the 'population-diagram' can be used to test the capability (limits) of the *GNN iW95000* tester, before starting to carry out any performance test with it. Therefore, we configured Port A of the tester as source (user) and Port B emulated the network part. We illustrated both layer 2 and layer 3 messages in the same diagram, while generating a burst of 1000 *SETUP* messages, and simultaneously releasing calls after a certain call duration (see Figure 4.13). The average inter arrival times within a burst are obtained and shown in the Figure 4.13 as well. The tester is capable of generating *SETUP* messages at a rate of 1377 messages/sec (i.e., an inter arrival time IAT of 0.726 milliseconds), while establishing calls at a rate of 588 calls/sec (i.e., an IAT = 1.7 msec of *CONNECT* messages). The IAT of layer 2 *POLL* messages is much lower here (20 msec) compared to the *HP BSTS 75000* (0.75 seconds).

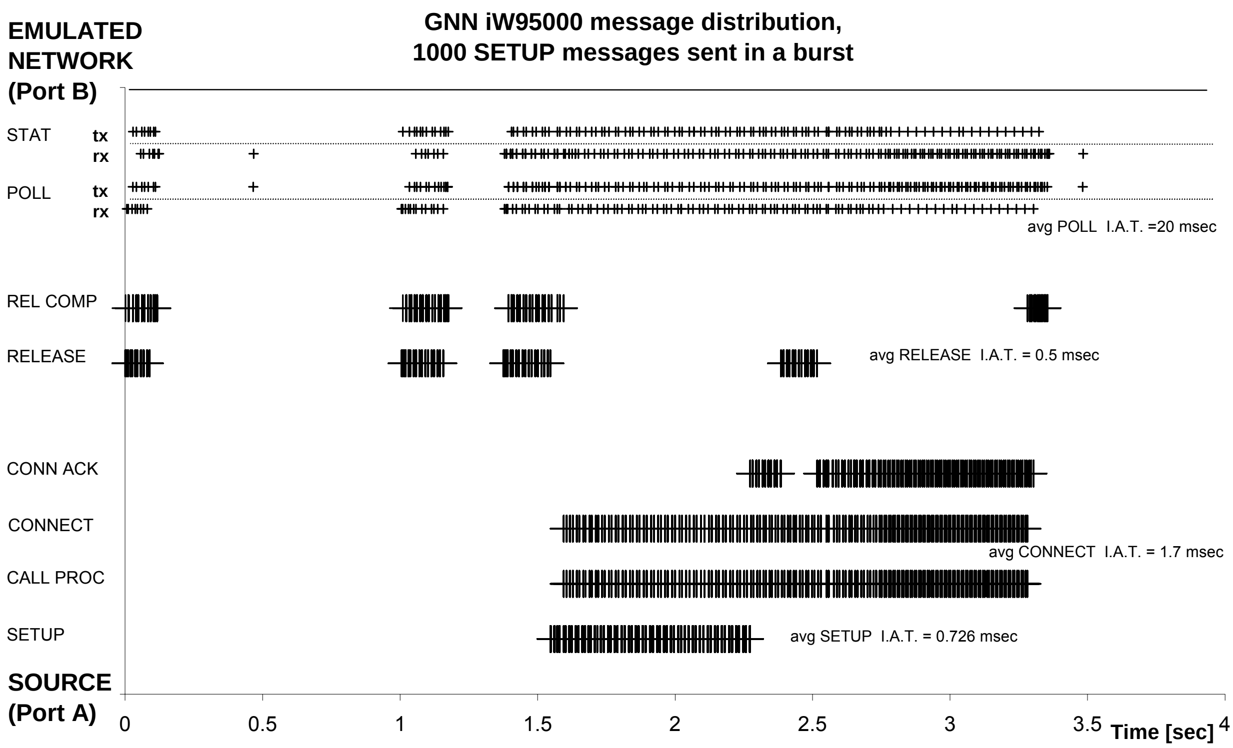


Figure 4.13 Population-diagram representing the capability test results of the tester GNN iW95000

4.2.3 Intrinsic properties of ATM signalling burst arrivals

After a detailed analyses of the burst measurement results on four switches, I have collected my conclusions in this section, describing *the properties of the signalling burst structure*.

I have found that at burst arrival of SETUP messages the following properties can be observed for the call establishment and release processes:

4.2.3.1 The first call

- The first call has always 80-100% longer establishment time than the second one.

This happens because either an address resolution from IP to ATM has to take place before a call can be initiated (case of sending 'ping' messages, see [Nov99]) or an end-to-end signalling path has to be established first (exchange of *BGN* and *BGAK* messages needed at each link to establish a peer-to-peer reliable data link before layer 3 signalling messages can be sent. Thus the first *SETUP* message will be delayed). Once this mapping is stored in the cache and/or the SAAL link is established, the next calls are executed in shorter time interval than the first one, showing then a gradual increase again as the burst size increases (see Figure 4.14*a*).

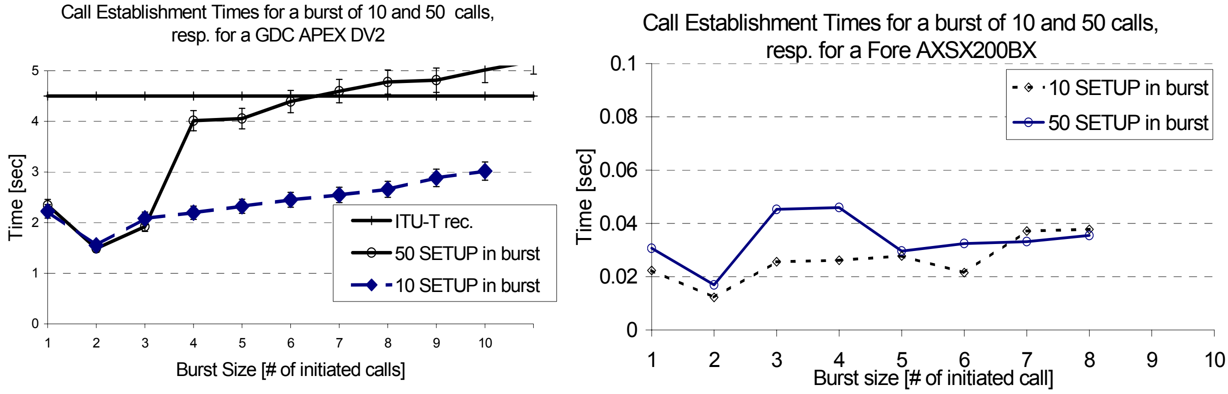


Figure 4.14 The first call establishment time is always longer than the second one a) GDC APEX; B) Fore ASX200

The first call establishment time will be always longer, regardless that the slope of the curve will be different as the burst size increases (except the Fore ASX200BX switch, where a flow control mechanisms blocks the transmission of messages after the credit window was used up, thus introducing a jump of 0.7 seconds starting with the 9^{th} message, not represented here in Figure 4.14*b*, but in Figure 4.7 previously).

4.2.3.2 The change in the CONNECT delay

- Opposite to the steady-state behaviour, in the first part of the burst the *SETUP delay* is lower than the *CONNECT delay*, but the gradient ($tg\ \alpha$) of these curves is the same until equation (11) is fulfilled, then the slope of the *CONNECT delay* starts to decrease with $tg(-\alpha),\ 0 < \alpha < \pi/2$.

In Figure 4.15 two examples are shown: a burst with 10 and 100 calls, respectively. The two main constituents of the call establishment time are shown, namely the *SETUP* delay and the *CONNECT* delay. More interesting is the evolution of the *CONNECT* delay in the burst. Due to the fact that all messages have the same priority, before the first *CONNECT* is sent back from the destination, the signalling buffer of the switch is already filled with the subsequent *SETUP* messages, thus the *CONNECT delay* will be longer than the *SETUP delay* (these details are illustrated again later in a 'population-diagram' in Figure 4.1). Later, when no more *SETUP* message arrives, then the *CONNECT delay* will gradually decrease. Its gradient (noted with $tg\ \alpha$) changes its sign at the point where the "signalling burst" ends, according to equation (11) in Def 1.3.

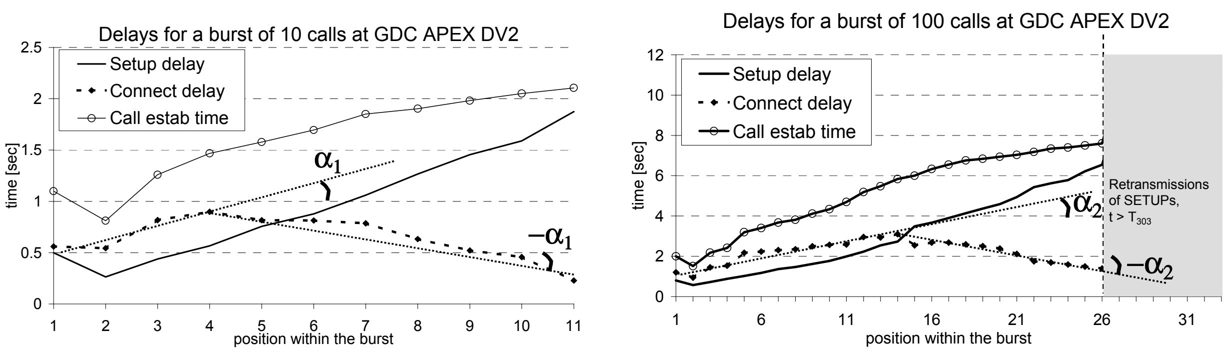


Figure 4.15 Change in the slope of CONNECT delays for a burst of a) 10 calls; b) 100 calls

4.2.3.3 Approximation formula of the Call Establishment Burst Time

In Figure 4.15*b* the marked region (right) shows a different characteristic, which is due to the apparition of the retransmitted *SETUP* messages, as their timer T_{303} expired. Thus we can define 5 phases of a signalling burst:

- the first phase contains the first call with a longer establishment time due to the phenomenon described above in Section 4.2.3.1;
- in the second phase the *SETUP* delay and *CONNECT* delay show an increase with the same gradient;
- in the third phase the *SETUP* delay further increases, but the slope of the *CONNECT* delay starts to decrease with gradient of (- $tg\ \alpha$);
- in the fourth phase the first retransmitted *SETUP* messages will appear causing a jump in the call establishment time (this phase is optional);
- the fifth phase occurs only by buffer overflow in the switch (e.g., the last 40 *SETUP* messages out of 100 in a burst were lost, as already shown in Figure 4.12).

The next property describes these five phases analytically:

- The *call establishment time* T_C is always longer for simultaneous call arrivals (transient state) than in steady-state of deterministic arrivals. Moreover, T_C depends on the size of the burst of the arriving *SETUP* messages that can be approximated as follows:

If the *burst size (b)* $\leq$ *max buffer size (BS)* of the signalling processor, then

$$T_{Ci}(b) = 2 \cdot T_{C2} \quad \text{for } i=1,$$
$$T_{Ci}(b) = T_{C2} + (i-2) \cdot tg\ \alpha(b), \quad \text{for } 1 < i \leq b \leq BS, \tag{12}$$

else $T_{Ci}(b) = 0,$ if the call is lost,

or $T_{Ci}(b) = T_{C2} + T_{303} + (j-1) \cdot tg\ \alpha(b),$ if the call is repeated,

where:

i = position of the message within the burst, $i = 1,2,...,b$ (for i=1 see Section 4.2.3.1);
$tg\ \alpha(b)$ = is the tangent to the curve of T_C, this gradient will be however different for different burst size (e.g., see Figure 4.14, and the effect of $\alpha 1$ and $\alpha 2$ in Figure 4.15);
T_{303} = 4 sec is the retransmission timer, see [Q2931];
T_{C2} = the second (*minimum*) call establishment time;
j = position of the message within the retransmitted burst, $j = 1,2,...,b$.

For example, in Figure 4.12 the burst size (100 messages) exceeds the maximum buffer size of the GDC APEX DV2 switch, which is approx. 60 frames, therefore 40 messages will be lost, furthermore we found 26 calls being successfully established (in the phase 2 and phase 3 of the burst, respectively). The remaining *SETUP* messages between 26 and 60 have been retransmitted, and some of them successfully established, but for the majority of them the *CONNECT* message came too late, therefore those calls were released (see also Figure 4.15*b*).

4.2.3.4 Approximation formula of the Call Release Burst Time

- Similarly, the *call release latency* T_{RN} can be approximated as follows:

If the *burst size (b)* $\leq$ *max buffer size (BS)* of the signalling processor, then

$$T_{RNi}(b) = T_{R1} + (i-1) \cdot tg\ \beta(b), \tag{13}$$

else $T_{RNi}(b) = 0,$

where:

i = position of the message within the burst, $i = 1,2,...,b$;
$tg\ \beta(b)$ = is the tangent to the curve T_{RN}.

In Figure 4.16*a* and Figure 4.16*b* the release delays, release times and release latencies are shown. For isolated switch measurements there is only a small difference between the release time, release delay and release latency, however this will be significant for a long chain of switches along a path. Figure 4.16*c* and Figure 4.16*d* show therefore the release times of calls in a burst of 10, 20 and 50, respectively. The flow control mechanism shows once again its effect in Figure 4.16*b*, therefore these measurements were repeated with iW95000 in Figure 4.16*d*.

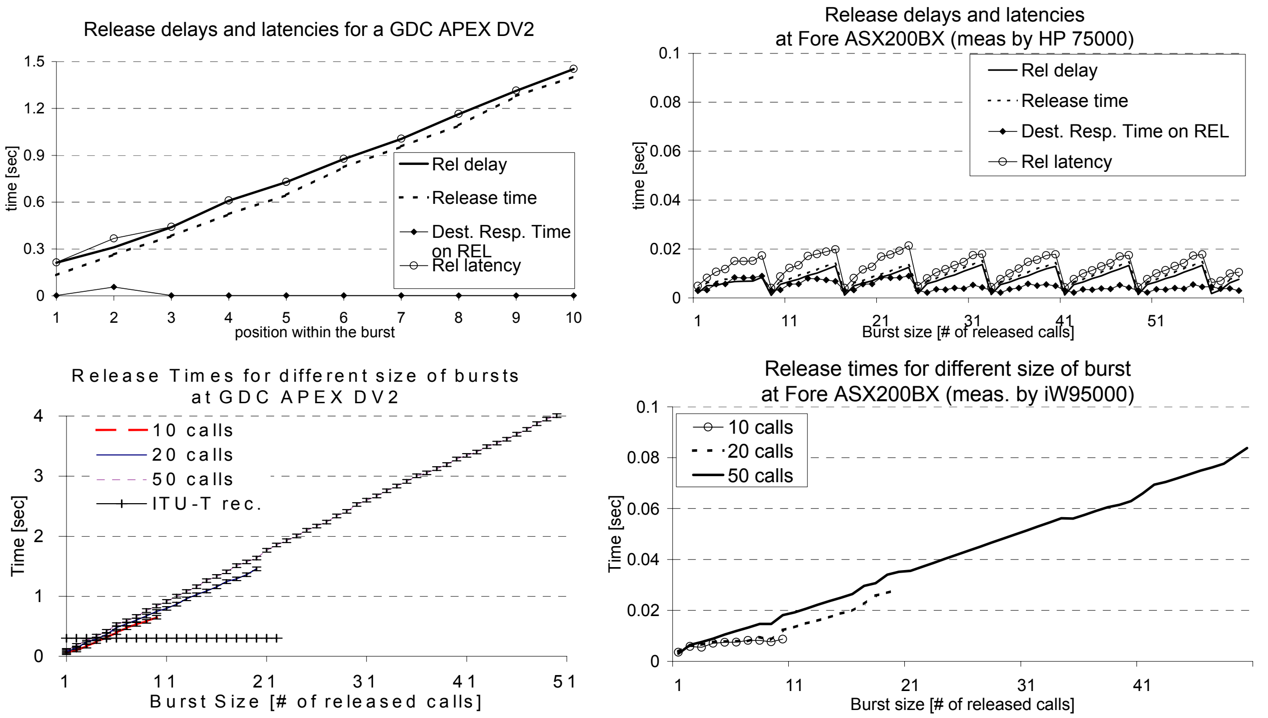


Figure 4.16 Call release burst times of 10, 20 and 50 calls, respectively on a) GDC; b) Fore; c) GDC; d) Fore

The call establishment burst time T_C *(b)* is the time that it takes to establish a burst of connections (see its definition in the Terminology section, chapter 10). From the equation (12) and Figure 4.15 it can be concluded that:

$$\frac{T_C(b_{10})-T_{C2}}{b_{10}-2}<\frac{T_C(b_{20})-T_{C2}}{b_{20}-2}<\ldots<\frac{T_C(b_{100})-T_{C2}}{b_{100}-2}<\ldots$$

or

$$tg\alpha(b_{10})<tg\alpha(b_{20})<\ldots<tg\alpha(b_{100})<\ldots$$

Similarly, the call release burst time T_R *(b)*, is the time it takes to release a burst of connections. From the equation (13) and Figure 4.16 it can be concluded that:

$$\frac{T_R(b_{10})-T_{R1}}{b_{10}-1}<\frac{T_R(b_{20})-T_{R1}}{b_{20}-1}<\ldots<\frac{T_R(b_{100})-T_{R1}}{b_{100}-1}<\ldots$$

or

$$tg\beta(b_{10})<tg\beta(b_{20})<\ldots<tg\beta(b_{100})<\ldots$$

The tangents to the curves of T_C and T_R ($tg\ \alpha, tg\ \beta$) depend on the following parameters: burst size, message type, call profile and processing capacity. All these dependencies can be collected in a tabled form e.g., as shown in Table 4.10. The burst size is increased gradually from 10 to 100 messages, the message type is either *SETUP* or *RELEASE*, the call profile is changed from simple calls ("default" type) to complex multi-media calls (e.g., CPP type *n*). As the ($tg\ \alpha, tg\ \beta$) depend on the capacity of the switch, therefore these values do not represent a general property of the ATM switches.

Table 4.10 General form of representing the (tg α, tg β) for different switches vs. burst size and call profile

Switch type Control Module type	***SETUP*** **type "default"**	***SETUP*** **CPP type 1**	**…**	***SETUP*** **CPP type *n***	***RELEASE***
HW version SW version Max. BS Port A type Port B type	T_{C2}; $tg\ \alpha(b_{10})$	T_{C2}; $tg\ \alpha(b_{10})$		T_{C2}; $tg\ \alpha(b_{10})$	T_{R1}; $tg\ \beta(b_{10})$
	T_{C2}; $tg\ \alpha(b_{20})$	T_{C2}; $tg\ \alpha(b_{20})$		T_{C2}; $tg\ \alpha(b_{20})$	T_{R1}; $tg\ \beta(b_{20})$
	…	…		…	…
	T_{C2}; $tg\ \alpha(b_{100})$	T_{C2}; $tg\ \alpha(b_{100})$		T_{C2}; $tg\ \alpha(b_{100})$	T_{R1}; $tg\ \beta(b_{100})$
	…	…		…	…

4.2.4 Further applications of the 'population-diagram'

4.2.4.1 Case study 3: Deterministic arrival of calls. Visual inspection.

The 'population-diagram' can be used to analyse the steady-state measurements as well. E.g., let us represent a test scenario where calls were generated with 2 calls/sec to a GDC APEX DV2. The parameters of the diagram are (m=8; n=1). It can be observed that the offered call rate equals the accepted call rate, i.e., 2 calls/sec (see Figure 4.17). Everything looks just perfect. No irregularity of message inter arrival times, the first call is released after 2.5 seconds, *STAT ENQ* and *STAT* messages are exchanged periodically every second.

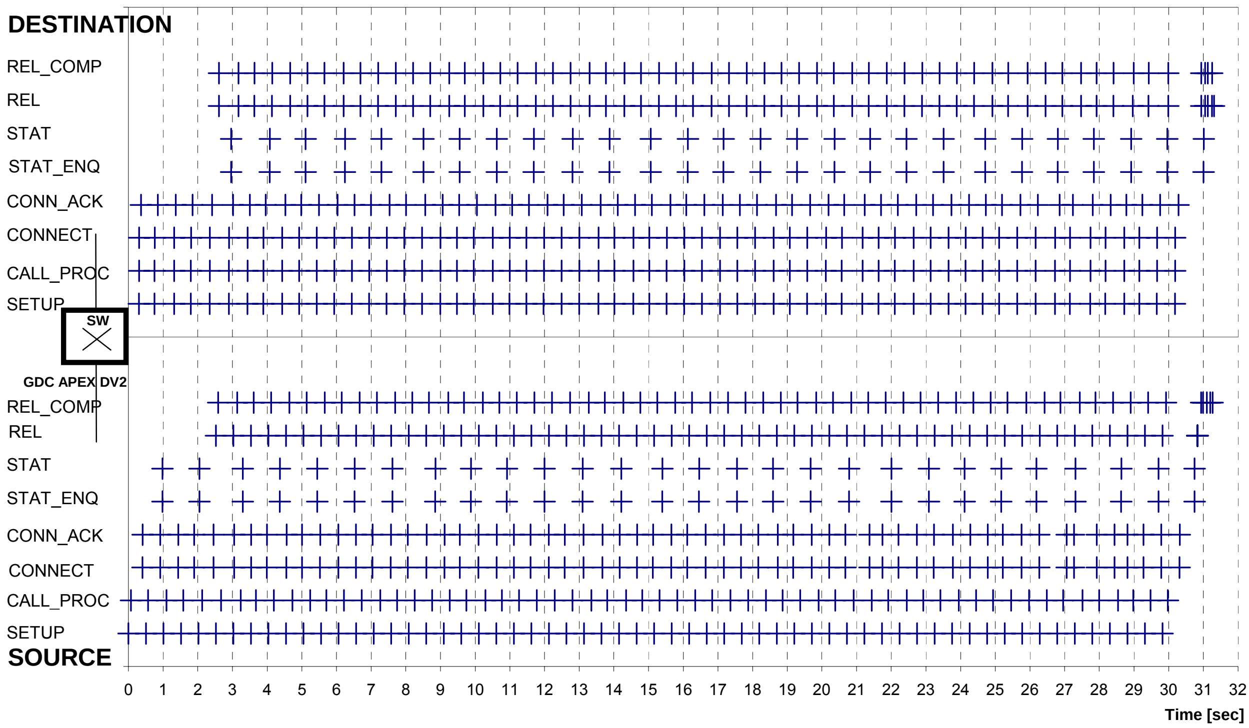


Figure 4.17 Deterministic call arrival with 2 calls/sec rate at the GDC APEX DV2

In Figure 4.18 it is shown that the offered call arrival rate of 6 calls/sec will be reduced to an accepted call arrival rate of 3 calls/sec on the link, when the HP75000 tester generates calls to a GDC APEX DV2 (generally, 3 *SETUP* messages can be count in each second). This problem was already discussed at the beginning of this chapter (see Figure 3.2). Furthermore, please note that the *STAT ENQ* and *STAT* messages are missing at the lower interface for the whole duration of the test, they appear also delayed by 3 seconds in the upper interface. The parameters of this population-diagram are (m=8; n=1).

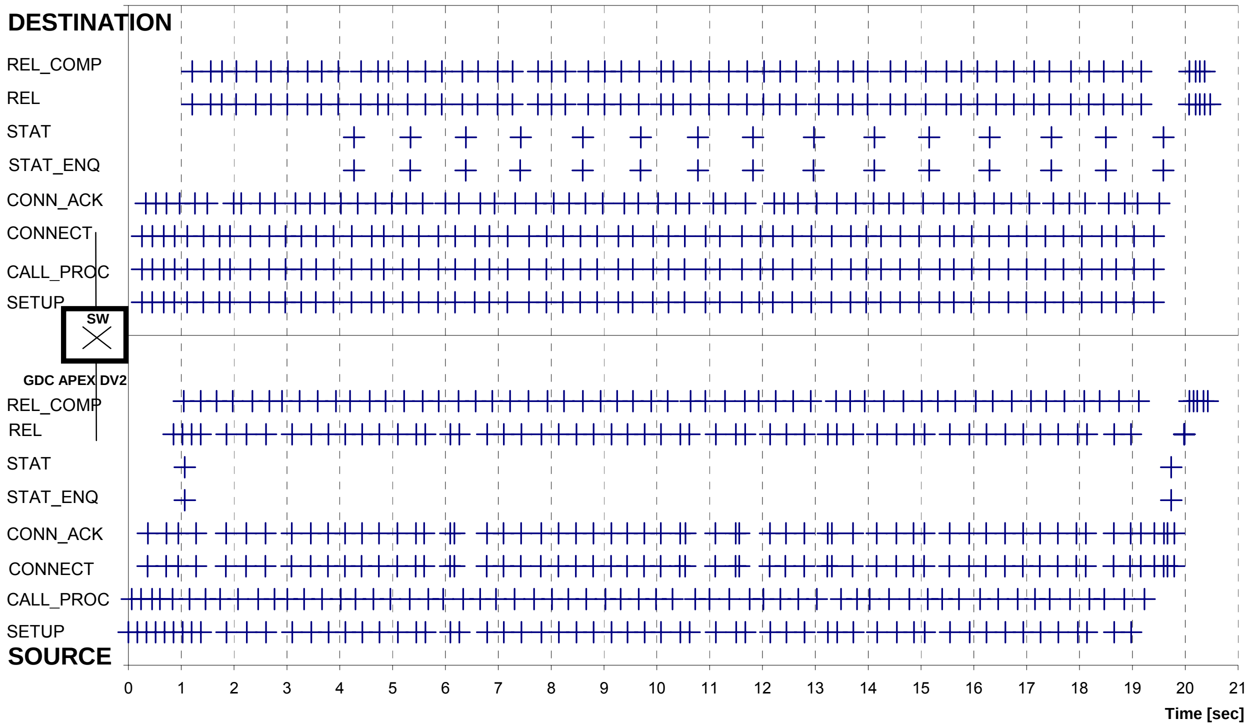


Figure 4.18 Example of accepted CAR of 3 calls/sec rate when originally offered at 6 calls/sec (HP75000)

Finally, in Figure 4.19 it is shown that the offered call arrival rate of 6 calls/sec will be reduced to an accepted call arrival rate of 4 calls/sec for the HP75000 tester generating calls to the Fore ASX200BX. The ‘population-diagram’ helps us again to show that the same link layer flow control problem as discussed in case study 1 is responsible here for the limitation of the accepted call arrival rate. Except the first couple of seconds, the chart shows a similar behaviour of the switch as in the “chopped” burst case studied in Figure 4.8 (*m*=8; *n*=1).

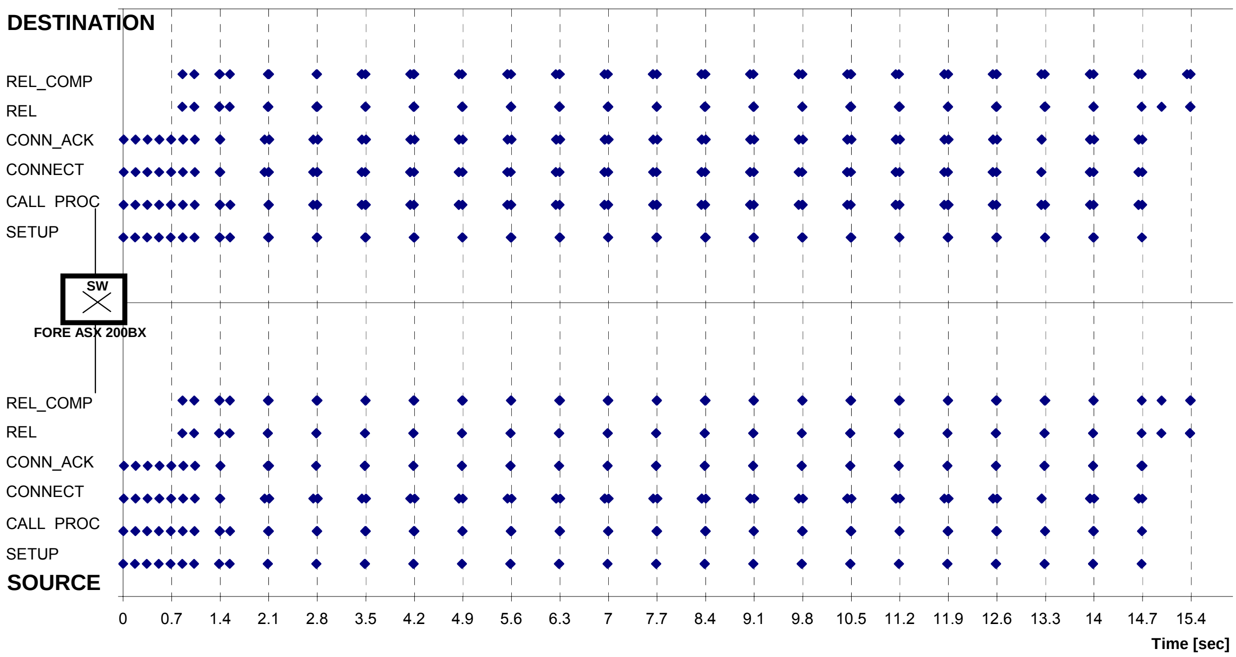


Figure 4.19 Example of accepted CAR of 4 calls/sec rate when originally offered at 6 calls/sec (HP75000)

4.3 Validation of the measurement results

The properties of the ATM switches presented in Chapter 4 represent the results of a very "unpopular" research, the measurements were collected during the last four years by our research team on four different generation of switches manufactured by different vendors. Moreover, the results were obtained with two different testers, and in addition, independent of our work five other research groups (see [Nie97], [Nov99], [Pil99], [Far01] and [Mau01]) obtained very similar results with other types of ATM switches, thus confirming that these properties are general to a wide range of ATM switches.

All of our measurements were repeated 10 times, one set containing 60 to 100 calls. The minimum, average, maximum values, the standard deviations and correlation coefficients were evaluated on these sets of measurements.

Unfortunately our tester (*HP BSTS 75000*) did not capture more than 100 calls at one run. Moreover, it allocated the same time-stamp for all PDUs belonging to one message travelling up or down the UNI, SSCOP, and AAL5 layers. Therefore, we had to use an indirect way to prove our statement in Section 4.1.1.

The statements in Section 4.1.5 and Section 4.1.6 could not be all tested on the Fore ASX200BX switch with the HP75000 tester due to the known flow control problem, therefore we have repeated these tests later with the GNN iW95000. Basically, the call intensity tests over 3 calls/sec were all supported by the iW95000. We have carried out detailed analyses of all message delays and we have also calculated the accepted call arrival rates each time. Thus we have shown that the nature of the accepted call arrival rate changes from constant rate to an on-off type of arrival at a certain threshold for the HP75000, but the average values of the call establishment times and release latencies remain the same as before (see Figure 4.2*f*).

Representing the measured results in a 'population-diagram' gives us immediately a '*visual inspection*', a first-step validation of our results. Secondly, it helps us discovering interdependencies which are not visible measuring the call establishment times and release latencies only (e.g., retransmission of lost messages, unexpected messages in a certain state, biased results due to the layer 2 flow control mechanism, etc.)

4.4 Guidelines for signalling performance measurements

I have given some evaluation hints, proposals and several practical conclusions to be considered during the measurements of ATM signalling, which can be used by ATM network designers, network operators and those they intend to carry out signalling performance measurements in the future (see [C-10]).

- Before starting the tests, the performance of the test equipment has to be validated first. In many cases the switch (SUT) might be faster than the tester. It is recommended that the offered and accepted call arrival rates will be always compared.

- To increase the precision and performance of the measurement tool it is recommended to perform off-line decoding and analyses. The usage of special signalling software tools which emulates the users might be a useful extension of the testbed, but usually there is no possibility for synchronised clock between independent PCs, and thus not all performance parameters can be obtained.

- The timestamps of all signalling messages have to be captured at both sides of the switch (input and output port). The results are valid only together with the input parameters. They must be reproducible any time. To measure the performance of the control plane (and not the bottleneck of the user plane), available bandwidth has to be always guaranteed on the links.

- It is insufficient using only the call establishment time (T_C) to characterise the call processing, but T_{CN}, T_{DS}, T_R, T_{RN}, T_{DR} and γ_R need to be measured as well. Moreover, the dependency on the message content has to be taken into account (e.g., Complex Call Profile type '*n*'). The measured results must not be accepted if $\Delta T_{DS}/T_{CN}$ or $\Delta T_{DR}/T_{RN}$ are increased over 10% during the given test case (impact of the destination on the measured performance). Timer settings may also influence the throughput (see e.g., the effect of timer T_{303} in Figure 4.12).

- In addition to deterministic arrivals of calls, burst arrivals of calls must be investigated, and if possible, poissonian arrivals as well.

- By burst arrival of calls, analysing the call establishment time only cannot answer the questions related to the complexity of the burst structure (e.g., biased results, unexpected messages). Therefore the representation of the whole burst event in a 'population-diagram' is highly recommended (see Section 4.2.1).

- If the representation in a 'population-diagram' is not applied, at least it is recommended to verify the formula: $T_C = T_{CN} + T_{DS}$ to detect if the measured layer 3 message delays are biased by the layer 2 flow control mechanism.

4.5 Conclusions

By a detailed analyses of burst arrivals of signalling messages (see Section 4.2), I had to conclude that the methodology used by other researchers (e.g., [Pil99], [Mau01]) is not sufficient to analyse burst arrivals of calls and therefore I have introduced a new method of representing the complete set of messages belonging to a burst in a so called 'population-diagram', thus opening new dimensions of analysing the structure of such bursts. Basically, this method can be used not only for burst arrivals, but any kind of signalling measurements. Section 4.2 then analysed the performance simultaneous (burst) arrival of signalling messages based on a new definition of the signalling burst.

However, while all companies and research groups interested in ATM are united in their interest, there is less consistency among the *evaluation metrics* and *measurement methods* applied. This makes it difficult to compare switches from different vendors, which is particularly important for service providers. I have also pointed out (delivering examples in Section 3.1) that there is no consensus in the current research papers for using the same terminology for the evaluation metrics, which makes it quite complex to compare their results to ours. I was trying to explain and adapt all these divergences to the newest standard adopted in 2000 by the ATM Forum [ATMF00]. A couple of new definitions were also necessary for a better characterisation of signalling performance in large networks (see Section 4.1.1 and later in Section 6.3).

The importance of these new properties (only 20% was known before) is reflected in the fact that they provide detailed analyses of each component of the call establishment time, release latency and burst structure. Moreover, I have shown that these properties are intrinsic to ATM signalling, but they are very general regardless of different architecture and processing power of ATM switches. Of course, there are features which depend on software implementations and hardware architectures (e.g., see [Nie97]), but those were not subject of my investigations. The only exceptions are the parameters ($tg\ \alpha, tg\ \beta$), analysed in Section 4.2.2, which depend, as mentioned, also on the processor's capacity (HW and/or SW).

Collecting the results was the most time consuming activity as it lasted for more than 4 years investigating four different generation of switches and comparing them to several others.

These results presented in Chapter 4 form a basis for development of a new signalling processing model in Chapter 5.

CHAPTER 5

5 Construction of a generic call processing model for ATM networks

While tests addressing the properties of a single switch are important and clearly relevant to LAN performance, it is obvious that tests addressing larger LAN and much larger WAN configurations are also relevant. But the problem is the unavailability of such large ATM networks with signalling capabilities. To buy a representative number of ATM switches (e.g. 30 nodes) just for testing purposes is too expensive and not arguable. Analytical performance evaluation like flow analysis supplies good results for the mean values only. Emulation of signalling protocols allows a deep insight in the protocol behaviour, but implies limitations in the performance, and is especially difficult to emulate all nodes of different manufacturers. Therefore, I have turned to *simulations*[3]. This approach supplies results on a higher abstraction level, but the tool is able to simulate all different vendors' nodes. It provides call establishment times, release latencies, bandwidth utilisation of links, call throughput and signalling load of nodes. My investigations include a construction of a new message flow model for one signalling node and then analysis of cascaded and arbitrary network topologies. The results are validated by measurements on 2-to-4 cascaded switches and on the 7-node TEN-155 network, see [Nov99].

The construction of a call model for UNI and Private Network Node Interface [PNNI] has been motivated by the fact that many service providers want to introduce dynamic call establishment in their access Digital Subscriber Line (*x*DSL) and backbone ATM network. However, today is almost impossible to have access to large ATM networks with signalling capabilities. Some very basic results have been obtained on the *TEN-155* Pan-European ATM network, reported by [Nov99]. This network consists of seven ATM signalling nodes, the longest path (diameter) contains only four nodes. Therefore, our experiences gained in a small network have been extended by simulation to real size networks. Moreover, as in Chapter 4, Section 4.1.2 we have shown that in many cases the test equipment may be the bottleneck, therefore we have developed an adequate model for the end systems as well.

5.1 Construction of the model

5.1.1 The architecture of the model

I have developed a new call processing model for an ATM signalling node based on the specifications [Q2931], [UNI40], [PNNI] and the measurement results presented in Chapter 4, respectively.

[3] The simulation study was carried out based on the simulation software called ACCEPT, developed by the author and his colleague I. Moldován, HSN Laboratory, Hungary, for further details see the Appendix B.

The [Q2931], [UNI40] and [PNNI] standards do not specify any call model, but describe the basic functionalities, features, format of the signalling messages and the protocol how they interact. The information gained from these documents was complemented by the results of Chapter 4.

For example, from the results presented in Section 4.1.2, 4.1.4 and 4.1.6, it is clear that the average call release latency is always shorter than the average call establishment latency, furthermore, the ratio between these parameters varies as the call arrival rate exceeds the service rate of calls. These things suggest somehow, that the *SETUP* and the *RELEASE* messages visit different paths inside the call processor. Another example is given in Figure 5.1 for a burst arrival of 10 and 100 calls, respectively.

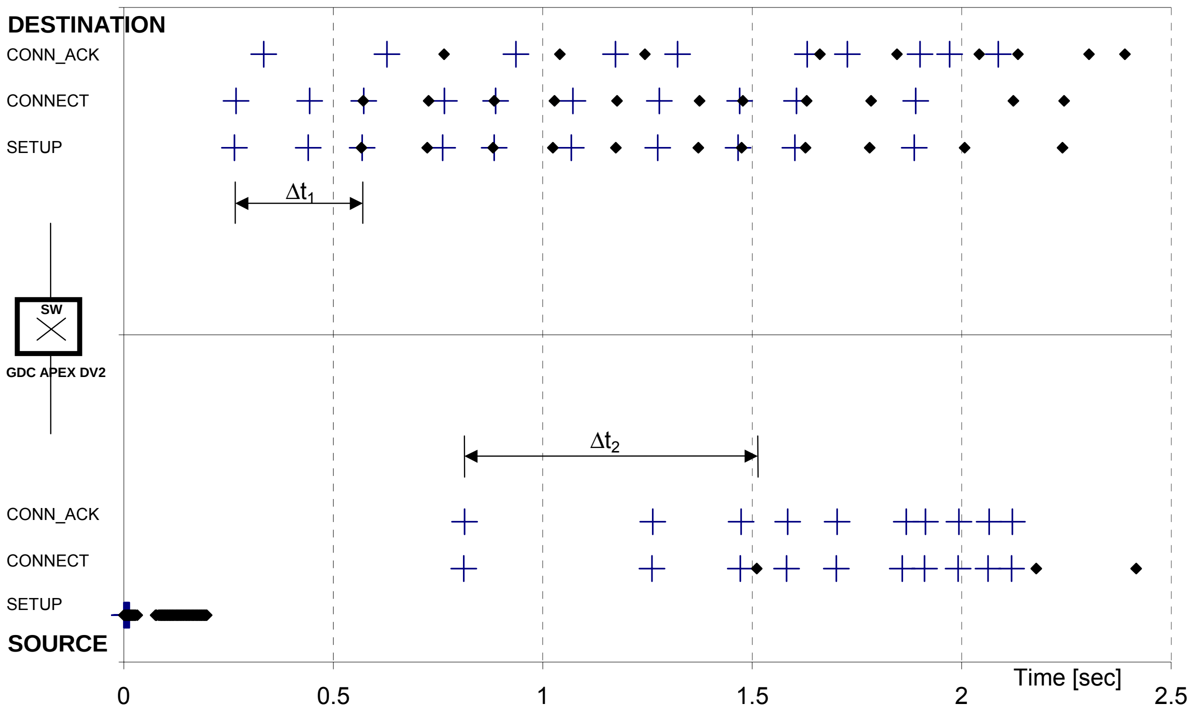


Figure 5.1 Comparison of a burst of 10 and 100 messages at the GDC APEX DV2 (first 2.5 seconds)

When we superpose Figure 4.11 and Figure 4.12 we obtain Figure 5.1. The first *SETUP* message out of 100 suffers a delay Δt_1 compared to that one from a burst of 10 messages passing through the switch. Similarly the first *CONNECT* message out of 100 suffers a delay Δt_2 compared to that one from a burst of 10 messages passing the switch. The fact that $\Delta t_2 > \Delta t_1$ suggests a *FIFO* queueing mechanism for all incoming calls.

These features are all reflected in my new model. I have investigated two different mechanisms in the proposed model: *FIFO queueing* and *priority queueing.* The *FIFO* queueing is investigated in this chapter, while the investigations for priority queueing are presented later in Chapter 7. Moreover, at one node I have distinguished *separate processor phases* according to the jobs to be done (see Figure 5.2). These are as follows: *UNI* and *PNNI* to decode/encode the incoming/outgoing messages, *CC* to create and update the objects related to one call, *RT* for path selection, *BW* for bandwidth allocation/de-allocation on the outgoing link, and finally, *CCP* for execution of a complex call profile, buffer allocation and Quality of Service issues.

E.g., there are 5 processes visited by a '*default' SETUP* message (UNI→CC→RT→BW→PNNI) and one more (CCP) by a '*complex' SETUP* with additional capabilities (according to Section 4.1.4). Instead, *CONNECT* and *RELEASE* messages visit only 3 processes (UNI↔CC↔PNNI). The local acknowledgements (e.g., *CALL PROCEEDING*, *CONNECT ACKNOWLEDGE*) are not included in this model due to the fact that they do not have a substantial effect on the performance measures. The only difference to the standards is that our *RELEASE COMPLETE* message has end-to-end meaning, visiting the following path: PNNI→CC→BW→UNI (this step was necessary to avoid buffer de-allocation problems in the model). Furthermore, we implemented the layer 2 handshaking process (*POLL*, *STAT*) between adjacent nodes in our model for flow control, except the SAAL establishment (*BGN, BGAK*) and release (*END, ENDAK*) as we consider that the SSCOP layer connection is always alive. The ATM specific part of this model is given by the followings: different bandwidth requirements can be served in the *BW* block, different internal paths are visited according to the message types, and different service rates are obtained due to specific call profiles, routing, QoS service guaranties, buffer allocation for CBR and VBR traffic, etc.

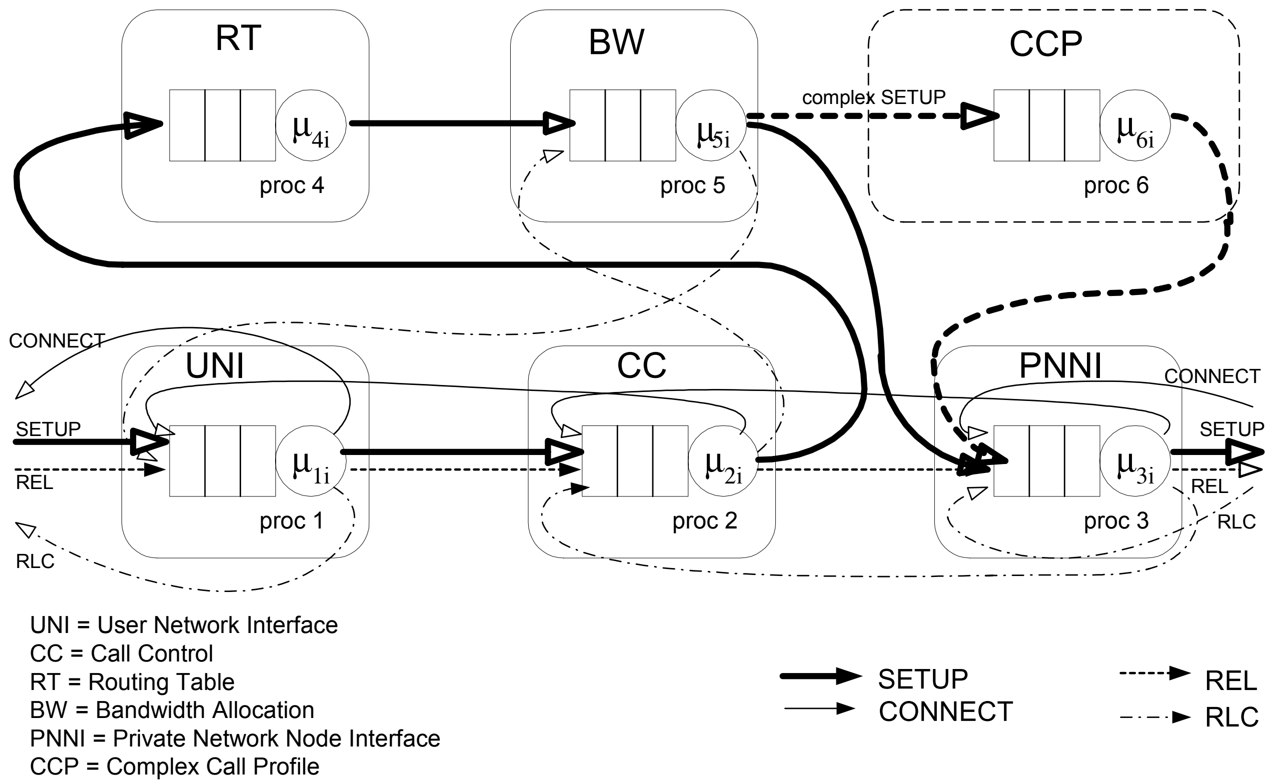


Figure 5.2 Call processing model of an ATM signalling point

The service rate of *RT* is dependent on the size of the routing table (see Section 4.1.3) and on the level of PNNI peer-group hierarchy. A higher PNNI level gets a lower service rate (according to the results of [Nie97]).

Finite buffer sizes and deterministic service rates are considered at each process. The sum of the buffer sizes of each process represents the *buffer size (BS)* of the processor. Some messages may be lost at overload due to the finite buffer capacity. Therefore, a protection mechanism is introduced in our model: the *CONNECT*, *RELEASE* and *RELEASE COMPLETE* messages are retransmitted any time they fail to attend the queue of the next process due to buffer overflow. This retransmission is practically a reinsertion of these messages at the end of the same queue after a certain delay. This delay is called the *retransmission delay of lost messages (RDLM)*. In contrast, a *SETUP* message is never reinserted at the end of a queue at buffer overflow, it is lost and a *RELEASE COMPLETE* message is sent on the path back to release the allocated resources of that given unsuccessful call. The *SETUP* may only be retransmitted by the source after expiry of the T_{303} timer (optional).

Two other call processing models have been developed in [Wu98] and [Gel97]. While the model in [Wu98] is even more complex than ours and considers different service times for CBR, VBR and UBR calls respectively (which is irrelevant according to our measurements, see Section 4.1.4), none of the two other models considered the release phase. However, as described in Section 4.1.5, the T_C is increased by 15-20% when release messages are also present in the network. Furthermore, the model in [Gel97] consists of a single *FIFO* queueing model, which cannot capture the differences between the call establishment times and release latencies (see also Section 5.2).

5.1.2 Parameter settings of the model

The problem can be formulated as follows:

- The known parameters are the measured message delays (*SETUP delay, CONNECT delay, RELEASE time* and *RELEASE delay*), the call throughput of the switch (γ_R), the call establishment times and the release latencies vs. call arrival rate (e.g., see Table 4.3*a-h* and Figure 4.5*a-d*);
- It has to be determined the service rate (μ_i) and the buffer size (BS_i) of each process of the signalling processor and the retransmission delay of lost messages *(RDLM)*[4].

I have shown that the *service time* of each process can be derived from the system of equation (14). The *buffer size (BS_i)* of each process and the *retransmission delay of lost messages (RDLM)* are derived by simulation. The input parameters needed for simulation settings are the service rates, obtained by solving the system of equation (14), then the call throughput of the switch, the call establishment times and release latencies obtained by measurements at different call arrival rates.

I have developed an algorithm to obtain the μ_i, $BS = \sum_{i=1}^{6} BS_i$ *and RDLM, which consists of four steps:*

I. *Set the service rates according to the system of equations (14).*

$$\sum_{i=1}^{5}\frac{1}{\mu_i} = \min\ SETUP\ delay \qquad \sum_{i=1}^{6}\frac{1}{\mu_i} = \min\ 'complex'\ SETUP\ delay$$

$$\sum_{i=1}^{3}\frac{1}{\mu_i} = \min\ CONNECT\ delay^{5} \qquad \sum_{i=1,2,3,5}\frac{1}{\mu_i} = \frac{1}{2}\cdot\min(RELEASE\ \ time + RELEASE\ \ delay) \qquad (14)$$

$1/\mu_1 = a/\mu_2 = 1/\mu_3$, where $a \in (1,2]$ is a correction factor. Modify $a \in (1,2]$ until: $|T_{CN}^{meas} - T_{CN}^{sim}| < 0.05 \cdot T_{CN}^{meas}$ and $|T_{RN}^{meas} - T_{RN}^{sim}| < 0.05 \cdot T_{RN}^{meas}$, while γ_R=1.

II. *Set the length of the bottleneck buffer BS_{j^*}, $j^* = \min\{j \in \{1,...,6\} \mid \forall k \in \{1,...,6\} : q_j \ge q_k\}$ so that the threshold values $\left|\lambda_{Th}^{sim} - \lambda_{Th}^{meas}\right| < 0.05 \cdot \lambda_{Th}^{meas}$, where $\lambda_{Th} = \min_{i \in I}\{\lambda_i \mid \frac{T_{CN,i+1} - T_{CN,i}}{\lambda_{i+1} - \lambda_i} \ge 1\}$, I is a finite set of measured/simulated data and q_j is the message length of queue j.*

III. *Set the length of the remaining buffers until the throughput $|\gamma_R^{meas} - \gamma_R^{sim}| < 0.05 \cdot \gamma_R^{meas}$ $, \forall \lambda > \lambda_{Th}$.*

IV. *Set RDLM to further minimise the errors between the three measured and simulated curves in the overload region $(\lambda > \lambda_{Th})$.*

5.1.3 Numerical results

Table 5.1 gives a couple of numerical examples of calculating the service times according *step I*, the system of equation (14), where the minimum message delays are obtained from Table 4.3*a-h* (except the switches *Ascend CBX500* and *CBX550*, obtained from [Nov99]). The introduction of a correction factor '*a*' was necessary as the coefficient matrix of the system of equation (14) is singular. The last two columns in the Table 5.1 indicate the value of this factor '*a*' and the *RDLM* parameter obtained in *step IV* of the algorithm, respectively. The maximum service time of the CCP process has been obtained for the last row of Table 4.7 (adding optional higher layer IEs). For 'default' *SETUP* messages the CCP process has to be set to null. The bottleneck buffer obtained in *step II* is marked with the grey field for each modelled switch (except the *Seabridge XP140* and *Newbridge MSX36170*, the other switches have the buffer of the RT process as a bottleneck).

[4] *RDLM* is the delay a lost message (other than *SETUP*) is regenerated at the switch in case of buffer overflow.
[5] or min *RELEASE delay*

Table 5.1 The service rates and buffer sizes obtained from the algorithm 2.1.2 for different switches

Switch type	UNI		CC		PNNI		RT		BW		CCP max (default)			
	1/μ(1) [ms]	BS(1)	1/μ(2) [ms]	BS(2)	1/μ(3) [ms]	BS(3)	1/μ(4) [ms]	BS(4)	1/μ(5) [ms]	BS(5)	1/μ(6) [ms]	BS(6)	RDLM [sec]	a
GDC APEX DV2	30	5	20	5	30	5	200	6	40	6	120 (0)	6	3.5	1.5
Fore ASX200BX	0.6	30	0.3	20	0.6	30	2.7	110	0.7	50	3 (0)	60	0.5	2
Seabridge XP140	3.5	10	2	10	3.5	10	3	10	12	12	3 (0)	10	2	1.7
Newbridge MSX36170	1.8	30	1	30	1.8	30	6	30	7	40	3 (0)	30	1	1.8
Ascend CBX500	2	5	1	5	2	5	8	5	2	5	n.a. (0)	5	1	2
Ascend CBX550	5	5	3	5	5	5	13	5	6	5	n.a. (0)	5	2	1.6

Once the service rates of the processes are set (obtained in *step I*), the simulation has to be run repeatedly in *step II*, starting at large buffer sizes and reducing them (while keeping all equal) until the conditions set for the threshold values of λ_{Th} are fulfilled (thus obtaining the bottleneck buffer size). In *step III* then the remaining buffer sizes will be set to match the characteristics of the throughput curve. Finally, in *step IV* the retransmission delay *RDLM* will be set to further minimize the errors of the simulated curves vs. the measured ones. The effect of selecting different buffer sizes is illustrated in Figure 5.3 for a *GDC APEX DV2* and a *FORE ASX200BX* switch, respectively. As the buffer size (sum of the buffer sizes of individual processes) is decreased, the call establishment time suffers a more abrupt jump at the overload region (due to retransmissions). According to *step II* and *step III,* we have selected the *BS=33* for *GDC APEX DV2* and *BS=300* for *FORE ASX200BX* , respectively.

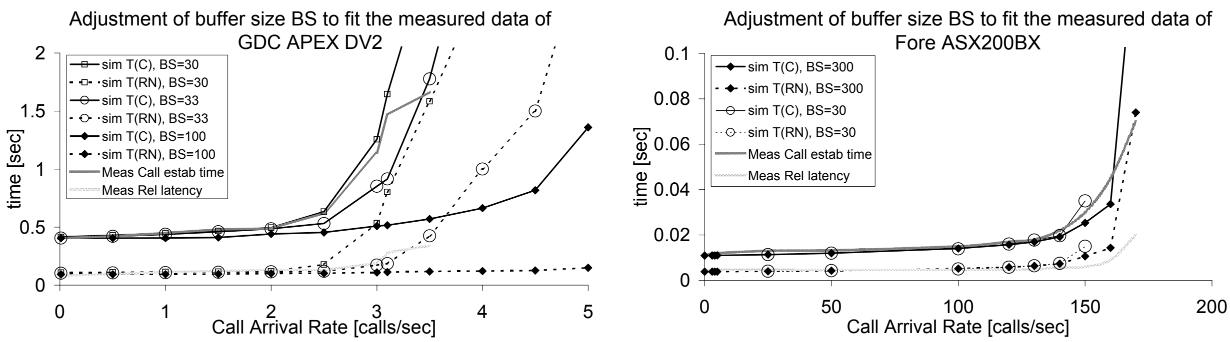


Figure 5.3 Setting the BS of the model (step II and step III of the algorithm)

The output parameters are determined such that the absolute error between simulation results and measurement results is less than a predefined value (e.g., 5%). It is important that all three curves obtained by simulation (the curves of the call establishment time, throughput and release latency) have to converge simultaneously to the measured results. The effect using multiple hosts and the impact of selecting different values for the *RDLM* parameter is illustrated in Figure 5.4 for a *GDC APEX DV2*.

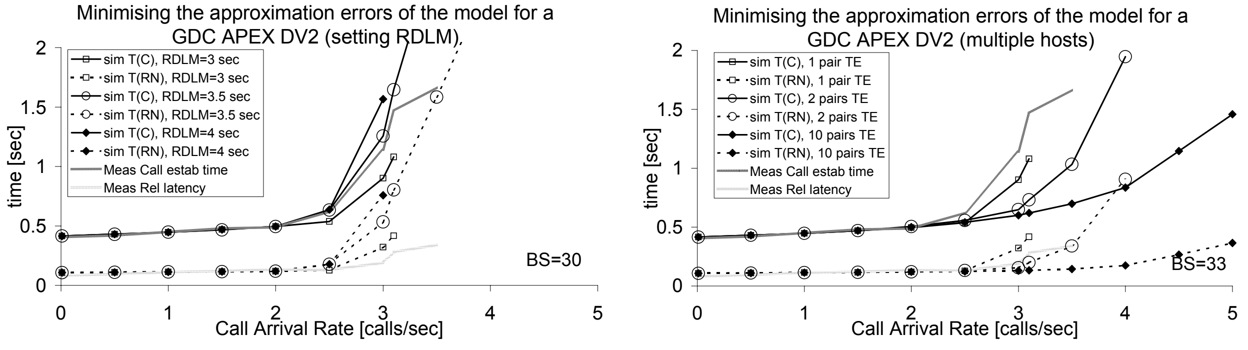


Figure 5.4 Use of multiple hosts and settings of the RDLM of the model (step IV of the algorithm)

Use of multiple hosts (e.g., 10 independent source-destination pairs) was beneficial at higher call intensities (e.g., over 100 calls/sec) as a single host introduced much larger delays at such high call arrival rates. As illustrated in Figure 5.3 and Figure 5.4, the must delicate problem is to adjust the parameters of the model in the overload region.

5.1.4 Validation of the results

First of all, I have shown that the simulated results of all three parameters (T_{CN}, T_{RN}, γ_R) can be adjusted to the measured ones for all four switches we have studied (e.g., see Figure 4.5*a-d*). Secondly, I have validated my model against a 7-node network, described in [Nov99].

To reduce the effect of the destination response time, we have replaced the (1 source; 1 switch; 1 destination) configuration by that of (10 source; 1 switch; 10 destination). In our simulation studies we have generated constant and Poisson arrival rates, respectively. The average values were obtained from 10000 calls.

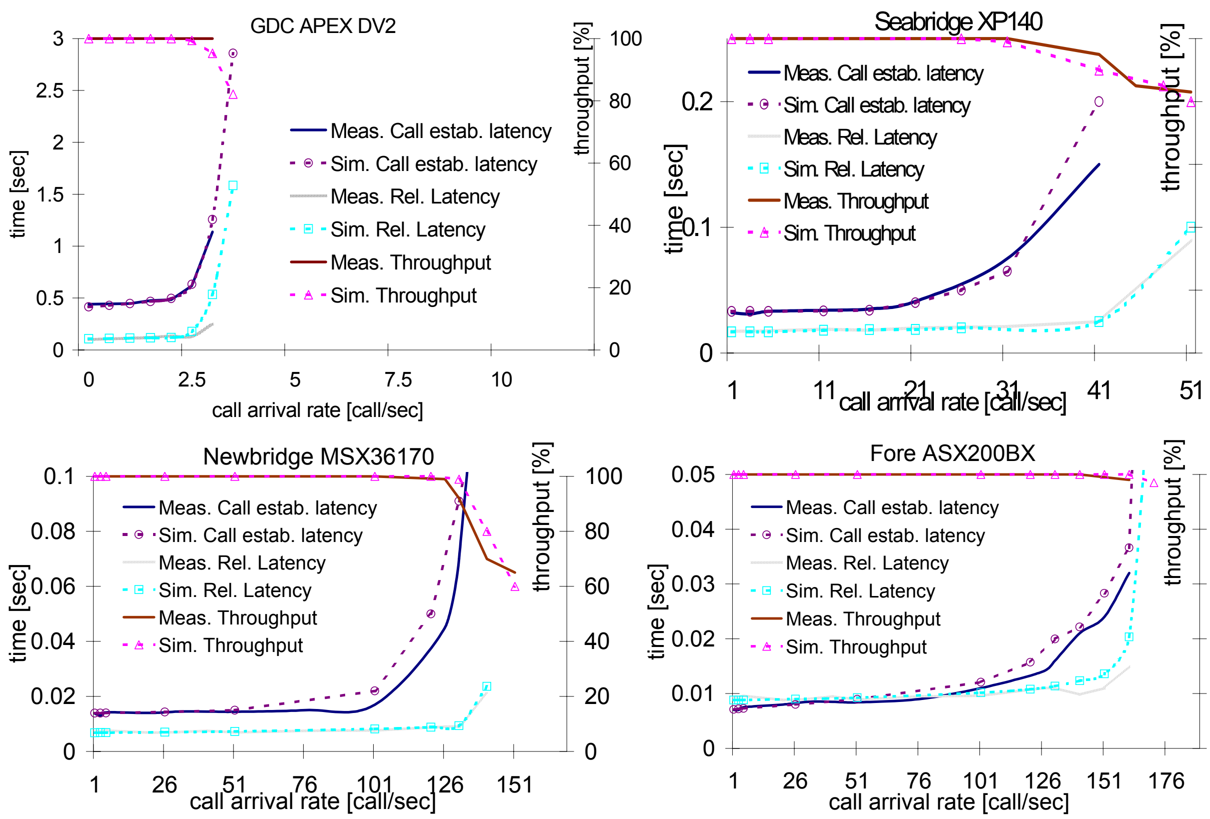


Figure 5.5 Validation of the model for 'default' SETUP messages:
a) GDC APEX DV2 switch; b) Seabridge XP140 switch; c) Newbridge MSX36170 switch; d) Fore ASX200BX switch

To validate our model in a real environment, we have selected a 7-node ATM network (see Figure 5.6), representing the *TEN-155* network offering service to the European research community [Nov99]. The backbone nodes are all *LUCENT* (formerly *ASCEND*) *CBX500* (with a signalling service time of 25 msec, i.e. 40 calls/sec) except the nodes is Sweden (SE) and Italy (IT), where a lower speed *CBX550* (50msec) is placed. The 8th node from Switzerland (CH) has not been used for signalling. The workstations at the sites are all SUN workstations with different processor capacities using SUN ATM CLIP software to generate ATM signalling. The SVC network paths consist of 2, 3 or 4 hops, respectively, but we have observed 12 different call establishment times due to different switches and workstation speeds. The measured call establishment results are presented for each node-pairs in [Nov99]. Each measurement result was an average of 5-6 measurements, each consisting of a stream of 'ping' messages. Our simulated results consisted of 3000 calls at 1 call/sec rate. The results are compared in Table 5.2.

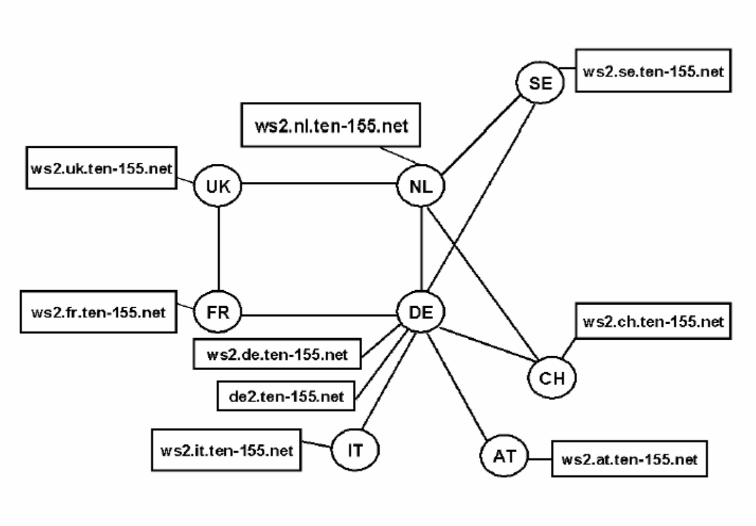


Figure 5.6 The backbone nodes of the TEN-155 network

The measured results are placed in the lower triangle, while our simulated results in the upper one. For the measured results we have found only the average values in [Nov99] (*x(A)..x(B)*, for both directions) and a standard deviation of 4.4, while for the simulated results we have obtained the Min/Avg/Max values and the standard deviations.

Table 5.2 Validation of our model against network-level measurements in the TEN-155 network

Simulated results → Min/Avg/Max (StDev)	DE	FR	NL	UK	AT	SE	IT
Measured results ↓ Avg	[msec]	[msec]	[msec]	[msec]	[msec]	[msec]	[msec]
DE	———	45/45.1/60.7 (0.62)	45/45.1/60.7 (0.62)	75/75.2/90 (1.2)	55/55.2/68.7 (1.05)	90/90.6/125.8 (2.47)	80/80.4/104.6 (1.57)
FR	44..45	———	65/65.2/73 (0.78)	55/55.2/69 (1.08)	75/75.3/91 (1.19)	110/110.6/132.6 (2.49)	100/100.4/127 (1.62)
NL	40..45	65..66	———	55/55.2/69 (1.08)	75/75.3/91 (1.19)	90/90.6/125.8 (2.47)	100/100.4/127 (1.62)
UK	79..80	54..66	41..52	———	105/105.4/118 (1.36)	120/120.8/143.7 (2.9)	130/130.5/152.8 (1.73)
AT	53..54	80..85	75..76	115..118	———	120/120.7/149.7 (2.7)	110/110.5/133.5 (1.72)
SE	90..98	124..125	90..92	109..112	132..133	———	145/145.9/178 (3.07)
IT	80..83	90..107	87..96	124..141	110..112	143..160	———

We have observed that in some cases, the measured average values in the two opposite directions are very different (see e.g., IT-FR, IT-UK, IT-SE, UK-NL). This is due to the fact that the call establishment time was measured as the difference between the round-trip-time (RTT) of the first and second 'ping' packet when there is no SVC built up a-priori, but the ARP cache is still populated (IP to ATM mapping is kept for max. 600 sec. Otherwise, when the ARP cache is empty, the address resolution time will be added to the call establishment time). We have to admit that these type of measurements cannot be so accurate as with a calibrated special test equipment used in our measurements. However, in more than 80% of the cases, the simulated results are very closed to the measured ones. There are a few deviations, due to the inaccurate "ping-type" measurement results, we think. For example, between nodes UK-AT and FR-SE the measured values are 10% over the simulated average values; between nodes NL-IT, NL-UK and UK-SE the measured values are 5-10% under the simulated ones. There was no greater deviation observed between the simulation and measured results.

Performance analyses of large ATM network topologies, using the call model presented in Section 5.1, will be presented in Chapter 6, based on simulation results. But before, I will try to simplify this above model to be able to solve it analytically. Let's see how good such an analytical model can approximate the measured results (see Section 5.2).

5.2 Simplified signalling flow models for analytical studies

In this section I have shown that the message flow model given in Section 5.1 can be simplified in a way that it allows analytical solutions. Due to the fact that the call establishment time T_C of a cascaded network will always slightly overestimate that of an arbitrary network configuration (as shown later in Chapter 6), I have offered analytical solutions for the cascaded network only.

Let us assume a cascaded network with *'r'* intermediate nodes, one calling party and one called party. The selected network model is a special case of the BCMP model (off-the-shelf technique, see [BCMP75]), where the node model consist of *one equivalent central server and one queue* with *FIFO* discipline irrespective of class. The messages represent the classes of the BCMP network, and there is a mandatory class change at the end users: *SETUP* (class 1) changes to *CONNECT* (class 2), *CONNECT* to *RELEASE* (class 3), *RELEASE* to *RELEASE COMPLETE* (class 4). The only external call pattern for class 1 (*SETUP*) calls at the first node is assumed to be state independent Poisson arrival with a mean rate of λ_N [calls/sec].

5.2.1 The M/M/1/K node model

I have shown that the call establishment time of a cascaded network with r-nodes can be approximated analytically by using an M/M/1/K queueing model for call processing in each node and an infinite service model (M/M/∞) for modelling the call duration.

Proof: We consider a BCMP network consisting of 4 classes of jobs. Our sample network consists of *'r'* intermediate nodes with finite queues (M/M/1/K) and one infinite server to model the mean holding time of open calls (see Figure 5.7). There are two terminals (TE1 and TE2) connected to the ends of the cascaded switches. Only four types of layer 3 messages are considered here: *SETUP*, *CONNECT*, *RELEASE*, *RELEASE COMPLETE*. Each of them is represented by different classes of jobs. Previous papers assumed all signalling messages to be of the same time duration [Baf93]. Based on the results of Section 4.1, here we assume different processing times for different messages. We focus only on the effect of message processing in the signalling network, therefore we assume that the total bandwidth requested by calls is always available.

E.g., the flow of jobs through the $r = 2$ nodes *M/M/1* queueing network is as follows:

1. Class 1 calls (*SETUP* messages) arrive to node 1 (switch) at a rate of $\lambda_N = \lambda \cdot p_{0,11}$ (calls/sec), where $p_{0,ij}$ is the probability of external arrival to node '*i*' as class '*j*' job.
2. The job is served in FIFO order at rate μ_{11} (μ_{ij} is the service rate at node *i* of class *j* jobs) and joins node 2 with probability $p_{11,21}$ ($p_{ir,js}$ is the probability that class *r* customer from node *i* joins node *j* as class *s* job). Then it is served in FIFO order with rate μ_{21} and joins node 3 with probability $p_{21,31}$.
3. After being served with rate μ_{31}, class 1 job changes to class 2 (*CONNECT* message) and joins node 2 again. Here a class 2 job is served with rate μ_{22} and joins node 1 with probability $p_{22,12}$.
4. Class 2 job is served with rate μ_{12} in node 1, then joins node 4, the infinite server (server-per-job strategy). After a mean holding time class 2 job is changed to class 3 (*RELEASE* message) and re-attends node 1 queue.

5. Class 3 job travels the same path as class 1, but it is served with rate μ_{13}, μ_{23}, μ_{33}, respectively. Then class 3 job changes to class 4 (*RELEASE COMPLETE* message) in node 3, joins node 2, then served by μ_{24} joins node 1. Finally, in node 1 being served with rate μ_{14}, class 4 leaves the system with probability $p_{14,0}$.

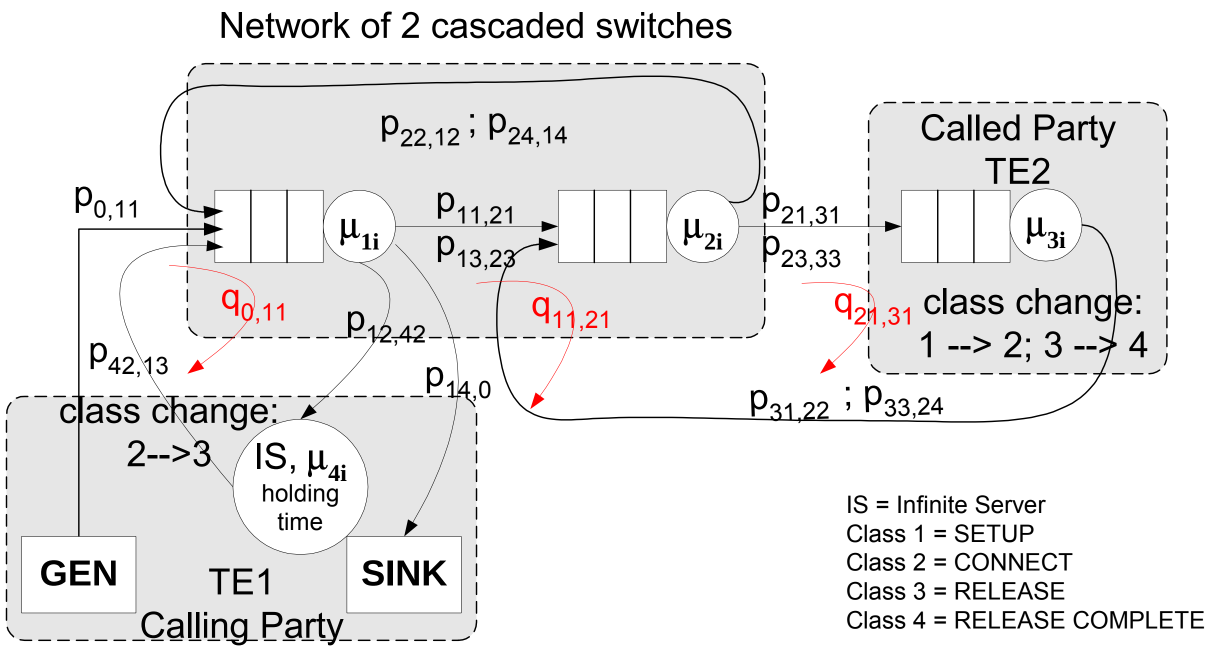


Figure 5.7 Queueing model for a cascaded network with two intermediate nodes

5.2.2 The BCMP method applied to a chain of 'r' nodes

5.2.2.1 M/M/1 queueing nodes

In general, in the '*r*'-node cascaded queueing network the load distribution can be examined by solving the traffic equations:

$$e_{js} = p_{0,js} + \sum_{i,q} e_{iq} \cdot p_{iq,js}\,, \quad j = 1,\dots,r+2\,\text{(nodes)} \quad s = 1,2,3,4\,\text{(classes)} \tag{15}$$

where: $p_{0,11} \neq 0$, $p_{0,js} = 0\ \forall j,s > 1$. The Poisson arrival assumption is a fairly valid one and is adopted in this analysis. The assumption of the exponential service time μ_j is grossly inadequate. Using general service-time statistics complicates matters too much, therefore it is not subject here. All we can say is that the exponential service-time assumption is usually a conservative one, so that we at least achieve the worst-case result.

The routing probability in each node satisfies the condition: $\sum_{j,s} p_{iq,js} = 1$. The matrix form of this system contains an **A** = $[s_{\max} \cdot (r+2)\ \text{x}\ s_{\max} \cdot (r+2)]$ routing matrix: $\underline{x} = \underline{\underline{A}} \cdot \underline{x} + \underline{b}$, where **A** is a sparse matrix.

The condition for an open queueing network $\sum_{j,s} p_{0,js} = \sum_{j,s} p_{js,0}$ in our case leads to a simple relation: $p_{0,11} = p_{14,0}$, where $p_{ij,0}$ is the probability that a class *j* job leaves the system after being served in node *i*.

By definition, a job state is a pair (*i,r*) associated with a job at node *i*, as class *r* job. The set of job states is split into one or more non-intersecting subsets (or "sub-chains"). We have only one sub-chain (***E***), because there is a non-zero probability that a job will be in all possible job states during its life in the network: $(i,r)\in E \ \forall i,r$.

According to [Gel80], if we have only one completely open sub-chain, the service time is independent of node states and the external call arrival is independent of the network state, then a very simple expression can be obtained for the distribution of the aggregate states.

$$\rho_i = \begin{cases} \sum_{s=1}^{4} \dfrac{\lambda_N \cdot e_{is}}{\mu_{is}}, & i = 1,\dots,r+1 \\ \dfrac{\lambda_i}{\mu_i}, & i = r+2 \end{cases} \tag{16}$$

Therefore, we have '*r*+1' nodes with geometrical distribution and one with Poisson distribution (for the call duration server).

$$p_i(k) = P(n_i = k) = \begin{cases} (1-\rho_i)\cdot \rho_i^k, & i = 1,\dots,r+1 \\ \dfrac{e^{-\rho_i}}{k!}\cdot \rho_i^k, & i = r+2 \end{cases} \tag{17}$$

where n_i is the total number of jobs of all classes in node *i*. The stability condition is $\rho_i < 1$.

5.2.2.2 M/M/1/K queueing nodes

Let us assume now an '*r'*-node cascaded *M/M/1/K* queueing network. As the nodes have finite queue lengths, there is a non-zero probability that some messages leaving a node will be blocked due to a full buffer in the next node. Let us note this blocking probability with $q_{iu,js}$ (see Figure 5.7).

The load distribution can be examined by solving the traffic equations:

$$e_{js}' = (p_{0,js} - q_{0,js}) + \sum_{i,u} e_{ir}' \cdot (p_{iu,js} - q_{iu,js}), \quad i,j = 1,\dots,r+2; \quad r,s = 1,2,3,4 \tag{15'}$$

where: $p_{0,11} \neq 0, \quad p_{0,js} = 0 \ \forall j,s > 1$.

Let us have a closer look at the load distributions of the first three nodes for class 1 messages:

$$e_{11}' = p_{0,11} - q_{0,11}$$
$$e_{21}' = (p_{0,11} - q_{0,11})\cdot(p_{11,21} - q_{11,21}) = p_{0,11}\cdot p_{11,21} - p_{0,11}\cdot q_{11,21} - q_{0,11}\cdot p_{11,21} + \cancel{q_{0,11}\cdot q_{11,21}}$$
$$e_{31}' = (p_{0,11} - q_{0,11})\cdot(p_{11,21} - q_{11,21})\cdot(p_{21,31} - q_{21,31}) =$$
$$= p_{0,11}\cdot p_{11,21}\cdot p_{21,31} - p_{0,11}\cdot p_{11,21}\cdot q_{21,31} - p_{0,11}\cdot q_{11,21}\cdot p_{21,31} - q_{0,11}\cdot p_{11,21}\cdot p_{21,31} +$$
$$+ \cancel{p_{0,11}\cdot q_{11,21}}\cdot q_{21,31} + \cancel{q_{0,11}\cdot p_{11,21}}\cdot q_{21,31} + \cancel{q_{0,11}\cdot q_{11,21}}\cdot p_{21,31} - \cancel{q_{0,11}\cdot q_{11,21}}\cdot q_{21,31}$$
$$\dots$$

Suppose that: $p_{iu,js} > 10\cdot q_{iu,js}$, $p_{iu,js} + q_{iu,js} = 1$. Then, in the equations above all the terms containing a product of at least two '*q*'-s, i.e., $(q_{iu,js}\cdot q_{kv,lw})$ can be neglected. Furthermore, note that: $q_{\max} = \max_{i,j,u,s}\{q_{0,11}, q_{iu,js}\}$. Then these equations can be rewritten under the following form:

$$e_{11}' \geq p_{0,11} - q_{\max}$$
$$e_{21}' \geq p_{0,11} \cdot p_{11,21} - (p_{0,11} + p_{11,21}) \cdot q_{\max}$$
$$e_{31}' \geq p_{0,11} \cdot p_{11,21} \cdot p_{21,31} - (\underbrace{p_{0,11} \cdot p_{11,21}}_{\leq p_{0,11}} + \underbrace{p_{0,11} \cdot p_{21,31}}_{\leq p_{21,31}} + \underbrace{p_{11,21} \cdot p_{21,31}}_{\leq p_{11,21}}) \cdot q_{\max} \geq$$
$$\geq p_{0,11} \cdot p_{11,21} \cdot p_{21,31} - (p_{0,11} + p_{11,21} + p_{21,31}) \cdot q_{\max}$$

...

Generalising, for node 'r+1' and class 1 we have (using a notation $p_{0,11} = p_{01,11}$):

$$e_{(r+1)1}' \geq \prod_{i=0}^{r} p_{i1,(i+1)1} - q_{\max} \cdot \sum_{i=0}^{r} p_{i1,(i+1)1}$$

At this node (called party), there is a mandatory change from class 1 to class 2 messages, followed by a reverse propagation of class 2 messages along the same path.

Thus (using a notation $p_{(r+1)1,r2} = p_{(r+1)2,r2}$) we have:

$$e_{r2}' \geq p_{(r+1)2,r2} \cdot \prod_{i=0}^{r} p_{i1,(i+1)1} - q_{\max} \cdot (p_{(r+1)2,r2} + \sum_{i=0}^{r} p_{i1,(i+1)1})$$

$$e_{(r-1)2}' \geq p_{r2,(r-1)2} \cdot p_{(r+1)2,r2} \cdot \prod_{i=0}^{r} p_{i1,(i+1)1} - q_{\max} \cdot (p_{r2,(r-1)2} + p_{(r+1)2,r2} + \sum_{i=0}^{r} p_{i1,(i+1)1})$$

...

$$e_{12}' \geq \prod_{i=1}^{r} p_{(i+1)2,i2} \cdot \prod_{i=0}^{r} p_{i1,(i+1)1} - q_{\max} \cdot (\sum_{i=1}^{r} p_{(i+1)2,i2} + \sum_{i=0}^{r} p_{i1,(i+1)1})$$

$$e_{(r+2)2}' \geq \prod_{i=0}^{r} p_{(i+1)2,i2} \cdot \prod_{i=0}^{r} p_{i1,(i+1)1} - q_{\max} \cdot (\sum_{i=0}^{r} p_{(i+1)2,i2} + \sum_{i=0}^{r} p_{i1,(i+1)1})$$

Continuing these calculations, including the changes from class 2 to class 3 at the calling party, then later from class 3 to class 4 messages at the destination, finally we arrive to the term e_{14}':

$$e_{14}' \geq \prod_{i=0}^{r} p_{(i+1)4,i4} \cdot \prod_{i=0}^{r} p_{i3,(i+1)3} \cdot \prod_{i=0}^{r} p_{(i+1)2,i2} \cdot \prod_{i=0}^{r} p_{i1,(i+1)1} -$$
$$- q_{\max} \cdot (\sum_{i=0}^{r} p_{(i+1)4,i4} + \sum_{i=0}^{r} p_{i3,(i+1)3} + \sum_{i=0}^{r} p_{(i+1)2,i2} + \sum_{i=0}^{r} p_{i1,(i+1)1})$$

which can be 'compressed' as follows:

$$e_{14}' \geq \prod_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} \cdot \prod_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is} - q_{\max} \cdot (\sum_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} + \sum_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is})$$

In this expression we need to find the conditions for $q_{\max}$ so that the second term can be neglected:

$$q_{\max} \cdot (\sum_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} + \sum_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is}) \leq 10^{-2} \cdot \prod_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} \cdot \prod_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is}$$

Thus,

$$q_{\max} \le \frac{10^{-2} \cdot \prod\limits_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} \cdot \prod\limits_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is}}{\sum\limits_{\substack{i=0 \\ s=1,3}}^{r} p_{is,(i+1)s} + \sum\limits_{\substack{i=0 \\ s=2,4}}^{r} p_{(i+1)s,is}}$$

If we find an upper bound for the denominator, and a lower bound for the numerator, then we arrive to even more severe conditions:

$$q_{\max} \le \frac{10^{-2} \cdot (p_{\min})^{4\cdot(r+1)}}{4\cdot(r+1)}$$

where, $p_{\min} = \min\{p_{iu,js} > 0,\ \forall i,j = \overline{0,r};\ \forall u,s = \overline{1,4}\}$.

Case studies:

1) $p_{\min} = 0.9$, $r = 8$ $\qquad q_{\max} \le \dfrac{10^{-2} \cdot (0.9)^{36}}{36} = 6 \cdot 10^{-6}$

2) $p_{\min} = 0.99$, $r = 8$ $\qquad q_{\max} \le \dfrac{10^{-2} \cdot (0.99)^{36}}{36} = 2 \cdot 10^{-4}$

3) $p_{\min} = 0.999$, $r = 8$ $\qquad q_{\max} \le \dfrac{10^{-2} \cdot (0.999)^{36}}{36} = 2.7 \cdot 10^{-4}$

Case 2 is a realistic scenario for signalling networks under study for a light overload $(p_{\min} = 0.99)$. If we admit the condition of case 2, then we can neglect the term containing $q_{\max}$, thus arriving to the same formula for equation (16) as with M/M/1 queues.

Equation (17) instead will suffer the following changes:

$$p_i(k) = P(n_i = k) = \begin{cases} \dfrac{(1-\rho_i)\cdot \rho_i^k}{1-\rho_i^{K+1}}, & i = 1,\dots,r+1;\ k \le K \\ 0, & i = 1,\dots,r+1;\ k > K \\ \dfrac{e^{-\rho_i}}{k!}\cdot \rho_i^k, & i = r+2 \end{cases} \tag{17'}$$

where n_i is the total number of jobs of all classes in node *i*. The stability condition is $\rho_i < 1$.

As shown in Section 4.1.5, a finite call duration does not have any influence on the call establishment time (T_C), therefore we can neglect the effect of the node (i=r+2). The product form solution is obtained only if we have $\mu_{is} = \mu_i \quad \forall \mathrm{s} = 1,2,3,4$ (i.e., Jackson's theorem, see equation (16)), a condition which is not satisfied here. Let's assume an empirical relationship between the service rates of different classes at one node, according to Section 4.1.2:

$$\frac{1}{\mu_{i1}} = 3\cdot\frac{1}{\mu_{i2}} = 3\cdot\frac{1}{\mu_{i3}} = 12\cdot\frac{1}{\mu_{i4}}, \quad \mathrm{i} = 1,\dots,r,$$

and denote $\frac{1}{\mu_{i1}} = \alpha_i(S)$, $\frac{1}{\mu_{i2}} = \alpha_i(C)$ which is equivalent to the minimum *SETUP delay* and minimum *CONNECT delay*, respectively (see Section 5.1.2).

Moreover, because $\frac{1}{\mu_i^C} = \alpha_i(S) + \alpha_i(C) = \frac{4}{3} \cdot \alpha_i(S)$, we can express ρ_i in terms of μ_i^C, where $1/\mu_i^C$ is the *call establishment latency* in node *i*, thus we can get back to the product-form solution.

The expected number of jobs of all classes appearing in queue *i* can be derived from the definition of the mean value of random variables:

$$E(n_i) = \overline{N}_i = \sum_{k=0}^{K} k \cdot p_i(k), \quad i = 1,...,r+1$$

$$\overline{N}_i = \sum_{k=0}^{K} k \cdot \frac{1-\rho_i}{1-\rho_i^{K+1}} \cdot \rho_i^k = \frac{1-\rho_i}{1-\rho_i^{K+1}} \cdot \sum_{k=0}^{K} k \cdot \rho_i^k$$

where:

$$\sum_{k=0}^{K} k \cdot \rho_i^k = \sum_{k=1}^{K} k \cdot \rho_i^k = \rho_i \cdot \sum_{k=1}^{K} k \cdot \rho_i^{k-1} = \rho_i \cdot \sum_{k=1}^{K} (\rho_i^k)^{/} = \rho_i \cdot (\sum_{k=1}^{K} \rho_i^k)^{/} = \rho_i \cdot (\sum_{k=0}^{K} \rho_i^k)^{/} = \rho_i \cdot \left(\frac{1-\rho_i^{K+1}}{1-\rho_i} \right)^{/}$$

Then applying the Little's formula we have:

$$E(T_i) = \frac{\overline{N}_i}{\lambda_i} = \frac{1}{\lambda_i} \cdot \left[\frac{\rho_i}{1-\rho_i} - \frac{(K+1) \cdot \rho_i^{K+1}}{1-\rho_i^{K+1}} \right], \quad i = 1,...,r+1 \tag{18}$$

The stability condition is $\rho_i < 1$, and note that: $\lambda_i = \begin{cases} 4 \cdot \lambda_N, & \text{for } i = 1,2,...,r \\ 2 \cdot \lambda_N & \text{for } i = r+1 \end{cases}$.

As the load on the system increases, the throughput increases as well (desirable characteristic). An increase in the mean queue occupancy translates itself into increased time delay in the queue *i* (undesirable characteristic) and into increased call establishment time, since:

$$E(T_C) = \sum_{(i,s) \in E} E(T_i), \quad i = 1,2,...,r+1 \text{ (nodes)}; \quad s = 1,2 \text{ (classes)}$$

Therefore there is a typical trade-off in performance. Usually, we can neglect the transmission delay between nodes, according to Section 4.1.1, so we sum up only the processing delays. From this formula we can obtain the call establishment time (and similarly the release latency) for the cascaded network of *r*-nodes (let us consider the *ideal case,* when all call attempts are successful and suppose that all nodes are identical):

$$\begin{aligned} E(T_C)\big|_{'r'\text{switches}} &= \sum_{s=1,2} \sum_{i=1}^{r+1} E(T_i) = 2 \cdot r \cdot E(T_r) + E(T_{r+1}), \\ E(T_{RN})\big|_{r'\text{switches}} &= \frac{1}{2} \cdot \sum_{s=3,4} \sum_{i=1}^{r+1} E(T_i) = r \cdot E(T_r) + \frac{1}{2} \cdot E(T_{r+1}), \end{aligned} \tag{19}$$

where T_r is the time a messages spends in one switch, $E(T_i)=E(T_j)=E(T_r)$ $\forall i,j=1,\ldots,r$, while T_{r+1} denotes the response time of a terminal equipment ($i=r+1$).

5.2.3 Case study: Analysing a chain of 2 cascaded nodes

Let us consider the queueing network in Figure 5.7, where the terminal TE1 is a generator of call attempts and release of active calls are also initiated by TE1, while the terminal TE2 automatically accepts all calls. The network consists of $r = 2$ intermediate nodes. Equation (15) then results in a system of 16 equations (4 nodes x 4 classes). The matrix form of this system contains a (16x16) routing matrix:

$$\underline{x}_{16} = \underline{\underline{A}}_{16\times16} \cdot \underline{x}_{16} + \underline{b}_{16}$$

A is a sparse matrix. There is at most one non-zero element in a row or column. It can be reduced to a matrix of size (11x11), because of the 5 zero rows and columns in the matrix (***A*** is a singular matrix). The 'reduced' system has one unique solution, because there is only one completely open sub-chain. The solution can be obtained very simply by the Gaussian elimination method.

The number of terms in the product-form solution of e_{ir} is equal to the number of job states a job visits before arriving to that particular state (*i,r*). Particularly, the terms $e_{32} = 0$ and $e_{34} = 0$, because *CONNECT* and *RELEASE COMPLETE* messages are never received by the called party. Only the *CONNECT* message arrives to the call duration server (IS), therefore only the term e_{42} is non-zero out of $e_{4s}, s = 1,2,3,4$. The rest of the terms e_{ir} is given below:

$$\begin{cases}
e_{11} = p_{0,11} \\
e_{12} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \\
e_{13} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \cdot p_{42,13} \\
e_{14} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \cdot p_{42,13} \cdot p_{13,23} \cdot p_{23,33} \cdot p_{33,24} \cdot p_{24,14} \\
e_{21} = p_{0,11} \cdot p_{11,21} \\
e_{22} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \\
e_{23} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \cdot p_{42,13} \cdot p_{13,23} \\
e_{24} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \cdot p_{42,13} \cdot p_{13,23} \cdot p_{23,33} \cdot p_{33,24} \\
e_{31} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \\
e_{32} = 0 \\
e_{33} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \cdot p_{42,13} \cdot p_{13,23} \cdot p_{23,33} \\
e_{34} = 0 \\
e_{41} = 0 \\
e_{42} = p_{0,11} \cdot p_{11,21} \cdot p_{21,31} \cdot p_{31,22} \cdot p_{22,12} \cdot p_{12,42} \\
e_{43} = 0 \\
e_{44} = 0
\end{cases}$$

The overall traffic intensity for node *j* is a sum of per-class traffic intensities (see equation (16)), where nodes 1, 2, 3 are *type 1* nodes, while node 4 is a *type 3* node, according to [Gel80]. More precisely,

$$\begin{cases} \rho_1 = \sum_{s=1}^{4} \frac{\lambda_{1s}}{\mu_{1s}} = \frac{\lambda_N \cdot e_{11}}{\mu_{11}} + \frac{\lambda_N \cdot e_{12}}{\mu_{12}} + \frac{\lambda_N \cdot e_{13}}{\mu_{13}} + \frac{\lambda_N \cdot e_{14}}{\mu_{14}} \\ \rho_2 = \sum_{s=1}^{4} \frac{\lambda_{2s}}{\mu_{2s}} = \frac{\lambda_N \cdot e_{21}}{\mu_{21}} + \frac{\lambda_N \cdot e_{22}}{\mu_{22}} + \frac{\lambda_N \cdot e_{23}}{\mu_{23}} + \frac{\lambda_N \cdot e_{24}}{\mu_{24}} \\ \rho_3 = \sum_{s=1}^{4} \frac{\lambda_{3s}}{\mu_{3s}} = \frac{\lambda_N \cdot e_{31}}{\mu_{31}} + \frac{\lambda_N \cdot e_{33}}{\mu_{33}} \\ \rho_4 = \sum_{s=1}^{4} \frac{\lambda_{4s}}{\mu_{4s}} = \frac{\lambda_N \cdot e_{42}}{\mu_{42}} \end{cases}$$

As we can see, node 1 and node 2 (network nodes) are visited by all 4 classes of jobs, node 3 (the called party) receives only two classes of jobs, while the call duration server (at the source) handles only class 2 jobs, and reinserts them later as class 3 jobs in node1.

Let us consider the *ideal case* when all the call attempts are successful (this was the assignment we made in Section 4.1), i.e., $p_{0,11} = 1,\ p_{ir,js} = 1$ for all the non-zero elements in the ***A*** matrix (11 terms), which results in:

$$e_{js} = \begin{cases} 0 & \text{for job states (3,2), (3,4), (4,1), (4,3), (4,4)} \\ 1 & \text{for the rest of the 11 job states} \end{cases}$$

Accordingly,

$$\lambda_{js} = \lambda_N \cdot e_{js} = \begin{cases} 0 & \text{for job states (3,2), (3,4), (4,1), (4,3), (4,4)} \\ \lambda_N & \text{for the rest of 11 job states} \end{cases},$$

and,

$$\lambda_i = \begin{cases} 4 \cdot \lambda_N, & \text{for } i = 1,2 \\ 2 \cdot \lambda_N, & \text{for } i = 3 \\ \lambda_N, & \text{for } i = 4 \end{cases}.$$

The service rate for handling different types of messages μ_{js} is determined taking into account the message delay measurement results from Section 4.1.2:

$$\frac{1}{\mu_{i1}} = 3 \cdot \frac{1}{\mu_{i2}} = 3 \cdot \frac{1}{\mu_{i3}} = 12 \cdot \frac{1}{\mu_{i4}}, \quad \mathrm{i} = 1,2,3$$

Thus we can express the traffic intensities in terms of the call establishment latency of each node:

$$\begin{cases} \rho_1 = \frac{21}{16} \cdot \frac{\lambda_N}{\mu_1^C} \\ \rho_2 = \frac{21}{16} \cdot \frac{\lambda_N}{\mu_2^C} \\ \rho_3 = \frac{\lambda_N}{\mu_3^C} \\ \rho_4 = \frac{\lambda_N}{\mu_4} \end{cases}$$

One can observe that a call attempt generates thus approx. $1.5 \times \frac{\lambda_N}{\mu_i^C}$ traffic at one network node.

Finally, the equation (18) will be split in the following two sub-cases (for identical cascaded switches):

$$E(T_i) = \frac{1}{4 \cdot \lambda_N} \cdot \left[\frac{\rho_i}{1-\rho_i} - \frac{(K+1) \cdot \rho_i^{K+1}}{1-\rho_i^{K+1}} \right], \quad i = 1,2$$

$$E(T_3) = \frac{1}{2 \cdot \lambda_N} \cdot \left[\frac{\rho_3}{1-\rho_3} - \frac{(K+1) \cdot \rho_3^{K+1}}{1-\rho_3^{K+1}} \right], \quad i = 3$$

The maximum buffer size '*K*' has to be determined such that the formula in equation (18) will best approximate the measured results.

5.2.4 Numerical results

In Figure 5.8 the *M/M/1* and *M/M/1/K* models for different *K* values are compared to the measured results of the *GDC APEX DV2* and *Fore ASX200BX* switches, respectively taken from the Figure 5.5.

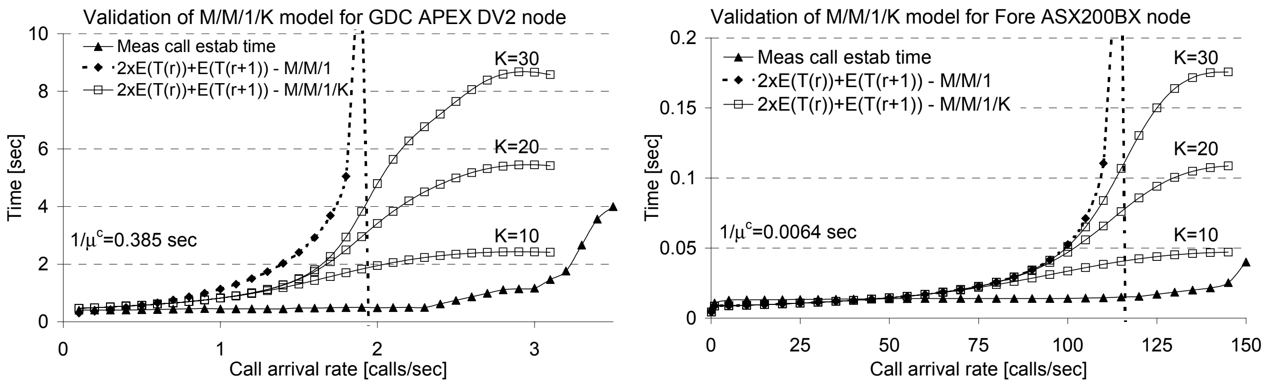


Figure 5.8 Validation of the M/M/1/K model against measured results

An *M/M/1 infinite buffer* solution is a particular case of the equation (18), i.e. when $K \to \infty$ then the second term converges to zero. It gives a fair approximation of the call establishment time only if the load $\lambda_N/\mu^C < 0.65$. For higher load region ($\lambda_N/\mu^C > 0.65$) the *M/M/1* model is not appropriate. Similarly, the *M/M/1/K* model over-estimates the measured call establishment time, but for small *K* values (e.g., *K* < 20) could be used as a conservative approach.

As a conclusion, we can state that the application of the BCMP queueing network provides a flexible and effective framework for modelling and analysing the call establishment and release procedures, however only for loads $\lambda_N/\mu^C < 0.65$. According to the BCMP theorem, the distribution of each type of messages in progress and the associated mean processing delay at arbitrary node was derived using a product-form solution. Finally, the expected call establishment time using an *M/M/1/K* node model was compared to the measured one in two cases. This *M/M/1/K* model may be useful for signalling network design in the first place in order to compute a rough estimate, however it is inaccurate when estimating the call release latency. In Section 5.1 we have already shown, that the simulation results (based on a more complex model, of course) are more accurate.

5.2.5 A new approximation formula

If there are many different types of switches in the network, or more complex calls are generated and more than one PNNI levels are defined, then the equation (19) is difficult to be applied. For this case I have constructed another approximation formula that can be used for the network design (if the T_C and T_{RN} of one node is a-priori known) to estimate the call establishment latency and release latency of cascaded networks:

As the *M/M/1/K* model did not perform well above $\lambda_N/\mu^C > 0.65$, therefore I have been looking for another function with the following characteristics:

-relatively flat, for $\lambda_N/\mu^C < 0.9$;

-sharp increase for $0.9 \le \lambda_N/\mu^C \le 1$;

-slowly decreasing, for $\lambda_N/\mu^C > 1$ (i.e., in the overload region),

which contains also dependency on complexity of call profiles, number of switches and PNNI hierarchy. I did not found any simple algebraic function that satisfies all the above conditions. A more complex function is described below:

I have shown that the inequality in equation (10) regarding the call establishment latency of calls that go through 'r' cascaded switches can be approximated by the following formula:

$$E(T_{CN})\big|_{'r'} \approx \begin{cases} (1+s)\cdot\dfrac{m+1}{2}\cdot[1-0.1\cdot\log(r)]\cdot\exp\left(\dfrac{\lambda_N}{r\cdot\mu_0^C}\right)\cdot\sum\limits_{i=1}^{P}(n_i\cdot\overline{T}_{CN}^{\,\mathrm{type}\,i}), \quad r=\sum\limits_{i=1}^{P}n_i\,,\ \mu_0^C=\min\limits_{i=1..P}(\mu_i^C),\ \lambda_N<0.9\cdot\mu_0^C \\ (1+s)\cdot\dfrac{m+1}{2}\cdot[1-0.1\cdot\log(r)]\cdot\dfrac{C\cdot\left(\dfrac{\lambda_N}{\mu_0^C}\right)^2}{\sqrt{\exp\dfrac{\lambda_N}{\mu_0^C}}}\cdot\sum\limits_{i=1}^{P}(n_i\cdot\overline{T}_{CN}^{\,\mathrm{type}\,i}), \quad \lambda_N\ge 0.9\cdot\mu_0^C \end{cases} \tag{20}$$

Similarly, for the estimation of the T_{RN}, I have found the following formula:

$$E(T_{RN})\big|_{'r'} \approx \begin{cases} [1-0.1\cdot\log(r)]\cdot\exp\left(\dfrac{\lambda_N}{r\cdot\mu_0^R}\right)\cdot\sum\limits_{i=1}^{P}(n_i\cdot\overline{T}_{RN}^{\,\mathrm{type}\,i}), \quad r=\sum\limits_{i=1}^{P}n_i\,,\ \mu_0^R=\min\limits_{i=1..P}(\mu_i^R),\ \lambda_N<0.9\cdot\mu_0^R \\ [1-0.1\cdot\log(r)]\cdot\dfrac{C\cdot\left(\dfrac{\lambda_N}{\mu_0^R}\right)^2}{\sqrt{\exp\dfrac{\lambda_N}{\mu_0^R}}}\cdot\sum\limits_{i=1}^{P}(n_i\cdot\overline{T}_{RN}^{\,\mathrm{type}\,i}), \quad \lambda_N\ge 0.9\cdot\mu_0^R \end{cases} \tag{21}$$

where, the term containing *log(r)* expresses the message overlapping,

m represents the PNNI peer-group hierarchical level,

s represents the signalling overhead, $s = 0$ for the default *SETUP* message, $0 \le s < 1$,

P is the number of different types of ATM switches in the network, and

n_i is the number of switches of one type,

(μ_i^C, μ_i^R) are the average values of the equivalent service rates of the switches for the call establishment and release, where $\mu_i^C < \mu_i^R$, and

λ_N is the call arrival rate.

Finally, the term $C = \exp\left(\dfrac{0.9}{r}\right)\cdot\dfrac{\sqrt{\exp(0.9)}}{(0.9)^2}$ is the normalising factor.

It can be observed that the T_{RN} is independent of the call complexity and of the PNNI hierarchy, respectively.

5.2.6 Numerical results

A comparison of the three models is shown in Figure 5.7 for two switches. The three curves represent the estimated call establishment latency for a cascaded network of r=10 identical switches (where s=0, m=1, P=1, $n_i = r$).

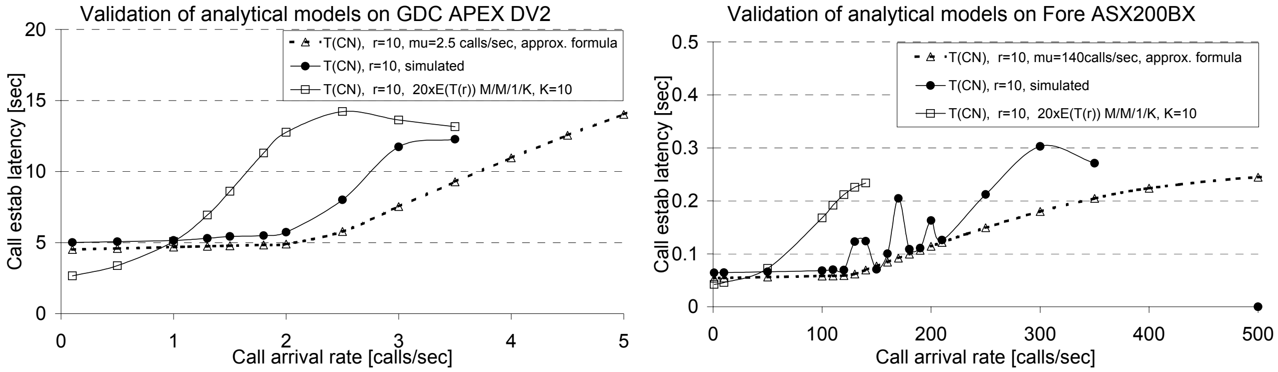


Figure 5.7 Comparison of the M/M/1/K model and of the formula in equation (20) against simulated results of 10 cascaded switches

It can be seen that the *M/M/1/K* model (K=10) provides an underestimation for $\lambda_N/\mu^C < 0.5$, then overestimates the results obtained either by simulation or with the formula in equation (20) for $\lambda_N/\mu^C > 0.5$. The approximation formula of equation (20) gives very close results to those of simulated ones. It delivers a slight under-estimation in the overload region compared to the simulated results.

As another validation, let us now compare the numerical values derived by the equation (20) for 1 call/sec against the measured results of 2, 3 and 4 cascaded switches presented in Section 4.1.7 (Table 4.8 and Table 4.9). The equivalent service rates and call establishment latencies for individual switches are obtained from Figure 5.5*a-d*. As shown in Table 5.3, the analytical results are very closed to the measured results, except two cases (see measured values marked with [Mau01]) for mixed type of switches. The error here is due to fact that we obtained a different value for call establishment latency of the MSX36170 switch (i.e., 24msec, see Figure 5.5) than it was obtained by [Mau01] (i.e, 33msec, see Table 4.9).

Table 5.3 Analytical versus measured results of cascaded switches at 1 call/sec rate (average)

Cascade	Type	*Call establishment latency* [msec] ANALYTICAL, eq.(20)	*Call establishment latency* [msec] MEASURED
2 switches	2x XP140	69.1	68
2 switches	2x ASX200BX	14.6	14.6
2 switches	XP140 + MSX36170	58.2	71 [Mau01]
3 switches	3x XP140	101.2	101
3 switches	3x ASX200BX	21.5	21
3 switches	XP140 + MSX36170 + AXD311	79.9	85 [Mau01]
4 switches	4x XP140	132.7	133
4 switches	4x ASX200BX	28.2	28

Both formulas of Section 5.2 have been validated against measured results of $r \leq 4$ cascaded switches for 1 call/sec, and versus simulated results of r=10 switches (for many different call arrival rates, at overload as well). The model in Section 5.2.1 can be used as a rule of thumb, for the first estimations in network planning. The disadvantage of this method is that it gives the average value estimation only. It has lost its ATM specific characteristics due to the single FIFO queueing model. Moreover, the model cannot handle priority, because then the product form solution will disappear. The sparse matrix is already of size [16x16] for 2 cascaded switches, with one calling and one called party, respectively. Each additional node adds 4 rows and 4 columns to the matrix, while each generator adds 8 of them. The *M/M/1/K* finite buffer model gives an overestimation of the call establishment time of cascaded switches for $\lambda_N/\mu^C > 0.5$, moreover this model is also inaccurate when estimating the call release latency (it gives $T_C \approx 2T_{RN}$). In Section 5.2.4, I have found that the proposed approximation formula, i.e. equation (20) has acceptable accuracy in practice. Except the impact of the PNNI hierarchy, which has been only checked for m=1 and m=2, the other parameters have been validated against different types of cascaded switches (by measurement and simulation).

5.3 Conclusions

Section 5.1 presented the construction of a generic ATM call processing model for UNI and PNNI networks, giving details of its architecture and an algorithm how to obtain the system parameters. However, the proposed generic model in Section 5.1 was too complex to be solved analytically. Therefore, I have developed a simplified call model as well. Of course, this simplification had its drawbacks, as it was shown in Section 5.2. To solve it *analytically,* I have used the results of the queueing networks theory, i.e. a special case of the BCMP networks [BCMP75]. The analytical studies in Section 5.2 were complemented by an approximation formula for calculating the call establishment time and release latency in cascaded networks.

CHAPTER 6

6 Performance analysis of signalling in large ATM Networks

The model defined in Section 5.1 is quite complicated, it is not easily tractable analytically, especially when using priority mechanism or when studying a large network, therefore in the followings I have used *simulation* to estimate the call establishment times, release latencies and signalling load/switch in real size networks. The message latencies introduced by the signalling nodes in an ATM network are additive, as shown in Section 4.1.7. This property of the measurement implies that as more nodes are traversed on a signalling message path, the signalling message latency increases. The latency measured across a small number of nodes can be used *to predict* the performance of a larger network of similar nodes. Thus one can obtain the upper bound (diameter, Φ) of the network in order not to exceed the maximum latencies set by the ITU-T recommendations [ITU97], which is of 4.5 seconds as shown in Figure 4.14*a*.

Def 6.1 I have defined the network diameter as the shortest path between the two furthest nodes of the network:

$$\Phi = \max\{L(i,j) \mid L(i,j) < L'(i,j), \forall i,j \in \{1,...,N\}\}, \tag{22}$$

where *L(i,j)* is the path length between nodes *i* and *j*.

6.1 Investigated network topologies

I have studied the call establishment times and release latencies for 2, 3, …, 10 cascaded switches versus different signalling load, respectively. A short chain of nodes has been validated by measurements (2, 3 and 4 switches, respectively). Furthermore, I have estimated these parameters for *N*=4-node *fully meshed*, *N*=7-node (having a 4-node backbone *ring,* see Figure 5.6), *N*=30-node and *N*=35-node sample ATM networks (see Figure 6.1) in a *typical xDSL topology* with 2-to-5 DSLAMs connected to each access node, having 200 subscribers. This later topology contains all known basic topologies (e.g., access multiplexers, redundant links, backbone ring, etc.). These sample networks contain all source-destination path lengths from 2 to 8 nodes (hops), so I have compared the call establishment times T_{CN}^{L} and release latencies T_{RN}^{L} (for all path lengths, *L(i,j)*=1,…,8) to those obtained for cascaded switches.

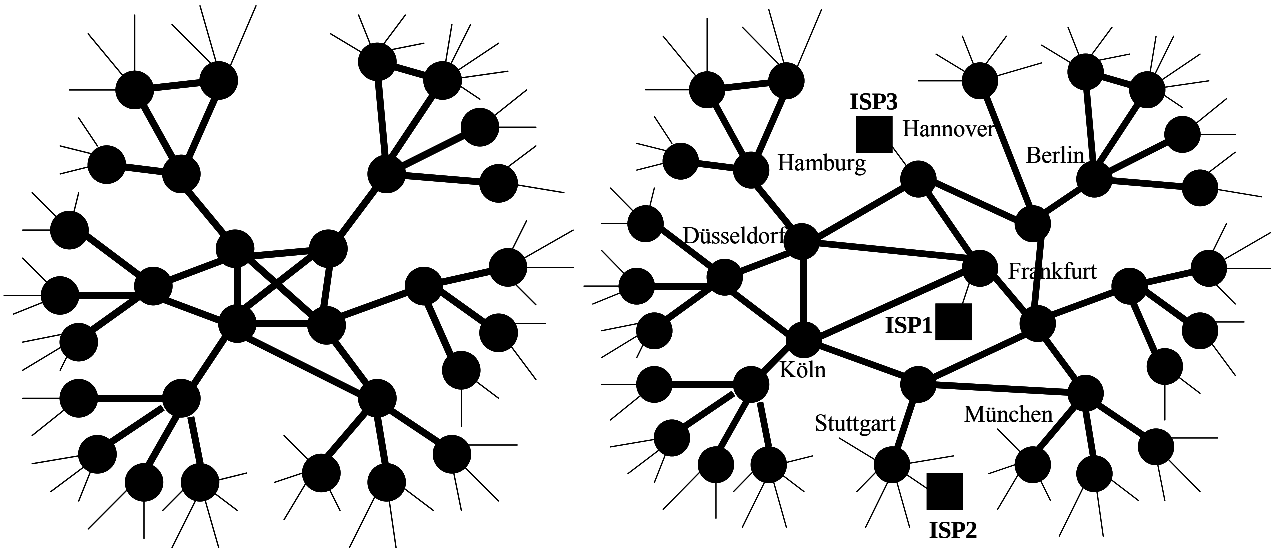


Figure 6.1 Sample networks: a) 30-nodes with Φ=6; b) 35-node network of the COLT Telecom Germany with Φ=8

6.2 Benefits of analysing a cascaded network vs. arbitrary network topology

I have shown that an r-node cascaded network can be used to estimate the signalling parameters (T_{CN}, T_{RN}, γ_R) of a large, N-node network (N > r) having a diameter $\Phi = r$ nodes.

First, I have investigated a *homogeneous network* (all network nodes are identical). I have shown that the minimum values of T_{CN}^{L} and T_{RN}^{L} for each path-length (L) are equal in both network topologies. The average and the maximum values of T_{CN}^{L} and T_{RN}^{L} for the cascaded topology slightly overestimate those of the network's (see Figure 6.2), i.e. with 1-5% the average values and with 15-20% the maximum values respectively, for a range of network load of $0 < \lambda_N < \lambda_r^{\max}$, where $\lambda_r^{\max}$ is the maximum call arrival rate without any call rejection in a cascaded network, r is the number of cascaded nodes (e.g., $\lambda_r^{\max} = 140$ calls/sec for a *FORE ASX200BX*).

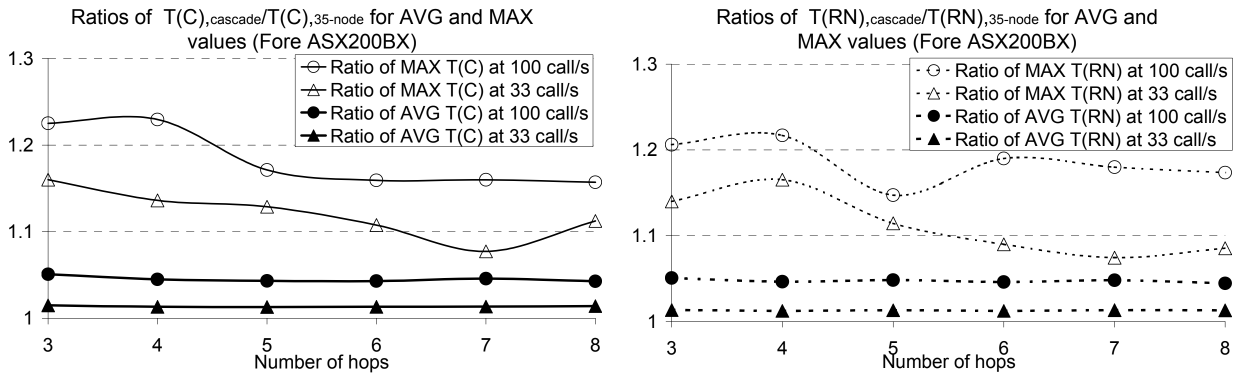


Figure 6.2 Ratio of the call establishment times (and release latencies) in cascaded vs. 35-node network with Φ=8

There is one exception for 2 hops (not shown in Figure 6.2), where the ratio for the maximum values is even higher (i.e., approx. 30%), but the ratio of the average values remains still in the range of 1-5%. This is due to the fact that we have a small number of calls with 2 hops in the 35-node network, which is approx. 2% of the maximum load (see Figure 6.3), while the cascaded network of 2 nodes is tested under the maximum load (33 and 100 calls/sec, respectively). Therefore we consider the results being in the above mentioned range starting at 3 hops.

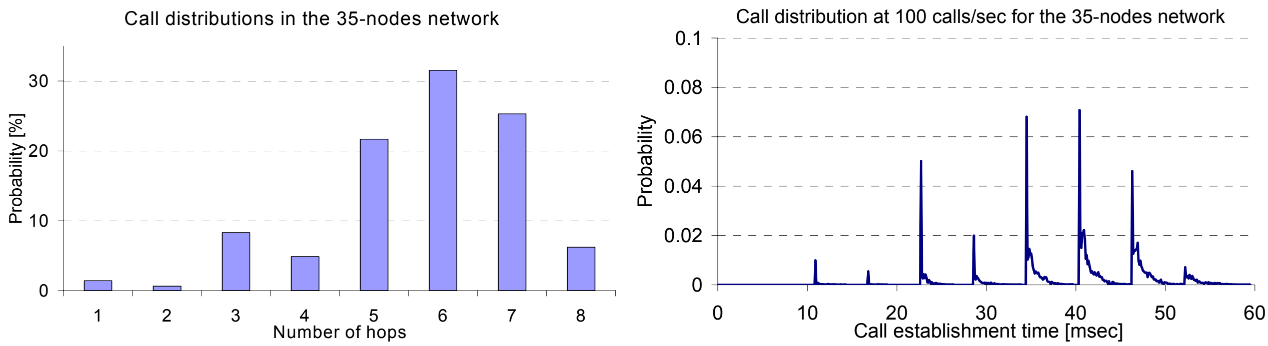


Figure 6.3 Call distribution vs. number of hops and call establishment time in the 35-node network, respectively

In Figure 6.3*a* we have shown the distribution of the calls in this 35-node network. It can be seen e.g., that 30% of the calls go through 6 network nodes, 25% travel along 7 nodes, while 20% visit 5 hops. Less than 10% of the calls are traversing 3, 4 and 8 nodes, respectively. Furthermore, the call distribution versus call establishment time (at 100 calls/sec arrival rate) is shown in Figure 6.3*b*, in the case of *FORE ASX200BX* switches. It can be observed, that at this call arrival rate the time a call is waiting in the queue at a node is less than the delay introduced by an additional node. For higher rates than 100 calls/sec arrival rate we are getting closer to $\lambda_r^{\max} = 130$ calls/sec for cascaded *FORE ASX200BX* switches, however in this 35-node network maximum 30% of the injected network load is distributed to the switches which are situated along the network path with 6 hops. Therefore, it is an interesting question, what the maximum network-level call arrival rate for our 35-node network is?

6.3 Investigating $\lambda_N^{\max}$ in a homogeneous network

It is quite obvious that the maximum network-level call arrival rate $\lambda_N^{\max}$ depends on the network topology as follows:

$$\lambda_1^{\max} \approx \lambda_r^{\max} \le \lambda_N^{\max} \qquad \text{or}$$

$$\max\{\lambda \mid r \ge 1, \text{cascade}, \ \gamma_R = 1\} \le \max\{\lambda \mid \Phi(\text{N - node}) = r > 1, \ \gamma_R = 1\}$$

The term $\lambda_1^{\max}$ is the maximum call arrival rate measured at one isolated switch without having any rejected call (γ_R=1). As usually, we make sure that the user plane has always sufficient bandwidth.

So, in cascaded networks, where users are connected only to the ends, the maximum call arrival rate is limited to that of an isolated switch's. In real-size networks the loads are usually distributed, thus the $\lambda_N^{\max}$ will be higher than $\lambda_1^{\max}$. I will try to approximate $\lambda_N^{\max}$ below. To be able to do that I will introduce first a couple of definitions.

The average call path is defined by the following equation: $\overline{L} = \sum_{r=1}^{\Phi} p_r \cdot r$, where p_r is the probability that a call goes through r nodes.

6.3.1 New definitions

Def 6.2 I have defined the *average user density* ($\overline{D}_N$) of an *N*-node network:

$$\overline{D}_N = (1 - \frac{N_{tr}}{N}) \cdot \frac{\overline{L}}{\Phi}, \tag{23}$$

where N_{tr} is the number of transit nodes in the network (i.e., no end-user directly connected to it). The user density is an important network-level parameter, because it is indicating the distribution of the end users among the network nodes. In general, $0 < \overline{D}_N \le 1$. The bigger the value, the denser the users populate the network. In our sample networks (N = 4, 30, 35) this parameter is $\overline{D}_4 = 0.93$, $\overline{D}_{30} = 0.59$, $\overline{D}_{35} = 0.45$, while for an isolated switch $D_1 = 1$ (for more examples, see Table 6.1) .

Def 6.3 Furthermore, I have defined the *signalling load balance* of a given switch in the network:

$$\theta_i = \frac{\lambda_i - \frac{1}{N} \cdot \sum_{i=1}^{N} \lambda_i}{\max\limits_{1 \le i \le N} (\lambda_i)}; \quad -1 \le \theta_i \le 1; \quad 1 \le i \le N \tag{24}$$

where λ_i is the signalling load of a given switch. The maximum signaling load balance can be defined as:

$$\theta_N^{\max} = \frac{\max\limits_{1 \le i \le N} \left| \lambda_i - \frac{1}{N} \cdot \sum_{i=1}^{N} \lambda_i \right|}{\max\limits_{1 \le i \le N} (\lambda_i)}; \quad 0 \le \theta_N^{\max} \le 1; \quad 1 \le i \le N$$

A similar definition can be found in the literature by [Csel99], so called *signalling intensity balance*, but in his definition the average load is to be found in the denominator instead of the maximum load, thus $\theta_N^{\max}$ can be also higher than 1, which is not appropriate in my case.

6.3.2 Approximation formula

I have found that $\lambda_N^{\max}$ *can be approximated analytically by the following formula:*

$$\lambda_N^{\max} \approx \left(\lambda_1^{\max} \cdot \left(1 - \frac{\theta_N^{\max}}{N/(1+N_{tr})} \right) + \left| \frac{N_{ISP} - 1}{N_{ISP} + 1} \right| \cdot \frac{\lambda_1^{\max}}{\overline{D}_N} \right) \cdot \frac{1}{1 + \frac{\overline{s}}{2 - p_s}}$$

After resolving D_N, we obtain:

$$\lambda_N^{\max} \approx \left[1 - \frac{\theta_N^{\max} \cdot (1 + N_{tr})}{N} + \left| \frac{N_{ISP} - 1}{N_{ISP} + 1} \right| \cdot \frac{\Phi}{(1 - \frac{N_{tr}}{N}) \cdot \overline{L}} \right] \cdot \frac{\lambda_1^{\max}}{1 + \frac{\overline{s}}{2 - p_s}} \tag{25}$$

where, N_{ISP} is the number of internet access servers (ISP = Internet Service Provider),

$\overline{s} = \frac{1}{m} \cdot \sum_{i=1}^{m} s_i$ is the average signalling overhead for the "complex" call profile,

m is the number of different types of "complex" call profiles,

p_s is the probability that a call has a "complex" call profile.

This formula reflects the dependency of the $\lambda_N^{\max}$ on the:

- maximum capacity (speed) of the processor of individual switches ($\lambda_1^{\max}$);
- network topology (Φ, $\overline{L}$, N, N_{tr});
- user density (D_N), topological asymmetry ($\theta_N^{\max}$), bottleneck nodes;

- call complexity (signalling overhead (s, p_s));
- applications (LANE: branch offices connected to headquarter; internet access (N_{ISP})).

The applicability of this formula is dependent on how easy the average call path $\overline{L}$ and the maximum signalling load balance $\theta_N^{\max}$ can be found. I have obtained these parameters by simulation. However, for the first rough estimates the effect of these two parameters could be neglected if there is no possibility to compute them (i.e., $\overline{L} = \Phi$; $\theta_N^{\max} = 0$).

I have found lower and upper bounds of the $\lambda_N^{\max}$, respectively, i.e.,

$$\lambda_1^{\max} \le \lambda_N^{\max} \le \lambda_1^{\max} \cdot (1 + \frac{1}{D_N}) \le \lambda_1^{\max} \cdot (1 + \frac{N}{N - N_{tr}}).$$

If $N_{tr} = 0$, then $\lambda_1^{\max} \le \lambda_N^{\max} \le 2 \cdot \lambda_1^{\max}$**.** For realistic network scenarios the upper margin is usually larger, i.e., $\lambda_1^{\max} \le \lambda_N^{\max} \le 3 \cdot \lambda_1^{\max}$ (see e.g., Table 6.1).

6.3.3 Numerical results

Table 6.1 Comparison of simulated and estimated values of the $\lambda_N^{\max}$

If the max. call arrival rate for one isolated switch $\lambda_1^{\max}$ = 140 call/sec									max. network level call arrival rate $\lambda_N^{\max}$ [call/sec]		
N	N_{tr}	N_{ISP} (0=LANE) (0+1=mixed)	*L(i,j)*	Φ	D_N	$\theta_N^{\max}$	$\overline{s}$	p_s	simulated (x)	estimated with eq.(25) (y)	rel. error (y-x)/x [%]
4	0	0	1.86	2	0.93	0	0	0	335	290	-13.3
6	4	0	6	6	0.33	0	0	0	142	140	-1.4
7	0	2	2.53	4	0.63	0.56	0	0	198	203	+2.3
8	6	0	8	8	0.25	0	0	0	140	140	0
20	18	0	20	20	0.1	0	0	0	138	140	1.4
30	10	0	5.28	6	0.59	0.58	0	0	385	348	-9.4
35	13	0	5.72	8	0.45	0.75	0	0	355	410	+15.3
35	12	1	4.17	5	0.55	0.89	0	0	120	94	-21.9
35	12	2	4.67	6	0.51	0.78	0	0	250	191	-23.7
35	11	3	4.49	6	0.51	0.76	0	0	270	240	-11.1
4	0	0	1.86	2	0.93	0	1	1	150	145	-3.1
7	0	2	2.71	4	0.63	0.56	1	1	102	101	-0.7
30	10	0	5.28	6	0.59	0.58	1	1	207	174	-15.7
35	13	0	5.72	8	0.45	0.75	1	1	168	205	+21.9
35	12	1	4.17	5	0.55	0.89	1	1	50	47	-6.28
35	12	2	4.67	6	0.51	0.78	1	1	118	95	-19.2
35	11	3	4.49	6	0.51	0.76	1	1	130	138	+6.3
35	12	(0+1)	4.94	8	0.41	0.82	0	0	250	269	+7.6
35	11	(0+2)	5.20	8	0.45	0.76	0	0	300	313	+4.3

In the case of *FORE ASX200BX* switches I have obtained the following upper margins of $\lambda_N^{\max}$, where γ_R=1, $0 \le s \le 1$ (see Table 6.1). As the last column shows, the relative difference between simulated and measured values varies in the range of ± 20%. I have tried to cover many different scenarios, but of course it is unrealistic to believe that every possible network scenario can be investigated. The last two rows of Table 6.1 indicate a mixed scenario between LANE and internet access, the ratio is 50%-50% in both cases.

6.4 Investigating $\lambda_N^{\max}$ in a hybrid network

Secondly, I have studied the behaviour of the call processing in a *hybrid network* (not all network nodes are identical). In practice, we have often found two or three types of ATM switches in one network. In this case, the formula given in equation (25) will suffer some certain changes:

$$\lambda_N^{\max} \approx \left[\max[\min_{i=1..P}(\lambda_1^{\max})^{\text{type}\,i}, \min_{j=1..R}(\lambda_1^{\max})^{\text{type}\,j}] \cdot \left(1 - \frac{\theta_N^{\max}}{N/(1+N_{tr})}\right) + \left|\frac{N_{ISP}-1}{N_{ISP}+1}\right| \cdot \frac{\min_{i=1..P}(\lambda_1^{\max})^{type\,i}}{\overline{D}_N} \right] \cdot \frac{1}{1+\frac{\overline{s}}{2-p_s}} \tag{26}$$

where $(\lambda_1^{\max})^{\text{type}\,j}$ is the maximum call arrival rate of the switch connected to an ISP server ($1 \le j \le R$).

If $N_{ISP} = 0$, then it results in j=0, and thus the equation (26) will became simpler:

$$\lambda_N^{\max} \approx \left[1 - \frac{\theta_N^{\max}}{N/(1+N_{tr})} + \frac{1}{\overline{D}_N}\right] \cdot \frac{\min_{i=1..P}(\lambda_1^{\max})^{type\,i}}{1+\frac{\overline{s}}{2-p_s}} .$$

6.5 The impact of signalling overhead

Next, I have studied the evolution of the T_C and T_{RN} in the N-node network when instead of default *SETUP* messages (containing only the mandatory information elements) more complex calls are injected into the network. The effect of the complex call profile (*CCP*) for one switch was already described in Section 4.1.4 by a parameter '*s*', where $0 \le s < 1$.

6.5.1 Approximation formula

I have shown that, if $T_C^{CCP} = (1+s) \cdot T_C^{default}$ *for one node (see equation (6)), then the average call establishment time of the network will be increased to*

$$\overline{T}_C^{CCP} = (1 + \frac{\overline{s}}{2-p_S}) \cdot \overline{T}_C^{default}; \qquad 0 \le \overline{s} < 1, \tag{27}$$

while the average call release latency $\overline{T}_{RN}$ *remains unchanged:*

$$\overline{T}_{RN}^{CCP} = \overline{T}_{RN}^{default} . \tag{28}$$

where: $\overline{s} = \frac{1}{m} \cdot \sum_{i=1}^{m} s_i$ is the average signalling overhead for the "complex" call profile,

m is the number of different types of "complex" call profiles,

p_s is the probability that a call has a "complex" call profile.

6.5.2 Numerical results

When all calls in the network are of the same complexity then the $p_S = 1$, when none of them is complex then $p_S = s = 0$. A couple of examples are shown in Table 6.2.

Table 6.2 The impact of signalling overhead on call establishment times and release latencies in a 35-node network

N (LANE)	$\bar{s}$	p_s	*average* T_C [msec] (estimated with eq.(27)) (x)	*average* T_C [msec] (simulated) (y)	*rel. error* [%] (x-y)/y	*average* T_{RN} [msec] (simulated)
35	0	0	38.73	38.73	0	12.29
35	0.5	0.25	49.80	48.77	2.1	12.29
35	0.5	0.5	51.64	49.51	4.3	12.29
35	0.5	0.75	54.22	52.03	4.2	12.29
35	0.5	1	58.1	55.88	4.0	12.29
35	1	1	77.46	67.33	15.0	12.29

The values in Table 6.2 are obtained for 1 call/sec arrival rate. The analytical formula approximates the simulation results very well, with a relative error of 4% (except the last case, where $\bar{s} = p_S = 1$, which is anyway not realistic, but only un upper limit). Equation (27) and (28) is especially important in the network design to avoid signalling congestion in network nodes and underlines again the differences in the behaviour between T_C and T_{RN}.

6.6 A new method of determining the signalling load of a node

Unfortunately, at the moment, our simulator (ACCEPT, version 2.0) cannot indicate directly the signalling load of each individual switch in the network. Therefore, to be able to get $\theta_N^{\max}$ by simulation, I was forced to develop a method that enables us to obtain the signalling load of the switches. Similar situation may happen also in practice, that we do not know the loads in the network, but the call establishment times only (e.g., in global networks with network cross-domains owned by different providers). In such a case, the following method is very useful.

Let us note the equivalent service rates for the call establishment of a node $\mu^C = \mu_{eq}^{setup}$, and for the call release $\mu^R = \mu_{eq}^{release}$, respectively.

I have shown that replacing one node of a network ($(\mu_i^C;\mu_i^R)$*, i=1,...,N) by a switch having a signalling service rate* $(\mu_0^C \le \frac{1}{\Phi+1}\cdot\min_{i=1..N}\mu^C;\mu_0^R \le \frac{1}{\Phi+1}\cdot\min_{i=1..N}\mu^R)$*, can be used to determine the signalling load of this node. Replacing each node one-by-one will help us identifying the bottle-neck nodes in a given network topology.*

If it is not possible to monitor and analyse each network node individually, then the method described above helps us to obtain the signalling loads. All we need is to replace one node with the above given parameters and run the simulation at very low call arrival rate to obtain the call establishment times and release latencies. Then analysing the results, we should see that those calls with $T_C > \frac{1}{\mu_0^C}$ are processed by the test node as well. The number of these calls over the total number of generated calls gives the load of the investigated node. Thus, the signalling load of this node can be defined as follows: $\lambda_i = \lambda_N \cdot p_i$, *i*=1,...,*N*, where p_i is the probability that the call will be processed by node *i*. The bottle-neck node in the network has its load equal to:

$$\lambda_i^{\max} = \max\{\lambda_i | \lambda_i > \lambda_j, \forall i, j = 1,...,N\}.$$

6.7 Numerical results: The impact of network failures and of applications

6.7.1 Case study 1: The impact of link and node failures

I have investigated the impact of link and node failures in the core of the 35-node COLT TELECOM Germany's ATM network (in the case of pure LANE services). In this study all nodes are considered to be of type *FORE ASX200BX*. In reality, the access nodes are of type *Seabridge XP140*, which does not change the essence of the problem, but the call establishment times will be a bit longer. In general, one node failure in the backbone resulted in 3-to-5 adjacent link failures.

Statement 1 I have found that the failure of one link/node in the core of the 35-node sample network given in Figure 6.1*b* has the following impacts (see Table 6.3 and Table 6.4 and Figure 6.4, respectively):

- *it will result in a call blocking probability P_B of 20-45%, where there is no alternative path;*
- *it will increase the signalling load of the neighbour switches by 10-25%;*
- *it will increase the average load of the backbone by 5-10%;*
- *and it will result in a fluctuation of the average $\overline{T}_C$ and $\overline{T}_{RN}$ by maximum ± 5%.*

This statement shows that the luck of redundancy is especially critical for switches carrying a load over 30% of the total network load. The load of some neighbour switches may increase from 40% up to 60% of the total load of the network (which is a 50% increase, as shown in Figure 6.4). These margins should be taken into account by network planning. Another example shows that the failure of switch #11 or #14 will double or almost triple the load of the switch #16, respectively (see Table 6.3). The switch ID numbers are shown in Figure 6.5. The calls that travel along longer paths (4-to-8 hops) are primarily affected by a failure in the core network (LANE).

Table 6.3 The impact of node failure on the signalling load of the neighbour switches (check switch IDs in Figure 6.5)

Switch #	no failure, initial load/sw	failure of sw # 11 [%]	failure of sw # 14 [%]	failure of sw # 16 [%]	failure of sw # 20 [%]	failure of sw # 21 [%]	failure of sw # 22 [%]	failure of sw # 23 [%]
11	40.9	-	30.94	39.17	45.47	33.05	**59.74**	48.59
14	23.22	22.76	-	26.94	19.02	0	44.16	26.01
16	14	32.17	37.93	-	22.42	25.23	0	25.58
20	37.95	53.49	38.41	45.03	-	49.33	**61.42**	33.9
21	44.71	49.38	45.01	47.89	50.24	-	48.8	46.26
22	36.17	37.11	**59.64**	32.1	54.9	33.39	-	45.85
23	17.7	16.25	17.3	28.29	8.95	27.41	36.21	-
lost calls [%]	0	21.27	0	0	32.86	45.02	21.49	7.51
Avg load/sw [%]	30.66	35.19	38.21	36.57	33.50	28.07	41.72	37.70
avg T_C [msec]	38.73	38.42	39.62	39.04	37.27	36.54	40.47	39.77
avg T_{RN} [msec]	12.29	12.19	12.56	12.38	11.84	11.62	12.82	12.60

It can be observed that the switch #16 has its role to offer alternative path for calls in case of link or node failure in the backbone. We can conclude that by link and node failures it is very important to consider the high probability of lost calls if no alternative links are available (e.g., P_B =45% by failure of switch #21), and the important growth in the signalling load of adjacent switches (e.g., up to 25% increase of λ_N , i.e. 50% increase of λ_i , see Figure 6.4).

Table 6.4 The impact of link failure on the signalling load of the neighbour switches (check link IDs in Figure 6.5)

Switch #	no failure, initial load/sw	failure of L(21,22) [%]	failure of L(22,23) [%]	failure of L(20,23) [%]	failure of L(11,20) [%]	failure of L(11,14) [%]	failure of L(14,21) [%]	failure of L(13,22) [%]
11	40.9	43.18	40.9	45.74	40.9	40.9	40.9	34.88
14	23.22	41.05	23.22	23.22	23.22	23.22	0	11.79
16	14	29.56	14.01	27.28	22.56	33.32	26.51	9.14
20	37.95	40.23	37.95	35.07	37.95	37.95	37.95	31.67
21	44.71	44.71	44.71	44.71	45.20	44.71	44.71	24.27
22	36.17	33.89	36.17	39.03	37.11	37.12	**58.39**	26.22
23	17.7	17.7	17.7	15.96	17.7	17.7	17.7	16.01
lost calls [%]	0	0	0	0	0	0	0	27.13
Avg load/sw [%]	30.66	35.76	30.67	33.00	32.09	33.56	32.31	22.00
avg T_C [msec]	38.73	40.87	38.96	39.70	39.10	39.52	39.52	37.15
avg T_{RN} [msec]	12.29	12.94	12.36	12.58	12.40	12.53	12.53	11.81

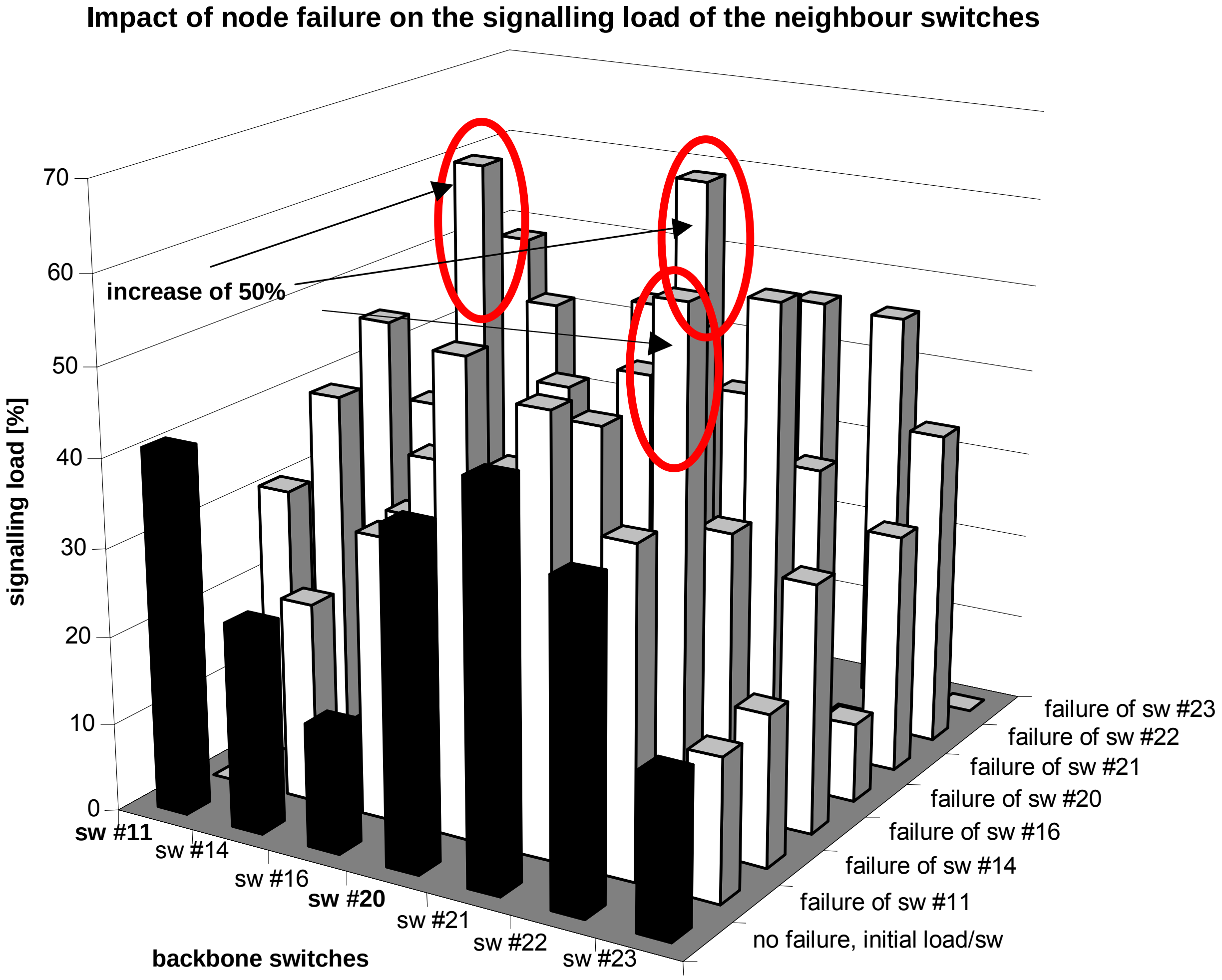


Figure 6.4 Impact of node failure in the backbone of the 35-node COLT-network

Furthermore, I have tested the impact of different user applications on the signalling load of switches in the backbone. Internet connectivity via one, two or three access servers, respectively has been studied vs. LAN Emulation service. The mixed scenario has been also investigated. I have found that the distribution of the network load λ_N strongly depends on the number of ISPs present in the network (see Figure 6.5).

6.7.2 Case study 2: The impact of user applications (LANE, Internet access via ISPs)

Statement 2 I have shown that the signalling load of switches in the same network topology is strongly affected by the types of user applications (Figure 6.5), i.e.:

- *one internet access only results in a 100% load of the switch connected to this server;*
- *more internet servers (connected to different nodes) share the load* λ_N *approx. equally;*
- *the signalling load is smoothly distributed among the backbone nodes in the case of LANE services, i.e.* $\lambda_i \approx (0.2-0.4)\cdot\lambda_N$, *and*

$$\theta^n_{n\in S} = \{0.23, -017, -0.37, 0.16, 0.31, 0.12, -0.29\};$$

$$S = \{sw\#11, sw\#14, sw\#16, sw\#20, sw\#21, sw\#22, sw\#23\}.$$

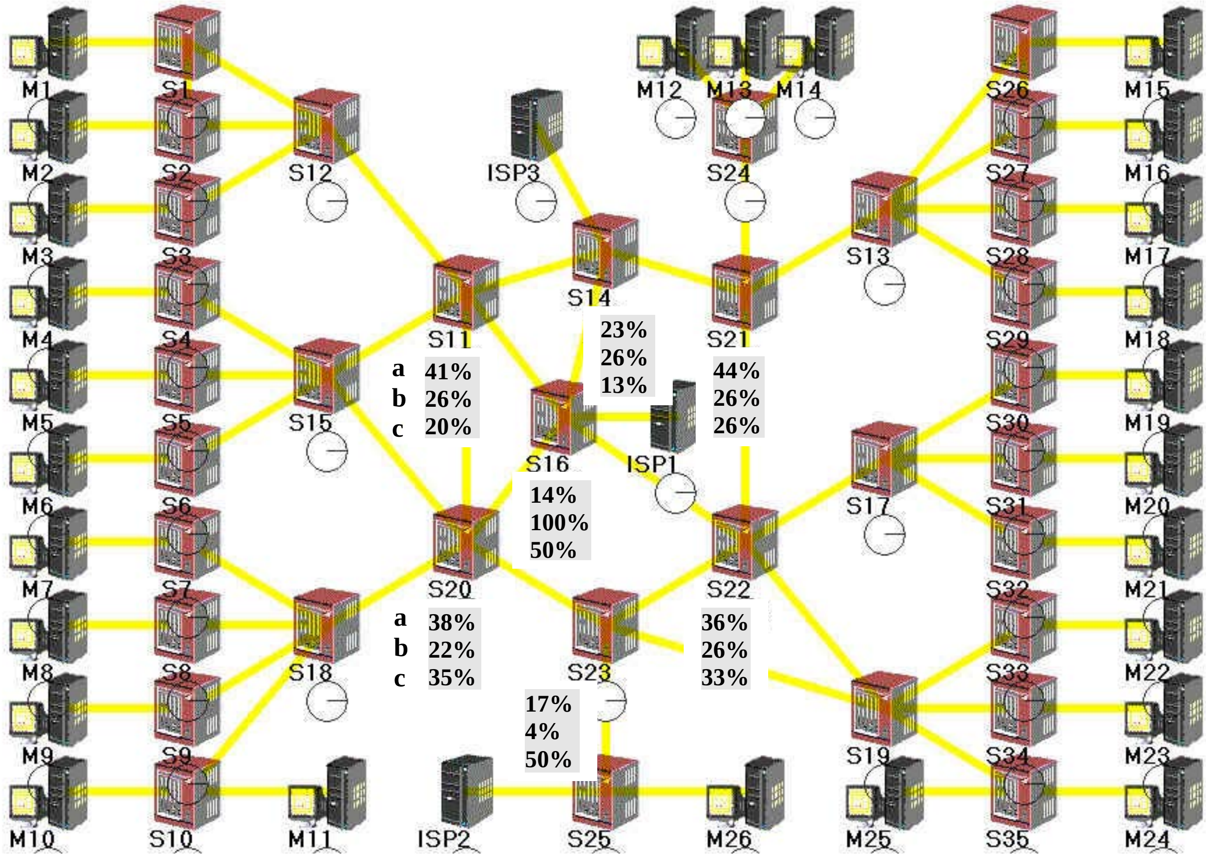
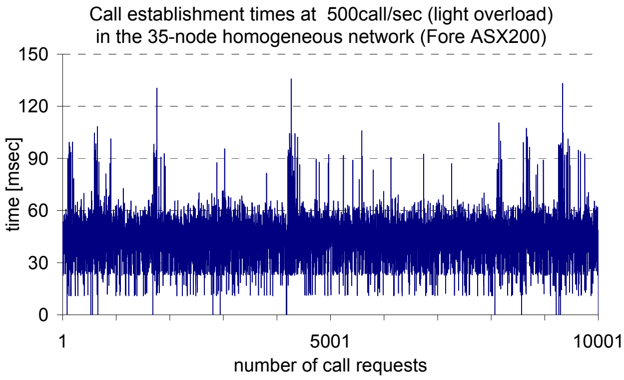


Figure 6.5 Measuring the relative signalling load (p_i) in the core of the sample 35-node network
a) LANE service (1st value); b) Internet access via one ISP (2nd); c) Internet access via 2 ISPs (3rd)

6.8 Validation of the results

We have considered that the call arrival rate is poissonian and the source-destination pairs are either uniformly distributed in the network (for LANE) or there are centrally located servers (ISP1,2,3) for Internet connectivity (see Figure 6.5). Please, consider also, that we have defined only one hierarchical level in our sample PNNI networks. Simulations have been carried out for the following network-level call arrival rates: 1, 5, 33, 100, 200, 400, 500 calls/sec (e.g., see Figure 6.6). Each time 10000 calls have been evaluated. The zero setup times represent the lost calls in the network. An arrival rate of 500 calls/sec resulted in 20 lost calls out of 10000 (P_B = 0.2%).

The simulation results have been validated for 2, 3 and 4 cascaded switches against measured data (e.g., see Figure 4.9 and Table 4.3 for *FORE ASX200BX*, *SEABRIDGE XP140* and mixed types, respectively).



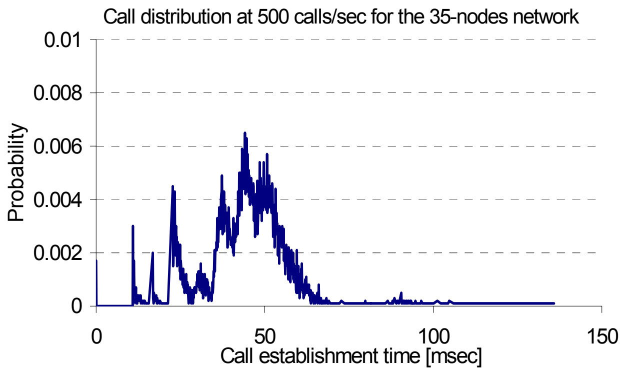


Figure 6.6 Distribution of call establishment times in the 35-node homogeneous network (0.2% of calls are dropped)

Furthermore, I have implemented by simulation the *TEN-155* network scenario as described in [Nov99], and validated my results against those measured data of T_C (see Table 5.2). There are two coupled papers in the literature, [Gel97] and [Man97], which studied the impact of the Call Admission Control in the network nodes via signalling message traffic for establishing calls, using two strategies: with or without resource reservation in a 10x10 grid and obtained the queue lengths of nodes via simulation. However, they did not consider the impact of the release messages on the queue lengths. As the current version of our simulator does not deliver the queue lengths of individual nodes, but the averages, there was no possibility to compare our results to those of [Gel97].

The importance of the statement in Section 6.2 is reflected in the fact that independent of the network topology, a cascaded network will always slightly over-estimate the mean and maximum of the call establishment times and release latencies in the network and thus it is a good estimate for network planning. As a validation, the sample 35-node network is transformed to a cascade along its longest path of $r = 8$ hops (see Figure 6.7). Instead, in our cascade, all nodes have a load of 100%, which explains why the maximum and average call establishment times are higher in that case.

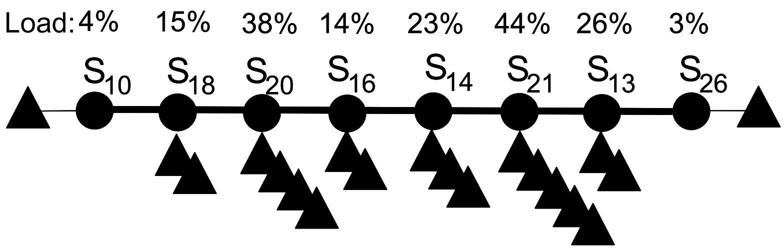


Figure 6.7 Translation of the 35-node network into a cascade, r=Φ=8.

The formula in equation (25) has been validated by simulating many different network scenarios and applications. In equations (27) and (28), I have pointed out again the different behaviour of the T_C and T_{RN} in an N-node network for complex calls. A case study for the impact of link/node failures and network applications on the signalling load of a 35-node network has been investigated and presented above to emphasize the applicability of the simulation model.

6.9 Conclusions

Chapter 6 focused on the performance analysis of signalling in large ATM networks. As a very important contribution, it pointed out that a cascaded network is always a worst case scenario compared to an arbitrary network topology when looking at call establishment times and release latencies. Furthermore, it presented details of finding the maximum network-level call arrival rate in both homogeneous and hybrid networks, gave a quantitative measure of signalling overhead due to the complexity of multimedia calls and presented a method how the signalling load of each network node can be determined by simulation. A case study of calculating the signalling load of each network element due to link or node failures and different applications (e.g., LANE, internet access) has been presented at the end of this chapter. Such investigations are of key importance for service providers in their network management system.

CHAPTER 7

7 Optimisation of the call processing architecture of the AAL2 switch in UMTS networks

7.1 Introduction

The ATM adaptation layer type 2 (AAL2) defined in [AALI97] has been selected as the transmission technology in the landline part of the radio access network of UMTS systems [Ene99]. It was briefly presented in Chapter 2, end of Section 2.1. To suit AAL2 to a network where support for soft handoff is essential, an additional switching level on top of ATM has been developed.

7.1.1 The investigated UMTS network scenario

The main architectural elements of a third generation mobile radio access network are shown in Figure 7.1.

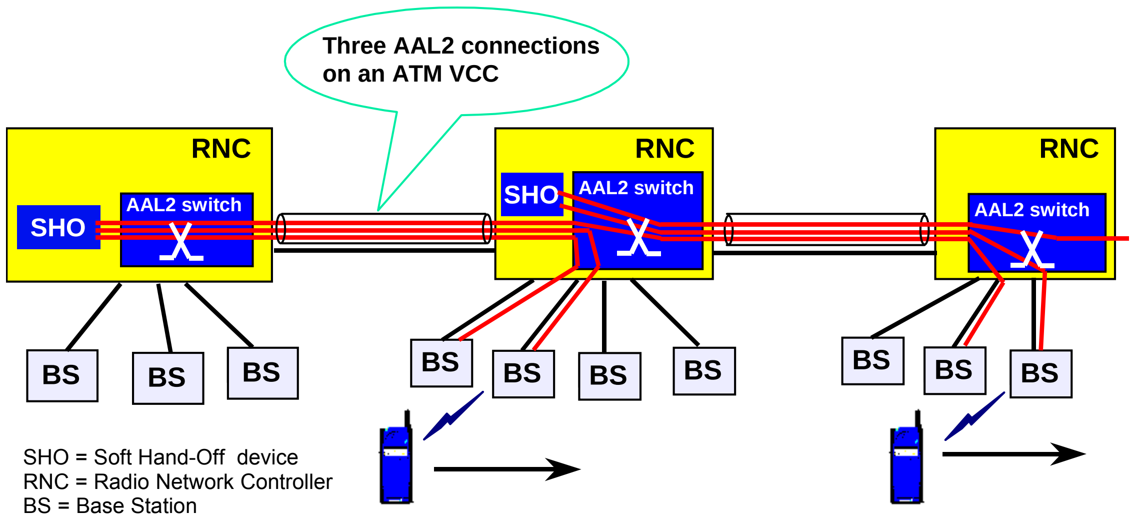


Figure 7.1 AAL type 2 connections in an UMTS Radio Access Network

The network architecture has the following properties:

- AAL2 switches are located in Radio Network Controller and Base Station nodes;
- There is no direct connection between the Base Stations;
- AAL2 connections are always initiated by the SHO unit and terminated in the BS;
- It is possible to establish AAL2 connection from any Radio Network Controller to any Base Station.

AAL2 Signalling is a new protocol which is capable of handling on-demand, switched AAL2 connections [AALQ99]. It makes possible carrying small data packets on top of an ATM infrastructure with low delay while using the bandwidth efficiently [Bal97]. In this chapter, I am dealing with certain design considerations of an AAL2 signalling point and propose techniques to further decrease the AAL2 connection establishment time for soft handoff legs [Won97]. It should be mentioned that at the time of writing and publishing my results (see e.g., [C-7]) there was no AAL2 switch available on the market (1999-2000).

7.1.2 General framework of the AAL type 2 signalling protocol

The AAL type 2 signalling protocol provides the signalling capability to establish, release and maintain AAL type 2 point-to-point connections across a series of ATM VCCs that carry AAL2 links. These services are accessible via the AAL2 served user service access point. The AAL2 signalling protocol also provides maintenance functions associated with the AAL2 signalling. An AAL2 signalling endpoint shall be able to control AAL2 links on more than one AAL2 path. These AAL2 paths may be contained on different ATM VPCs, which in turn may be carried on different ATM physical interfaces. Both peer AAL2 signalling entities provide the same set of services.

The AAL type 2 signalling is independent of the signalling transport, although an assured data transport is required and a message size limit applies. To adapt the generic signalling transport services to a specific signalling transport service, a signalling transport converter may be needed.

The AAL type 2 signalling entity provides the following services to the AAL2 served user (see Figure 7.2):

- establishment of AAL2 connections (*Establishment Request = ERQ; Establishment Confirm = ECF*); and
- release of AAL2 connections (*Release Request = REL; Release Confirm = RLC*).

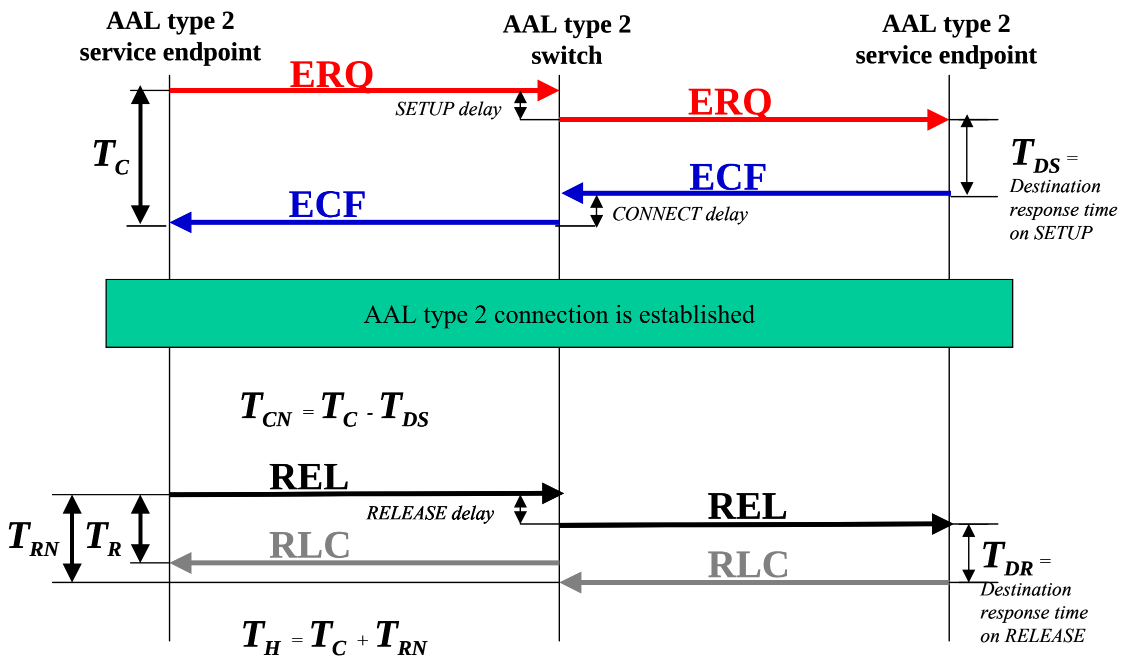


Figure 7.2 AAL type 2 Signalling message flows

An *ESTABLISH REQUEST (ERQ)* message describes the properties of the connection, and carries the Destination Service Endpoint Address. When processing it, the switch selects an outgoing link, which is able to accommodate the new connection and leads towards the specified destination, reserves resources on the selected link, and forwards the message. If a proper outgoing link is not available, then a negative acknowledgement (in the form of an *RLC* message) is returned back to the preceding switch.

7.2 Construction of the model

7.2.1 The architecture of the model

I have shown that the message flow model described in Section 5.1 can be applied to model an AAL type 2 signalling node with certain re-design according to the protocol specification in [AALQ99].

A re-design means, that there is no need for 'complex call profile' internal process (CCP) in the model, the messages have shorter formats and different notations (see Figure 5.2 and Figure 7.2). Other features are similar to the description in Section 5.1.

7.2.2 Parameter settings of the model

The problem can be formulated as follows: given that the overall capacity of the signalling processor is constant (fixed), we have to (re)allocate the service times of each process in a way *to minimise* either the AAL2 connection establishment time T_C or the overall handling time T_H (see Figure 7.2). A single AAL2 switch and two connection endpoints are used in the optimisation process, the Soft Hand-Off device can only initiate and the Base Station will terminate the connections. The connection arrival process is poissonian, and the distribution of the connection holding time is exponential.

7.3 The optimisation algorithm QUE

7.3.1 The description of the algorithm. FIFO queueing.

I have developed a simulation-based iterative algorithm, optimising the allocation of processor resources in order to minimise the call establishment times and release latencies of soft handoffs in UMTS networks.

The algorithm (see Figure 7.3) starts with assigning an initial service rate to each process, sets the connection arrival rate and the step size $\Delta\mu$, as being half of the maximum service rate. The buffer sizes are set to infinity, to avoid connection rejection due to buffer overflow. The second phase is to run the simulation, and compute $Q_{tot}(n)$ and $T(n)$, where n represents the number of correct steps. $Q_{tot}(n)$ is the sum of the average queue length measured in the buffers of individual processes in step n. $T(n)$ is equal to the connection establishment time, or to the overall connection handling time in step n respectively, depending on the optimisation criterion. In the next phase, the processor capacity is reallocated in a way that the service rate of the busiest process is increased by $\Delta\mu$, and the service rate of the process with the shortest average queue length is decreased by the same value. Thus the overall processor capacity is kept constant. We repeat the simulation run, and compute $Q_{tot}(n+1)$ and $T(n+1)$. If the last step improved the performance ($T(n+1)<T(n)$), the step is considered correct, n is incremented, and the capacity allocation is updated. If $T(n+1)>T(n)$, then the step size ($\Delta\mu$) is halved subsequently, until the readjusted capacity allocation results in $T(n+1)<T(n)$, or we arrive to the predefined minimum value of $\Delta\mu$, when the algorithm is stopped. The comparison of $Q_{tot}(n+1)$ to $Q_{tot}(n)$ allows the algorithm to find the global optimum point instead of a local one.

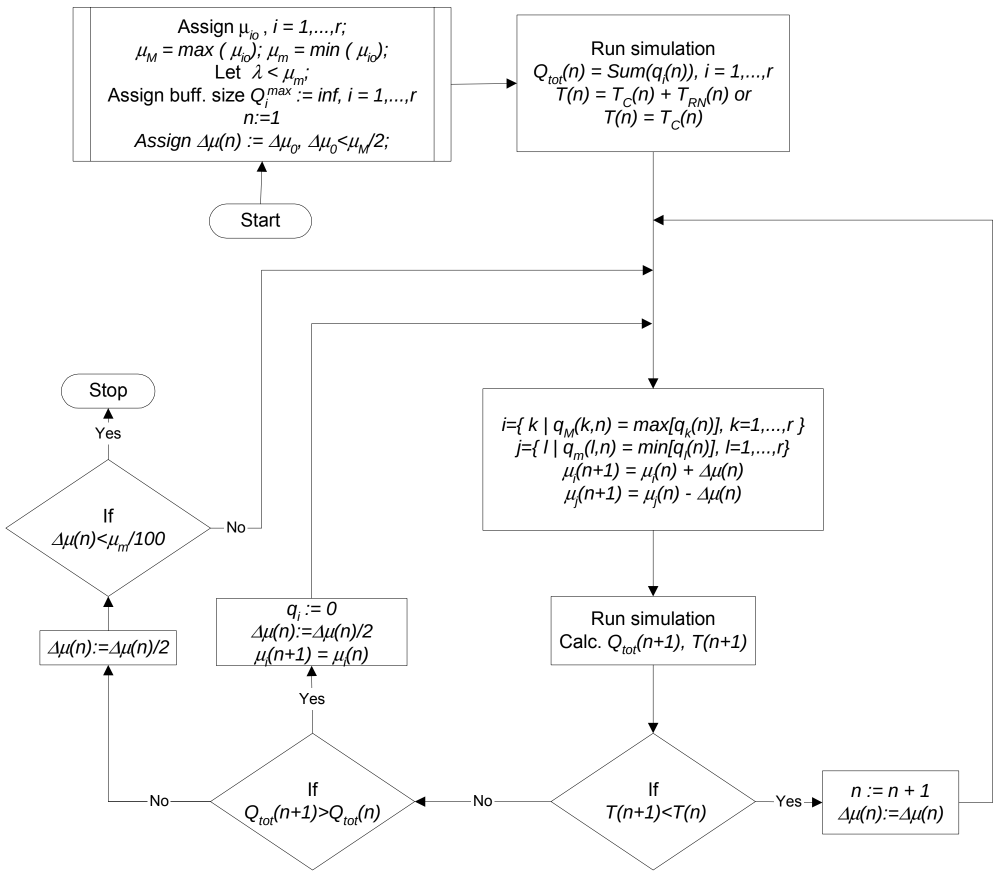


Figure 7.3 Block diagram of the optimisation algorithm QUE (FIFO queueing)

7.3.2 Numerical results. Case study 1: Starting from the EXTREME_x state (x=1,…,5)

A case study is shown below in Table 7.1 in order to help understanding the steps of the algorithm QUE, the functionality of the three loops and of the decision points. We have selected a fixed capacity of the processor under design to be 166 instructions/time unit (e.g., t.u. = 1 millisecond). For the start we have distributed this capacity in an extremely unbalanced way, in a so called *EXTREME_x states*, i.e., four sub-processes get only one unit and the remaining one gets 162 units, x is equal to the index of the sub-process which gets the maximum units (e.g., $x = 4$ in Table 7.1). These first settings of the processor will result in an average call establishment time of 13280 seconds (i.e., 3 hours and 41 minutes!), due to the simple fact that we have four bottleneck queues with very low capacities. After completing the optimisation process the call establishment time will be reduced to order of hundreds of millisecond, i.e., a ratio of order of 10^4.

After circa 7 runs the algorithm QUE drives into a relatively balanced distribution of the resources, where the average call establishment time could be reduced to approx. 300 milliseconds. As the last column shows, not all runs are considered being a correct step. The last correct step (e.g., step 10) is considered the final state. Of course, if we select another criteria for T(n), then we will arrive into a different final state. In run 6 and 18 respectively, the condition $Q_{tot}(n+1) > Q_{tot}(n)$ is satisfied, therefore the maximum queue length will be set to null, and the last correct distribution is reloaded. These actions are represented in an additional row in Table 7.1. The algorithm is stopped once we have reached the predefined condition for $\Delta\mu$.

Table 7.1 Starting from EXTREME_4 state, $\lambda = 1$call/sec, FIFO queueing, the algorithm QUE is applied for T(n)=T(H)

RUN	μ1	μ 2	μ 3	μ 4	μ 5	q1	q2	q3	Q4	q5	T_C [ms]	T_{RN}	T(n) = T_C + T_{RN}	Q_{tot} (n)	T (n+1) < T (n)	Q_{tot} (n+1) > Q_{tot} (n)	q max ID	q min ID	Δμ(n)	CORRECT steps (n)
1	**1**	**1**	**1**	**162**	**1**	1015	1569	2203	0.05	0.7	13.28 M	7.00 M	20.28 M	4788	-	-	3	4	81	1
2	1	1	82	81	1	1805	2945	0.35	0.06	0.59	12.79 M	7.02 M	19.81 M	4752	yes	-	2	3	81	2
3	1	82	1	81	1	1804	0	2947	0	0.60	12.80 M	7.03 M	20.83 M	4752	no	no	3	2	40.5	-
4	1	41.5	41.5	81	1	4621	0	0	0	0.37	11.29 M	6.97 M	18.26 M	4621	yes	-	1	4	40.5	3
5	41.5	41.5	41.5	40.5	1	0.008	0	0.002	0.018	3046	3.046 M	1.879 M	4.925 M	3046	yes	-	5	2	40.5	4
6	41.5	1	41.5	40.5	41.5	0.002	4621	0	0	0	11.29 M	6.97 M	18.26 M	4621	no	yes	-	-	20.25	-
	41.5	41.5	41.5	40.5	1	0.008	0	0.002	0.018	Ø	Set q5:= Ø; μ(5):=μ(4)						4	2	20.25	
7	41.5	21.25	41.5	60.75	1	0.01	0.02	0	0.006	3046	3.047 M	1.879 M	4.926 M	3046	no	no	5	3	10.12	-
8	41.5	21.25	31.38	60.75	11.12	0.146	0.167	0.08	0.022	0.217	338.2	160.9	499.1	0.632	yes	-	5	4	10.12	5
9	41.5	21.25	31.38	50.62	21.25	0.069	0.231	0.088	0.021	0.147	294.8	137.1	431.9	0.556	yes	-	2	4	10.12	6
10	41.5	31.38	31.38	40.5	21.25	0.092	0.161	0.097	0.028	0.128	264.5	119.2	383.7	0.506	yes	-	2	4	10.12	7
11	41.5	41.5	31.38	30.38	21.25	0.096	0.064	0.121	0.055	0.143	256.9	110.3	367.2	0.480	yes	-	5	4	10.12	8
12	41.5	41.5	31.38	20.25	31.38	0.081	0.098	0.136	0.075	0.079	257.9	102.1	360.0	0.471	yes	-	3	4	10.12	9
13	41.5	41.5	41.5	10.12	31.38	0.088	0.062	0.086	0.121	0.111	292.3	92.8	385.1	0.470	no	no	4	2	5.06	-
14	**41.5**	**36.44**	**41.5**	**15.18**	**31.38**	0.086	0.100	0.076	0.054	0.135	263.4	96.2	359.6	0.467	yes	-	5	4	5.06	10
15	41.5	36.44	41.5	10.12	36.44	0.079	0.146	0.052	0.095	0.090	294.2	94.4	388.6	0.463	no	no	2	3	2.53	-
16	41.5	38.97	38.97	10.12	36.44	0.061	0.149	0.092	0.098	0.060	293.7	94.3	388.0	0.461	no	no	2	5	1.27	-
17	41.5	40.24	38.97	10.12	35.17	0.060	0.109	0.104	0.099	0.094	292.9	93.9	386.8	0.467	no	no	2	1	0.63	-
18	40.87	40.87	38.97	10.12	35.17	0.061	0.063	0.129	0.121	0.093	292.9	93.9	386.8	0.468	no	yes	-	-	0.32	-
	41.5	36.44	41.5	15.18	31.38	0.086	0.100	0.076	0.054	Ø	Set q5:= Ø; μ(11):=μ(10)						2	4	0.32	
19	41.5	36.76	41.5	14.86	31.38	0.088	0.101	0.076	0.053	0.135	264.2	95.9	360.1	0.452	no	no			0.16	STOP

7.4 The optimisation algorithm QUE/MU

The algorithm QUE is just one method to minimise the connection times. In order to further decrease the call establishment time, we have introduced *prioritised handling* of signalling messages in the AAL2 switch. Particularly, we have assigned higher priority to the messages involved in the call establishment phase (*ERQ* and *ECF*, respectively). Two parallel queues are implemented in all the processing elements of the AAL2 switch. The first queue collects the *ERQ* and *ECF* messages. The messages in this queue get absolute priority over messages in the second queue, which means that the server starts fetching messages from the second queue only if the first one is empty. The second queue collects the messages that are used for connection release. Some maintenance messages are also defined in AAL2 Signalling. We ignore these messages in our study because they are related to exceptional cases, and their rate is very small compared to the rate of traffic handling messages.

7.4.1 The description of the algorithm. Priority queueing.

I have found that the optimisation algorithm QUE will not always find the global optimum when priority handling is used, therefore another criterion should be applied here.

I have shown that replacing $Q_{tot}(n) = \sum_{i=1}^{r} q_i(n)$ *by* $Q_{tot}(n) = \sum_{i=1}^{r} \frac{q_i(n)}{\mu_i(n)}$ *will enhance the performance of the algorithm QUE and will be applicable to both FIFO queueing and priority handling of the signalling messages.*

While the algorithm QUE minimises the sum of the queue lengths of processes, the algorithm QUE/MU minimises the average time the jobs spend in one node. Therefore in the case of FIFO queueing we obtained very closed results with both algorithms. The definition of a global minimum depends on the objectives: do we want to obtain a minimum for the call establishment time or for the overall handling time? As an example, Figure 7.4*a* compares the results of the above two algorithms obtained step-by-step when no priority for messages is applied. The call establishment time T_C is shorter in the case of algorithm QUE/MU than for the algorithm QUE, but (at the same time) the call release time T_R is longer, so that the overall handling time T_H is very close in both cases. Let us illustrate these in the next case study.

7.4.2 Numerical results. Case study 2: Starting from the EQUILIBRIUM state

This time I have chosen the starting point of the algorithms as being the *EQUILIBRIUM state* (this state will be defined as a *reference point* for further comparisons), where the initial distribution of the resources is uniform among the five processes. E.g., if the total capacity C=166 instructions/msec, then $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$ = 33.3 instructions/msec, as shown in the first row of Table 7.2, which presents the steps of the algorithm QUE/MU in the case of FIFO queueing, starting from the *EQUILIBRIUM* state and aiming to minimise the overall handling time T_H. The algorithm converges in 15 steps to the final state, while visiting 7 correct states (i.e., in these states $T(n+1)<T(n)$). One can observe, that starting from step n=3, the index of the q/μ(min) is different from the index of q(min), thus the two algorithms will take different ways. This fact can be followed in Figure 7.4*a*. Basically, the result (the final distribution of the resources) delivered by the algorithm QUE/MU is independent of the initial state and of the processor capacity.

Table 7.2 Start from EQUILIBRIUM state, λ = 5calls/sec, FIFO queue, algorithm QUE/MU to minimise T(n)= T(H)

RUN	μ1	μ 2	μ 3	μ 4	μ 5	q1	q2	q3	q4	Q5	T_C [ms]	T_R [ms]	T(n) = T_C + T_R	Q_{tot} = Σq/μ	T(n+1) < T(n)	Q_{tot}(n+1) > Q_{tot}(n)	q/μ max	q/μ min	q max	q min	Δμ(n)	CORRECT step (n)
1	**33.3**	**33.3**	**33.3**	**33.3**	**33.3**	1.05	0.75	1.04	0.16	0.44	402	377	779	3.44	-	-	1	4	1	4	16.7	1
2	50	33.3	33.3	16.7	33.3	0.48	0.98	1.17	0.38	0.22	401	352	752	3.23	yes	-	3	5	3	5	16.7	2
3	50	33.3	50	16.7	16.7	0.5	0.98	0.28	0.34	1.15	407	359	766	3.25	no	yes	-	-	-	-	8.33	-
	50	33.3	33.3	16.7	33.3	0.48	0.98	Ø	0.38	0.22	Set q3:= Ø; μ(3):=μ(2)						2	5	2	5	8.33	
4	50	41.6	33.3	16.7	25	0.49	0.64	1.30	0.38	0.40	397	337	733	3.21	yes	-	3	1	3	4	8.33	3
5	33.3	41.6	50	16.7	25	1.3	0.36	0.51	0.35	0.50	384	318	701	3.02	yes	-	1	2	1	4	8.33	4
6	41.6	33.3	50	16.7	25	0.89	0.93	0.50	0.37	0.42	382	329	711	3.02	no	no	2	1	2	4	4.16	-
7	37.5	37.5	50	16.7	25	1.08	0.53	0.55	0.33	0.51	375	314	689	3	yes	-	1	3	1	4	4.16	5
8	41.6	37.5	45.8	16.7	25	0.9	0.61	0.62	0.38	0.47	373	312	685	2.98	yes	-	4	3	1	4	4.16	6
9	41.6	37.5	41.6	20.8	25	0.86	0.62	0.72	0.28	0.54	367	323	690	2.98	no	no	3	4	1	4	2.08	-
10	41.6	37.5	43.7	18.7	25	0.88	0.63	0.68	0.29	0.52	368	318	686	3.01	no	yes	-	-	-	-	1.04	-
	41.6	37.5	45.8	16.7	25	Ø	0.61	0.62	0.38	0.47	Set q1:= Ø; μ(7):=μ(6)						3	4	3	4	1.04	
11	41.6	37.5	44.8	17.7	25	0.88	0.62	0.66	0.32	0.53	371	315	686	2.98	no	no	4	3	1	4	0.52	-
12	41.6	37.5	45.3	17.2	25	0.9	0.60	0.65	0.37	0.46	372	313	685	2.98	no	no	4	3	1	4	0.26	-
13	**41.6**	**37.5**	**45.6**	**16.9**	**25**	0.89	0.61	0.63	0.38	0.46	372	311	683	2.98	yes	-	3	4	1	4	0.26	7
14	41.6	37.5	45.3	17.2	25	0.9	0.60	0.65	0.37	0.46	372	313	685	2.98	no	no	4	3	1	4	0.13	-
15	41.6	37.5	45.4	17.0	25						372	313	685		no							STOP

Figure 7.4*a* shows a comparison of the algorithm QUE vs. the algorithm QUE/MU, when both start from the *EQUILIBRIUM* state, having a total capacity of C=166 instructions/msec, and aiming to minimise T_H in the case of FIFO queueing. It can be observed, that the first couple of steps are common in both cases. Figure 7.4*b* illustrates the difference in the initial and final distributions of the processor resources for the algorithm QUE/MU, for the case study presented in Table 7.2.

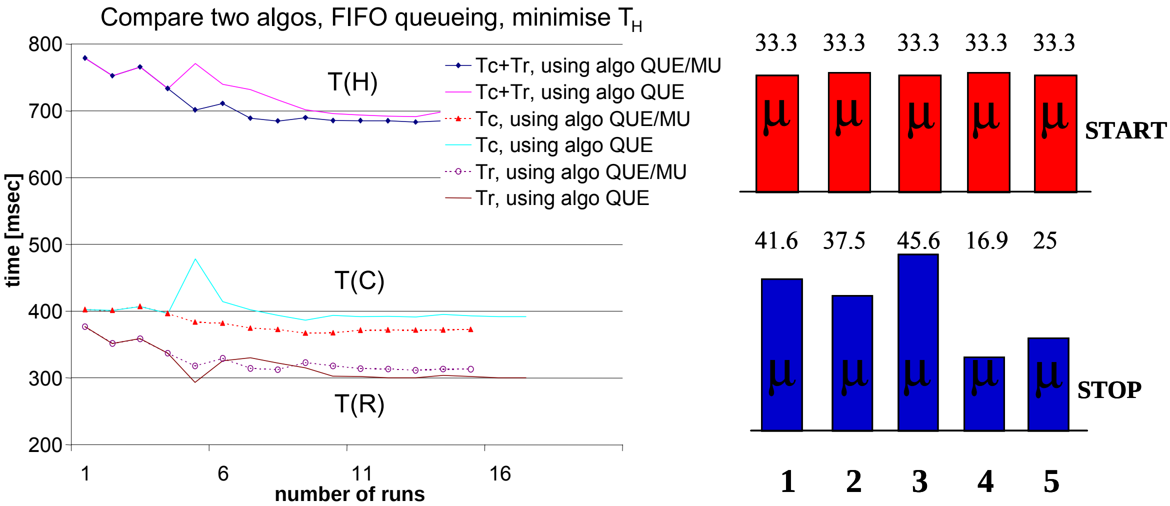


Figure 7.4 a) Effect of two algorithms on T_H, T_C, and T_R starting from the EQUILIBRIUM state (FIFO queueing); b) Initial and final distributions of the processor resources for the algorithm QUE/MU

7.5 Performance evaluation

7.5.1 The impact of the optimisation algorithm and that of priority handling

I have investigated the optimisation of the processor resources in the following four cases (see Table 7.3), using the optimisation algorithm QUE/MU.

Table 7.3 Definition of four case studies

Priority handling	Minimise	
	T_C of an AAL2 connection	T_H of an AAL2 connection
no priority (FIFO)	Case 1	Case 2
priority for setup phase	Case 3	Case 4
priority for release phase	n.a.	n.a.

I have not investigated the case when priority for messages in the release phase is given, because there was no argument in practice for such scenario.

According to the equations (20) and (21), the latencies in a homogeneous network are proportional to the delays in one switch, therefore the following investigations will refer to just one isolated switch. The reference point (*EQUILIBRIUM state*) is not the worst case situation, but it was obvious to select for the starting point a state with uniformly distributed resources among the sub-processes.

During my investigations, I have found the following properties (this properties are independent of the total capacity C of the processor):

I have found that prioritised handling of the signalling messages ERQ (≡SETUP) and ECF (≡CONNECT) results in a noticeable 8-12% decrease of the call establishment time T_C **(in any state!)** *compared to the FIFO case. However, at the same time, an increase in the call release time T_R of 9-11% can be also observed.*

Based on the above results, I have concluded that the bandwidth utilisation on the outgoing link cannot be increased by introducing priority mechanism for the setup phase, as the overall handling time T_H remains constant.

Furthermore, I have shown that applying the algorithm QUE/MU for the Case 1 to Case 4, leads to different optimum points, but in all these points the overall handling time T_H is at least 15% less than in the EQUILIBRIUM state.

The above statements are illustrated in the next section through a case study implementing all four cases listed in Table 7.3.

7.5.2 Numerical results. Case study.

Figure 7.5 illustrates all three parameters T_C , T_R and T_H , when starting from the *EQUILIBRIUM* state, and optimising a processor with a capacity of *C*=166 instructions/msec by implementing Case 1, Case 2, Case 3 and Case 4 listed in Table 7.3 (e.g., Case 3 = 'min Tc only, S&C hiprior', i.e., *SETUP* and *CONNECT* messages are given higher priority, while minimising the call establishment time).

Having a detailed look at our case study, one can see that applying the algorithm QUE/MU to all four cases, the call establishment time T_C is reduced by 10% if no priority was applied (Case 1&2), T_C is reduced by 6% in the Case 3 and by only 3% in the Case 4 (see Figure 7.5*a*).

Furthermore, in the Case 3 we obtained the maximum gain in reducing the call establishment time T_C (see Figure 7.5*a*), but the minimum gain in reducing the overall handling time T_H (see Figure 7.5*c*). The decrease of T_C by introducing priorities is more than what could be achieved by applying the algorithm in any case.

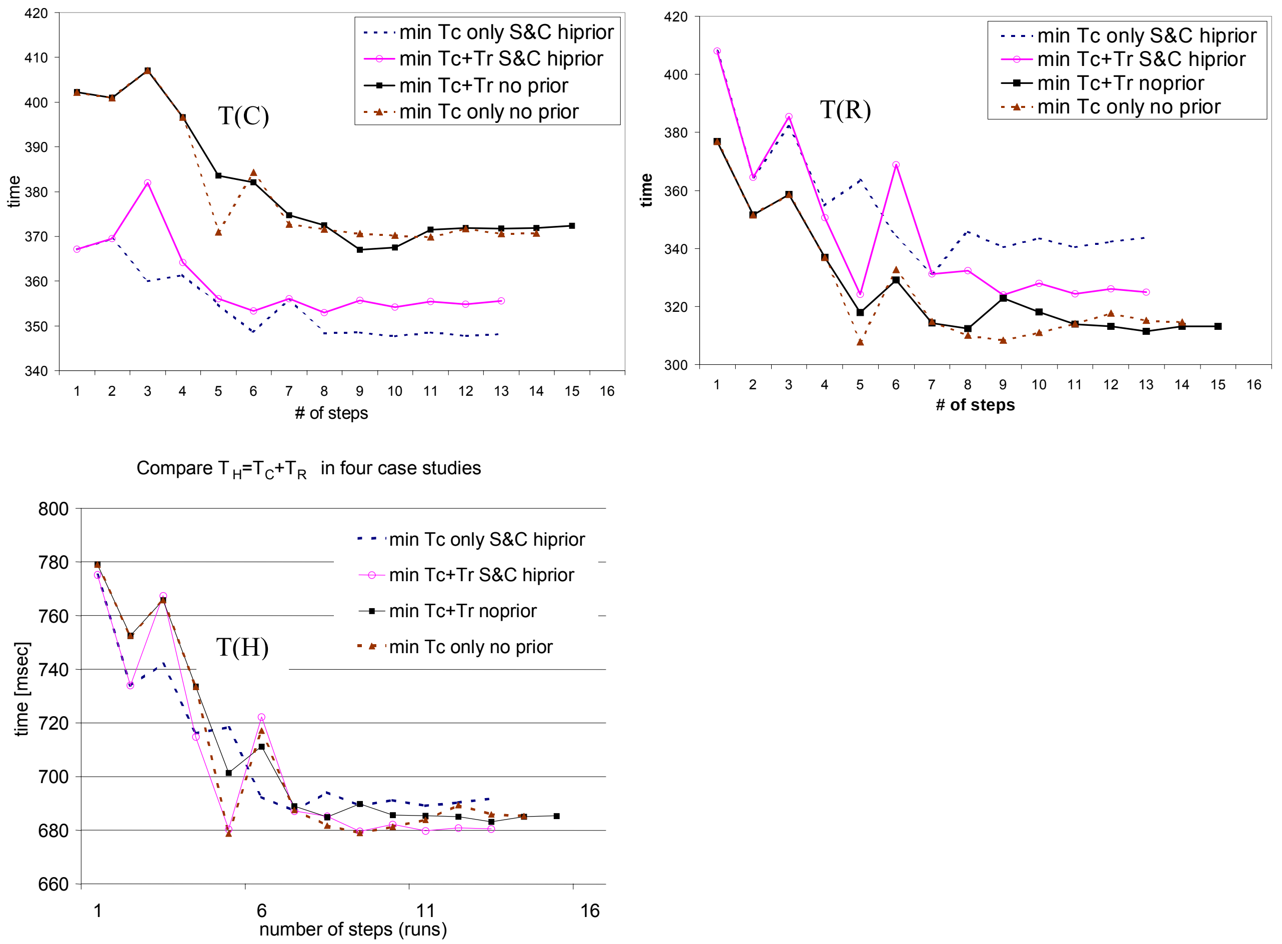


Figure 7.5 Case studies 1 to 4, using the algo QUE/MU a) Effect on the T_C ; b) Effect on the T_R ; c) Effect on the T_H

The call release time T_R is decreased by 25–32% in all cases compared to the *EQUILIBRIUM* state (but e.g., the gain in the Case 3 is with 7% less than in the Case 4). The call release time T_R is significantly decreased also in those cases when the algorithm was aiming to reduce the time needed for the connection establishment. This is due to the fact that the setup and release messages partly travel along the same path within the processor (see Figure 5.2).

Figure 7.5*c* clearly shows that the overall handling time T_H does not change when introducing priorities compared to the values of the *FIFO* case ($\Delta T_H \approx$1-3%, $\forall i$=1,...,15).

7.6 Validation of the results

We have tested both algorithms starting from different initial states (extreme states as well), in all cases the three parameters (T_C , T_R and T_H) converged to the same optimum values, only the convergence speed varied (from 10 to 20 steps). The algorithm converges usually in 13-15 steps, in worst case (starting from *EXTREME_x* values) this could go up to 20 steps. Also when trying for different capacities *C*, the ratios in the final distribution among the processes remained the same.

A measure of the efficiency of the algorithms is the ratio between the number of runs and the number of correct steps (*n*). In both investigated cases this is approximately equal to 2 (see e.g., Table 7.1 and Table 7.2). The convergence speed of the algorithms could be probably further improved by a more adequate selection of decreasing the $\Delta\mu$, but this was out of the scope of my investigation.

The algorithm QUE should not be applied for the case when certain messages get higher priority, because it will not necessarily deliver the global optimum. Instead, the algorithm QUE/MU converges always to the same values, regardless of *FIFO* queueing or priority handling. I haven't found any similar paper in the literature to compare these algorithms to. [Wu97] presented a model for the setup phase only and minimised it using the lagrangean method, however it did not presented any numerical results of this optimisation method. In addition, we have carried out network level simulations with the optimised processor architecture obtained in the Case 3 and Case 4, for the UTRAN network topology presented in Figure 7.1 (for both flat and tree topologies), but these results are out of the scope of my dissertation (for details, please check [C-7]).

7.7 Conclusions

In Section 7.1 we have seen that the ATM call processing model from Section 4.1 can be applied to model an AAL2 signalling node in UMTS networks with a certain re-design, then we have investigated two algorithms which minimise the call establishment times of soft handoffs. The difference in these two algorithms is only one decision criterion which makes in turn the second algorithm (QUE/MU) to be applicable to priority queueing solutions as well. It should be mentioned that the priority mechanism we discussed in Section 7.4 is different from the priority procedures presented in Figure 2.4 (ATMF, CPP1.0) and discussed in Section 2.3.1, which allocate higher priority to all messages of a certain call against other calls.

Section 7.5 investigated the impact on the overall handling time of the optimisation algorithm QUE/MU and that of the application of high priority service of setup messages against release messages, respectively.

The importance of the statements in Section 7.5 is to show that the application of a priority mechanism for setup messages reduces the call establishment time and thus advantageous for handling soft handoff procedures, but cannot save bandwidth on the outgoing link, because of the increased release time. At the same time it is pointed out that applying the algorithm QUE/MU the processor states can be optimised so that it will reduce the overall handling time significantly, thus reducing the bandwidth usage on the outgoing link.

My overall recommendation is to use both mechanisms (introducing priority handling and applying the algorithm QUE/MU) in the UTRAN network design.

CHAPTER 8

8 Application of the Blocked Call Queueing (*BCQ*) mechanism to wide-band calls in ATM networks

8.1 Introduction

8.1.1 Traditional loss networks

Since ATM is the switching and multiplexing technology of broadband networks, it is essential for its signalling system that it supports the co-existence of narrow- and wide-band services. Broadband networks have been traditionally modelled on the call scale by the theory of multi-rate loss networks [Ros95]. A call request requiring a certain amount of bandwidth between a given originating - destination pair is blocked and disappears from the system if sufficient resources are not available at the time the call request arrives to the network, see e.g. [Rit94] and [Chu93]. Adopting this rule to a multi-rate environment with a large difference in bandwidth requirements between traffic classes implies that calls requiring a large amount of bandwidth will experience a much higher blocking probability than calls requiring only a small amount of bandwidth [Ber96]. By applying either trunk reservation or class limitation it is possible to level out the blocking probabilities. However, in most cases, the disadvantage put on the narrow-band traffic is much bigger than the advantage obtained for the wide-band traffic, because network utilisation is inherently low at multi-rate pure loss networks, when request sizes may differ with orders of magnitude, as it has been shown e.g., in [Syk91] and [Bla96].

8.1.2 Mixed queueing and loss networks

Since with an advanced signalling protocol it may be possible to allow calls requiring a large amount of bandwidth to wait in a queue until resources become available, there is a hope to significantly reduce the blocking probabilities of these calls. These types of systems constitute an important generalisation of the pure loss systems and have already been studied in the literature as "mixed delay and loss" or "mixed queueing and loss" systems. In particular, numerical examples indicate that per-class blocking probabilities of wide band services decrease at the expense of a short delay during call set up and a slight increase of narrow-band service class blocking [Bla96]. It is concluded that letting wide-band calls to queue can decrease wide-band blocking and increase network revenue and thus advantageous for both users and network operators [C-2], [Fod99]. In fact, these papers do not take into account the effect of signalling message flows and secondly, they look at a rather small network only (4 node fully connected network).

8.1.3 Queueing of blocked wide-band calls at the access nodes (BCQ)

The queueing of wide-band calls has been found as being efficient at the access node of the network, but it requires enhancement of the Call Admission Control (CAC) function to queue the unsuccessful wide-band calls. In the case that a wide-band call attempt arrives to the access node of the network, the CAC function tests whether the network resources are available and proceeds either with resource allocation or the unsuccessful call joins the waiting queue. In Figure 8.1 a general case is presented with all the possible decisions made by the CAC function. The queueing functionality is added here as a new possible solution.

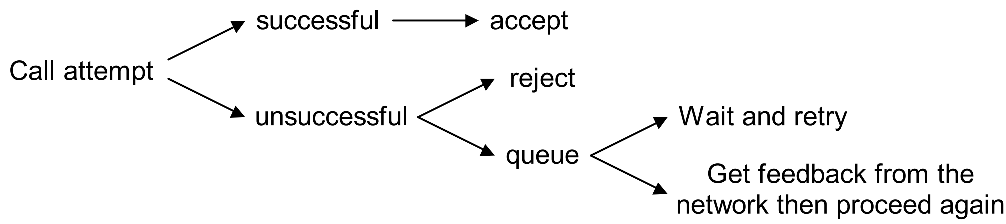


Figure 8.1 Possible outcomes for a wide-band call attempt at the access node

Once a wide-band call has been queued, it is not trivial to determine which mechanism is better: to retry after *'d_1'* delay (according to the mean holding time of call type *'t'*: $w_t=E(MHT_t)$), go to the end of the processing queue again or get feedback from the network first, then proceed again. We are mainly interested in the signalling overhead it generates on intermediate and access nodes and particularly its effect on the call establishment time. We assume here that the signalling channel is never congested (but it may create overload on the signalling CPU's and suffer long delays).

As a solution for Blocked Call Queueing (BCQ) we have been using the existing UNI and PNNI protocols (UNI4.0, PNNI1.1).

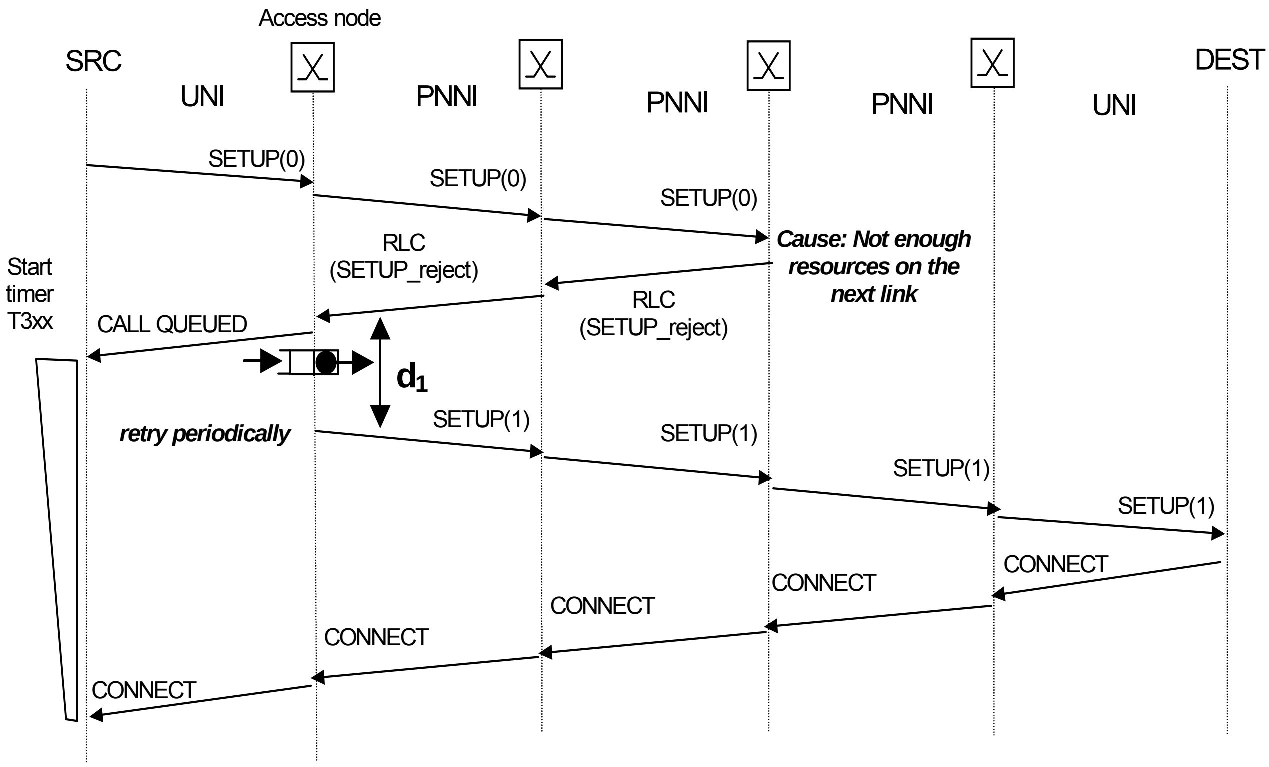


Figure 8.2 Successful call establishment with Blocked Call Queueing (simplified)

8.1.4 A need to extend the signalling standards

The Blocked Call Queueing mechanism holds some blocked calls by storing their signalling information in a separate buffer at the access ATM switch. These calls will be later connected when network resources become available. My first contribution was to investigate the possibility of supporting this BCQ mechanism by using signalling protocols (i.e., proposal for extension of existing protocols). The signalling overhead associated with wide-band *BCQ* and questions related to grade of service are also in focus of my investigations. I was interested only in a part of the repeated call attempts, when rejection of wide-band calls is caused by the network (neglecting those refused by the called party). According to my proposal, in such a case there is no need for any action by the calling party. A simplified example is given in Figure 8.2, when a call is blocked at the first attempt somewhere in the network because of unavailable resources on a link, rejected, then queued at the access node, waiting for a certain time and finally succeeding at the second trial. The *CALL QUEUED* message is not part of the standard UNI specification [Q2931], but an extension of that and it is described in Figure 8.2 and Figure 8.3.

8.2 Extension of the Finite State Machine graph with the BCQ mechanism

I have shown that it is possible to apply the Blocked Call Queueing (BCQ) mechanism by extension of the current signalling protocols [Q2931], [UNI40]. I have introduced a new state (U), a new message (CALL QUEUED) and a new timer (T_{3xx}) to implement the BCQ mechanism (see Figure 8.3).*

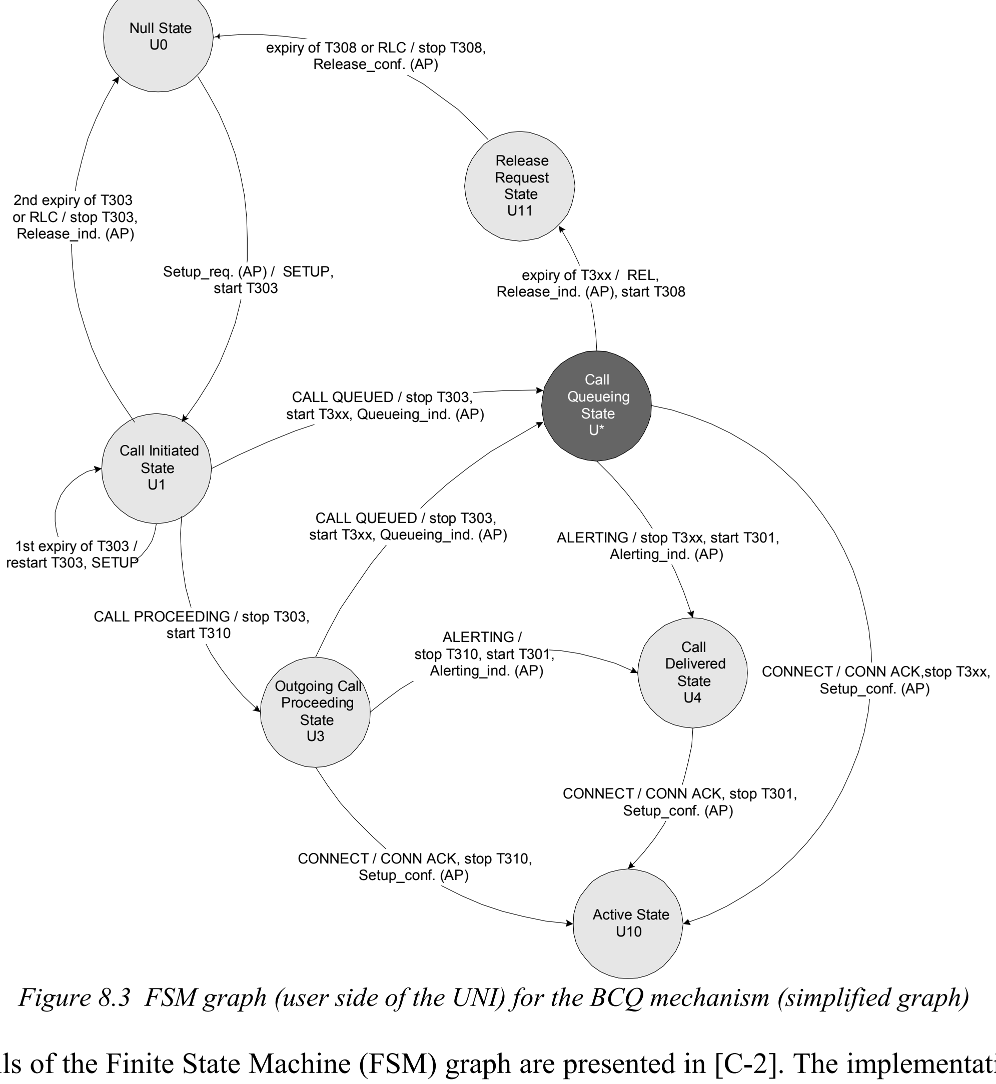


Figure 8.3 FSM graph (user side of the UNI) for the BCQ mechanism (simplified graph)

Further details of the Finite State Machine (FSM) graph are presented in [C-2]. The implementation of wide-band *BCQ* mechanism is quite simple, no complexity arises when extending the flow charts based on [Q2931], [UNI40]. Five original states are affected by introducing a new state for BCQ. All the new elements are represented in Figure 8.3. I have extended our simulation model with this *BCQ* mechanism for the access nodes and activated for the blocked wide-band calls. While in the previous chapters I have supposed that all call attempts are successful, in this chapter I have analysed the bandwidth requirements of calls, and focused especially on the blocked wide-band calls.

8.3 Performance evaluation of the BCQ mechanism

In this section I have tried to find an analytical solution to the call establishment time of the blocked and retransmitted wide-band calls. The study of the call release time is out of interest, as this parameter is identical to the non-blocked calls presented in the previous chapters.

After a blocked call has been queued, to indicate the best algorithm for retransmitting the *SETUP* message has been subject of our investigations in [C-3]. Hence, we assume that these messages are retransmitted within a *'d_0'* delay as many times as necessary until success, i.e., a positive acknowledgement (*CONNECT* message) arrives back to the source or until expiry of the timer T_{3xx}.

I have given an analytical solution for the lower and upper bounds of call establishment time of the blocked and retransmitted wide-band calls (T_C^{BCQ}):

$$ET_C^0(i,j) \le ET_C^{BCQ}(i,j) \le ET_C^0(i,j) + T_{3xx} \tag{29}$$

where: $ET_C^0(i,j)$ is the expected call establishment time of narrow-band calls along the main path between nodes *i* and *j*. The new timer (T_{3xx}) for the blocked calls has to be shorter than the time defined for the grade of service parameter, $T_{3xx}<T_{GoS}$, but long enough to allow a couple of retrials.

Suppose that *E(N)* is the expected number of retrials and d_n is the expected time spent in the access queue at the n^{th} retrial.

$$N = \max\{n : t_1 + t_2 + ... + t_n < T_{3xx}\},$$

where: $t_1 = d_1 + T_C^1(i,j); t_2 = d_2 + T_C^2(i,j); ...; t_n = d_n + T_C^n(i,j)$,
and $T_C^k(i,j)$ is the average call establishment time of a narrow-band call between nodes *i* and *j* along the alternative path $\pi_k(i,j)$, *k*=1,...,*n*.

According to Lorden's Theorem [Lord70], we can find lower and upper bounds for *E(N)*:

$$\frac{T_{3xx}}{E\left[T_C^0(i,j)+d_0\right]} - 1 \le E(N) \le \frac{T_{3xx}}{E\left[T_C^0(i,j)+d_0\right]} + \frac{E\left[T_C^0(i,j)+d_0\right]^2}{\left[E\left[T_C^0(i,j)+d_0\right]\right]^2}$$

where $T_C^0(i,j)$ is the average call establishment time of a narrow-band call between nodes *i* and *j* along the main path $\pi_0(i,j)$ and d_0 is the expected time spent in the access queue at any retrial (using BCQ).

Furthermore, conditional on N=n, I have found an approximation formula for the expected call establishment time when using wide-band BCQ mechanism:

$$ET_C^{BCQ}(i,j) \approx (1-p_0^{n+1}) \cdot T_C^0(i,j) + \frac{p_0}{1-p_0} \cdot (1-p_0^n) \cdot d_0, \tag{30}$$

where p_0 is the call blocking probability of wide-band calls along the main path $\pi_0(i,j)$ between nodes *i* and *j*, $T_C^0(i,j)$ is the expected call establishment time of narrow-band calls between nodes *i* and *j* along the main path $\pi_0(i,j)$, and d_0 is the waiting time between two trials at the access node after a wide-band call being rejected.

Proof:

The expected value of the call establishment time for blocked and retransmitted wide-band calls is given by (e.g., see Figure 8.2),

$$ET_C^{BCQ}(i,j) = (1-p_0)\cdot T_C^0(i,j) + p_0\cdot d_1 + p_0\cdot(1-p_1)\cdot T_C^1(i,j) + p_0\cdot p_1\cdot d_2 +$$
$$+ p_0\cdot p_1\cdot(1-p_2)\cdot T_C^2(i,j) + \ldots + p_0\cdot p_1\cdot\ldots\cdot p_{n-1}\cdot d_n + p_0\cdot p_1\cdot\ldots\cdot p_{n-1}\cdot(1-p_n)\cdot T_C^n(i,j)$$

where: p_0 = P[unsuccessful call establishment on the main path between nodes *i* and *j*];

p_n = P[unsuccessful call establishment on the alternative path '*n*' between nodes *i* and *j*, if the main path is blocked];

d_n is the waiting time at the n^{th} retrial at the access node after a wide-band call is being rejected.

For simplifications, let us consider:

$$T_C^0(i,j) = T_C^1(i,j) = \ldots = T_C^n(i,j)$$
$$p_0 = p_1 = \ldots = p_n$$
$$d_1 = d_2 = \ldots = d_n = d_0$$

Then it results,

$$ET_C^{BCQ}(i,j) \approx \left[(1-p_0) + p_0\cdot(1-p_0) + p_0^2\cdot(1-p_0) + \ldots + p_0^n\cdot(1-p_0)\right]\cdot T_C^0(i,j) +$$
$$+ (p_0 + p_0^2 + \ldots + p_0^n)\cdot d_0 \,.$$

and soon we arrive to equation (30):

$$ET_C^{BCQ}(i,j) \approx (1-p_0^{n+1})\cdot T_C^0(i,j) + \frac{p_0}{1-p_0}\cdot(1-p_0^n)\cdot d_0 \,.$$

The $T_C^0(i,j)$ depends on the processing capacity of the signalling processor on each switch, on the signalling traffic load in the network, on the number of switches in the path, on the call complexity, etc., as already shown in Chapter 4.

Further on, if p_{kZ} is defined as the call blocking probability on the link *k* directed from node *Z*, and $\pi_0(i,j)$ is the main path between nodes *i* and *j*, then:

$$p_0 = 1 - \prod_{k\in\pi_0(i,j)} (1-p_{kZ}) \,.$$

To find the probabilities p_{kZ} for all nodes is a non-trivial problem. If all links in the backbone and external network have the same call blocking probability 'p_t' for a wide-band call of type *'t'*, and $\overline{L}(i,j)$ is the average length of the path in the network (computed in equation (24)), then:

$$p_0 = 1 - (1-p_t)^{\overline{L}(i,j)} \,.$$

The average number of calls on the link *k* directed from node *Z* is:

$$E_{kZ}(m) = \rho_{kZ}(1-p_{kZ}) = (1-p_t)\cdot\rho_{kZ} \,.$$

Further on, in a homogeneous network,

$$T_C^0(i,j) = 2 \cdot \sum_{k \in \pi_0(i,j)} E_{kZ}(m) \cdot E_k(T) = \frac{2}{\mu^C} \cdot \sum_{k \in \pi_0(i,j)} E_{kZ}(m), \tag{31}$$

$$T_C^0(i,j) = \frac{2}{\mu^C} \cdot (1 - p_t) \cdot \sum_{k \in \pi_0(i,j)} \rho_{kZ}$$

while, similarly (supposing that after blocking, the rejected call enters again the end of the queue in the access node after a waiting time w_t equal to the mean holding time of call type *'t'*: $w_t = E(MHT_t)$, procedure described in section 8.1.3):

$$d_0 = w_t + \frac{1}{\mu^C} \cdot \rho_A \cdot (1 - p_t) \tag{32}$$

where: $\rho_A = \frac{\lambda_A}{\mu^C}$ is the utilisation of one access node.

Finally, after some certain steps we achieve:

$$ET_C^{BCQ}(i,j) \approx (1 - p_0^{n+1}) \cdot \frac{2}{\mu^C} \cdot (1 - p_t) \cdot \sum_{k \in \pi_0(i,j)} \rho_{kZ} + \frac{p_0}{1 - p_0} \cdot (1 - p_0^n) \cdot \left[w_t + \frac{1}{\mu^C} \cdot \rho_A \cdot (1 - p_t) \right], \tag{33}$$

where: $0 < p_t < 1$ is the average link blocking probability of a wide-band call of type '*t*';

μ^C is the equivalent service time of one node in a homogeneous network;

and w_t is the waiting time equal to the mean holding time of call type *'t'*.

In particular, in a cascaded network of '*r*' nodes (which is always an overestimate of an arbitrary network in terms of T_C, as shown in Section 6.2), all nodes have the same utilisation: $\rho_{kZ} = \rho_A = \rho$, and $p_0 = 1 - (1 - p_t)^{\bar{L}(i,j)} = 1 - (1 - p_t)^r$, thus equation (33) will get a bit simplified form:

$$ET_C^{BCQ}(i,j) \approx (1 - p_0^{n+1}) \cdot \frac{2}{\mu^C} \cdot (1 - p_t) \cdot r \cdot \rho + \frac{p_0}{1 - p_0} \cdot (1 - p_0^n) \cdot \left[w_t + \frac{1}{\mu^C} \cdot \rho \cdot (1 - p_t) \right]$$

8.4 Numerical results

As a case study, I have selected the same 35-node network topology already discussed in Chapter 6. In the example given in Figure 6.7 we have already obtained the utilisation ρ_{kZ} of each node along the longest path (as shown in Figure 8.4, r = 8). Let us select the following input values: $1/\mu^C = 10$ msec (*FORE ASX200BX*); N=1,2,3; $0.01 < p_t < 0.05 \Rightarrow 0.07 < p_0 < 0.33$. The obtained results are shown in Table 8.1.

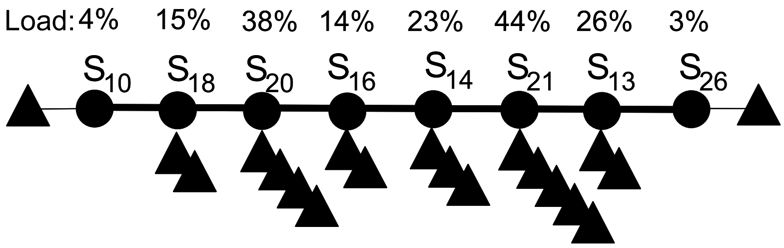


Figure 8.4 Translation of the 35-node network into a cascade, r=Φ=8

Table 8.1 Estimated call establishment times of blocked wide-band calls with activated BCQ mechanism

$w_t = 1\sec$			$w_t = 10\sec$			$w_t = 30\sec$		
N=n	p_t	Estimated $T_C^{BCQ}(r)$ [sec]	*N=n*	p_t	Estimated $T_C^{BCQ}(r)$ [sec]	*N=n*	p_t	Estimated $T_C^{BCQ}(r)$ [sec]
1	0.01	**0.109**	1	0.01	**0.805**	1	0.01	**2.350**
1	0.02	**0.181**	1	0.02	**1.524**	1	0.02	**4.508**
1	0.03	**0.246**	1	0.03	**2.193**	1	0.03	**6.518**
1	0.04	**0.308**	1	0.04	**2.815**	1	0.04	**8.387**
1	0.05	**0.364**	1	0.05	**3.393**	1	0.05	**10.125**
2	0.01	**0.116**	2	0.01	**0.865**	2	0.01	**2.529**
2	0.02	**0.203**	2	0.02	**1.747**	2	0.02	**5.177**
2	0.03	**0.294**	2	0.03	**2.662**	2	0.03	**7.922**
2	0.04	**0.387**	2	0.04	**3.593**	2	0.04	**10.718**
2	0.05	**0.480**	2	0.05	**4.529**	2	0.05	**13.526**
3	0.01	**0.116**	3	0.01	**0.869**	3	0.01	**2.543**
3	0.02	**0.207**	3	0.02	**1.780**	3	0.02	**5.277**
3	0.03	**0.305**	3	0.03	**2.763**	3	0.03	**8.226**
3	0.04	**0.409**	3	0.04	**3.810**	3	0.04	**11.367**
3	0.05	**0.519**	3	0.05	**4.911**	3	0.05	**14.671**

Finally, I have investigated the range of usability of the *BCQ* mechanism by simulation in different network topologies and network scenarios.

I have concluded that the wide-band BCQ mechanism is most beneficial when $0.5 \le \rho_{kZ} \le 0.67$ *and the percentage of wide-band calls in the network is higher than 10%. When* $\rho_{kZ} < 0.5$ *the BCQ is rarely utilised, while for* $\rho_{kZ} > 0.67$ *the BCQ is not effective (*T_C^{BCQ} *>>,* p_0*>>).*

8.5 Validation of the results

To validate my analytical results, I have conducted simulation studies to obtain the call establishment time of narrow-band and wide-band calls, call blocking probabilities (p_0) and queue lengths for two different scenarios of call mixture in a number of network topologies using the wide-band *BCQ* mechanism. These case studies are described in details in [C-4]. An example is shown in Figure 8.5.

I have shown that using wide-band *BCQ*, the blocking probability p_0^{WB} is significantly reduced (approx. 50%) for wide-band calls (2-10Mbps) at the expense of a small increase (max. 20%) for narrow-band calls (0.1Mbps) in a large network configuration. Thus an increase in network revenue is obtained, which is advantageous for both users and network operators. The dependency on the utilisation of the access nodes, call distribution and mean holding time is also shown in [C-4]. Furthermore, I have shown that the average call establishment time (T_C^{WB}) is not significantly longer (<10%) in case of *BCQ*, compared to the case without using the queueing mechanism for $p_0 \le 0.05$, independent of the lengths of w_t. The applicability of the formula in the equation (33) is limited by the fact that it is difficult to obtain the probabilities p_{kZ} on the links.

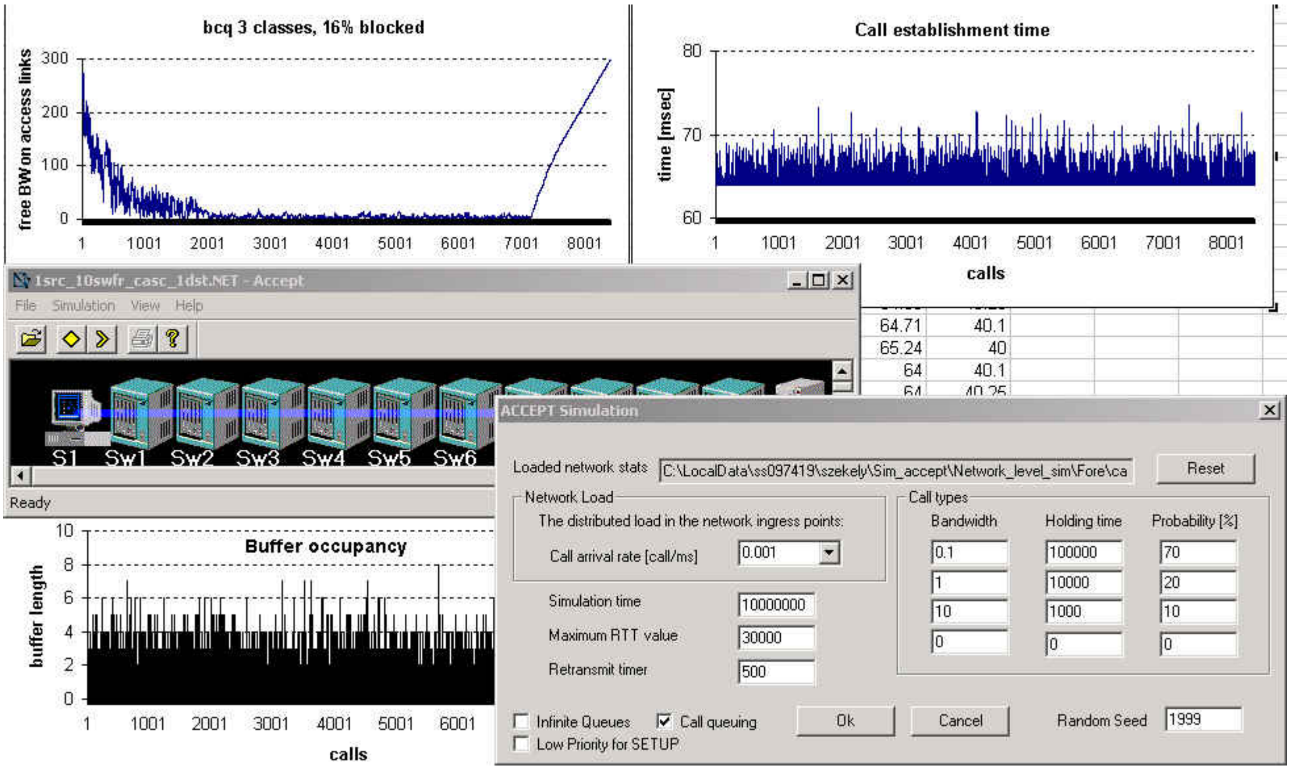


Figure 8.5 An example of simulating the effect of the BCQ mechanism in a cascaded network

8.5 Conclusions

In this chapter, first I have presented an extension to the standard signalling protocols in order to support the retransmission of blocked wide-band calls at the access nodes of the network. This is a so called Blocked Call Queueing (BCQ) mechanism. Next, in Section 8.2 an analytical study has been given for obtaining the call establishment times of these blocked wide-band calls using the presented BCQ mechanism. Due to the complexity of the problem, we have introduced a lot of approximations in the formula, first of all for the link blocking probabilities and node utilisations. Then some numerical results have been presented. I have carried out both *analytical studies* and *simulations* to validate my results.

Finally, I have investigated the range of usability of the *BCQ* mechanism and I have concluded that the wide-band BCQ mechanism is most beneficial when the utilisation of the majority of nodes in the network is close to 0.5 and the presence of wide-band calls in the network is higher than 10%.

CHAPTER 9

9 Concluding Remarks

This dissertation covered various fields of performance evaluation of broadband signalling network. It was particularly devoted to a very unpopular research of performance measurements of ATM signalling, that has taken several years, further on evaluated of the signalling load in arbitrary network topologies, investigated the optimisation of signalling processor's architecture and enhancement of current signalling protocols.

9.1 Summary of the dissertation

In this dissertation we have carried out the following studies:

- we have measured the message delays in the signalling processor on four commercial ATM switches and identified the most significant parameters that influence the message latencies through the switches and thus the call establishment times and release latencies;
- we have analysed the complexity of calls and identified new intrinsic properties of point-to-point ATM signalling (3 out of the 11 presented properties were only known before);
- we have developed and analysed a new way of representing the signalling flow, in a so called 'population-diagram', used first of all to analyse burst arrivals of signalling messages;
- we have developed and analysed a new call model that reflects accurately all the ATM specific aspects of calls, both FIFO and priority queues have been investigated;
- we have compared the performance of cascaded nodes vs. arbitrary network topology, defined some new performance metrics and analysed the signalling load distribution in the network;
- we have developed two simulation-assisted optimisation algorithms for minimising the signalling queues and call establishment times;
- we have developed and analysed an architectural extension of the signalling model in order to enable queueing of wide-band blocked calls in access nodes.

9.2 Applications of the new results

The measurement results presented in Chapter 4 can be widely used by network operators to design their ATM signalling network. With the rapid expansion of the xDSL applications today (e.g., in Europe, USA and South Korea) between residential customers and business users (SMEs), there is more and more obvious to introduce signalling in the existing ATM networks. The today's applications are either PPPoA or PVCs. One of the driving factors is that full provisioning with PVCs over a single STM-1 path (155Mbps) from the DSL Access Multiplexer (DSLAM) to the ATM switch is not possible for more than 200 customer lines (with Integrated Access Devices, IADs) each having 0.66Mbps CBR channel for its voice traffic (8x64kbps channels) and 0.1Mbps nrt-VBR channel for its inband management. The bandwidth needs of the data channels (UBR) are not even considered here. In such a case the most economical solution is the introduction of ATM signalling, because there is no need for hardware change. When introducing signalling in an existing networks, the presented intrinsic properties play a significant role in the design of these networks.

The method and the "population-diagram" described in Section 4.2.1 has been widely used by the author and his colleagues at the 'System Test Department' at Siemens AG in their every-day work by testing Voice over DSL, Voice over IP (MGCP, H.323, SIP) and virtual trunking applications (SS7→ MGCP (or SIP) → SS7) in the new generation "SURPASS" network (in this last scenario, the diagram will be split into 3 parts).

Results of the Chapter 6 can be used first of all by service providers in network design and in network maintenance, i.e., to determine the impact of a failure of a node in a live-network on the signalling load of adjacent nodes, without being necessary to mirror the live-network in laboratory conditions, but using simulation. It is of particular importance, when cross-domain problems occur due to interconnected networks of different service providers.

The model described in Section 5.1 has been adjusted in Section 7.1 to be used for network design in the radio access network of UMTS systems. These investigations are important, as I have shown that our optimised AAL2 switch model achieves the fastest possible connection setup, which is the most important requirement when it comes to supporting soft handoff. This model can be a basis for AAL2 switch implementations.

Another application of ATM signalling is created by the flexible bandwidth allocation on demand for efficient Video Contribution Services between TV-Studios and Broadcast Centers. The applicability of Section 8.2 is quite obvious here for blocked wide-band calls.

CHAPTER 10

Terminology

Accepted SETUP Rate
The average SETUP message rate as observed on the interface. This rate may differ from the offered call establishment rate due to the link layer flow control mechanism. The SUT may cause the Tester to stop transmitting temporarily during the POLL-STAT procedures defined in the SAAL layer. This blocking can influence the rate at which messages are observed on the physical interface (see Figure 3.2).
Note: the rate at which SETUP messages are seen on the interface is based on the link layer (flow control) protocol between the Tester and the SUT.

Average Signalling Message Rate
The rate at which signalling messages are transmitted or observed on an interface is of great importance to the measurement of a system's performance. The signalling message rate is observed in one direction on a particular signalling interface for a single signalling message type, and it is based on the average time between individual message frames. Message rates can be reported for any signalling message type, such as SETUP and CONNECT messages. It is important to note the difference between the offered and accepted rates for SETUP messages.

Blocked Call Queueing
A mechanism to enable the blocked wide-band calls in the network to be reinserted in a separate queue at the access points of the network, and later retransmitted, without the user being invited to take any action. Thus it is aimed to reduce the blocking probability of wide-band calls in the network.

Call Cycle Time
The amount of time that it takes for an ATM system to establish a call and then immediately release all resources allocated to the call on a signalling interface is the call cycle time.
Note: This definition is similar to that of Overall Handling Time, except the second term (see Def 4.2 in Section 4.1.).

Call Cycle Burst Time
Call cycle times are used to measure how long it takes to establish individual switched virtual connections and tear each one down as soon as it is established. The call cycle burst time measurement yields the time that it takes to cycle a burst of connections on an ATM interface. A tester cannot simply take the call cycle time for a single connection, and conclude that it would take twice as long for two connections with the same parameters. This extrapolation does not take into account the overlapping of signalling messages.

Call Arrival Rate
The rate at which calls are generated or observed on an interface is of great importance to the measurement of a system's performance. Usually, the call arrival rate can be identified with the arrival rate of SETUP messages.

Call Duration
The length of time that the call remains in the active state at the originating side.
Note: this definition is equivalent to the Call Holding Time.

Call Establishment Burst Time
Call establishment times are used to measure how long that it takes to establish individual switched virtual connections. The call establishment burst time measurement yields the time that it takes to establish a burst of connections on an

ATM interface. A tester cannot simply take the call establishment time for a single connection, and conclude that it would take twice as long for two connections with the same parameters. This extrapolation does not take into account the overlapping of signalling messages.

Call Establishment Latency
The total time taken by the network to establish a connection from source to destination is the call establishment latency. For short duration VCs, call establishment latency is an important part of the user perceived performance.

Call Establishment Time
The amount of time that it takes for an ATM system to establish a switched virtual connection between network components is a fundamental signalling performance metric. The larger the call establishment time, the fewer calls can be established during any fixed period.

Call Release Burst Time
Call release burst times yield the amount of time taken to release a set of calls in a burst on a user-network interface. A tester cannot simply take the call release time for a single connection, and conclude that it would take twice as long for two calls with the same parameters. This extrapolation does not take into account the overlapping of signalling messages.

Call Release Latency
The total time taken by the network to release all resources allocated to an active connection from source to destination is the call release latency. The acknowledgement by the destination is also included in this parameter (see Figure 3.3). For short duration VCs, call establishment latency is an important part of the user perceived performance.

Call Release Time
The amount of time that it takes for an ATM system to release all resources allocated to an active call on a signalling interface is the call release time. The larger the call release time, the longer it takes for a network to reclaim resources on an interface. In a situation where the network is under extreme load, this could result in delays or failures in establishing new connections. This may result in ambiguous results being reported to the user.

Call profile
A particular set of Information Elements.

Call Throughput
By definition, the throughput is $\gamma = \lambda \cdot (1 - P_B)$, where λ is the arrival rate of calls, P_B is the blocking probability of calls. The probability of successful calls (relative throughput) is: $\gamma/\lambda = 1 - P_B$.

“chopped” burst
It is a signaling message burst concatenated due to the blocking caused by the flow control mechanism.

Complex Call profile
A particular set of Information Elements that contains more Information Elements than only the mandatory ones.

“complex” SETUP message
SETUP message containing optional elements in addition to the the mandatory Information Element.

CONNECT delay
The time a CONNECT message spends between an ingress and egress point of an ATM switch.

“default” SETUP message
SETUP message containing the mandatory Information Element only.

Equilibrium state
A reference point where all resources of the signaling processor are equally distributed, i.e. equal buffer sizes, equal service rates.

Flow Control of SSCOP
This function allows an SSCOP receiver to control the rate at which the peer SSCOP transmitter entity may send information.

Keep Alive of SSCOP

This function verifies that the two peer SSCOP entities participating in a connection are remaining in a link connection established state even in the case of a prolonged absence of data transfer.

Message Distribution
The variation of the message initiation rate over time. Two modes are available:

- Constant Rate – The time between SETUP messages is fixed
- Burst – The time between SETUP messages varies

Message Rate
The rate at which messages are generated or observed, calculated over a reported period of time, expressed in messages/second. On any given interface, message rates are measured one direction at a time.

Offered Call Establishment Rate
The user specified rate at which the tester initiates call connection requests. Note: a call might contain several connections simultaneously (in the simplest case a call is equivalent to a connection).

Offered SETUP Rate
The number of SETUP messages defined by the user (SETUPs/sec) and initiated by the Tester. E.g., this rate may differ from the accepted SETUP rate due the link layer flow control mechanism (see Figure 3.2)

RELEASE delay
The time a RELEASE message spends between an ingress and egress point of an ATM switch.

Sequence number (SN) of SSCOP
This parameter indicates the value of N(S) in the received SD PDU, and may be used to support the data retrieval operation.

SETUP delay
The time a SETUP message spends between an ingress and egress point of an ATM switch.

SETUP Rate
The message rate for the SETUP message.

Signalling Congestion
A condition characterized by insufficient control plane resources to process the offered signalling and call processing load in a timely manner. This includes signalling link transmission bandwidth, internal switch message queues, CPU realtime, memory resources, etc. Signalling congestion specifically does not address the issue of insufficient user plane resources to accept new calls.

Signalling Message Delay
The time a signaling message spends between an ingress and egress point of an ATM switch.

Signalling Message Latency
The time that it takes for a network to propagate signalling messages will impact the signaling performance of the network. The message latencies introduced by nodes in an ATM network are additive. This property of the measurement implies that as more nodes are traversed on a signalling message path, the signalling message latency increases. The latency measured across a small number of nodes could be used to predict the performance of a larger network of similar nodes.
Note: For one network node, this definition is equivalent to that of Signalling Message Delay (e.g., SETUP delay, CONNECT delay, RELEASE delay)

T303 timer
It is the retransmission timer in outgoing state of a SETUP message. After its expiry (e.g. 4 seconds) the SETUP message will be retransmitted if no corresponding CALL PROCCEDING message received until this time.

Abbreviations (Acronyms)

AAL ATM Adaptation Layer
AAL2 ATM Adaptation Layer Type 2
ABR Available Bit Rate
AESA ATM End System Address
AINI ATM Inter Network Interface
ANSI American National Standards Institute
ATM Asynchronous Transfer Mode

BCQ Blocked Call Queueing
BICC Bearer Independent Call Control
BICI Broadband Inter Carrier Interface
B-ISDN Broadband Integrated Services Digital Network
B-LLI Broadband low layer information
B-HLI Broadband high layer information
BS Buffer Size

CAR Call Arrival Rate
CBR Constant Bit Rate
CCP Complex Call Profile
CDVT Cell Delay Variation Tolerance
CES Circuit Emulation Service

DSL Digital Subscriber Line
DSLAM Digital Subscriber Line Access Multiplexer
DSS2 Digital Service Signalling System No.2.

ECF Establishment Confirm, equivalent with a *CONNECT* message
ERQ Establishment Request, equivalent with a *SETUP* message
ETSI European Telecommunications Standardisation Institute

FIFO First in First out
FSM Finite State Machine

GDC General DataComm

H.323 Packet-based multimedia communication systems (ITU-T Recommendation)

IAD Integrated Access Device
IAT Inter Arrival Time
ILMI Integrated Local Management Interface
IE Information Element
IP Internet Protocol
ISP Internet Service Provider
ITU-T International Telecommunication Union – Telecommunication Standardization Sector (formerly CCITT)

LAN Local Area Network
LANE LAN Emulation Service
LIJ Leaf Initiated Join

Mbps Megabit per second
MBS Maximum Burst Size
MGCP Media Gateway Control Protocol
MIMO Message in Message out

N-ISDN Narrowband Integrated Services Digital Network
NNI Network-to-Network Interface, also known as Network-Node Interface
NSAP Network Service Access Point

OAM Operations Administration and Maintenance

PCR Peak Cell Rate
PDU Protocol Data Unit
PNNI Private Network-to-Network Interface
PPPoA Point-to-Point Protocol over ATM
p-to-p Point-to-point
p-to-mp Point-to-multipoint
PVC Permanent Virtual Connection

QoS Quality of Service

RDLM Retransmission Delay of Lost Messages
RLC Release Complete Message
RSVP Resource Reservation Setup Protocol

SAAL Signalling ATM Adaptation Layer
SAP Service Access Point
SAR Segmentation and Reassembly
SCR Sustainable Cell Rate
SDU Service Data Units
SIG 4.1 UNI Signalling version 4.1
SIP Session Initiation Protocol
SS7 Signalling System Number 7
SSCF Service Specific Coordination Function
SSCOP Service Specific Connection Oriented Protocol
SSCS Service Specific Convergence Sublayer
STM-1 Syncronous Transfer Mode – 1 (=155Mbps rate)
SUT System Under Test
SVC Switched Virtual Connection

UBR Unspecified Bit Rate
UMTS Unified Message Transfer System
UNI User-Network Interface
UTRAN UMTS Terrestrial Radio Access Network

VBR Variable Bit Rate
VCC Virtual Channel Connection
VCI Virtual Channel Identifier
VPC Virtual Path Connection
VPI Virtual Path Identifier
VoDSL Voice over DSL
VoIP Voice over IP

WAN Wide Area Network

xDSL Digital Subscriber Line, x=A (Asymmetric), x=S (Symmetric), x=V (Very High Rate)

Appendix A: Example of capturing and analysing (decoding) a *SETUP* message

An example of capturing a successful call establishment process with the related timestamps for each message is shown below (global view, no details of the message structure):

```
HP Broadband Series Tester Capture Data Record

-----Source UNI:
21:20:38.28314260  CPP:3 LIF:4  Tx  UNI Sig.  SETUP  1922 orig
    Called Number : 47 00 05 80 FF E1 00 00 00 F2 1A 51 52
21:20:38.28697250  CPP:3 LIF:4  Rx  UNI Sig.  CALL PROCEEDING  1922 dest
    Connection Identifier: 0/478
21:20:38.29361780  CPP:3 LIF:4  Rx  UNI Sig.  CONNECT  1922 dest
    Connection Identifier: 0/478
21:20:38.29610540  CPP:3 LIF:4  Tx  UNI Sig.  CONNECT ACKNOWLEDGE  1922 orig

-----Destination UNI:
21:20:38.28659870  CPP:6 LIF:7  Rx  UNI Sig.  SETUP  437136 orig
    Called Number : 47 00 05 80 FF E1 00 00 00 F2 1A 51 52
    Connection Identifier: 0/120
21:20:38.29097120  CPP:6 LIF:7  Tx  UNI Sig.  CALL PROCEEDING  437136 dest
21:20:38.29207810  CPP:6 LIF:7  Tx  UNI Sig.  CONNECT  437136 dest
21:20:38.29383280  CPP:6 LIF:7  Rx  UNI Sig.  CONNECT ACKNOWLEDGE  437136 orig
```

Let's select the first *SETUP* message of this signalling flow and analyse it in more details. Its protocol data unit (PDU) structure at the user side of the UNI is shown in Figure A.1.

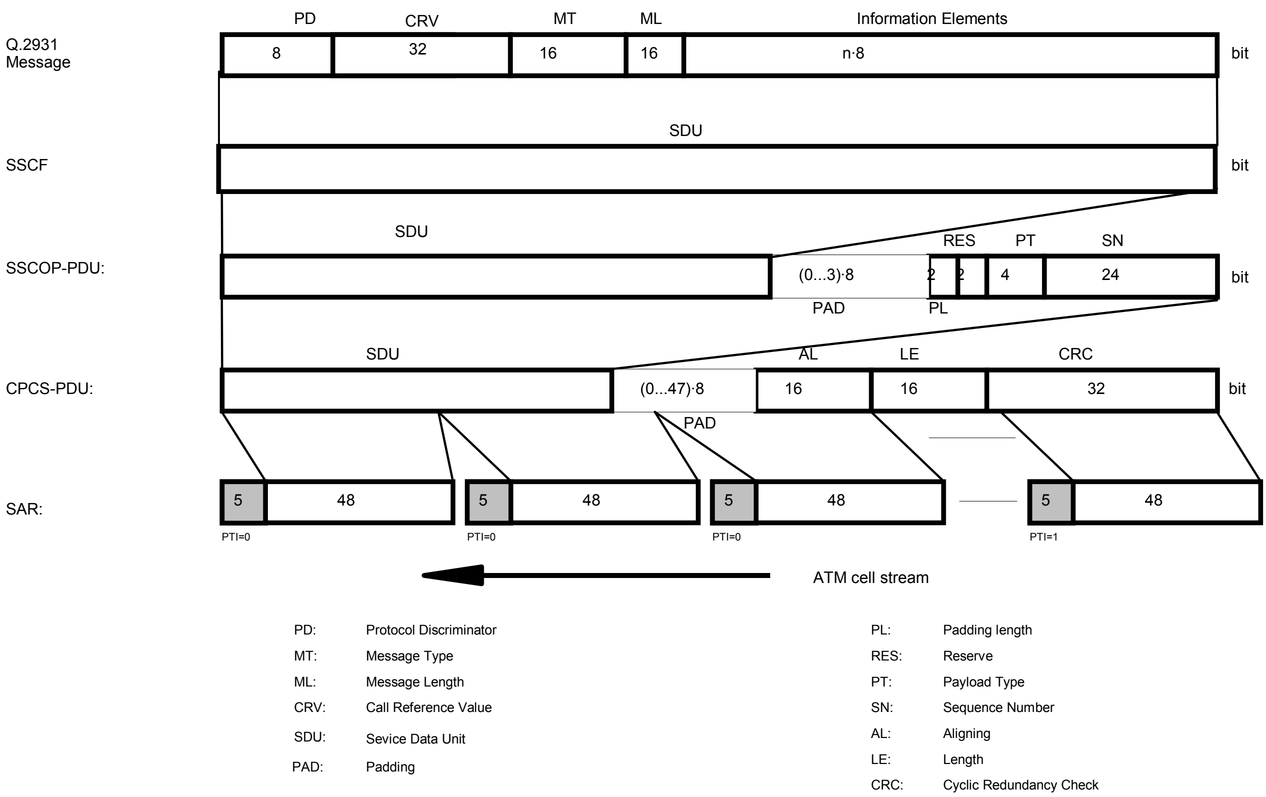


Figure A.1 Signalling message structure at a network node (NNI side)

Once the layer 2 connection is established, the *SETUP* message (i.e, 85 octets in our "default" case, containing the Mandatory Information Elements only and the Calling Party Number IE) may be sent to the SAAL layer using the AAL-DATA Request primitive. The SSCOP sublayer will attach 3 octets of padding for 32-bit alignment and 4 octets of trailer information (e.g., Payload Type and Sequence Number). The CPCS sublayer will then add 44 octets of padding and 8 octets of trailer (e.g., Length Indicator and Cyclic Redundancy Check). The CPCS PDU is now 144 octets in length, which will then be segmented into 3 SAR PDUs (payload for 3 ATM cells).

- At the ATM Layer, the payload of the cells which carry the *SETUP* message looks like the following (in hexa code):

Table A.1 Example of payload of ATM cells containing a SETUP message

1st cell

09	03	00	38	FA	05	00	00	7B	58	80	00	01	05	59	80
00	1A	82	00	04	D2	83	00	04	D2	88	00	00	7B	89	00
00	7B	A0	00	00	80	A1	00	00	0A	BF	00	5E	80	00	04

2nd cell

83	80	81	01	5D	80	00	01	80	63	80	00	01	82	70	80
00	09	82	00	00	00	20	37	58	13	01	71	80	00	06	80
00	05	00	00	55	6C	80	00	07	02	80	20	37	58	01	01

3rd cell

6D	80	00	06	80	00	05	00	00	55	5A	80	00	05	88	00
E4	00	20	5C	E0	00	02	02	02	62	80	00	01	A1	00	00
00	00	00	00	00	00	00	00	00	00	00	88	8E	AF	EF	AF
AAL5 Padding								Control		Length		CRC			

- According to the ITU-T Q.2931 protocol specification (see Figure 2.5), this *SETUP* message contains the following Header and Information Elements (type M=Mandatory, O=Optional):

Table A.2 Information Elements of the analysed SETUP message

Information Elements	Comments according to the decoded information	Type
Protocol discriminator	09 → Q.2931 UNI Message	M
Call reference	00 38 FA	M
Message type	05 → SETUP, 00 → initiated by the user	M
Message length	7B → 123 bytes	M
AAL parameters	AAL Type: 5	O
ATM traffic descriptor	Forward Peak Cell Rate-1234cell/s(CLP=0) Backward Peak Cell Rate-1234cell/s(CLP=0) Forward Sustainable Cell Rate 128cell/s(CLP=0) Backward Sustainable Cell Rate 128cell/s(CLP=0) Forward Maximum Burst Size 128cells(CLP=0) Backward Maximum Burst Size 128cells(CLP=0) Tagging not requested	M
Broadband bearer capability	Bearer Class-BCOB-C Not susceptible to clipping User plane connection configuration-point to point	M
Broadband high layer information	High Layer Information Type-ISO	O
Broadband repeat indicator	Prioritized list for selecting one possibility	O
Called party number	Addressing Identification-ISO NSAP NSAP address- 0x2037581301 Node-203758, Slot-13, Link-01 Type of number: unkwown	M
Called party subaddress	Type of Subaddress-NSAP (X.213/ISO 8348 AD2) Subaddress Info: 0x0005000055	O
Calling party number	Addressing Identification-ISO NSAP NSAP address- 0x2037580101 Node-203758, Slot-1, Link-1 Type of numbar: unknown	O
Calling party subaddress	Type of Subaddress-NSAP (X.213/ISO 8348 AD2) Subaddress Info: 0x0005000055	O
Quality of service parameter	QoS class (F/B)-2.	M
Connection Identifier	Explicit indication of VPCI, exclusive VCI	M
Broadband sending complete	Broadband sending complete indication	O

Appendix B: A short description of the simulation tool ACCEPT v2.0

Motivation

The simulator has been originally developed to model and simulate the behavior of retransmitted signalling messages due to blocked wide-band calls (BCQ). We simulated the call set-up phase of the ATM connection, including path selection, bandwidth reservation and call rejection, and the flow of call establishment messages in order to estimate the queue lengths of signalling messages in the ATM switches. Later, this single queue model has been extended to the model presented in Section 5.1.

Graphical User Interface

We can simulate many types of calls: phone calls, with a bandwidth requirement of 64Kbps, and multimedia calls of different call profile, etc. These calls can be mixed arbitrarily (see Figure B.1). The duration of each call can be set to 0…9999sec. The network load can be selected from a pull-down menu or set manually.

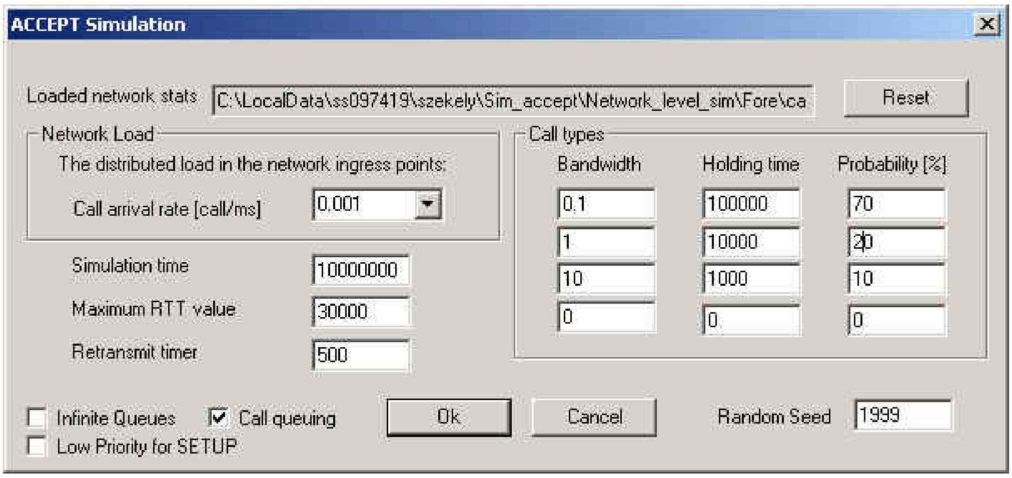


Figure B.1 Example of settings via GUI of the simulator ACCEPT for the BCQ mechanism

Descriptor file of network topology

The network topology can be defined using a *.net descriptor file. A simple example of an interconnected source, intermediate node and destination is given below:

```
-----------------------------------------------------------------
# Nodename(x,y)[Type,1/μ1,q1,1/μ2,q2,1/μ3,q3,1/μ4,q4,1/μ5,q5,1/μ6, q6];
FORE(320,240)[Switch,0.6,50,0.3,50,0.6,50,2.3,50,0.6,50,0,50];

# End systems (x,y)[Type,Buffer,setup_delay_income, setup_delay_outgoing,
# connect_delay, release_delay];
ES1(200,240)[Terminal,200,0,2,1,0];
ES2(440,240)[Server,200,4,2,0,0];

# {(Node_i, Node_j, Bandwidth1)(Node_k, Node_l, Bandwidth2)...}
{(ES1,FORE,155)(ES2,FORE,155)};
-----------------------------------------------------------------
```

Output file

The output parameters are: buffer size at access nodes, buffers size in core network, buffer size of each sub-process within a node, bandwidth usage in access nodes, bandwidth usage in core network, call establishment times, release latencies. The format of the output file is *.txt, which is portable to MS Excel.

Actions taken at the arrival of a new call

Processing of a call from the source node **A** to destination node **B** with a bandwidth requirement *BW* is the following:

1. A set of paths $\pi(i,j)$ which are candidates to carry this call are selected, and these are ordinarily indicated in some order of fixed priority so that the individual paths are denoted by $\pi_k(i,j)$, $k = 1, \ldots, K$. We assume that the number of possible paths K is fixed for all source-destination pairs. Paths may have different lengths, and $L(\pi_k(i,j))$ is the length of the k^{th} path between node i and j.

2. The *Network State Table* (NST) contains information regarding the availability of bandwidth on each link in the network. It is called upon to select the path $\pi(i,j)$ which is of highest priority and has enough bandwidth on all of its intermediate links to accommodate the call of bandwidth *BW*. Obviously, it may result in the network state table indicating that no path can accommodate the call, because for each path in $\pi(i,j)$ there exists at least one link with insufficient bandwidth. In this case, the call is rejected (reservation policy).

3. While selecting the path, the information in the network state table is assumed to be up-to-date and correct. Of course, there is a finite but hopefully small probability that this may not be true.

4. After the path is selected, the network state table is updated to reflect the reservation of bandwidth in the selected links of this call. Subsequently, a call request proceeds, hop by hop, from the source node through all nodes along the selected path, up to the destination node, reserving bandwidth on each intermediate link. At each intermediate node, this reservation message is executed either in *FIFO* or in some order of fixed priority with respect to other messages.

5. If the request reaches the destination node successfully, reserving bandwidth at each node, then a call confirmation will be transmitted back, acknowledging the "successful setup" state at each intermediate node along the same path (traversed in reverse order) and the call is established once the confirmation message reached the source node. The network state table is then updated to reflect the establishment of this call.

6. If a node being visited indicates that insufficient bandwidth available on the designated outgoing link (despite the fact that the network state table indicated the contrary), then the call establishment fails. A message is sent back, hop by hop, to the input node. An alternate path is then selected to establish this call. The alternate path can begin either from the node which could not provide the necessary bandwidth or from the input node. If the alternate path begins from the input node, then the call request goes back releasing the bandwidth until it reaches the input node.

7. If the call establishment fails after a specific number of tries, the call is rejected.

8. If the call is successfully established, then the bandwidth is reserved for the holding time of the call. Upon termination, the bandwidth is released at each link by a message which travels up the path, from source to destination, and the network state table is then updated.

The structure of a call event

We have implemented a classical event-controlled simulator. The main idea was to generate a linked list of events ordered in time, which contains events (calls) of type described in Table B.1.

Table B.1 The structure of a call event

a1	a2
b1	b2
Bandwidth	
Time	
Duration	
Route 1	
Route 2	
…	
Route k	
Current node in the call queue	
Pointer to the next event	

The list call events is generated using exponential distribution, while the source- and destination node is selected by uniform distribution. The initiating node is represented by the pair of numbers (a1,a2) and the destination node is represented by the pair (b1,b2). The routes are computed and filled at the initial moment of the call, when the call is generated at the initiating node. The current node field shows, in which node of the path the event is actually waiting for control, and the last part of the event represents the number of events waiting in the queue of the current node before this event. This list is ordered by time of initiation of the events.

The main loop of the simulator

The main loop of the simulator picks up the first event from the list, and starts processing it. There are several cases:

- If the event is situated at the initiating node, then the path selection algorithm is invoked. If a path could not be found, due to the saturation of the links in the network, the call will be rejected. If there is at least one path which can carry out the call, the route field in the event is filled in, and the event goes back in the queue of the initiating node.
- If the event is in the queue of a node, and there are one or more events waiting for control before this event, it will go back in the queue with a processing time added to his time stamp.
- If the event is on the top of the queue of a node, then it tries to reserve bandwidth on the outgoing link. If there is not enough bandwidth left to carry out the call, the call is rejected, and the bandwidth reserved at previous nodes are released as well (hop by hop back to the initiating node). If at the outgoing link there is enough bandwidth to carry out the call, the call reserves bandwidth, updates the network state table, enters the call queue of the next node in the path, and will go back in the list with a processing time added to its time stamp.
- If an event arrives to the destination node reserving bandwidth at the each intermediate node, the call will be accepted, and the connection established. The event will be sent back to the queue, with the duration of the call added to its time stamp. When the call is terminated, the event will go back hop by hop from the destination node to the source node, releasing bandwidth and updating the network state table.

History of Software Versions

- v1.0 – single queue model, FIFO queueing only, infinite buffer size, BCQ
- v1.1 – FIFO queueing, finite buffers, RDLM
- v2.0 – complex call model, FIFO + priority queueing for SETUP & REL
- v3.0 – bandwidth (re-)negotiation (planned)

In the current version, the simulator retries the blocked calls until success with the same bandwidth requirements, i.e. one can select the rejected wide-band calls to be retransmitted (optionally). In fact, there is not much work left to reduce the requested bandwidth for rejected calls to the half at each retrial, or according to a certain algorithm. These planned extension of the simulator to support renegotiation capabilities has been stopped. The current version of the simulator is available for free, however there is officially no support offered for further extension.

Bibliography

Referenced papers

[AALI97] ITU-T Recommendation, *"B-ISDN ATM Adaptation Layer Type 2 Specification"*, I.363.2, Geneva, Switzerland, 1997

[AALQ99] Draft ITU–T Recommendation, *"AAL type 2 Signalling Protocol (Capability Set 1)"*, Q.2630.1, March 1999

[ATMF00] ATM Forum Technical Committee, *"UNI Signalling Performance Test Suite"*, (ed. J. Orvis, A Francis) *af-test-0158.000*, October 2000

[Baf93] M. Bafutto, P.J. Kuhn, G. Willmann, *"Modelling and Performance Analysis of Common Channel Signalling Networks"*, Hirzel-Verlag, AEU, Vol.47, No.5/6, 1993, pp. 411-419

[Bat96] A. Battou, *"Measurement of UNI-3.0 Connection Setup Times"*, DARPA Workshop Wide-Area ATM Performance, University of Kansas, USA, June 1996

[Ber96] S.A. Berezner, A.E. Krzesinski, *"Call queueing in circuit switched networks"*, *Telecommunication Systems* No.6, 1996, pp. 147-160

[Bal97] J. H. Baldwin, B. H. Bharucha, B. T. Doshi, S. Dravida, *"AAL-2 – A New ATM Adaptation Layer for Small Packet Encapsulation and Multiplexing"*, Bell Labs Tech Journal, April 1997, pp. 111-131

[BCMP75] F. Baskett, K.M. Chandy, R.R. Muntz, F.G. Palacios, *"Open, Closed and Mixed Networks of Queues with Different Classes of Customers"*, *Journal of ACM*, 22 (2), 1975, pp. 248-260

[Bla96] S. Blaabjerg, G. Fodor, A.T. Andersen, *"Reducing Wide Band Blocking by Allowing Wide Band Calls to Queue"*, *COST 244 Technical Report*, TD(1996/14)

[Blac98] U. Black, *"ATM, Vol II., Signaling in Broadband Networks"*, Prentice Hall, New Jersey, ISBN 0-13-571837-6, 1998

[Chen98] X. Chen, *"Scalable Call Processing Architectures for ATM Switches"*, *Bell Labs Technical Journal*, January-March 1998, pp.62-80

[Cyp94] D. Cypher, *"Analysis of the Call Setup Delay for the Proposed B-ISDN Call Control Procedure"*, http://isdn.ncsl.nist.gov/misc/hsnt/NISTIR/sigdelay/sigdelay.html

[Csel99] I. Cselenyi, R.Szabo, *"Service Specific Information Based Resource Allocation for Multimedia Applications",* Informatica Journal, September 1999, pp. 317-324

[Dow96] N. Downey, A. Patel, *"A Proposed Architecture for an Advanced ATM Signalling Protocol"*, *Journal on Communications*, (periodical of the Scientific Soc. for Telecommunication, Hungary), Vol. 47, "Special Issue on ATM Networks", Jan.-Feb. 1996, pp. 24-33

[Ene99] G. Eneroth, G. Fodor, G. Leijonhufvud, A. Rácz, I. Szabo, *"Applying ATM/AAL2 as a Switching Technology in Third-Generation Mobile Access Networks"*, IEEE Communications Magazine, June 1999, pp. 112-122

[Far01] S. Farraposo, E. Monteiro, *"Evaluating PNNI Performance", Proceedings* of 4th IEEE International Conference on ATM and High Speed Intelligent Internet Symposium, ICATM, Seoul, South Korea, April 22-25, 2001, pp. 295-299

[Fod99] G. Fodor, S. Blaabjerg, A.T. Anderson, *"Modeling and Simulation of Mixed Queueing and Loss Systems"*, Kluwer Acad. Publisher, Personal Wireless Communications, 1999, pp.???

[Fore97] ForeRunner™ *ATM Switch Network Configuration Manual*, MANU0148-03 – Rev. A – 12/19/97, Software Version 5.1.x, Fore Systems, 1997

[GDC95] GDC APEX® *ATM Switch System, Operation and Installation*, Version 4.1, GDC 032R115-V410, Volume 1, General DataComm Inc., 1995

[Gel80] E. Gelenbe, I. Mitrani, *"Analysis and Synthesis of Computer Systems"*, Academic Press, 1980, pp. 94-109

[Gel97] E. Gelenbe, S. Kotia, D. Krauss, *"Call Establishment Overload in Large ATM Networks"*, *Proceedings* of the ATM'97 Workshop, Lisbon, Portugal, May 26-28, 1997, pp. 560-569

[GNN00] interWatch® 95000, *Performance and Verification Systems*, (CD ROM) DOC 96103695RH, GN Nettest, Release 3.3.0, 2000

[Hen95] T.R. Henderson, *"Design principles and performance analysis of SSCOP: a new ATM Adaptation Layer Protocol"*, ACM SIGCOMM, Computer Communication Review, 1995, pp.47-58

[HP96] HP Broadband Series Test System, *UNI Signalling User's Guide*, Hewlett-Packard Co., 1996

[Jain97] R. Jain, G. Babic, *"Performance Testing Effort at the ATM Forum: An Overview"*, *IEEE Communications Magazine*, August 1997, pp. 110-116

[Kan95] K. Kant, *„Analytic Modeling of SSCOP"*, Bellcore Technical Magazine, July 1994, pp. 1-13

[Kar98] G. Karagiannis, V.F. Nicola, I.G.M.M. Niemegeers, *"Quantitative Evaluation of Scalability in Broadband Intelligent Networks"*, Chapman&Hall Publisher, IFIP TC6 WG6.3 Conference "Performance of Information and Communications Systems", PICS'98, Lund, Sweden, 25-28 May 1998, pp. 65-82

[Kaus97] A. Kaushal, S. Shumate, R. Hill, S. Murthy, D. Niehaus, V. Sirkay, B. Edwards, *"Performance Benchmarking of ATM Signaling Software"*, *Proceeding* of OPENSIG Workshop, Columbia University, USA, October 1997, http://www.ittc.ukans.edu/~niehaus/

[Kim96] Y.B. Kim, S.S. Lee, C.H. Oh, Y.S. Kim, C. Han, C.H. Yim, *"An Architecture of Scalable ATM Switching System and Its Call Processing Capacity Estimation", ETRI Journal,* Vol. 18, No. 3, October 1996, pp. 107-125

[Kol98] G.T. Kolyvas, S.E. Polykalas, I.S. Venieris, *"Performance Evaluation of Integrated IN/B-ISDN Signalling Platforms", Computer Communications* Vol. 21, 1998, pp. 606-623

[Man97] X. Mang, E. Gelenbe, R. Önvural, *"Bandwidth Allocation and Call Admission Control in High-Speed Networks"*, *IEEE Communications Magazine*, Vol.35, No.5, May 1997, pp. 122-129

[Mau01] M. Maurogiorgis, N. Papadoukakis, E. Sykas, G. Tselikis, *"ATM Signalling Overview and Performance Measurements in a Local Area ATM Network", IEEE Symposium on Computers and Communications, ISCC'01,* Hammamet, Tunisia, July 3-5, 2001, pp.635-640

[Mer00] A. Mertz, M. Pollakowski, "*xDSL & Access Networks", Grundlagen, Technik und Einsatzaspekte von HDSL, ADSL und VDSL, Kapitel 6, Prentice Hall,* ISBN 3-8272-9593-9, Germany, 2000

[Min91] S. Minzer, *"A Signalling Protocol for Complex Multimedia Services", IEEE Journal on Selected Areas in Communications,* Vol. 9, No. 9, December 1991, pp. 1383-1394

[MSX99] MainStreetXpress 36170 *General Information Book and Technical Practices*, System Release 4.1, NNP 95-4936-01-00-B, Newbridge, October 1999

[Nie97] D. Niehaus, A. Battou, A. McFarland, B. Decina, H. Dardy, V. Sirkay, B. Edwards, *"Performance Benchmarking of Signaling in ATM Networks," IEEE Communications Magazine*, Vol. 35 No. 8, August 1997, pp. 134-143

[Nov99] J. Novak, A. Pouélé *"Interim Report on the Results of the Quantum Test Programme"*, November 1999, http://www.dante.net/quantum/qtp/QUA-99-070.pdf

[Onv97] R.O. Onvural, R. Cherukuri, *"Signaling in ATM Networks"*, Artech House, ISBN 0-89006-871-2, 1997

[Per01] H.G. Perros, *"An Introduction to ATM Networks, Part 4: Signalling in ATM Networks"*, John Wiley&Sons Ltd., New York, ISBN 0-471-49827-0, January 2001

[Pil99] R. Pillai, K.L. Su, J. Biswas, C.K. Tham, "*Call Performance Studies on ATM Forum UNI Signalling Implementations*", Computer Communications, Elsevier Science, Vol. 22, Issue 5, April 1999, pp. 463-469

[PNNI] ATM Forum Technical Committee*, "Private Network Node Interface Specification",* ver. 1.1, af-pnni-0055.002, April 2002

[Pri95] M. de Pricker, *"Asynchronous Transfer Mode, Solution for Broadband ISDN"*, *Prentice Hall*, U.K., third edition, 1995

[Q2931] ITU-T Recommendation Q.2931, *"B-ISDN. Dig. Subscriber Sign. Syst. No.2 (DSS2). UNI Layer 3 Specification for Basic Call/Connection Control", COM 11-R 78-E*, October 1994

[Schw97] A. Schwarz, P. Domschitz, A. Van Der Vekens, *"Performance Modelling and Simulation Studies of a Broadcast Service Switching Point"*, 5th International Conference on Performance and Management of Complex Communication Networks, PMCCN'97, Workshop 3: "Queueing Theory and its Applications", Tsukuba, Japan, November 1997, pp.46-57

[Sti95] B. Stiller, *"A Survey of Signaling Systems and Protocols for ATM Networks"*, ACM Computer Communications Review, Vol.25, No.2, April 1995, pp. 21-33

[UNI40] ATM Forum Technical Committee, *"ATM User-Network Interface (UNI) Signalling Specification",* Version 4.0, *ATM Forum/95-1434R8*, April 1996

[Won97] D. Wong, T. J. Lim, *"Soft Handoffs in CDMA Mobile Systems"*, IEEE Personal Communications, December, 1997

[Wu98] C. S. Wu, J. C. Jiau, K.J. Chen, M. Choy, *"Minimizing Call Setup Delay in ATM Networks via Optimal Processing Capacity Allocation"*, *IEEE Comm. Letters*, Vol.2, No.4, April 1998, pp. 110-113

[XP01] XpressPass 140/140HD/142/144 *General Information Book and Technical Practices Manual*, System Release 4.3, CD-ROM (P/N: CD-1400043), Seabridge, June 2001

Additional reading

[Chu93] C-P. Chung, K.W. Ross, *"Reduced Load Approximations for Multi-Rate Loss Networks", ACM/IEEE Trans. on Networking*, 1993, pp. 1222-1231

[Mar98] P. Martins, C. Rigault, N. Raguideau, *"A Signaling Based Approach to Broadband Service Control. Application to Access Signaling",* Proceed. of Broadband Communications, (eds. P.J. Kuhn, R. Ulrich), *Published by Chapman&Hall,* pp. 183-194

[Rit94] M. Ritter, P. Tran-Gia (Ed.), *"Multi-Rate Models for Dimensioning and Performance Evaluation of ATM Networks", COST 242 Interim Report*, Commission of the European Communities, June 1994

[Schwt88] M. Schwartz, *"Telecommunication Networks: Protocols, Modeling and Analysis"*, *Addison-Wesley Publisher*, ISBN 0-201-16423-x, 1988

ITU-T Recommendations

ITU-T Recommendation I.113 - B-ISDN Vocabulary of Terms

ITU-T Recommendation I.150 - B-ISDN Asynchronous Transfer Mode Functional Characteristics

ITU-T Recommendation E.164 - Numbering Plan for the ISDN era,

ITU-T Recommendation I.211 - B-ISDN Service Aspects

ITU-T Recommendation I.311 - B-ISDN General Network Aspects

ITU-T Recommendation I.321 - B-ISDN Protocol Reference Model and its Application

ITU-T Recommendation I.327 - B-ISDN Functional Architecture

ITU-T Recommendation I.361 – B-ISDN ATM Layer Specification, 1993, updated 1999

ITU-T Recommendation I.362 - B-ISDN ATM Adaptation Layer (AAL) Functional Description, 1993

ITU-T Recommendation I.363.1: B-ISDN ATM Adaptation Layer Type 1 Specification, 1993

ITU-T Recommendation I.363.2: B-ISDN ATM Adaptation Layer Type 2 Specification, 1997

ITU-T Recommendation I.363.5: B-ISDN ATM Adaptation Layer Type 5 Specification, 1996

ITU-T Recommendation I.413 - B-ISDN User-Network Interface, 1993

ITU-T Recommendation Q.700 - Introduction to CCITT Signalling System No. 7

ITU-T Recommendation Q.701 - Functional Description of the Message Transfer Part of Sign. Syst. No. 7

ITU-T Recommendation Q.702 - Signalling Data Link

ITU-T Recommendation Q.703 - Signalling Link

ITU-T Recommendation Q.704 - Signalling Network Functions and Messages

ITU-T Recommendation Q.705 - Signalling Network Structure

ITU-T Recommendation Q.706 - Message Transfer Part Signalling Performance

ITU-T Recommendation Q.707 - Testing and Maintenance

ITU-T Recommendation Q.708 - Number of International Signalling Point Codes

ITU-T Recommendation Q.709 - Hypothetical Signalling Reference Connection

ITU-T Recommendation Q.710 - Simplified MTP Version for Small Systems

ITU-T Recommendation Q.2110 - BISDN - ATM Adaptation Layer - Service Specific Connection Oriented Protocol (SSCOP) Specification, March 1994

ITU-T Recommendation Q.2130 - B-ISDN Signalling ATM Adaptation Layer - Service Specific Coordination Function for Support of Signalling at the UNI Interface (SSCF at UNI), December 1993

ITU-T Recommendation Q.2140 - B-ISDN Signalling ATM Adaptation Layer - Service Specific Coordination Function for Signalling at the Network-to-Network Interface (SSCF at NNI), Sept 1994

ITU-T Recommendation Q.2610 - Usage of Cause and Location in B-ISDN User Part and DSS2

ITU-T Recommendation Q.2650 - B-ISDN User Part to DSS2 Interworking

ITU-T Recommendation Q.2660 - B-ISDN User Part to N-ISDN User Part Interworking

ITU-T Recommendation Q.2726.2 - B-ISDN user part – Call priority, 1996

ITU-T Recommendation Q.2761 - Functional Description of the Broadband ISDN user part of Signalling

ITU-T Recommendation Q.2762 - General function of messages and signals

ITU-T Recommendation Q.2763 - Formats and codes

ITU-T Recommendation Q.2764 - Signalling procedures

ITU-T Recommendation Q.2931 - B-ISDN DSS2 UNI Layer 3 Specification for Basic Call/Connection Control, 1995

ITU-T Recommendation Q.2951 - Stage 3 Description for Number Identification Supplementary Services Using B-ISDN DSS2 Basic Call

ITU-T Recommendation Q.2957 - B-ISDN, Stage 3 Description for Additional Information Transfer Supplementary Services Using B-ISDN DSS2 Basic Call, 1995

ITU-T Recommendation Q.2959 - Digital Subscriber Signalling System No. 2 – Call Priority, 1996

ITU-T Recommendation Q.2961.1 - B-ISDN DSS2 Negotiation/Modification: Additional Traffic Parameter Indications

ITU-T (draft) Recommendation Q.2971 - B-ISDN DSS2 UNI Layer 3 Specification for Point-to-Multipoint Call/Connection Control, 1995

ATM Forum Recommendations

ATM Forum Technical Committee, ATM UNI Specification Version 3.0, September 1993

ATM Forum Technical Committee, ATM UNI Specification Version 3.1, September 1994

ATM Forum Technical Committee, Interim Inter-Switch Signalling Protocol Specification, December 1994

ATM Forum Technical Committee, ATM UNI Specification Version 4.0, June 1996

ATM Forum Technical Committee, Private Network-Node Interface (PNNI) Specification version 1.0, ATM Forum, April 1996

ATM Forum Technical Committee, ATM Broadband Inter-Carrier Interface (B-ICI) Specifications v2.1, af-bici-068.001, November 1996

ATM Forum Technical Committee, ATM Inter-Network Interface (AINI) Specifications, af-cs-0125.000, April 1999

ATM Forum Technical Committee, “UNI Signalling Performance Test Suite”, (ed. J. Orvis, A Francis) af-test-0158.000, October 2000

ATM Forum Technical Committee, User-Network Interface (UNI) Signalling Specification, Version 4.1, af-sig-0061.002, March 2002

ATM Forum Technical Committee, Call Processing Priority Version 1.0 , af-cs-0182.000, April 2002.

ATM Forum Technical Committee, Private Network-Network Interface Specification v1.1, af-pnni-0055.002, March 2002

ATM Forum Technical Committee, Private Network-Node Interface (PNNI) Specification version 1.1, ATM Forum, April 2002

Own Publications

[J] *Journal papers*

[J-5] S. Székely, G. Szűcs, S. Maly: Performance Benchmarking of Data Transmission over Symmetrical DSL Networks, Communications Magazine, Special Issue on Broadband Access, re-submitted on June 30, 2004 for a 2nd review, at the first review being rejected, number of pages: 13

[J-4] S. Székely, G. Szűcs, Cs. Simon, I. Moldován, S. Molnár: On the Impact of Link/Node Failures and Network Applications on the Load and Call Processing Times in ATM Networks, Periodica Polytechnica Electrical Engineering, Journal of the B.U.T.E ., Vol.46, No.1-2, 2002, pp.99-122

[J-3] O. Pop, S. Székely, M. Nafornita: Performance Evaluation of Forward Error Correction Schemes in ATM Networks, Buletin Stiintific of T.U. Timisoara, 42 (56), 1997, (in Romanian), pp. 89-98

[J-2] I. Cselényi, P. Haraszti, S. Székely: The First Hungarian ATM Pilot Network, Selected Papers from the Hungarian Telecommunication Periodical, 1997, pp. 6-10

[J-1] M. Otesteanu, S. Székely: User-Network Interfacing Problems in ATM Networks, Buletin Stiintific of T.U. Timisoara, Special Edition, 40 (54), 1995, (in English), pp. 153-160

[C] *Conference and Workshop papers*

[C-10] S. Székely: Signalling Performance Evaluation of Large ATM Networks Based on Performance Measurements of Isolated Switches, 7th IEEE Symposium on Computers and Communications, ISCC'02, Taormina, Italy, 27-30 July 2002, pp.670-675, *avail. online in IEEE DL, www.ieee.org,*

[C-9] S. Székely, G. Szűcs, Cs. Simon: Modelling of Call Processing in ATM Switches Based on Performance Measurements, Proc. of the IEEE International Conference on Telecommunications, ICT'01, Bucharest, Romania, 4-7 July 2001, pp. 327-335

[C-8] Cs. Simon, S. Székely, K. Németh: Point-to-Multipoint ATM Signalling Performance Measurements, Proc. of the 7th IFIP Workshop on Performance Modelling and Evaluation of ATM/IP Networks, PMEAN'00 Ilkley, U.K., 16-19 July 2000, pp. W17/1-10

[C-7] I. Szabó, S. Székely, I. Moldován: Performance Optimisation of AAL2 Signalling for Supporting Soft Handoffs in UMTS Terrestrial Radio Access Networks, Proc. of the 5th IEEE Symposium on Computers and Communications, ISCC'00, Juan-les-Pins, France, 4-6 July 2000, pp.46-52, *available online in the IEEE Digital Library, www.ieee.org*

[C-6] S. Székely, G. Szűcs: Performance Measures of Call Establishment in ATM Networks, Proc. of the ATMTU'99 Int. Symposium, Kosice, Slovakia, 17-19 February 1999, pp.116-121

[C-5] S. Székely, Cs. Simon, G. Szűcs: Performance Testing on Switched Virtual Connections in ATM Networks, Proc. of the 6th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, PMEAN'98 Ilkley, U.K., 20-22 July 1998, pp. 102/1-10

[C-4] S. Székely, I. Moldován, Cs. Simon: Overload Generated by Signalling Message Flows in ATM Networks, IFIP TC6 WG6.3 Conference "Performance of Information and Communications Systems", PICS'98, Chapman&Hall Publisher (eds. A. Nilsson, U. Körner), Lund, Sweden, 25-28 May 1998, pp. 51-64, *available at DBLP Uni-Trier*

[C-3] S. Székely: On Bandwidth Allocation Policies in ATM Network using Call Queueing, Proc. of the 5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks, PMEAN'97 Ilkley, U.K., 21-23 July 1997, pp. 46/1-10

[C-2] S. Székely, G. Fodor, S. Blaabjerg: Call Queueing: The Design and Performance Analysis of a New ATM Signalling Functionality, Proc. of the IEEE Workshop, ConTEL B&MW, Zagreb, Croatia, 11-12 November 1996, pp.99-113

[C-1] A. Faragó, S. Blaabjerg, W. Holender, B. Stavenow, T. Henk, L. Ast, S. Székely: Enhancing ATM Network Performance by Optimizing the Virtual Network Configuration, International Conference on Data Communications and their Performance, Chapman&Hall Publisher (eds. R. Onvural, H. Perros), Istanbul, Turkey, 23-26 October 1995, pp. 401-414, *available at DBLP Uni-Trier*

[T] *Technical papers*

[T-2] S. Székely: Schaltungsanordnung und Verfahren zur Fehleranalyse (*Eine neue Methode für System Tester zur Fehlersuche: Repräsentierung von Signalisierungsnachrichten in einem Plot-diagramm*), Phase 1: Erfindungsmeldung – 2003 E18241 DE, Phase 2: 2004 P00559 DE, Germany, patent pending

[T-1] S. Székely: VINCE: A Tool for Protocol Verification and for Interfacing Between Different ATM Signalling Protocols, Technical Report, T.U. Karlsruhe, Institute of Telematics, Germany, pp.1-37, March, 1996, (http://www.telematik.informatik.uni-karlsruhe.de/forschung/atm-info/vince/), *avail. at: Citeseer.com*

[H] *Hungarian journal papers*

[H-3] G. Szűcs, S. Székely: Hívásszintű teljesítmény-mérések ATM hálózatokban, Magyar Távközlés, 99/5, május 1999, pp. 3-7

[H-2] S. Székely, A. Arató: Videókonferencia alkalmazások ATM hálózaton, Magyar Távközlés, 96/6, június 1996, pp. 6-10

[H-1] S. Székely, K. Szarkowicz: Bepillantás az ATM jelzésekbe, Magyar Távközlés, 95/5, május 1995, pp. 27-30

[R] *Academic works which cited my publications*

[R-6] G. Fehér: Resource Control in IP Networks, Ph.D. dissertation, Budapest University of Technology and Economics, April 2004, *available on request*, cited [J-4]

[R-5] S. Waller: New AAL2 signalling protocol to support the UMTS Terrestrial Radio Access Network, M.Sc. thesis, University of Cape Town, 2003, http://crg.ee.uct.ac.za/progress.html, cited [C-7]

[R-4] A. Schwarz: Modellierung und Bewertung von Verfahren zur Last- und Leistungsregelung in Steuereinheiten von B-ISDN/ATM Vermittlungssystemen, Ph.D. dissertation, University of Stuttgart, LDB #33612, 2000, *available on request*, cited [C-4] and [C-5]

[R-3] I. Cselényi, R. Szabó, I. Szabó, A.L.-Henner, N. Björkman: Experimental Platform for Telecommunication Resource Management, Computer Communications, Vol. 21, pp.1624-1640, 1998, http://qosip.ttt.bme.hu/papers/local/cselenyi98compcomm.pdf, cited [C-2]

[R-2] B. Kaan: A Virtual Path Routing Algorithm for ATM Networks based on the Equivalent Bandwidth Concept, M.Sc. thesis, Bogaziti University, 1998, http://mercan.cmpe.boun.edu.tr/~kaanbur/thesis.pdf, cited [C-1]

[R-1] G. Seres, F. Baumann, G. Gordos, T.Henk: ATM alkalmazása a lokális hálózatban. A BME-TTT tanszék helyi ATM hálózata, Networkshop'97, Keszthely, Hungary, May 27-29, 1997, http://www.iif.hu/rendezvenyek/networkshop/97/tartalom/NWS/1/7/index.htm, cited [J-2] and [H-2]