Performance Evaluation of the DTM Fast Circuit Switched Networking Technology

Csaba ANTAL
High Speed Networks Laboratory
Department of Telecommunications and Telematics
Technical University of Budapest

Ph.D. Dissertation

Supervised by

Dr. Tamás Henk   Dr. Sándor Molnár   Dr. József Bíró

High Speed Networks Laboratory
Department of Telecommunications and Telematics
Technical University of Budapest

Budapest, Hungary
1999

Acknowledgements

First, I would like to express my gratitude and appreciation to my advisors.

To Tamás Henk for his guidance throughout my Ph.D. studies, for encouraging my work, and for his valuable comments on the thesis.

To Sándor Molnár who joined the DTM project at the best time, and helped my work with his deep knowledge in the fields of traffic modeling and performance analysis.

To József Bíró, who supported my work with his knowledge in discrete time queuing theory.

I wish to thank to László Györfi for the inspiring discussions about different model of DTM networks, and about queuing analysis.

I also would like to thank András Faragó and Miklós Boda for initiating my work.

A special acknowledgement to Gábor Szabó, who participated in the design and development of the simulation software, and to József Molnár, who implemented the software and participated in the evaluation of simulation results.

Finally, I would like to thank my family and my wife Kati, who have helped and supported me during this time with their love and patience.

Table of contents

Chapter I: Introduction

In the age of information society, communication is becoming a more and more important part of our everyday life. In the background of the changes of communication, there is a rapid evolution of telecommunication technologies. The main directions of this evolution are determined by new applications and by the development of base technologies. The most important demands on telecommunication networks are high communication speed, support of multimedia applications and mobility.

In order to obtain **high end-to-end communication speeds** all parts of the communication path should be fast. The bottleneck, which is continuously changing, can be the link capacity, processing capacity in nodes and the operation of higher layer protocols.

Advances of fiber technology increased dramatically the link capacity available for fixed networks. Wavelength division multiplexing [Bra90, DGL90] and spatial bandwidth reuse [CO90, KWH94, MS97] resulted in further increase in the bandwidth of links.

The increased link capacity moved the bottleneck to other parts of the network. The processing capacity of routers and switches is one of the bottlenecks of today networks. Therefore, the layer 2 and the layer 3 protocols are undergoing an integration to replace layer 3 functions - i.e. per packet routing with large processing requirements - with layer 2 functions - i.e. switching of data flows. IP switching [NEH96, rfc1987], tag switching [rfc2105] and ARIS [Wou96] are different implementations based on this idea.

Support of **multimedia applications** [J5] is a general term and involves requirements in many different areas.

First, multimedia traffic includes real-time data flows like voice and video. Real-time data should be delivered to the destination with low delay and low delay variation. Multimedia traffic also includes components that are not sensitive to delay variation but are very sensitive to data loss. The integration of these sources into a single network is an essential need today. Implementations of this integration are solved differently in different kinds of networks.

In the local area real-time delivery can be provided by overdimensioning the network capacity and using protocols, which do not introduce large delay variation. However, metropolitan and wide are networks should differentiate between different flows with different requirements. In the metropolitan area, isochronous and best effort traffic is differentiated. Technologies in that area are FDDI-II [Cal91], DQDB [Mar94], iso-ethernet [Gre96] and Metaring [Ofe94]. In wide area networks, several service classes are introduced [DBL93]. The most important technologies supporting several service classes are ATM [ATM96, MS94, Pry91, J3, C5, C6], RSVP [Res96] and UMTS [Mar97] at different areas of telecommunication. Packets belonging to different service classes are buffered in separate queues, which are served according to the priority of the related service class.

Multimedia applications (teleconference, internet radio, video on demand) also need networks with broadcast and multicast capability. Handling multi-party connections needs new signaling and routing methods, which is another rich research area.

Mobility is an essential need for future access devices. Broadband services already available in fixed networks are appearing now in mobile networks. Third generation mobile systems, which are standardized by ITU as IMT-2000 [IEEE97] and by ETSI as UMTS [IEEE98], are based on packet switching and support multiple service classes. Developments in radio technology [JSAC90] allow high speeds also across the radio interface.

Besides the above technological challenges, a technology should fulfil many other requirements in order to be successful on the market. Among many factors, the two most important ones are:

- reasonable price for good quality
- interoperability with existing technologies

So economists always have to select the technology, which provides the highest quality at an acceptable price. In the long term, integration and simple protocol reduces prices.

- With **integration** the network can provide wider service range, so one infrastructure can replace several parallel solutions. The main cost reduction of integration is due to the reduced maintenance requirements.
- **Simple protocol** reduces the implementation costs of equipment or allows higher speeds and therefore higher quality.

New technologies can not replace the entire existing infrastructure at once. Therefore, **interoperability** with existing devices is a very important need for all new technologies.

This thesis is about Dynamic Synchronous Transfer Mode (DTM) [Kah98], which is a new integrated-services networking technology. It is a fast circuit switched TDM system based on shared media. Its implementations [Netins, Dynarc] provide *high speeds*, *integrated* services with *real-time* support. *Interoperability* is provided trough IP technology, which is the unifying concept in today networks. IP over DTM and DTM LAN Emulation [whp99a, whp99b, Hol98] are the basic capabilities of DTM switches.

However, DTM, as a circuit switched network, has inherently two disadvantages:

- channel utilization: Bursty traffic uses the reserved bandwidth only in a small fraction of the time.
- scalability: The number of connections a DTM network can parallel support is limited.

There are two ways to increase the utilization and the number of allowed parallel connections with *multiplexing* in a DTM network:

- connection level multiplexing (burst switching): DTM connections are used only for the duration of data burst. If there is an idle period, resources are released. When data transmission starts again new DTM connection is established. That is, user connections are split to several successive DTM connections.
- slot level multiplexing: Multiple connections with low bandwidth demand are multiplexed into a DTM channel. That is, several parallel user connections share a DTM connection.

The dissertation deals with both performance improvement methods.

If *burst switching* is applied, the most important performance characteristics of the network *are average set-up time* and *blocking probability*. If different bursts have considerably different set-up times and bursts are blocked within the call then there is less QoS guarantee for the whole connection. That is, the main benefits of circuit switching (like low and deterministic delay during the connection) are lost. Consequently, optimizing the mentioned characteristics is advantageous for burst-switching as well. Set-up time and blocking probabilities are related to calls, therefore they are referred to as *call level characteristics* in this work.

If several connections are multiplexed into a successfully established DTM channel, the system should be analyzed on another scale. Small data items (referred to as messages) are buffered in nodes and the queues are served according to the multiplexing methods. Two important measures of performance are message loss probability and queuing delay of messages. Good performance in terms of these characteristics is also elementary. Advantages of circuit switching, i.e. guaranteed delivery and low delay and delay variation, can also be lost with a multiplexing method with poor *message level performance characteristics.*

1.1 Objectives of the Dissertation

The dissertation is about the performance and fairness of DTM networks. Results can be categorized into two fields:

- Call level characteristics of channel allocation algorithms in DTM
- Message level characteristics of multiplexing methods in DTM

The main goal of the evaluation of call level characteristics of channel allocation algorithms in DTM is to improve the effectiveness of channel allocation algorithms applicable for DTM networks.

For this purpose I have carried out the following studies:

- development and evaluation of new channel allocation algorithms to improve the aggregate performance characteristics and the fairness of the DTM network
- comparison of the performance of channel allocation techniques in DTM, in order to select the significant parameters influencing performance characteristics
- identification of the main factors causing the unfairness of a DTM dual-bus (including parameters of channel allocation algorithms, physical properties of the network and traffic profile of sources)

The goal of the evaluation of message level characteristics of multiplexing methods for DTM is to define and analyze new methods that *increase the effectiveness of a DTM channel*, while the network provides the required service parameters to all multiplexed sources.

I have carried out the following studies in that area:

- development and analysis of the most appropriate models for the examined prioritized multiplexing methods
- evaluation of the significant parameters influencing required buffer-size and message delay
- comparison of the effectiveness of multiplexing methods

I used simulations and mathematical analysis for the performance evaluation of DTM networks.

Simulation was used to evaluate call level characteristics of channel allocation algorithms in DTM. The DTM group in the High Speed Networks [HSNLab] developed a simulation software under my supervision in 1996-98. The simulator is based on the DTM model published in [BLRS96].

I have analyzed the message level characteristics of multiplexing methods by mathematical means, more specifically, by discrete time queuing theory [BrKi93]. The goal of the derivations is to obtain closed form expression for the probability generating function (pgf) of system content, system time of messages and unfinished work.

1.2 Outline

In Chapter II, the DTM network architecture and protocol is presented. The chapter starts with an outline of networking technologies to show the place of DTM among them. Then the detailed description of the DTM transport mode follows. It includes the description of the basic features of DTM, different modifications of the basic protocol as well as the interoperation methods with IP networks. The last part of this chapter summarizes previous performance studies of DTM.

Chapter III presents the simulation work of the dissertation. First, the simulation software is described including the modeling assumptions, some of the implementation details and the testing methods we used. Then the models of the simulated networks follow. Simulation results are presented in the two following sections. Fist, aggregate performance and fairness of set-up-time slot allocation algorithms are discussed. Then the characteristics of the new smoothing algorithms are analyzed.

In Chapter IV, the results of the mathematical analysis of different multiplexing methods are discussed for DTM networks. After the introduction, which presents the analysis of a simple queuing problem, two multiplexing methods are described in separate sections. First, the description and the models of “time division on two time scales with priorities” technique, and the results obtained from the models are presented. Next, “packet switching with priorities” multiplexing is analyzed using more models. Finally, the comparison of the multiplexing methods and the conclusion of the chapter follows.

Finally, Chapter V summarizes and concludes the dissertation.

Chapter II: Dynamic Synchronous Transfer Mode

This chapter gives an overview about Dynamic Synchronous Transfer Mode technology. Section 2.1 points out the place of DTM among other media access protocols. Section 2.2 presents the detailed description of DTM protocol including several development directions. Finally, Section 2.3 overviews the available performance evaluation studies of DTM.

2.1 Networking Environment of DTM

DTM is a *fast circuit switched technology* using *shared media* and *dual-bus* topology. To clearly point out the position of DTM among other networking protocols, a short introduction is given about other switching and media access methods illustrated with a few examples.

2.1.1 Switching Methods

Switching methods [Tan89] are divided to two basic classes: *circuit switching* and *packet switching*.

In packet switching data to be transmitted is segmented into packets, which have headers. The packet is routed through the network up to the destination based on its header information. Packets can be treated in two different ways.

- In *datagram packet switching*, the header includes the address of the destination (and source) node, and packets are routed through the network based on this address, independently of the other packets.
- In *virtual circuit packet switching*, a virtual connection is set up (using signaling or management) between the sender and the receiver before data is transmitted. Once the virtual connection is established – via a fixed route through the network – data packets are switched through the network based on the identifier of the virtual connection, which is located in the header of each packet. The address of the destination is used only during connection establishment. Service guarantees cannot be implemented without the virtual circuit concept because link capacity and buffer space should be allocated for connections.

Figure 2.1.1 shows some examples for each category.

Local and metropolitan area networks (e.g. Ethernet, FDDI, Token ring, SMDS) use datagram packet switching at media access control (MAC) layer. Virtual circuit packet switching is used in wide area networks (X.25) as layer 2 protocol.

Fast packet switching is a subcategory within virtual circuit packet switching. The difference is only in the implementation: The functions of the header are minimized, which allow fast processing in the switches. Fixed packet length [J3] further increases the processing speed, which allows the application of more complex buffer management strategies and faster switching. Frame relay and ATM are example for this category.

In *circuit switched networks*, nodes reserve fix bandwidth channels (circuits) for the whole duration of the connection. Each circuit switched system relies on a signaling system, which establishes and releases network resources accordingly.

Dedicated and fixed bandwidth is advantageous for real-time applications because it yields small delay and low delay variation. Computer generated data traffic, however, is bursty, so the usage of fixed bandwidth channels results in low efficiency.

As the overhead of a connection is independent of its duration (set-up, release), circuit switching is efficient for long connections. For short connections the overhead of resource reservation (both in volume and time) becomes large in contrast to packet switching.

Circuit switching also has a fast implementation: *fast circuit switching*. Here, resources are released during idle periods of the connection. Burst switching [Ams89, Ore88, MM88, HU90] (a specific technology in contrast with the term burst switching in the Introduction of Chapter I) and DTM are examples of fast circuit switching.

In burst switching, port processors are monitoring the link to notice activity. Whenever a port processor determines that a burst has begun, it prefixes a header to the information. The destination address in header is used to route the burst to its destination. A burst looks like a packet, but there are significant differences between burst switching and packet switching:

- the length of a burst is not determined before the start of transmission
- a burst is sent in a time-division channel of fixed bit-rate, i.e. it is interleaved with other bursts (in contrast to packet switching, where packets are sent one at a time with full link bandwidth)

The main difference between burst switching and DTM, which is introduced in the next section in detail, is that in DTM data and control channels are separated.

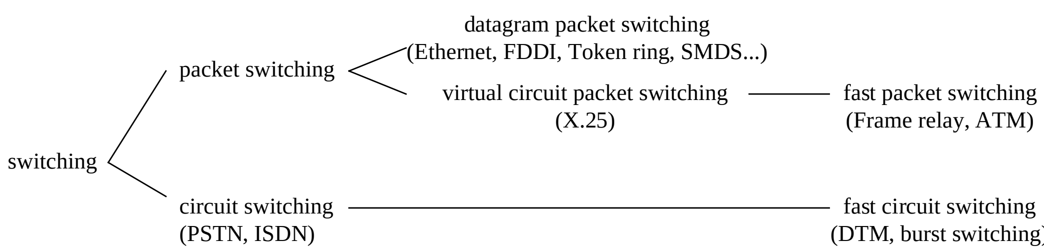


Figure 2.1.1 – Switching categories of media access protocols

It is not straightforward, which one is the best switching method for an integrated network.

Circuit switching is a better solution for real-time and non-bursty traffic, while it is inefficient for short sessions and best effort connections.

Packet switching provides high utilization because statistical multiplexing can efficiently share the bandwidth among different connections. The challenge for packet switching is to provide service guarantees for real-time applications.

2.1.2 Media Access Methods and Topology

The classification of DTM according to the media access method is interesting because the usual access methods are:

- **shared media** for datagram packet switching
- **point-to-point dedicated link** between switches for virtual circuit packet switching and circuit switching

DTM is an exception from this general rule because it is based on *shared media* despite of its *circuit switched* operation. The most prevalent topologies for a few shared media MAC protocols are listed in Figure 2.1.2.

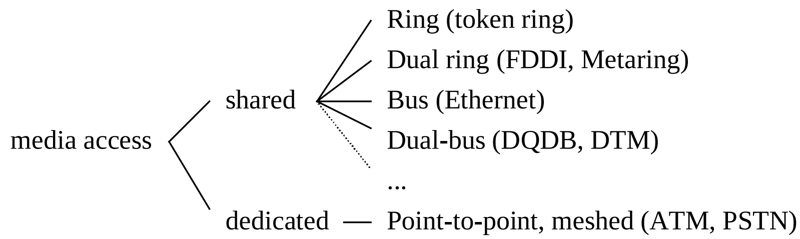


Figure 2.1.2 – Media access methods and topologies

Media access method and topology has special importance for DTM because these factors have a definite influence on fairness among nodes.

Fairness of protocols using dedicated links is controlled in a centralized manner. Therefore, providing fair share for connections can be solved with well-known scheduling algorithms. In shared media protocols, where due to efficiency reasons usually distributed protocols are used, equity among connections and nodes is not automatic.

Previous experience on dual-bus architectures suggests that the fairness of DTM should also be examined. For example, several algorithms were proposed to correct inequality among DQDB nodes [KWH94, MS97] because the basic architecture was inherently unfair. Providing fair operation becomes more difficult when inter-node distances are large, the system is overloaded (data, control or processing capacity) or spatial bandwidth reuse technology is applied.

After this short introduction to the classification of DTM within the family of media access protocols, a detailed description of DTM protocol follows.

2.2 Description of DTM

The detailed description of the family of DTM protocols is described in this section. First, the most important characteristics of DTM are presented. In Section 2.2.2, the resource management related topics – like slot allocation, QoS provisioning, fragmentation - are discussed in detail. Section 2.2.3, highlights the various development directions of the DTM media access protocol. Finally, Section 2.2.4 is about methods that enable DTM to become the transport protocol IP, and thus the Internet.

2.2.1 History and General Description

2.2.1.1 History

The first ideas for DTM were developed at Ericsson in the middle of the 80’s in the framework of Duper design [Hag85a, Hag85b, Hag86]. The DTM protocol, switching mechanisms and topologies are developed from these initial ideas. The DTM development has started at Royal Institute of Technology (Kungl Techniska Högskolan, KTH) in Stockholm as a part of the MultiG research program [PPG92, PRL92a, PRL92b, GHP92, PS93, Kar93] in 1990. In parallel with the development of the architecture, a prototype implementation was designed. The work on the prototype implementation and network architecture was reported in a number of publications and technical reports [AH93, BHL94, BLR93, BLR94, Goh94, LB94, Lin94, BL95, BLR96]. In 1996, the most active members of the DTM group at KTH received their Ph.D. degrees [Boh96, Lin96, Ram96a] and Licentiate degree [Hid96]. Two companies were established that time to produce DTM devices [Dynarc, Netins]. The work related to DTM started at Technical University of Budapest in 1996. I have participated in all activities related to DTM at TUB since 1996 [J1, J2, J3, C1, C2, C3, C4, C7, P1, P2, HSNLab]. North Carolina State University started DTM research in 1997 [CN98a, CN98b]. Product development has been focused on IP technology [whp99a, whp99b, Hol98, Kah98]. It clearly shows the current status of DTM that many patents are filed, products are available [Hey98] on the market and standardization has started [Dynar, NetIns].

2.2.1.2 General Description

The operation of DTM is based on multirate and either unicast or broadcast channels. It is designed for unidirectional medium with multiple access. The total medium capacity is shared by all connected nodes. Previous proposals and implementations are based on dual-bus topology, but folded bus and ring are also feasible. The architecture can be extended to include a large number of connected buses using switching nodes.

The most important elements of a DTM network are the nodes and the hosts. *Nodes* are networking devices connected to the dual bus. *Hosts* are end-devices with a simple interface that connects them to a node. Host-host communication is based on the assistance of nodes. Nodes are responsible for resource allocation, connection establishment and release along the bus. Figure 2.2.1 shows the set-up of a dual-bus network.

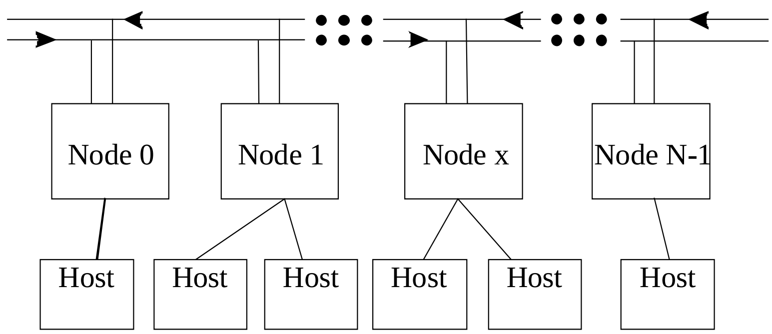


Figure 2.2.1: Structure of a DTM bus

The communication on the physically shared medium is realized by time division multiplexing scheme. The total capacity of the bus is divided into cycles of 125 microseconds, which are further divided into slots. A slot consists of a 64-bit data-word and some additional management bits. The sequence of slots at the same position in successive cycles is called DTM channel.

There are two types of slots (and so DTM channels): *data and static slots*. Data slots are used for data transfer. The number of data channels specifies the bit-rate of a DTM connection. There is a token for each DTM channel, which is assigned to one of the nodes. Both free and used data channels have one and only one owner at a time. If a node has the right to use a channel, then it has full control on it: it can set-up a connection on it, send data within the connection, release a connection using the channel, or give the channel ownership to another node.

In the DTM protocol, the sender node is responsible for channel reservation even if it is the initiator or not. This is the most obvious solution if point-to-multipoint (multicast) connections are used.

At system start-up, data channels (tokens) are allocated to nodes and they are transferred dynamically during the operation. Nodes can ask others for free data channels, if they do not have enough to serve a new request. This procedure is called *channel (re)allocation or slot (re)allocation*.

The other type of slot, called static slot is used for broadcast control channels between the nodes. Nodes send control information in their static slots and listen to all the other static channels to receive control information.

DTM uses a distributed channel reallocation algorithm [BLR93, BLR94]. A procedure for channel reallocation was proposed in [BLR96], which is referred to as **KTH algorithm** in the dissertation. In this method, nodes maintain a status table about the amount of free channels of other nodes. Nodes update their tables from messages captured from the control slots. The administration of status tables is a low priority task; therefore it can happen that tables are outdated.

2.2.2 The DTM Protocol

In [Boh94, Boh96, Hid96, Lin94, Lin96, Ram96a] the whole DTM protocol suite is described in detail. The DTM protocol suite allows other protocols to use DTM as a carrier network, and also supports native DTM applications that use DTM without any intermediate protocol. The architecture of DTM protocol can be seen on Figure 2.2.2

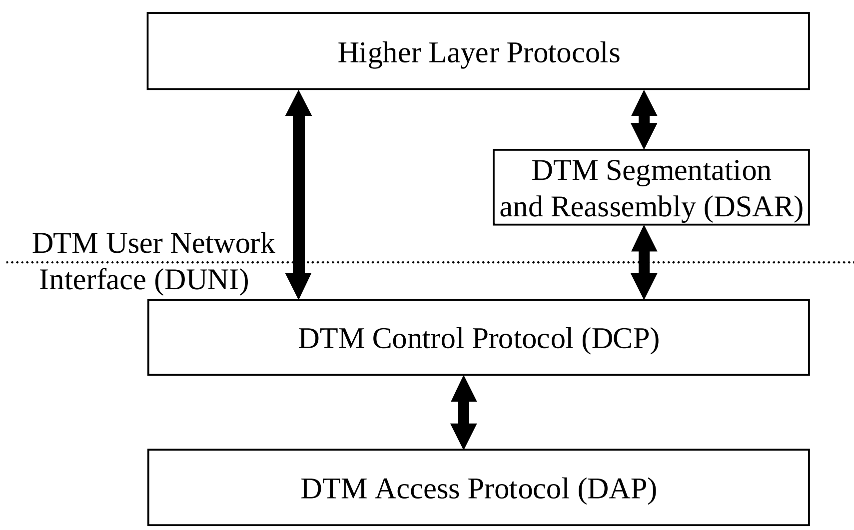


Figure 2.2.2 DTM protocol suite

Most of the existing networking applications use protocol-suites (like TCP/IP, IPX, Appletalk or ATM) of packet switched networks for communication. The easiest way to support these applications is to carry the packets transparently, i.e. without any direct interaction with the application, through the network. The protocol element, which is called **DTM Segmentation and Reassembly (DSAR)** layer, transforms (segments) the larger packets of higher layer protocols to 64 bit DTM protocol data units (PDU) at the sender, and reassembles packets at the receiver. Due to DSAR the transmission of packets over DTM can be done transparently for all upper layer protocols.

DSAR uses three types of PDUs: *head slot*, *data slot* and *idle slot*. When an upper layer packet arrives to DSAR, first a *head slot* is generated. It contains

- a *Length* field that shows the number of successive data slots for that portion of the packet
- and an *End of Packet* field that tells the receiver if that is the last part of the packet

Then the *data slots* are transferred. If there is nothing to transmit on the channel then the sender *transmits idle slots*.

The next protocol element in Figure 2.2.2 is the **DTM User Network Interface (DTM UNI)**. This interface defines how the user accesses the service of the DTM network. It also describes the service primitives between hosts and nodes. The DTM UNI service primitives are messages between nodes and hosts for connection set-up, connection release, change of bandwidth and data transmission.

Nodes have to communicate with other nodes in order to serve DTM UNI requests. The protocol that handles the node-to-node signaling and located below the user network interface is the **DTM Control Protocol (DCP)**. The main tasks of DCP are slot allocation, slot-to-connection mapping, sender/receiver synchronisation and management. Nodes communicate using DTM Protocol Data Units (PDUs). DTM control PDUs are transmitted in control slots and data PDUs are sent in data slots.

To illustrate the co-operation of UNI and DCP primitives Figure 2.2.3 shows the set-up of an acknowledged point-to-point connection.

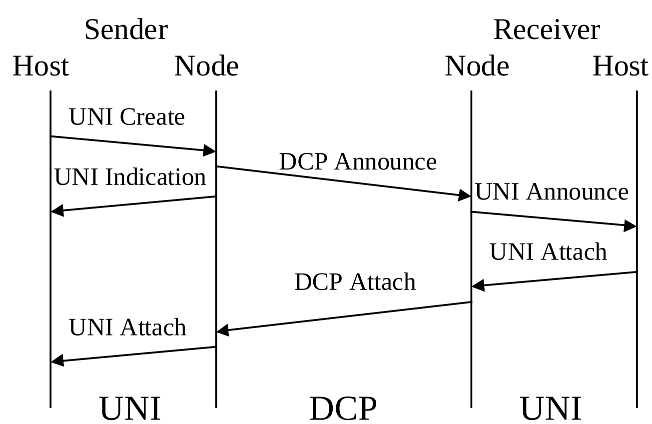


Figure 2.2.3 Service primitives used during the set-up of a point-to-point connection

Once a host wants to set up a connection, it sends a *create* primitive to its node via the user network interface. If the node can collect the requested number of slots via DTM Control Protocol, it sends a *DCP announce* message to the other node of the connection and it indicates (*UNI indication*) to its host that the requested resources have been allocated. The receiver node then forwards the *DCP announce* message to the destination host (*UNI announce*). If the destination host accepts the call, it sends a *UNI attach* primitive to its node, which transmits a *DCP announce* to the node of the sender host. Finally, the sender host receives the confirmation, its node forwards the *attach* message via the UNI in a *UNI attach* primitive.

The procedures for the following tasks are also described in [Lin96, Boh96]:

- Receiver initiated multicast connection set-up
- Rejection of connection set-up
- Sender initiated connection release
- Receiver initiated connection release
- Change of bandwidth
- Data to connection mapping
- Requested slot allocation
- Direct slot transfer
- Status message sent between nodes

The lowest protocol in the DTM protocol suite is the **DTM Access Protocol**. While the previously introduced protocols fit into the second OSI layer (Datalink layer), this is located at the Physical layer. It defines the access to the physical medium. The main units of the time division multiplexing scheme already introduced in Section 2.2.1 can be seen in Figure 2.2.4. A *frame* (or base frame) consists of *multiple cycles* and used for multiplexing multiple sources in a DTM channel. Frames are similar to multiframes and superframes in GSM [Rah93]. The description and analysis of multiplexing methods using frames will be described later in Chapter IV.

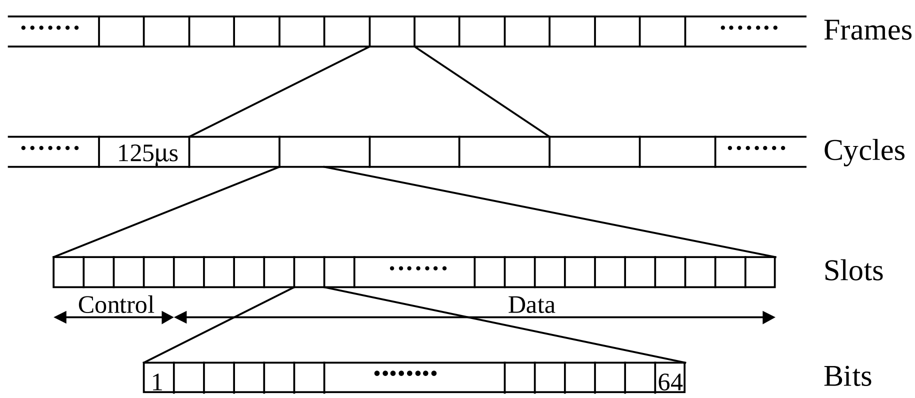


Figure 2.2.4 Definition of frame, cycle and slot of DTM

2.2.3 Resource Management

One of the most important questions is what kind of channel allocation algorithms to use in DTM. In this section, the basic types of allocation algorithms are introduced. The variants of distributed channel allocation algorithms are not discussed here in detail because the whole Chapter III is dedicated to their analysis.

2.2.3.1 Slot Allocation

Two basic types of channel allocation algorithms were developed in DTM: *centralized and distributed*.

In the centralized scheme there is a slot server. When a node wants to set up a connection, it has to ask the slot server for free channel. In the distributed scheme, each node has its own pool of slots. They only request channels from other nodes if local channels are not enough to serve a new request.

The disadvantage of using the distributed approach is that the synchronisation of nodes (status tables) causes an additional control load. Its advantage is that in case there are enough free slots locally the connection set-up time is shorter (no slot request). The distributed scheme is more fault-tolerant because it does not rely on a single slot server. The third disadvantage of the centralized scheme is that the slot server may become the bottleneck in the system. If distance between nodes increases the distributed solution outperforms the centralized one.

2.2.3.2 Fragmentation

The DTM Control Protocol handles channels in blocks. A block is a set of consecutive slots. A control message can only transfer one block at a time. If e.g. a connection consists of N blocks then N *DCP announce* messages should be sent to the receiver. Therefore, it is desirable to keep free channels in large blocks to reduce the signaling load and connection set-up time.

A fragmentation-avoiding algorithm is proposed in [Lin96]. In the algorithm, each slot has a home node. Home node of slots is set at start-up time, so that each node has a single block. There is a counter associated with each slot. The counter can for example store the time elapsed since the slot left its home node or the number connections the slot was used by. When the counter reaches a certain limit, the slot is transferred back to its home node.

The home node associated with a slot can change during the operation of the network, so more slots can be assigned to high capacity nodes.

2.2.3.3 Quality of Service

As the bandwidth of connections in DTM can be any multiple of 512 kbps, the primary performance measure in DTM is the bandwidth of the connection. In the basic protocol of DTM, there is no statistical multiplexing between connections, i.e. a connection uses its whole bandwidth up to the peak. So - with ATM terminology [ATM96] - there are constant bitrate connections (CBR) in DTM. The allocated bandwidth can be renegotiated during the connection, so to refine our naming, DTM connections are dynamic constant bitrate connections (dynamic CBR).

In the basic DTM protocol [BLR96] rejection policies can also be used to distinguish between connections. Rejection policy is used when a node can not allocate the resource requested by its host. There are three different rejection strategies [BLR93]:

- Fixed: The node rejects the connection immediately if available resources are not enough, and signals it to the host.
- Flexible: The connection is set up with resources that were available. If the host does not accept the offer, it can remove the connection.

- Negotiated: A negotiation takes place between the host and the node to decide if the connection should be set up.

In the case of the negotiated policy, there is a minimum acceptable bandwidth parameter. The network rejects the request if the allocated resources are below this parameter.

Chapter IV proposed slot level multiplexing strategies using priorities. In those proposals, QoS classes are separated with priorities.

2.2.4 DTM Enhancements

There are a number of DTM features that have been developed since the implementation of the first DTM prototype [BLR93, AH93, Kar93]. The following enhancements are introduced in this subsection:

- Fast Channel Creation
- Fast Channel Establishment over Several Hops
- Dynamic Signaling
- Virtual Networks
- Slot reuse
- Parallel DTM

2.2.4.1 Fast Connection Establishment [LB94]

In case of long distances confirmation based protocols are not effective because the propagation time of the acknowledgement message is too long. Fast channel creation is a non-confirmed connection set-up method. Data is sent directly after the *DCP announce* message, without waiting for the *attach* message. The advantage of this solution is that the set-up time of unconfirmed connections are shorter with the double of the propagation time between the sender and receiver. Its disadvantage is that the receiver can not reject the connection without data loss or buffering at the sender side.

This solution operates as a packet switched network: The *announce* message can be thought of as the packet header, the data as the payload of the packet, and the *remove* message as the packet trailer.

2.2.4.2 Fast Connection Establishment over Several Hops [LB94, Lin96]

The other procedure that slows connection establishment down is slot allocation. In case there are a number of switching nodes between the sender and the receiver, the usual procedure of the set-up is the following:

When a switching node receives an *announce* message one of the connected dual-buses, it first tries to allocate the requested number of slots on the other dual-bus. If the allocation was successful it sends an announce message to the next switching node along the path to the receiver.

Fast connection establishment over several hops accelerates this procedure. Switching node operating according to the improved protocol sends immediately a special message (DCP create) to the next hop after it receives a request (DCP create) from the previous hop without allocating the slots for the connection. The *announce* message is sent in the same way as it was in the previous version: when the node receives the announce message from the previous switch and the slot reallocation is successful. The advantage of this solution is that switching nodes along the path can allocate slots parallel to a DTM connection.

2.2.4.3 Dynamic Signaling [Lin96]

In the prototype the number of control slots was constant during the operation of the network. Dynamic signaling allows nodes to change the number of their control slots. Nodes therefore can use the optimal number of control channels. This is a very important feature because if the signaling capacity is insufficient it effects the performance of the node significantly. On the other side too many control slots degrade the performance because the bandwidth of unused control slots is wasted.

Nodes with low signaling requirements share one channel using frames. Each node uses on slot in a frame, in other words it has access to a control slot in every M^{th} cycle, where M is the number of cycles in a frame. In M is equal to 8 the capacity of the signaling channel assigned to one node is 64 kbps instead of the 512 kbps capacity of the whole DTM channel.

2.2.4.4 Virtual Networks [Lin96, whp99b]

Virtual networking, or building several logical networks on a common physical network is supported by changing the operation of control channels. In DTM virtual networks, signaling messages are not broadcasted, they are directed to (observed by) nodes that belongs to the same virtual network. The extreme case of virtual networking is a point-to-point control channel between a server and its client.

2.2.4.5 Slot reuse [Ram96a, Ram96b]

Slot reuse is a means to better utilize multiple access synchronous systems. It allows physically non-overlapping connections to use the same slots for communication.

Figure 2.2.5 presents the map of connections (with grey). Nodes in the order of physical location are shown on the horizontal axis, and the slots are displayed on the vertical axis. For example there is a connection between node 1 and node 7 that uses slots 1, 2 and 3. Without slot reuse the connection between node 10 and node 15 would not be able to use the same slots.

Slot reuse is implemented in hardware in most of the other technologies like DQDB [MS97], CRMA-II [ALS94], ATMRing [WR97, RW96, IHK90, IIK94] and Metaring [CCO92]. DTM provides a software solution by extending the block token format. Control messages include the segment information (physical part of the bus between two nodes) along with the slot number dimension when reallocating slots, establishing and releasing connections.

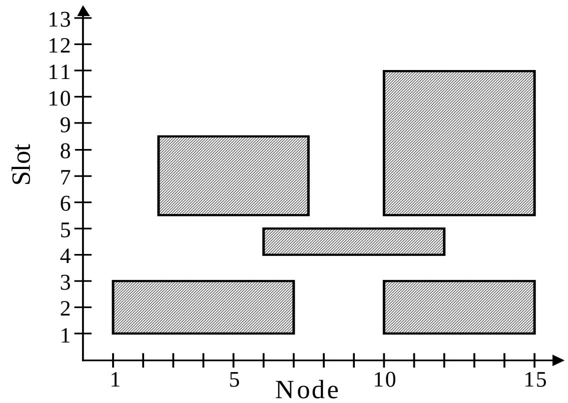


Figure 2.2.5 - Slot reuse

2.2.4.6 Parallel DTM [BL95,Lin96]

Another way to increase the performance of a DTM network is to parallelize its operation. In [BL95, Lin96], the use of wavelength division multiplexing (WDM) and space division multiplexing (SDM) is shown. In the examined implementation it was assumed that a node could send on one frequency (or physical fiber in SDM) and receive on all of them. To type of nodes are used: partially equipped and fully equipped nodes. A partially equipped node can only receive on a subset of frequencies (or physical fibers in SDM) simultaneously while fully equipped ones can listen to all of them. The usage of partially equipped nodes introduces destination conflict. That is, a call is blocked because the receiving node is the receiver of another connection that uses the same slot on another frequency (or physical fiber). So a connection can be blocked even if the sender can allocate the necessary resources.

Similarly to slot reuse, in case of WDM networks the wavelength can also be reused in physically non-overlapping segments of the network. [Lin96] discusses the performance issues of all aspects of parallel DTM.

2.2.5 Interoperation [Hid96, whp99a, Hol98]

As DTM is connection-oriented technology that can provide very high speeds, it has many common features with ATM. Due to its connection-oriented operation, a special protocol is required to interconnect it to broadcast based multiple access networks (like Ethernet, Token ring and FDDI). Two protocols are presented here: *DTM LAN Emulation and IP over DTM.*

2.2.5.1 DTM LAN Emulation

The operation of DTM LAN Emulation (DLE) protocol is very similar to that of ATM LAN Emulation [FM96]:

- It allows DTM to be used as a bridge between different segments of an Ethernet network
- and it integrates Ethernet and DTM nodes in the same local area network.

The basic elements of DLE are DLE server (DLES) and DLE client (DLEC). Their function corresponds to that of LES and LEC in ATM LAN Emulation [SM98, Min96].

When the DTM network connects Ethernet segments, Ethernet gateways are required at the border of the DTM and Ethernet networks. The operation has the following steps in this case:

- When an Ethernet frame arrives to an Ethernet gateway, the DLEC function in the gateway looks at the destination address of the frame:
- If it is a broadcast address, the DLEC forwards the packet (on a DTM channel) to the DLES for broadcast.
- If the destination is at the same Ethernet segment as the gateway, it does not have to do anything with the frame.
- If the destination is not at the local Ethernet segment, DLEC looks its table for the DTM address of the gateway that belongs to the destination.
- If there is and entry, the gateway forwards the packet.
- If the DLEC does not know the address it sends a MAC-to-DTM address resolution request to the DLES, and when it receives the reply it builds up a DTM connection to the destination DTM address.

2.2.5.2 IP over DTM

IP over DTM – similarly to IP over ATM [rfc1483, rfc1577, SM98] at ATM protocol - is aimed to specify how IP packets are transferred through a DTM network. Its operation consists of two levels.

First, there is a conventional IP network. Routers and their DTM connections define the structure of the logical network. IP packets are forwarded from router to router (hop-by-hop) until they reach the router connected to the subnet of the destination. Each router has to look at the network layer destination address of each packet and choose where to forward it. This store and forward operation requires high processing capacity at routers and is not able to guarantee low delay and delay variation needs of IP flows.

Consequently - as the second level - a protocol allows the establishment of shortcut DTM connection between sender and receiver nodes. It works at the presence of the following conditions:

- the application signaled its QoS demand so that it needs a dedicated (shortcut) connection between the sender and the receiver
- the router detected a large flow, and the IPOD system establishes a direct DTM connection to its destination.

The operation of shortcut establishment is very similar to that of Multiprotocol over ATM (MPOA) standard [MPOA]. The main difference is that IP over DTM does not rely on the concept of emulated LANs.

IP-to-DTM address resolution, which is needed during shortcut establishment, is based on Next Hop Resolution Protocol (NRHP) [NHRP].

The IPOD system has three types of nodes:

- IPOD client
- IPOD router
- IPOD border router

The IPOD client node is directly connected to the DTM network. It is an end node without any routing functionality. It has direct connection to one or more IPOD router where it sends the packets for hop-by-hop forwarding. It decides when to create shortcut for a flow. It also contains an NHRP client to request for NHRP address resolution.

The main tasks of the IPOD server are packet forwarding and exchange of IP level routing information using standard routing protocols (e.g.: OSPF, BGP). As it has also NHRP server functionality, serving IP-to-DTM address resolution requests is also its task.

The IPOD border router is an IPOD server with additional functions. It is connected to non-DTM networks. It handles shortcuts on behalf of these networks.

Finally, Table 2.2.1 presents a functional comparison of DTM LAN emulation and IP over DTM:

	OSI layer	Applicable network protocols	main purpose
DTM LAN Emulation	Operates at layer 2; deals with MAC addresses	all	integration of DTM with Ethernet networks
IP over DTM	Operates at layer 3 deals with IP addresses	only IP	interoperation with non-DTM networks

Table 2.2.1 – Comparison of DTM LAN Emulation and IP over DTM

2.3 Performance Studies of DTM

This section covers the most important publications related to DTM performance analyzes, which is the background of the performance study that to be presented in Chapter III. As Chapter IV is devoted to the analysis of multiplexing methods in time division multiplexing systems, which is not directly related to DTM, the related literature will be discussed there.

The most extensive performance studies of DTM were carried out based on the basic channel allocation protocol (i.e. without WDM and slot reuse) [BLR96] with dual-bus topology. Most of the studies analyzed the aggregate throughput and average access delay.

[BLR96], the first performance study of DTM focused on the effect of **overloaded signaling and data capacity**. The traffic model of the simulations was simple. Transfer requests were generated by Poisson processes and source and destination addresses were uniformly distributed. The slot allocation method was distributed using status tables and closest first request order during retry.

It was shown in [BLR96] that there is no break-down in throughput and access delay *at high offered loads* if there is *enough signaling capacity*. The lack of signaling capacity, however, resulted in large performance degradations:

- When the network was loaded with short but frequent connections (1kB data transfers), the throughput decreased, and access delay increased dramatically above 0.4 offered load.
- When the limit on channel reallocation retries was increased to 20, the same effect was observed at less frequent calls (16kB data transfers).

The effect of distance on performance was also studied. It was found that below a bus-length of 100 km, the performance is independent of the bus-length. At a bus-length of 1000 km the throughput decreased, and the access delay increased significantly.

The paper used a traffic model (Poisson arrival, constant holding time) that is not applicable for bursts of real data traffic. Poisson arrival process is less bursty than the arrival process of bursts in a real network is. Therefore, performance results presented in the paper are too optimistic (see later in Section 3.4.3.2).

The fairness of the used channel allocation procedure was not studied in the paper at all, therefore, the main weakness of the used algorithm remained hidden for the reader.

[Boh96] focused on the performance of **rectangular topologies**. The effects of different factors like processing, control and data capacity; distance of nodes; and number of nodes.

[Boh96] also gave a short performance study of **slot allocation techniques** in DTM. It compared the centralized slot allocation to three distributed slot allocation techniques. *Closest first*, *Most slot first* and *Broadcast* distributed slot allocation algorithms (all with status tables) were simulated. It was shown that at large bus-length the *Closes first* algorithm had the best performance. The arrival of connection requests were modeled with Poisson process for smooth sources, and with two-state Markov Modulated Poisson Process (MMPP) [GH85, Kle75] for bursty sources [JR86, PF95].

The assumption of the paper, namely having a network with regular rectangular topology is very specific, and the performance results are hard to apply to a real scenario. The paper did not considered fairness, which is a very important aspect of channel allocation.

[CN98a] and [CN98b] also analyzed the performance of a DTM dual-bus with slot allocation algorithms using status tables.

[CN98a] presented an **analytical model for DTM access nodes** based on the multi-rate Engset model. The mathematical model provided an estimate for the access blocking probability of DTM nodes in ideal conditions.

The model used in the paper provides valuable results when the *duration of the connection set-up time* and the *load of signaling traffic* are negligible. As these are two of the main differences between channel allocation algorithms, the presented model hides all differences between channel allocation methods, and can not be applied for their comparison.

[CN98b] carried out a simulation study to compare centralized and distributed **slot allocation methods**. A new factor, not studied in [Boh96], was examined: the *requested bandwidth of connections*. Both the algorithms and the model assumptions were adopted from previous studies.

[Ram96a] and [Ram96b] gave a performance study of DTM networks **using spatial bandwidth reuse**. Both central and distributed slot allocations were considered. The effect of bursty traffic, distance of nodes and length of transfers were simulated. It was shown that in most of the cases slot reuse improves the throughput of the network with a factor of two.

To develop a *fair* spatial bandwidth reuse algorithm in dual-bus networks is a very hard problem (see for example: DQDB [MS97]). The aspect of fairness was not addressed in these works. That is, the results of a potential fairness study might completely change their conclusions.

In [Lin96] various performance aspects of **parallel DTM** were studied. Bursty (MMPP) and Poisson sources, unicast and broadcast connections, distributed and centralized slot server, sender and receiver blocking, partially-equipped and fully-equipped nodes were also simulated.

Although fairness was not studied in detail in [Lin96], it was mentioned that due to receiver blocking the network is inherently unfair.

None of the previous performance studies of DTM provided conclusions about fairness, therefore my contributions are the first ones in this area.

Most of the papers dealing with channel allocation methods compared the centralized and the distributed methods. Comparison of different distributed slot allocation methods was addressed only in [Boh96], so this aspect of the dissertation is also a novelty in the DTM literature.

Therefore, the main objectives of the dissertation, which are related to call level analysis, are not addressed by these performance studies.

Chapter III: Performance of Call Level Characteristics of DTM

3.1 Introduction

This chapter is devoted to the performance analysis of channel allocation algorithm of DTM.

The basic methodology of the performance analysis to be presented in this section is simulation. Measurements were not carried out due to the lack of DTM devices. I did not analyze channel allocation algorithms with mathematical means because it would have required significant simplifications [CN98a], which are only applicable to an ideal DTM system where the delay of channel allocation is zero. The main focus of my work is the comparison of different channel allocation methods where this ideal operation can not be assumed because the main difference between the considered algorithms is in the channel allocation delay (set-up time).

As stated among the objectives of the dissertation one of the goals of this work is to propose new channel allocation algorithms to improve the aggregate performance characteristics and the fairness of DTM networks. I proposed two minor modifications to the channel allocation algorithm, which is used in the DTM prototype implementations in [J2] and [C3]. These variants are analyzed in Section 3.4. I also proposed a new algorithm, which is called smoothing algorithm in [J2] and [C1]. It is analyzed in Section 3.5.

The variants of the algorithm used in the prototype implementation are analyzed in two steps. First, a fairness study is presented, which requires the analysis of per node performance characteristics. Then the aggregate performance is analyzed. The main achievement of this section is the result of the fairness analysis.

A common drawback of the algorithms analyzed in Section 3.4 is that there is significant difference between the performance of nodes with different traffic load. The main achievement of smoothing algorithms, which are proposed to enhance the performance of channel allocation, is that they are able to correct this undesirable property.

Before presenting the results in Section 3.4 and 3.5, an overview is given in Section 3.2 about the simulation platform developed for DTM network in HSNLab. The model of the simulated DTM network, which includes three network loading profiles and two source models, is described in Section 3.3.

3.2 Simulator

3.2.1 Overview of the Development of the DTM Simulator

As our goal was to evaluate the characteristics of the DTM network and develop new algorithms and operation methods, we decided to develop a DTM simulation in 1996. There was no available simulation environment for us that time, so a completely new environment was developed in C++. There were many versions of the simulator and it was completely rewritten two times.

The first version of the software was designed by Gábor Szabó and me, and was implemented by József Molnár and Ákos Erdődi. The input was read and the output was written to files, the code was written in ANSI C, so the program was source compatible across many platforms. Practically, two platforms were used: the development was in MS-DOS operating system and the simulations ran on a Linux machine.

Later, József Molnár and I made the continuos improvement and development of the software. In 1997, the simulator was rewritten to support a more sophisticated node and bus model (supporting input and output queues, exact propagation delay calculation etc.). More and more variants of the existing channel allocation algorithms and also new ones were implemented. The code was also optimized for speed (by improving the operation of the scheduler of the global event queue).

The third version of the software intended to improve the user interfaces. The program was moved to Windows95/NT platform and a graphical user interface was added by the support of Visual C++. Now, there is no need to directly edit the input configuration files. All configuration parameters can be set using a user-friendly GUI.

The development of the user interface has not finished yet. Now, the simulator generates a lot of verbose output files that need to be further processed in e.g. Gnuplot or Microsoft Excel to obtain the final graphs and tables.

All versions of the simulation software have object-oriented design, and have a global event queue for intra-node and inter-node messages.

3.2.2 Modeling Assumptions

This section summarizes the main assumptions, which the development of the simulator was based on. The models of the DTM system are discussed in three groups. First, it is clearly stated, which DTM variant was implemented in the simulator from the family of DTM systems. Then the model of nodes follows. Finally, the models of hosts – which are seen as traffic generators here – are described.

3.2.2.1 General Model

The simulator is tailored to the main focus of my work, namely to the study of channel allocation methods. This topic is too broad, therefore the studied system has the following properties:

- Only one node can reserve a given slot a given time. That is, spatial slot reuse , which was presented in Section 2.2.4.5, is not implemented in the simulator.
- Signaling capacity is allocated to nodes statically in 512 kbps steps. That is, base frames and dynamic signaling, which was described in Section 2.2.4.3, is not considered.
- Both set-up and release messages need acknowledgement. That is, fast channel creation, which was introduced in Section 2.2.4.1, is not implemented.
- Wavelength division multiplexing, which was discussed in Section 2.2.4.6, is not used.

These properties of the general model of the simulator are based on the DTM model published in [BLR96].

The most extensively studied topology of DTM networks is the dual-bus, therefore the simulator is also based on dual-bus. The number of nodes and the number of hosts connected to a node, and the distance between nodes can be arbitrary.

3.2.2.2 Node Model

A proper node-model is necessary to analyze the operation of the network in the case of overload situations. The node model of the simulator can be seen in Figure 3.2.1.

If processing capacity is overloaded then messages waiting for the node processor are stored in input control buffers. If control capacity is too low, output control buffers are needed to delay control messages until free control slots are available.

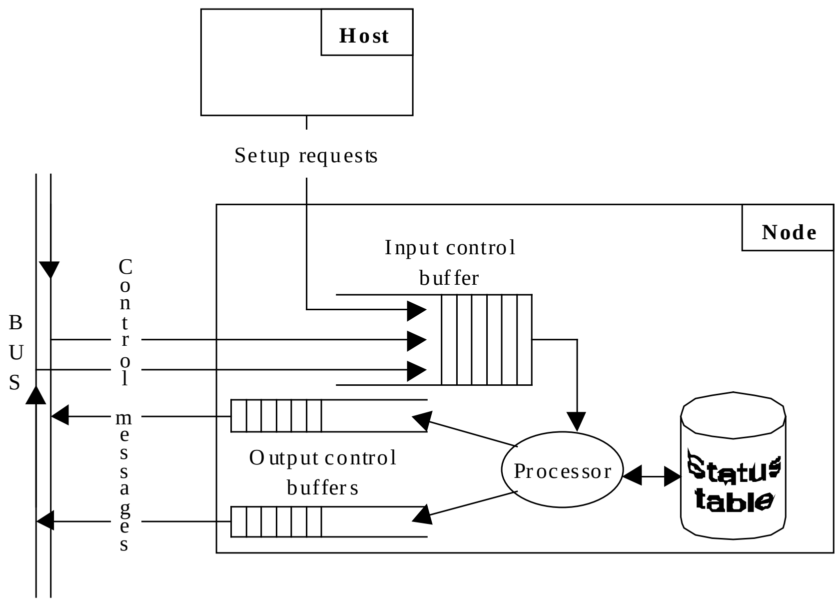


Figure 3.2.1 - DTM node model

The following assumptions are made on the parameters of the elements of the node model:

- In order to avoid overflow of input buffers, they should be large enough to store control messages arrived within a few cycles.
- Based on the assumptions in [BLR96], we also assume that the processing time for all control messages is the same ($5\mu s$)
- Each control message could be transmitted in a single time-slot (64 bits).
- One control channel belongs to each node.

In order to keep even a congested node in operation, message dropping and call blocking mechanisms have to be applied at the node. The following rules are in effect independently of the used channel allocation algorithms.

Control messages from other nodes that require a reply (connection set-up request, channel request, connection release request, background channel allocation request), or necessary for the node to continue its operation (connection set-up reply, channel request reply, background channel allocation reply) are never dropped even if the input buffer overflows. If these types of messages were discardible, only time-outs would solve the problem of closing broken channels, which should be avoided in a high-speed network.

Control messages from other nodes that don't require a reply (e.g.: status table updates) are dropped if the input buffer exceeds a given value. Though it causes small inconsistencies (e.g.: in status tables), it does not set back the operation of the nodes while the number of messages waiting for the node processor is decreased.

Auxiliary messages sent to all other nodes (e.g.: status table updates and balancing messages) are dropped if the output buffers exceeds a given value. This reduces the congestion in the control capacity, while it causes only a small inconsistency in the operation.

Set-up requests from a local host are blocked immediately, and they are not passed to other nodes, if the output buffer exceeds a given value. This rule moderates the congestion in the signaling capacity as well.

If the output buffer of an initiator node overflows, the calls being set up are blocked, if the node tries to send a connection set-up request for this call.

3.2.2.3 Host Model

Hosts are the traffic generators in the simulator. It is assumed that a host generates connections according to a given process with the same parameters during the whole simulation period. The distribution of three parameters can be configured for each host:

- the interarrival time of connections; It can be
    - the time between the end of a connection and the beginning of the next one
    - or the time between the beginning of two successive connections
- the holding time of the connections
- the bandwidth of the connection

The simulator allows using many kinds of distributions. The distribution of the host parameters depends on the applied traffic model. The detailed traffic models will be described later in Section 3.3.2.

It is assumed that the duration of a host-node communication is negligible. The connection set-up times do not include the delay coming from host-to-node messages.

3.2.3 Implementation

The simulator is designed with object oriented methodology. The main reason for using object-oriented design is that it is able to cope with high level abstraction of real elements of the communication like bus, node, host, connection and control messages. These network elements are mapped to Bus, Node, Host, Call and Event object classes. Several other object classes are defined in the program, [Mol98] includes their detailed description.

The simulator is based on event driven operation. That is, there is a scheduler, which ensures that the next event (e.g. control message) is always removed from the list of waiting events and its action is executed. The action of events usually generates new events, which are inserted to the event queue according to their attached time-stamp. The time-stamp of an event show the time when the action of the event is executed.

The simulation program makes it possible to study various features of the network. There is a list below with the characteristics written to output files:

Values logged for each node separately for both directions:

- number of blocked and served connections
- average and maximum connection set-up time
- average and maximum queuing delay in the output buffers
- number of connections that experienced a given number of channel reallocation retries, separately for succesful and blocked calls
- probability mass function of free channels
- average number of free and used channels

Values logged for each node:

- average queuing delay in the input buffer
- number of lost status table update messages

Values logged for each host:

- number of blocked and served connections

The simulator is checked thoroughly for errors. The simulator supports two tools to simplify this process.

- An output file can be generated where there is a list of all events related to a given node with the state of the node in the delivery time of the event. This tool is useful to test the implementation of new channel allocation algorithms.
- Another output file can be written with the generated random numbers (for interarrival time and holding time). The statistical parameters of the random number generators can be checked with this file.

To test the *correctness of the initial model* there are many characteristics of nodes and hosts that can be logged [Mol98].

3.3 Network Model

A DTM dual-bus contains nodes and hosts. The network model to be described has two parts: the network load profiles and the host models.

- Host models characterize the call level properties of traffic sources. Interarrival time, holding time and requested bandwidth are the three characteristics used in host models.
- Network load profiles define how hosts are distributed along the network. The number of hosts connected to each node and destination of connections generated by hosts are the main parameters of the network load profiles.

It is assumed throughout the dissertation that the DTM dual-bus has 622 Mbps line speed in both directions, so 1200 data slots are available for the channel allocation algorithms.

The next sections present three network load profiles and two host models.

3.3.1 Network load profiles

A great part of the following sections studies the fairness of DTM networks. It is hard to interpret fairness in an environment where the load of network nodes different. To avoid this problem, during the fairness studies the analyzed nodes generate calls with the same statistical parameters regarding both the arrival intensity and the holding time distributions. To emphasize the characteristics of the dual-bus, however, it is allowed that nodes share their calls differently between the direction of the dual-bus. Consequently, even though the load of the dual-bus is even, the load of one direction of the dual-bus can be uneven.

Based on these concepts, three basic network load profiles are proposed in this section. All of them can be associated with a real network scenario. The load of a real network can be obtained as the superposition of the described network loadings.

The first traffic profile, called external traffic profile, assumes that nodes communicate with a node at the end of the bus. The second one, so-called client-server traffic profile, gives a scenario where nodes communicate with a node in the middle part of the bus. According to the peer-to-peer profile all nodes communicate with all other nodes attached to the same dual-bus.

The following subsections give a detailed description about the network scenarios. Most of the simulations were based on a dual-bus with 100 nodes that is why this number is used throughout the description.

3.3.1.1 External Traffic

The first network scenario considers connections to external nodes.

A complex DTM network consists of many connected dual-buses. Two dual-buses are synchronized by a switching node, which is attached *to the end of both dual-buses* [Lin96, Boh96]. The interest of this work is a single dual-bus, therefore a connection to a host outside the simulated dual-bus is modeled by a connection to the switching node. Blocking probability and set-up delay of a real external connection are higher because here only the part of connection blocking and connection set-up delay affected by the conversations between nodes on the observed dual-bus are considered. The results, however, can be used to compare the characteristics of the nodes. The relative values of the main characteristics are enough to examine the fairness of the dual-bus network.

It is assumed that hosts initiate bi-directional high level connections to the switching node. A bi-directional connection should be set-up as two unidirectional connections at the DTM access control level. The backward direction of the connections, where the initiator is the receiver, is replaced to a unidirectional connection between the same hosts and with the same direction but with sender initiation. According to this replacement 99 virtual hosts are connected to the switching node. The simplified model is:

- 1 host is attached to each node except the switching node and it generates connections to the switching node at the end of the bus
- 99 hosts are attached to the switching node and each of them generates connections to one of the hosts attached to other nodes.

The statistical parameters of the hosts are the same, so the intensity of the switching node is 99 times higher than the intensity of the other ones.

The offered load of the unidirectional buses can be seen in Figure 3.3.1. Nodes have the same offered load on bus 0 (going towards the switching node) except the switching node, which is idling. On bus 1 only the switching node reserves channels.

This is the very basic load scenario from fairness point of view, because the load of the unidirectional buses is also evenly distributed.

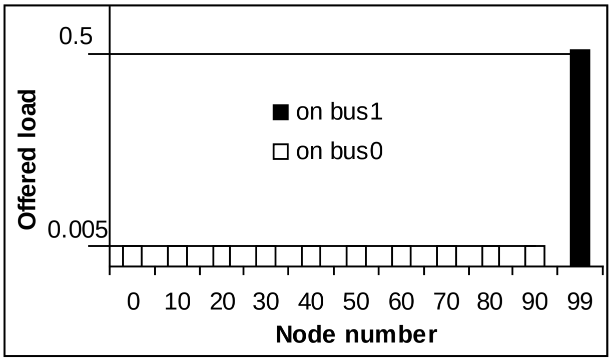


Figure 3.3.1 - Offered load in the external traffic scenario

3.3.1.2 Internal Traffic - Client-Server Model

The second network scenario considers connections between clients and a server.

A large part of the traffic in a LAN is directed to the server. In this model, it is assumed that this is the only traffic type in the dual-bus.

Because of the physical properties of the dual-bus, the maximal throughput of a node in the middle of the dual-bus is twice as much as it is at the end of the bus. Therefore the optimal place for a server is the middle part of the dual-bus. Apart from this fact, the traffic scenario is like in the case of the external model. If the receiver-initiated parts of the connections are substituted with similar but sender initiated connections between the same hosts, the following hosts are needed:

- 1 host is attached to one client node and they generate connections to the server in the middle of the bus
- 99 hosts are attached to the node of the server and each of them generates connections to one of the hosts attached to client nodes.

The statistical parameters of the hosts are the same, so the intensity of the node of the server is 99 times higher than the intensity of the nodes connected to client hosts.

If one has a look at one of the unidirectional buses in the client-server set-up, it can see that half of the client nodes generate with the same (non-zero) intensity and the other half of them are idling. The offered load of the server directed to this bus is 50 times higher than the load of the sender client nodes.

As it can be seen in Figure 3.3.2 the load profile of the bus in the other direction is the same but mirrored.

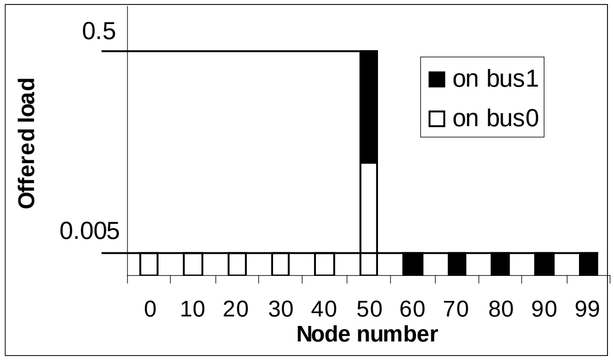


Figure 3.3.2 - Offered load in the client-server traffic scenario

3.3.1.3 Internal Traffic - Peer-to-Peer Model

In the case of peer-to-peer model there is no special node. Each node has 99 hosts. Each host generates unidirectional connections to one of the hosts attached to another node. The destination node is different for each host of a node. The traffic parameters are the same for every host along the dual-bus.

The traffic profile of the unidirectional buses can be seen in Figure 3.3.3. If only one of the buses is considered the offered load of the nodes is proportional to the distance from the end of the bus.

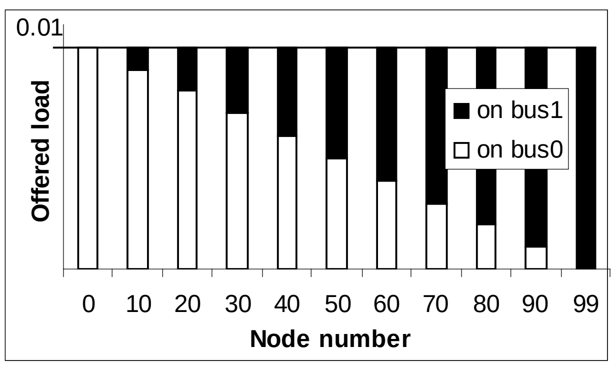


Figure 3.3.3 - Offered load in the peer-to-peer traffic scenario

3.3.2 Host Model

In our model hosts are traffic generators. Most of the simulations used a bursty traffic model as current data traffic has bursty characteristics [CB96, KA97]. The model of traditional networks, the Poisson model [Gir90] is also used in a few cases to examine dependence of performance on burstiness of traffic.

The host models define the distribution of three characteristics in both cases:

- interarrival time, which is the time between the connection set-up requests
- holding time, which is the duration of a connection
- bandwidth, which is the bandwidth reserved for the connection

Both in Poisson model and in the bursty model all hosts initiate bi-directional point-to-point connections and they require the bandwidth of one channel in both directions. That is, the requested bandwidth is deterministic and its value is 512 kbps.

3.3.2.1 WWW Model

The traffic model of the bursty traffic is based on the analysis of World Wide Web traffic. The burstiness of the WWW traffic at the connection level is due to the operation of the current version of the HTTP protocol (http 1.0) [http]. The protocol establishes a separate TCP connection for each object on a http page. If for example there are 5 graphics on the page then 6 TCP connections (1 for the body text and 5 for the graphics) are used. It introduces burstiness into the traffic because the interarrival time between page downloads depends on the reaction time of the user (typical greater than 3 s) and interarrival time between connections for the objects of the same page depends on the protocol.

The analysis of Web traffic showed that the user-initiated TCP session arrival process could be well modeled by Poisson processes like in classical telephony [PF95]]. However, the Poisson process cannot be used for modeling the arrival of WWW requests because it contains several non user-initiated requests. Several studies suggest the use of long-tailed distributions such as Weibull or Pareto distributions for modeling the arrival process of WWW and for estimating the size of requested documents [CB96, Den96, Vic97]. The WWW host model is based on these studies.

The inter-arrival time of the WWW requests (*X*) is modeled by a Weibull distribution given by the probability density function

$$f(x) = \lambda^{\beta} \beta x^{\beta-1} e^{-(\lambda x)^{\beta}} \qquad (3.3.1)$$

where the parameter β and the parameter λ depend on the generated traffic profile. Analytical studies of arrival process of WWW requests suggested the use of parameter $\beta = \frac{1}{3}$ [Den96]. With this value the mean of the inter-arrival time is

$$E(x) = \frac{6}{\lambda} \qquad (3.3.2)$$

The holding time *T* of a request is modeled by the Pareto distribution given by the probability density function

$$f(t) = \frac{\alpha k^{\alpha}}{t^{\alpha+1}} \qquad (3.3.3)$$

where the parameter is chosen to be $\alpha = 1.9$. The parameter *k* depends on the assumed mean size of the files to be transmitted.

The mean holding time *T* of a requested connection is

$$E(t) = \frac{\alpha}{\alpha - 1} k \qquad (3.3.4)$$

The parameters were selected based on the analysis of measured WWW traffic [Den96].

3.3.2.2 Poisson Model

The second type of host model assumes that DTM calls are generated according to a Poisson process [Gir90] with exponential holding time distribution. The exponential distribution is given by the probability density function

$$f(t) = \lambda e^{-\lambda t} \qquad (3.3.5)$$

The mean of the exponential distribution is

$$E(t) = \frac{1}{\lambda} \qquad (3.3.6)$$

3.4 Characteristics of Set-up Time Channel Allocation Algorithms

3.4.1 Description of Set-up Time Channel Allocation Algorithms

In the basic distributed channel reallocation algorithm of DTM [BLR96], nodes maintain a status table about the amount of free channels of other nodes. Nodes update their tables from messages captured from the control slots. The administration of status tables is a low priority task for nodes, therefore tables might be outdated. If the signaling load of the DTM bus is high or the processing capacity of a node is overloaded, its status table becomes outdated. In this work two models are applied regarding processing and signaling capacity:

- In the first model, it is assumed that *signaling and processing capacity does not cause bottleneck*, and nodes send out status table update messages after each change.
- In the second model, it is assumed that there is *no signaling capacity for status table update messages*, i.e. status tables are useless. That is, nodes try to get free channels from others without any apriori knowledge.

3.4.1.1 Using Status Table: KTH-S algorithm

It is assumed in this algorithm that there is enough signaling bandwidth and processing power to keep status tables up-to-date.

The details of the operation are the followings:

There are two connection set-up methods: one for the case where the initiator of the unidirectional connection is the sender (it can be called write request) and one for the read request case where the initiator is the receiver of the data.

a, Sender is the initiator (write request)

If a host wants to send data to another host, it requires a connection with M channels from the connected node. The node first checks its local pool to see if it has enough channels to satisfy the request. If so, it immediately sends a connection establishment message to the destination node. Otherwise, if it has only N free channels where N<M, it sends out reallocation messages requesting M-N channels. The node first sends a request to a node, which has free channels according to the status table. The node that receives the request for K channels and has an

amount of J free channels will always offer min(J,K) channels. If the node transferred J channels (J<K), then the requester node sends a reallocation message to another node with free channels, and so on. The requester node sends out request messages until all nodes with free channels are asked or the number of retries reaches a limit (retry limit) or the necessary number of channels is collected. If this last one is the case, the node sends a set-up request to the destination node. After acknowledgement arrives from the destination, data transmission can start immediately.

b, Receiver is the initiator (read request)

The node of the initiator host forwards the connection request to the destination node. This node, which will be the sender within the connection, allocates the required number of channels according to the above-mentioned procedure. After channel reallocation is finished it sends a positive or negative acknowledgement to the initiator node, and so it can start transmitting data.

This procedure is almost the same to the previous one. The difference is that not the initiator node is responsible for channel reallocation.

During the simulations, three request orders where used. They are described in the next subsection - in Section 3.4.1.2. The algorithms according to the request orders are referred to as KTH-S-CF, KTH-S-LR and KTH-S-RA.

Now we finished the description of the connection set-up procedure of KTH-S algorithm.

3.4.1.2 Without Status Table

According to the model where it is assumed that there is *no signaling capacity for status table update messages*, the sequence of nodes at channel request and the value of retry limit become more important. Three possible algorithms regarding the order are examined. KTH-CF algorithm was proposed in [BLR93, BLR96]. I proposed KTH-LR in [J2] and KTH-RA in [C3]. The description of these algorithms can be found below.

Closest First : KTH-CF algorithm

In the KTH-CF (Closest First) algorithm operates based on the closest first rule. An additional rule is also applied to balance the signaling load evenly between the directions of the dual-bus. The resulted operation is the following:

1. The requester node first chooses one of the directions randomly (bus0).
2. The first node to be asked for channels is the *closest downstream* node on bus0.
3. The second node is the *closest upstream* node on bus0.
4. The third node is the *closest (and not requested) downstream* node on bus 0.
5. And so on.

Logical Ring : KTH-LR algorithm

In the KTH-LR (closest first on logical ring) algorithm nodes are ordered into a logical ring. The order of channel requests is based on the location in the ring instead of the bus.

If channels need to be requested, nodes always take the closest not requested neighbouring node along the ring. The order can be determined based on the same rule - closest alternating between upstream and downstream directions - as it was in KTH-CF, if we redefine the words closest and distance. Here distance of two nodes is equal to the number of nodes between them stepping on the logical ring. With this distance definition nodes take the closest node first when asking for free channels.

The ring is constructed so that the second neighbouring nodes on the bus are successive nodes on the ring except the edges of the bus. The two neighbouring nodes of outer nodes on the ring are the first and second neighbours on the bus. With this choice the average physical distance of ring neighbours is minimal. The structure is illustrated in Figure 3.4.1.

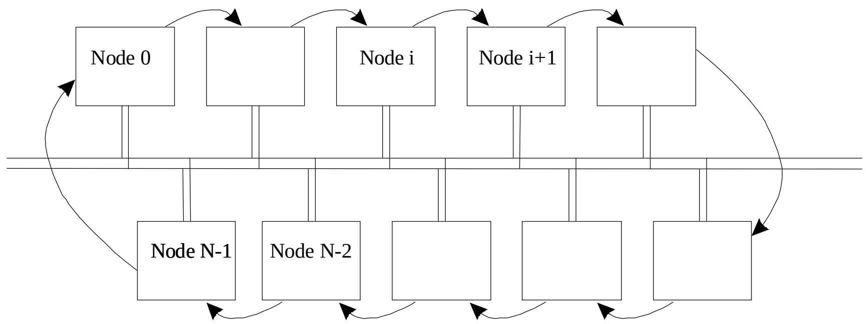


Figure 3.4.1: Logical ring structure

Random : KTH-RA algorithm

In KTH-RA (random order) algorithm nodes choose the next node randomly to ask for channels. The only restriction is that a node can be asked once during the channel reallocation for one connection.

3.4.2 Fairness Study

This section is about the fairness of set-up time channel allocation algorithms, which were described in Section 3.4.1.

The first subsection of fairness study introduces the methodology, where the definition of fairness (used in this work) and the confidence intervals of results can be found. Then in Section 3.4.2.2, the effects of the bus-length and effects of synchronization of the directions of the dual-bus are highlighted.

Then six sections follow where different scenarios are analyzed. All network load profiles, which were presented in Section 3.3.1, are evaluated at two bus-length settings.

First, in Sections 3.4.2.3-3.4.2.5, three scenarios belonging to the short-bus case are described. Then, in Sections 3.4.2.6-3.4.2.8, scenarios belonging to long bus-length are presented. The order of the description of scenarios is chosen so that homogeneous bus-load (external profile) is presented first, then more complex profiles (client-server and peer-to-peer) follow.

3.4.2.1 Methodology

Definition of fairness

Fairness is a dubious concept, which has several meanings depending on the context. To avoid misunderstandings a definition is described here, which shows the usage of this word in the dissertation:

A network is not fair if the differences of some specific performance characteristics of nodes, which are loaded with the same type and amount of traffic, are above some acceptable limits.

As fairness analyzes are usually based on the visual comparison of performance results, this definition might be sufficient. However, I aimed to provide quantitative results, which allow a more exact definition of fairness. The Jain fairness index [Jai91] is a good quantitative measure for fairness, therefore it is used for this purpose in this work.

If we denote the observed characteristics of node *i* (set-up time or blocking probability) by x_i and the number of nodes by *N* then Jain's fairness index can be calculated as:

$$f_{Jain} = \frac{\left(\sum_{i=0}^{N-1} x_i\right)^2}{N \cdot \sum_{i=0}^{N-1} (x_i)^2} \qquad (3.4.1)$$

If the performance of the nodes is the same then the index equals to 1. The less fair the network is the closer the index is to 0. Jain's fairness index reflects the number of nodes in unfair situation as well as the amount of differences [JCH84]. However, because the index of all systems should be mapped into the [0,1] interval, it is very hard to define the limit between unfair and fair operation based on the Jain index.

To make interpretation of the index easier let us take an example. Let us assume that there are two groups of nodes. The observed performance characteristic is the same for nodes within each group (denoted by x_1 and x_2). The characteristic of the first group is higher with *d* percent than that of group 2 ($x_1 = (1+n/100)x_2$). The first group contains *n* percent of the nodes ($N_1=Nn/100$; $N_2=N-N_1$). The fairness index of this system depends only on *n* and *d*. Table 3.4.1 shows the index for the network with the above assumptions at different *n* and *d* values.

	d=15%	d=20%	d=25%	d=30%	d=35%	d=40%	d=45%	d=50%
n=2%	1	0.999	0.999	0.998	0.998	0.997	0.996	0.995
n=4%	0.999	0.998	0.998	0.997	0.995	0.994	0.993	0.991
n=6%	0.999	0.998	0.997	0.995	0.993	0.991	0.989	0.987
n=8%	0.998	0.997	0.996	0.994	0.992	0.989	0.986	0.983
n=10%	0.998	0.997	0.995	0.992	0.99	0.987	0.984	0.98
n=12%	0.998	0.996	0.994	0.991	0.988	0.985	0.981	0.977
n=14%	0.997	0.995	0.993	0.99	0.987	0.983	0.979	0.974
n=16%	0.997	0.995	0.992	0.989	0.985	0.981	0.977	0.972
n=18%	0.997	0.995	0.992	0.988	0.984	0.98	0.975	0.97
n=20%	0.997	0.994	0.991	0.987	0.983	0.979	0.973	0.968
n=22%	0.996	0.994	0.99	0.987	0.982	0.977	0.972	0.966
n=24%	0.996	0.993	0.99	0.986	0.981	0.976	0.971	0.965
n=26%	0.996	0.993	0.99	0.985	0.981	0.975	0.97	0.964
n=28%	0.996	0.993	0.989	0.985	0.98	0.975	0.969	0.963

Table 3.4.1 – Jain's fairness index as the function of percentage of nodes with higher measured characteristics (*n*) and the difference between the performance of the two groups (*d*)

With a concrete example, if blocking probability is 0.3 for 10 nodes (x_1=0.3; N_1=10) and 0.2 for 90 nodes (x_2=0.2; N_2=90) then *d*=50% and n=10%, so the fairness index equals to 0.98.

According to the Table 3.4.1 I defined three fairness categories:

If the fairness index of nodes, which are loaded with the same type and amount of traffic, is above 0.995 then the network is ***fair***. If it is below 0.995 and above 0.98 then the network is ***unfair***. If the value of the index is below 0.98 then the network is ***very unfair***.

The fairness analysis of DTM is based on connection blocking probability and average connection set-up time. The results are based on an overloaded system where total offered load[1] is about 1.2 times higher than the system capacity.

Estimation of the steady state means and confidence intervals

The *replication/deletion approach* [LK91] is used in this work to estimate the steady-state mean of the observed characteristics. The idea of the replication/deletion approach is to delete the warmup period from the output data and to use n>1 replications of the simulation runs.

According to this approach the steady-state mean and its confidence interval is estimated as follows:

- Determine the length of the warmup period (denoted by l) using Welch graphical method [LK91]
- Choose the length of the simulation run (denoted by m) much larger than l
- Calculate means for each replication based on observations beyond the warmup period
- Now, n numbers are obtained (where n is the number of replications), which have normal distribution due to the central limit theorem
- Let $\overline{X}(n)$ denote the mean of the obtained n values and $S^2(n)$ their sample variance. Then $\overline{X}(n)$ is an approximately unbiased point estimator for the steady state mean, and an approximate $100(1-\alpha)$ percent confidence interval is given by

$$\overline{X}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \cdot \sqrt{\frac{S^2(n)}{n}} \qquad (3.4.2)$$

In our case the length of the warmup period is determined by the convergence speed of set-up time and blocking probability to their steady state means.

As the distributions of interarrival time and holding time of calls – Weibull and Pareto distributions – have heavy tails, their convergence should also be studied.

To determine l, i.e. the length of the warmup period, the mean of the set-up time of a given node (in Figure 3.4.1) and the mean of Pareto distribution (in Figure 3.4.2) can be found below as the function of the number of samples (generated calls).

[1] Offered load is the ratio of the sum of the requested volumes (holding time of the call * bandwidth of the call) during T and the maximum transmittable volume during T (T * total bandwidth of the bus) where T>>0.

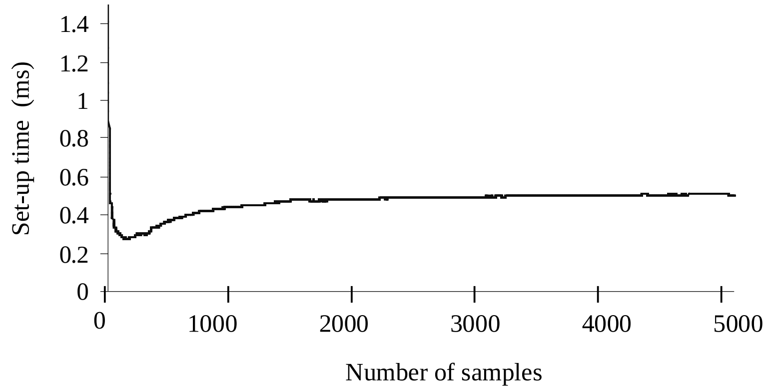


Figure 3.4.1 – Mean set-up time of a given node vs. the number of samples used during the calculation (KTH-LR algorithm, 120% offered load, client-server load profile, node 20)

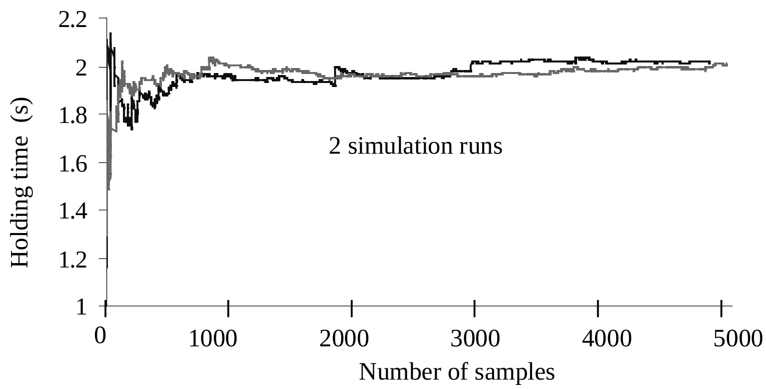


Figure 3.4.2 – Mean of holding time (Pareto distribution) vs. number of samples

Both figures show (and also the figures for other nodes and figures for blocking probability and interarrival time, which are not shown here) that *l*=1000 is a good estimate for the duration of the warmup period. Therefore, the length of the simulations was determined so that the least active node generated *m*=10000 calls. That is, the warmup period was – in worst case - 10% of the total simulation period. Instead of displaying the confidence intervals along with the simulation results, the worst case values are presented here. Using the above assumptions the half length of the 95 percent confidence interval of the results is always less than:

- 5% of blocking probability per node
- 1% of set-up time per node
- 0.5% of blocking probability per dual-bus
- 0.1% of set-up time per dual-bus

These worst case values at 10000 samples are valid for a slightly overloaded system, which is studied during the fairness analysis. When the offered load was lower longer simulation were needed to keep the confidence intervals shown here.

I also observed that the lower the retry limit was the slower the convergence was. So when the retry limit was 5, the simulation was run until the slowest host generated 50000 connections.

3.4.2.2 Effect of the Length of the Bus

Short bus vs. long bus

DTM protocol, like most of the communication protocols, is based on request-reply and transmission-acknowledgement message pairs. For example, a simple connection set-up contains a set-up request and a set-up reply message. The model used in the simulator takes into account both the *signal propagation times* and the so-called *response time*, which is the time elapsed between time instant when the requested node received the request and the time when the reply was sent. So the delay of a whole request-reply cycle consists of the following parts:

$$t_{whole\_delay} = t_{propagation} + t_{response} + t_{propagation}$$

where the response time is built from the following items:

$$t_{response\_time} = t_{input\_queue} + t_{processing} + t_{output\_queue}$$

So the message first waits in the input queue of the responding node until the control processor becomes available. Then the processor processes the message. And finally the message is put into the output queue of the node where it has to wait until it reaches the first place in the queue and the first control slot arrives.

Under light load the length of both the input and output queue is very low. The processing time is assumed to be 5μs. Therefore, waiting for the first available control slot, which can be 125μs long, dominates in the response time. So, unless the control channel or the processor is overloaded, the average response time is not more than a few microseconds.

Consequently, the delay of a whole request-reply cycle contains a distance dependent part (i.e. the propagation time) and a distance independent part (i.e. the response time).

- If the length of the bus is *short*, the propagation time is negligible compared to the response time. Therefore, t_{whole_delay} is the same for each node independently of the requesting and requested nodes if the network is well synchronized.
- If the length of the bus is *long*, the response time is negligible, so t_{whole_delay} is proportional to the distance between the requesting and requested nodes.

The operation of the DTM dual-bus is studied at two settings of inter-node distances. In the first case, the inter-node distance is 10 m, which corresponds to the short bus-length case. In the second case, the inter-node distance is 10 km, which is the long bus case.

Synchronization

Synchronization between the directions of the dual-bus relies on synchronizer nodes. One of the slot generator nodes, which are located at the ends of the dual-bus, synchronizes the timing of the two directions to each other. This node – so-called synchronizer node – starts a new cycle after an offset time from receiving the cycle start on the other bus.

In case the bus-length is short, synchronization needs to be designed with care.

Let us see an example: Suppose that there is 10µs between control slots on bus0 and bus1 at a given node, i.e. control slots on bus1 come always 10µs later than those on bus0. In this case the response times of messages arrived from bus0 are very short and messages coming from bus1 have to wait almost a whole cycle.

Take another example: Suppose that control slots on bus0 are very close to control slots on bus1. In this case a small difference between the slot timings can result into significant difference in the response time, which is called cycle-hop. That is, a node reaches the same cycle on the reverse bus when responding to messages, but its neighbouring node just misses that cycle due to the small difference caused by propagation. As a result, there is almost one cycle between the response times of two neighbouring nodes.

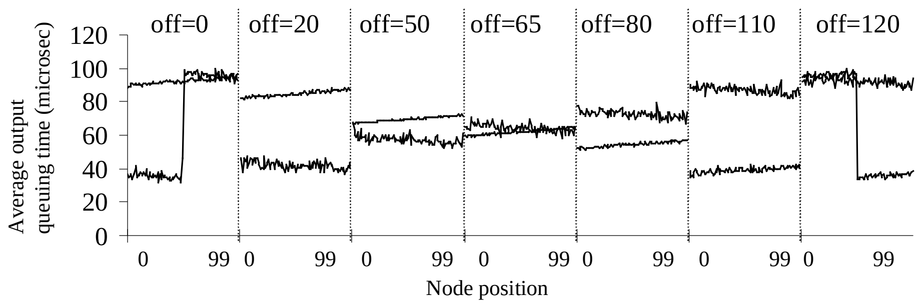


Figure 3.4.3 - Synchronization

Figure 3.4.3 shows the average output queuing time of messages for seven offset time configurations side by side. The utilization of the system where these results are obtained was low. Therefore, output queues were empty with high probability. That is, the queuing time of reply-type messages reflect the settings of synchronization.

The difference between offset settings can be clearly seen on Figure 3.4.3. A cycle hop can be observed when the offset is 0µs and 120µs. Difference between response times of the same node on two buses is the smallest when the offset is 65µs. For this reason, **the offset is set to 65µs** during the simulations.

3.4.2.3 Short Bus - External Load Profile

This is the first section among the ones, which present the six scenarios. Therefore, the structure of these sections, which have standard format, is shortly introduced here:

After the presentation of the configuration of the simulated network, the results of the fairness analysis are described shortly. Then detailed analysis of the most interesting questions follow in separate subsections. Finally, each section is closed with a very short conclusion.

In this section the external traffic profile at short bus-length is evaluated. Four algorithms are studied: KTH-S, KTH-CF, KTH-LR and KTH-RA.

Configuration

The main host and node parameters are summarized in Table 3.4.2.

Node parameters		
	Ordinary nodes	*Switching node*
Number of control slots	1	10
Length of input buffer	150	1500
Length of output buffers	100	1000
Processing time of control messages	5μs	5μs

Host parameters			
	Distribution	*Parameters*	*Mean*
Holding time	Pareto	α=1.9; k=3.79	8 s
Interarrival time	Weibull	β=0.33; λ=11.82	0.5 s
Bandwidth	Deterministic	-	1 slot/cycle

Table 3.4.2 - Configuration parameters, external model, short dual-bus

Results

According the simulation results, KTH-S algorithm is not sensitive for the request order at short bus length. That is, algorithms with different request orders resulted in the same performance. The reason is that if there are free channels in the network, nodes are aware of them, so they can get them. At KTH-S algorithm, blocking occurs only if there are no free channels in the system.

Retry limit of KTH-S algorithm is also a secondary question in this case because it turns out within a few request-cycle whether a node succeeds or fails with channel reallocation.

Due to these facts, only KTH-S algorithm is displayed only once in the figures and tables. It stands for KTH-S-CF, KTH-S-LR and KTH-S-RA algorithms.

Figure 3.4.4 shows connection blocking probability and average set-up time of nodes at different algorithms and retry limit settings. Both figures in Figure 3.4.4 include 4 graphs side by side, one for each algorithm variant (KTH-CF, KTH-LR, KTH-RA and KTH-S). Each line in the graphs represents a given algorithm variant with a given retry limit (5, 30 or 50) and displays the per node characteristics of nodes according to their physical location. The arrows on the graphs show the direction of increasing retry limit. These kinds of figures are used throughout the dissertation to visualize (un)fairness of nodes.

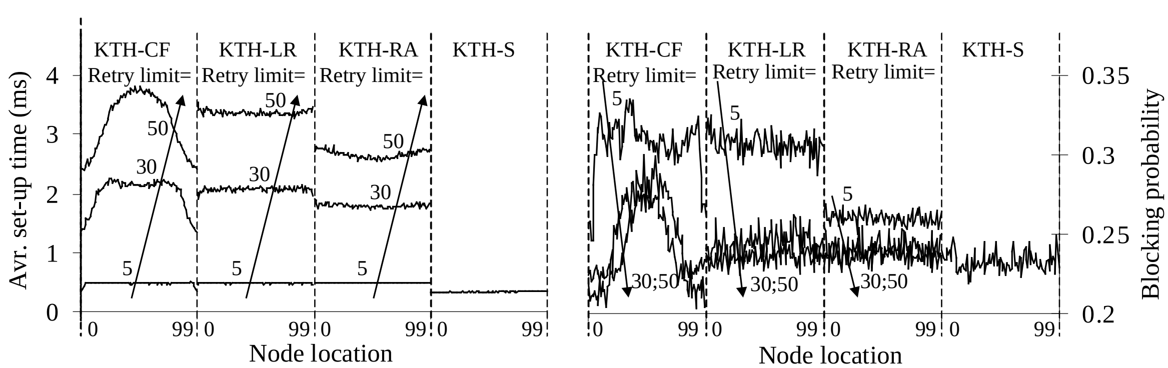


Figure 3.4.4 - Average set-up time and blocking probability, external model, short dual-bus, single direction

It can be seen in Figure 3.4.4 that all of the request orders are fair except KTH-CF algorithm. At KTH-CF algorithm, both blocking probability and connection set-up time is less for outer nodes than it is at the middle part of the dual-bus. The effect is more significant in case the retry limit is 30 and 50. Quantitative results, displayed in Table 3.4.3, show the same conclusions. That is, the fairness index of KTH-CF algorithm is less than 0.95 when retry limit is 30 and 50. In all the other cases the fairness index is greater than 0.95.

Algorithm	KTH-CF			KTH-LR			KTH-RA			KTH-S
Retry limit	5	30	50	5	30	50	5	30	50	-
Set-up time	0.997	0.983	0.978	1	1	1	1	1	0.999	1
Blocking probability	0.997	0.986	0.994	1	0.999	1	1	0.999	1	0.999

Table 3.4.3 - Fairness indices, external model, short dual-bus, both directions

Detailed analysis of KTH-CF algorithm

Unfairness of KTH-CF algorithm can be explained by an intuitive reasoning, using relations shown in Figure 3.4.5.

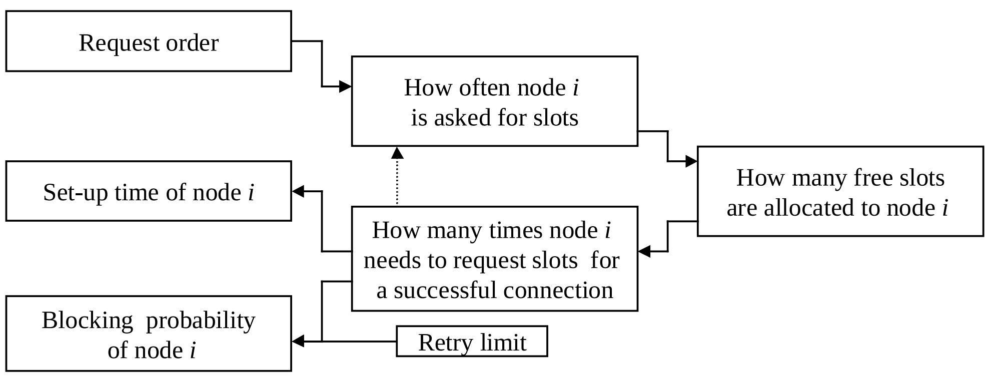


Figure 3.4.5 - Relation between different characteristics

The request order, which is different for different algorithms, determines how often are nodes asked for slots. In the case of KTH-CF algorithm, as we see it later, outer nodes are requested less frequently for slots. Nodes that are requested less frequently have more free slots in average. Nodes that have more free slots that the others have to ask slots more frequently form the others, which means that their set-up time is longer.

Due to the limit on the number of slot allocation retries, nodes are not able to ask all other nodes for slots. So calls can be blocked before the node have found a node with free slots. Therefore, blocking probability is less for nodes that ask nodes with – relatively – many slots before the retry limit is reached.

We have already seen the result of these effects in Figure 3.4.4. In the case of random and logical ring request order - and with external traffic load profile - nodes are asked for slots with the same frequency, so the resulting high level characteristics are fair. In the case of closest first request order nodes are exposed to uneven "slot request load", so the final characteristics are also uneven, i.e. the algorithm is unfair.

To illustrate these effects in the case of KTH-CF algorithm the following characteristics are *collected in the simulator*:

- $f(i,k)$ - the probability that node i has k free slots

- $F(i)=\sum_{k=0}^{\infty} k \cdot f(i,k)$ - the average number of free slots at node *i*
- *r(i,k)* - the probability that node *i* has to request slots *k* times during a successful connection set-up
- $R(i)=\sum_{k=0}^{\infty} k \cdot r(i,k)$ - the average number of requests needed for a successful connection set-up at node *i*
- *T(i)* - average set-up time of node *i*

With the wording of Figure 3.4.5 *F(i)* measures how many free slots are allocated to node *i* and *R(i)* shows how many times node *i* needs to request slots for a successful connection. The only missing box of Figure 3.4.5 should show how often node *i* is asked for slots. This number could be calculated knowing the operation of the algorithms. However, instead of using this number a new measure (average ordinal number) can be defined, which reflect the same effect. Average ordinal number can be calculated based on the definition of the algorithms.

To define the average ordinal number, first the concept of ordinal number should be defined. The ordinal number $n(i,j)$ is the number of nodes that node *j* asks in average before node *i* is reached. Note that $n(i,j)$ depends on the used algorithm. Averaging $n(i,j)$ over *j* gives the average ordinal number of node *i* (denoted by *N(i)*). In other words, an "average node" asks node *i* when it has already requested slots from other *N(i)* nodes. Small *N(i)* means that node *i* is often asked for slots.

Based on simple considerations $n(i,j)$ can be expressed from the definition of KTH-CF algorithm. Equation (3.4.3) shows $n(i,j)$ with the assumption that there are 100 nodes on the dual-bus.

$$n(i,j)=\begin{cases} i-1 & \text{if} \quad \mathrm{i>2j} \\ 99-i-1 & \text{if} \quad i<2j-99 \\ 2(i-j-1) & \text{if} \quad j<i \text{ and } \mathrm{i-j \le j} \\ 2(j-i-1) & \text{if} \quad j>i \text{ and } \mathrm{j-i \le 99-j} \\ 0 & if \quad i=j \end{cases} \tag{3.4.3}$$

Figure 3.4.6 gives hint of interpreting the different intervals by showing the first four ranges of expression (3.4.3). Nodes that are counted in $n(i,j)$ are marked in the figure (note that node *i* and node *j* never counts). To make the calculations easier, here we assumed that node *j* always asks first its neighbour in the direction of node *i*. That is, node *j* does not choose randomly between the directions in contrast to the definition in Section 3.4.1.2.

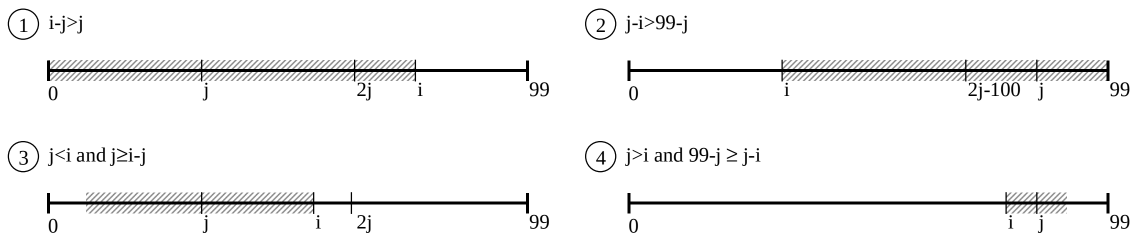


Figure 3.4.6 – Explanations to equation (3.4.3)

N(i) can be obtained with averaging over j: $N(i) = \sum_{j=0}^{99} n(i, j)/100$

Now *N(i)* is obtained from calculations and the other values shown in Figure 3.4.5 (*F(i)*, *R(i)* and *T(i)*) are known from simulations. Figure 3.4.7 shows the average values of each characteristic.

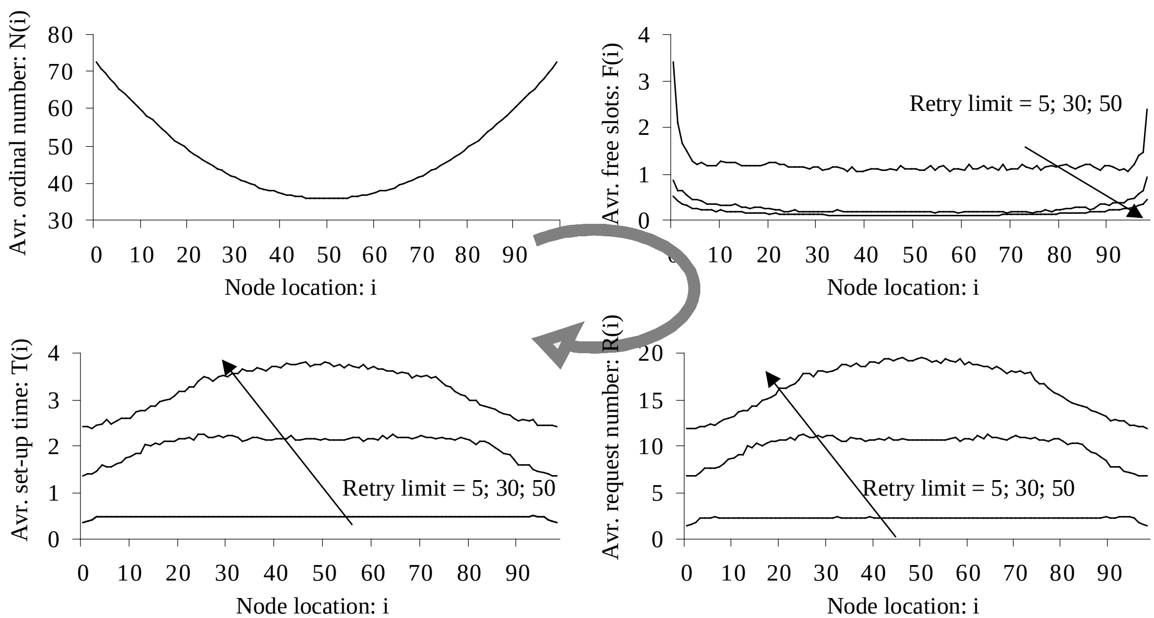


Figure 3.4.7 - N(i), F(i), R(i) and T(i) at the KTH-CF algorithm and retry limit of 5, 30 and 50

It can be seen in Figure 3.4.7 that nodes in the middle of the bus are oftener requested for slots than nodes at outer parts - *N(i)*. As a result middle nodes have less free slots in average than outer ones - *F(i)*. The average request number is lower at nodes closer to the ends of the bus than at middle nodes - *R(i)*. And finally, the curve of average set-up time has almost the same shape as that of average request number.

Conclusion

The conclusion of this section is that KTH-CF algorithm is *unfair* even in the case of the simple external load profile. All the other algorithms are *fair* in this scenario.

3.4.2.4 Short Bus - Client-Server Load Profile

This section analyzes the network operation based on the client-server network load profile and short bus. This is the best profile to examine fairness because it is complex enough to show unfairness and the results remain still interpretable. Therefore, this section gives the most detailed analysis.

Configuraion

The main host and node parameters are summarized in Table 3.4.4:

Node parameters		
	Client nodes	*Server node*
Number of control slots	1	10
Length of input buffer	150	1500
Length of output buffers	100	1000
Processing time of control messages	5μs	5μs

Host parameters			
	Distribution	*Parameter*	*Mean*
Holding time	Pareto	α=1.9; k=0.95	2 s
Interarrival time	Weibull	β=0.33; λ=20	0.3 s
Bandwidth	Deterministic	-	1 slot/cycle

Table 3.4.4 - Configuration parameters, client-server model, short dual-bus

Results

Figure 3.4.8 shows blocking probability of unidirectional connections directed to bus 0 for all algorithms. Results belonging to different algorithms are displayed side-by-side as in Figure 3.4.4. Average set-up times of the algorithms are displayed on Figure 3.4.9.

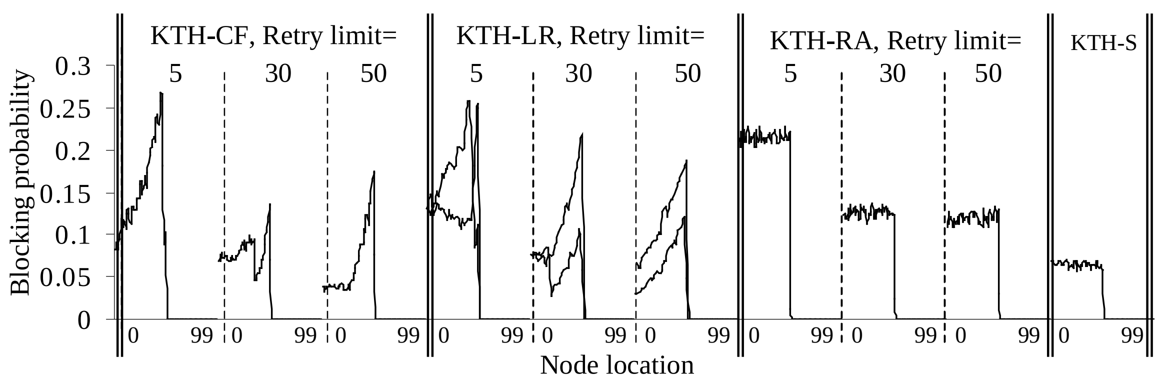


Figure 3.4.8 - Blocking probability, client-server model, short dual-bus, single direction

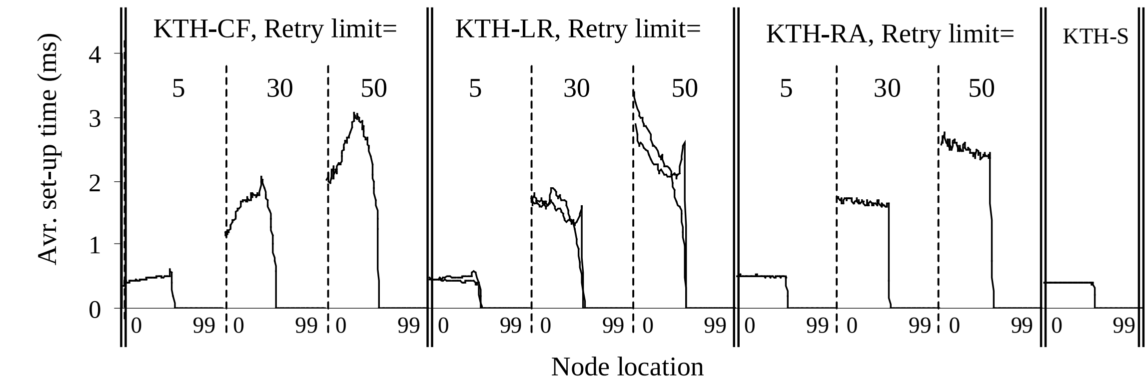


Figure 3.4.9 - Average set-up time, client-server model, short dual-bus, single direction

In Figure 3.4.10 and 3.4.11 the blocking probability and average set-up time values are based on all calls, i.e. directed to any direction of the dual-bus.

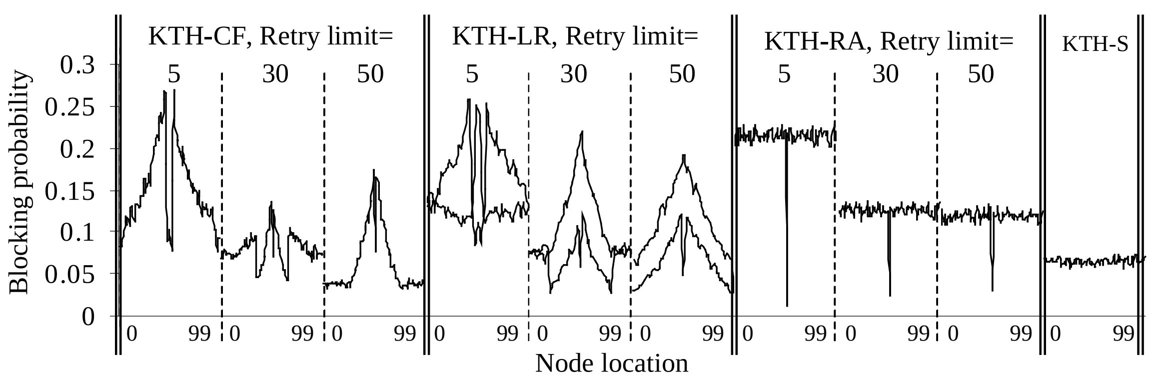


Figure 3.4.10 - Blocking probability, client-server model, short dual-bus, both directions

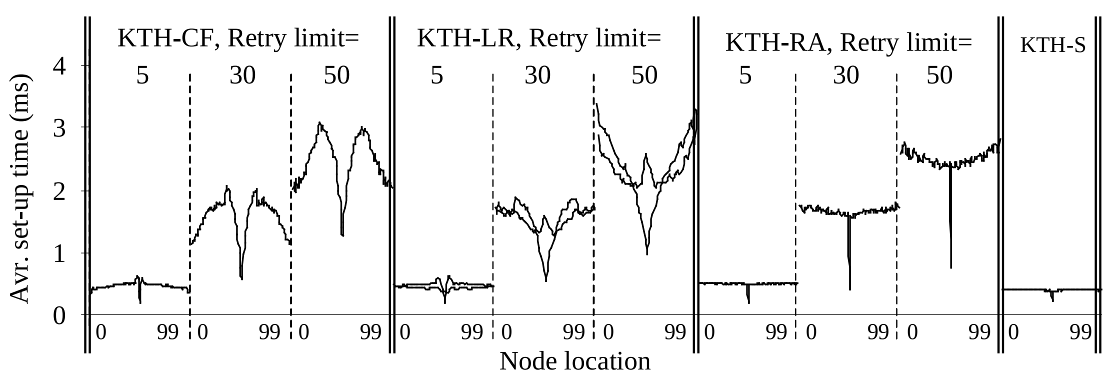


Figure 3.4.11 - Average set-up time, client-server model, short dual-bus, both directions

Table 3.4.5 shows fairness indices of the examined algorithms.

Algorithm	KTH-CF			KTH-LR			KTH-RA			KTH-S
Retry limit	5	30	50	5	30	50	5	30	50	-
Blocking probability	0.913	0.958	0.734	0.934	0.84	0.827	0.999	0.998	0.998	0.997
Set-up time	0.991	0.97	0.977	0.991	0.979	0.969	1	0.999	0.998	1

Table 3.4.5 - Fairness index based on client nodes, client-server model, short dual-bus, both directions

The characteristics of server node are hard to read in the figures, therefore it is extracted to Table 3.4.6 along with the average of client node characteristics. The averages coming from unfair algorithms are written with *italics* on grey background.

Algorithm	KTH-CF			KTH-LR			KTH-RA			KTH-S
Retry limit	5	30	50	5	30	50	5	30	50	-
Client blocking	*0.15*	*0.18*	*0.06*	*0.15*	*0.09*	*0.09*	0.21	0.13	0.12	0.065
Server blocking	0.10	0.07	0.08	0.11	0.06	0.05	0.01	0.02	0.03	0.06
Client set-up (ms)	*0.46*	*1.57*	*2.47*	*0.46*	*1.55*	*2.36*	0.50	1.66	2.51	0.39
Server set-up (ms)	0.18	0.57	1.25	0.18	0.53	0.97	0.18	0.41	0.74	0.2

Table 3.4.6 - Characteristics of server and client nodes, client-server model, short dual-bus, both directions

The two most important conclusions can be drawn from the above figures and tables. First, there is significant difference between the characteristics of client and server nodes. Blocking probability and average set-up time of the server node are lower than those of an average client. Second, significant unfairness can be observed at the KTH-LR algorithm, which was fair in the external load profile.

The first observation can be interpreted as asymmetry in the directions of the bi-directional connections, i.e. the downstream (from server to client) direction of the connections has better characteristics than that of upstream direction. With this interpretation, this difference is only asymmetry but not unfairness. This effect, however, could cause unfairness in other scenarios. The further discussion of the issue of asymmetry is postponed to Section 3.5.1 where the motivations of smoothing algorithms are described.

Client nodes have the same characteristics when using **KTH-S** algorithm. It is due to correct status tables, which yield to fair operation. Asymmetry can be observed here also: the server node has better characteristics than those of clients. The node of the server has much higher intensity then client nodes, therefore it collects most of the free channels. Having more free channels mean that less calls are blocked and set-up time is shorter. This effect, which causes asymmetry, is referred to as "cache" property of the DTM protocol.

Apart from asymmetry, **KTH-RA** algorithm is also *fair*.

KTH-LR and **KTH-CF** algorithms are *very unfair* in this scenario. Either blocking probability or average set-up time is different for clients with the same offered load at any retry limits.

Detailed analysis of KTH-LR algorithm

Although the detailed analysis of unfair algorithms does not change the main conclusion that they are unfair, an explanation of the results obtained for KTH-LR algorithm follows in this subsection. The reasons of unfairness of KTH-CF can be understood from that explanation and the detailed analysis of KTH-CF algorithm in Section 3.4.2.3.

The distance from the server node (very active node) and from idle nodes affects the behaviour of the client nodes of KTH-LR algorithm. It is hard to understand the reasons behind the characteristics of nodes based on Figures 3.4.8 and 3.4.9 because they display nodes at their physical locations. KTH-LR is based on a logical ring (ring was shown in Figure 3.4.1), therefore Figure 3.4.12 shows KTH-LR-30 algorithm in another view. In this figure nodes are displayed at their positions on the logical ring. There is no cut between node 0 and node 99 in the reality, they are neighbours on the logical ring.

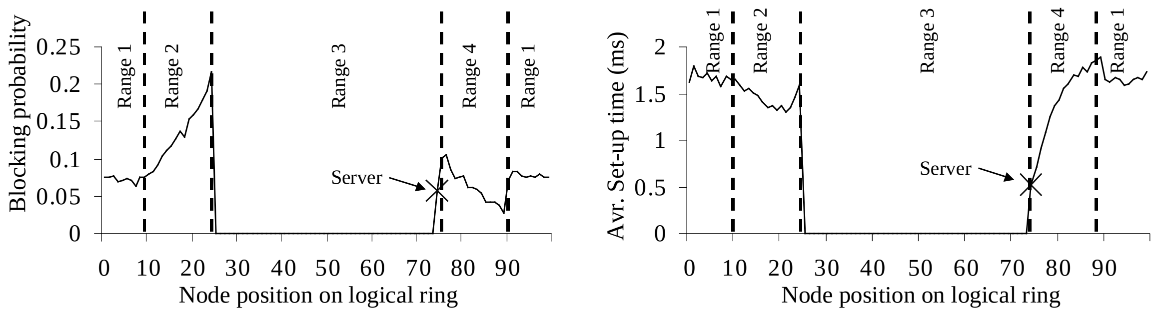


Figure 3.4.12: Characteristics of KTH-LR-30 algorithm, client-server model, short dual-bus, both directions

Based on this figure four ranges of client nodes can be identified apart from the server:

- range 1: active nodes which are far from the idle nodes and from the server (0-9, 91-99)
- range 2: active nodes which are close to the idle nodes and far from the server (10-24)
- range 3: idle nodes (25-74)
- range 4: active nodes which are close to the server (76-90)

The words far and close are used according to the following definitions:

- Two nodes are far in this context if they are not able to change slots directly (because of the limited number of retries).
- Nodes are close if they distance on the logical ring is less than the retry limit.

For the sake of better understanding the average number of free slots and slot requests in successful calls are displayed in Figure 3.4.13

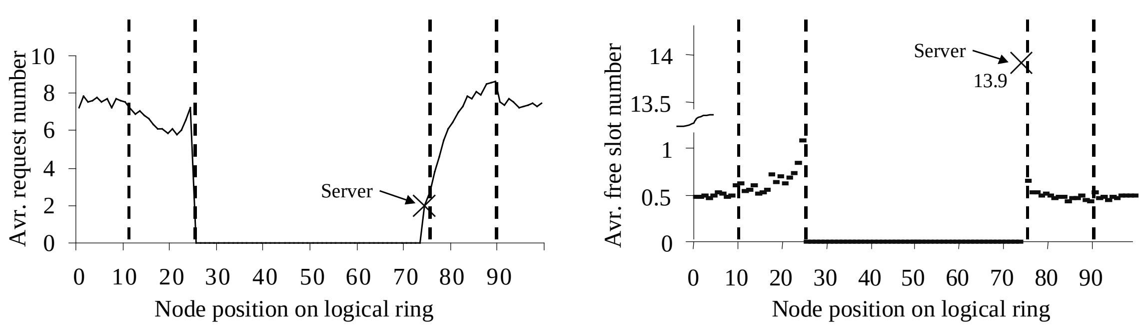


Figure 3.4.13 - Average number of free slots and slot requests (KTH-LR-30 algorithm)

In Figure 3.4.13 ranges, which were introduced above, are also indicated.

It is important to note that the server has 13.9 slots in average while this value is 0.6 for an average non-idle client node. So the server has more than 20 times more slots in average!

The shape of the request number curve is almost the same as that of connection set-up time. That is, buses of the dual-bus are synchronized correctly. Namely, the set-up time is the sum of the delay of the set-up message and the delay of the slot allocation process, which is the product of the slot-request number and the delay of a slot request-reply. In the case of short bus, the dominant factor in the delay of a slot request-reply is the waiting time for the next available control slot. If the dual-bus is properly synchronized then the response time is the same for each slot request independently of the location of the asked node.

Let us start the detailed analysis of the graphs with the relation between nodes in the same range and then have a look at the difference at range borders.

Range 3 is the range of idle nodes, so there is no attempt to establish connections. Blocking and set-up time has no value, but they are plotted as 0 in the figures.

In **range 4**, in the proximity of the server, the behaviour of nodes are effected by two facts:

1. Proximity of idle nodes: These nodes try to ask slots from idle nodes. It means that their effective retry limit, which counts only the non-idle nodes, is less than that of nodes in range 1. If the retry limit is 30, "effective retry limit" for node 76 (next to the server) is 16, because 14 nodes are idle in its request area. *The farther a node from the idle nodes is the lower the blocking probability is.*
2. Proximity of the server: Client nodes in range 4 can ask the node of the server for slots. The server node has 20 times more free slots due to the bursty traffic and the heavier activity. Therefore, it is very likely that client nodes in this range can get slot from the server. Therefore, the proximity of the server is advantageous for client nodes. *The closer a node to the server is the shorter the set-up time is.*

In **range 2**, due to the proximity of idle nodes, the farther a node from range 3 is the lower its blocking probability is. The shape of the average set-up time curve is caused by the border role (between range 1 and range 3) of range 2 nodes.

In **range 1** nodes are outside the range of effect of special nodes, therefore they have the same set-up time and blocking probability.

After having intuitive answers to differences of nodes within a range. Now let us see what is the cause of jumps at the borders of ranges.

Blocking of the **server** is lower than that of its active neighbours due to the ***cache effect***. It can also be called as starvation effect because the server node collects much more free slots than client nodes, therefore clients are starved for free slots.

At the **border of range 4 and range 1**, the node in range 4 has lower blocking probability because of the proximity of the server. The proximity of the server is the cause of the difference in the set-up times too. Node 90 has higher average set-up time than node 91 because it has many connections that are established with 30 retries while node 91 has no opportunity to ask slots from the server.

Ranges were presented based on KTH-LR-30 because in the case of retry limit of 30, ranges have nearly the same size. At other retry limits, some of the ranges are smaller or they are even missing:

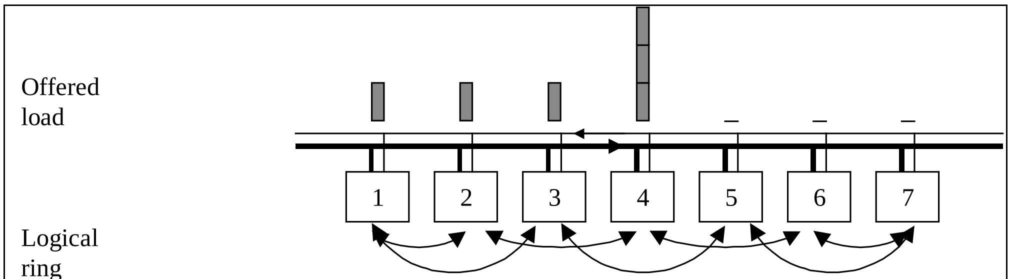
- if retry limit equals to 50 there is no range 1
- if there is no retry limit there is no range 1; and range 2 and 4 are merged to one range.

Tuning algorithms without status table

The effect of the proximity of idle nodes can be avoided if a small piece of intelligence is used in the algorithms operating without status tables. In the so-called *tuned* versions of the algorithms, nodes keep track of continuously idling nodes and they do not ask slots from them. Therefore, the "effective retry limit" is the same as the retry limit for each node.

Ordinal numbers of nodes from the viewpoint of the server (client-server load profile) are displayed in Figure 3.4.14 for all algorithms. Some of the nodes in the case of CF, KTH-LR and KTH-LR-t algorithms have two ordinal numbers. It is due to the random initial direction selection (see definition of algorithms in Section 3.4.1).

See the tuned version of KTH-LR algorithm. Assuming that the server starts with the right-hand nodes, the following order is obtained: As node 6,node 7 and node 5 have no hosts on the bus they are omitted (in that order); so the ring-neighbour of node 5, which is node 3, is taken first. The next one is node 2 because in the other direction along the ring it is the first node. The third one is node 1 as the only remaining.



Ordinal number measured from server node

CF	6 or 5	4 or 3	2 or 1	-	1 or 2	3 or 4	5 or 6
CF-t	3	2	1	-	-	-	-
LR	4 or 3	2 or 1	6 or 5	-	5 or 6	1 or 2	3 or 4
LR-t	3	2 or 1	1 or 2	-	-	-	-
RA	any	any	any	-	any	any	any
RA-t	any	any	any	-	-	-	-

Figure 3.4.14 - Order of nodes

Figure 3.4.15 shows the characteristics of tuned KTH-CF and KTH-LR algorithms.

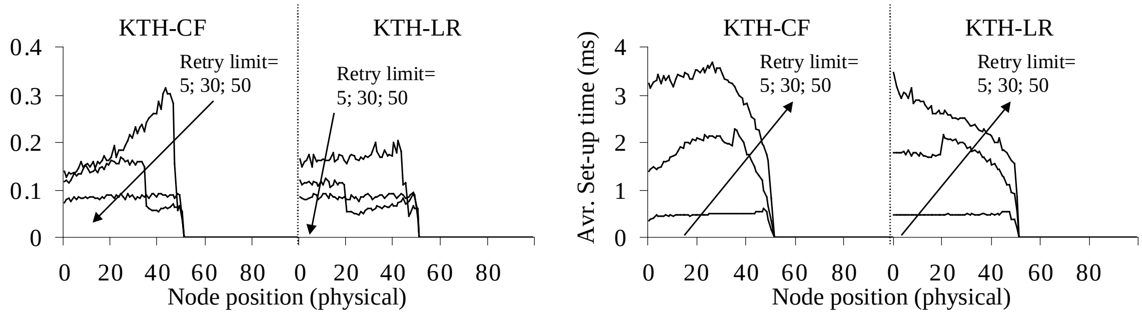


Figure 3.4.15 - Characteristics of tuned KTH-CF and KTH-LR algorithms

It can be seen that the effect of idle nodes disappeared from the plot of KTH-LR. Unfortunately, the algorithms remained unfair due to the effect of the server node.

Conclusion

It can be concluded from this scenario that in the case of the KTH-LR algorithm, very active nodes and very passive nodes can change the characteristics of their neighbouring nodes. Therefore, KTH-LR and KTH-CF algorithms are very unfair in this scenario. KTH-S and KTH-RA algorithms proved to be fair in this scenario too.

This section has shown that the drawback that is caused by the proximity of idle nodes can be avoided with a small improvement. Due to tuning of algorithms without status table, the "effective retry limit" of nodes is the same in all cases as the value of retry limit is.

3.4.2.5 Short Bus - Peer to Peer Load Profile

The last network load profile is based on peer-to-peer traffic. This section evaluates this profile at short bus-length.

The main properties of this load profile are:

- there is only a small difference between offered loads of neighbouring nodes directed to one direction (see later in Figure 3.4.17)
- offered load is different for every node

These properties are in contrast with client-server traffic, where there are three offered load levels (active client, passive client and server) and the server has much higher offered load than that of clients. Therefore, results here can not be explained by the influence of a single node or a small set of nodes.

Configuration

According to the description of the peer-to-peer model, each node has N-1 hosts where N is the number of nodes on the bus. That is, 9900 hosts are needed for a 100-node network. Due to the memory limitation of the computer running the simulation, the running time of simulations increased dramatically. To achieve acceptable running time, a 25-node network is simulated at the peer-to-peer load profile.

Node parameters			
Number of control slots	1		
Length of input buffer	150		
Length of output buffers	100		
Processing time of control messages	5µs		
Host parameters			
	Distribution	*Parameters*	*Mean*
Holding time	Pareto	α=1.9; k=3.79	8 s
Interarrival time	Weibull	β=0.33; λ=5.9	1 s
Bandwidth	Deterministic	-	1 slot/cycle

Table 3.4.7 - Configuration parameters of peer-to-peer traffic

The main configuration parameters are summarized in Table 3.4.7.

Results

First, Figure 3.4.16 shows the results, which are based on one of the directions of the dual-bus. The offered load in the observed direction decreases from node 0 to node 24. Retry limit is set to 5, 15 and 24 at algorithms without status table.

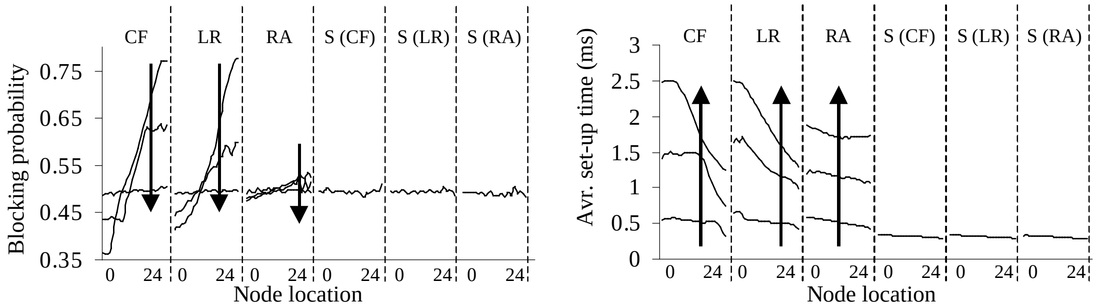


Figure 3.4.16 - Blocking probability and avr. set-up time, peer-to-peer model, short dual-bus, single direction

KTH-S and KTH-RA algorithm are *fair* so as in the previous scenarios. Both characteristics of KTH-CF and KTH-LR algorithms are uneven along the bus. Due to the cache effect described with the client-server profile, blocking probability is bigger for nodes with lower offered load. Set-up time, however, is lower at nodes with lower offered load.

Uneven free slot distribution can be in the background of the unfair blocking probability, so it is displayed in Figure 3.4.17.

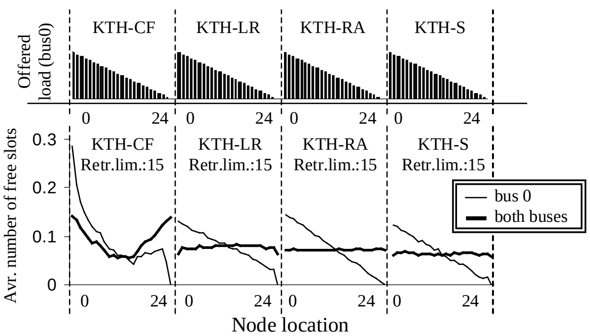


Figure 3.4.17 - Avr. number of free slots, peer-to-peer model, short dual-bus

Equilibrium free slot distribution on bus 0 is proportional to offered load at all algorithm except KTH-CF. At KTH-CF, average free slot distribution can be obtained as the superposition of a linear curve and the free slot distribution obtained at external load profile (Figure 3.4.7) due to the uneven average ordinal number. Considering both directions of the dual-bus, free slot distribution of KTH-CF algorithm is the worse, where outer nodes have almost twice free slots in average than the middle one.

According to the definition of fairness at the beginning of Section 3.4.2.1), the characteristics of nodes *with the same offered load*, should be compared. Figure 3.4.18 shows blocking probability and set-up time based on all connections. Characteristics are uneven at each algorithm without status tables. Both characteristics of outer nodes are better than those of middle ones are.

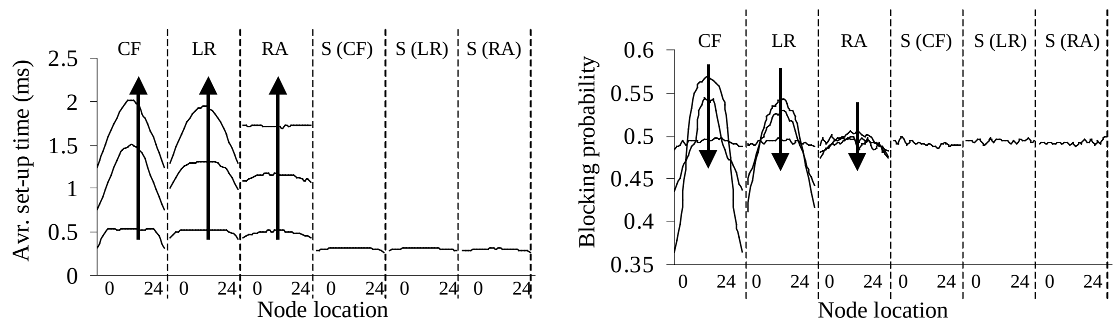


Figure 3.4.18 - Avr. set-up time and blocking probability, peer-to-peer model, short dual-bus, both directions

Finally, the fairness index of the algorithms can be calculated based on data displayed in Figure 3.4.18. Fairness values are shown in Table 3.4.8.

Algorithm	KTH-CF			KTH-LR			KTH-RA			KTH-S
Retry limit	5	15	24	5	15	24	5	15	24	-
Set-up time	0.982	0.956	0.977	0.997	0.993	0.998	0.997	0.999	1	0.998
Blocking probability	0.977	0.995	1	0.993	0.997	0.999	1	1	1	1

Table 3.4.8 - Fairness index, peer-to-peer model, long dual-bus, both directions

Conclusion

KTH-RA and KTH-S algorithms are *fair* at each setting. KTH-LR algorithm is *very close to fair*: only two *unfair* ratings are obtained, the other results are *fair*. KTH-CF algorithm is unfair: *very unfair* rating is obtained three times.

3.4.2.6 Long bus - External Load Profile

The fairness of a DTM dual-bus with inter-node distance of 10 m has been evaluated so far.

To evaluate the effects of longer propagation time, the following three sections go through the same steps as the previous three: It examines the fairness of the network in the case of external, client-server and peer-to-peer network load profiles. However, the inter-node distance is 10 km is these sections.

Configuration

This section is about external load profile at long bus. Configuration of nodes and hosts is the same as the configuration of the external profile in the case of short bus-length. Configuration parameters are summarized in Table 3.4.2.

Results

Blocking probability and average connection set-up time are shown in Figure 3.4.19 and Figure 3.4.20. Six algorithms are displayed side by side: closest first, logical ring and random request orders without status tables (KTH-CF, KTH-LR, KTH-RA) and those with status tables (KTH-S-CF, KTH-S-LR, KTH-S-RA). The slot allocation retry limit was configured to 5, 30 and 50.

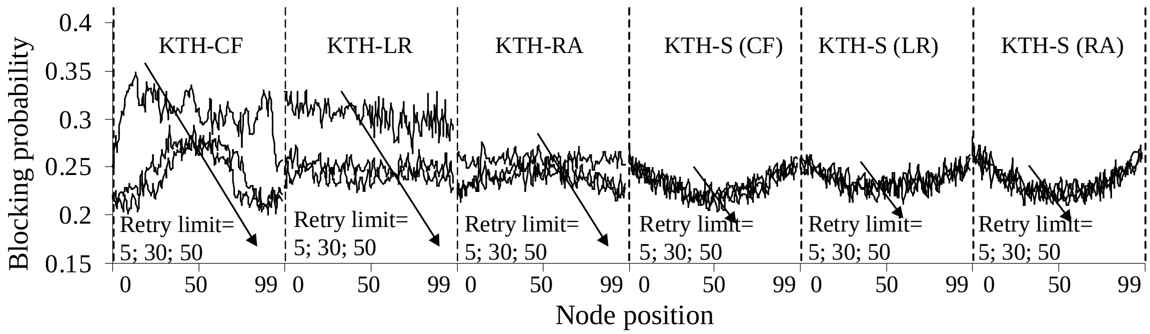


Figure 3.4.19 - Blocking probability, external model, long dual-bus, both directions

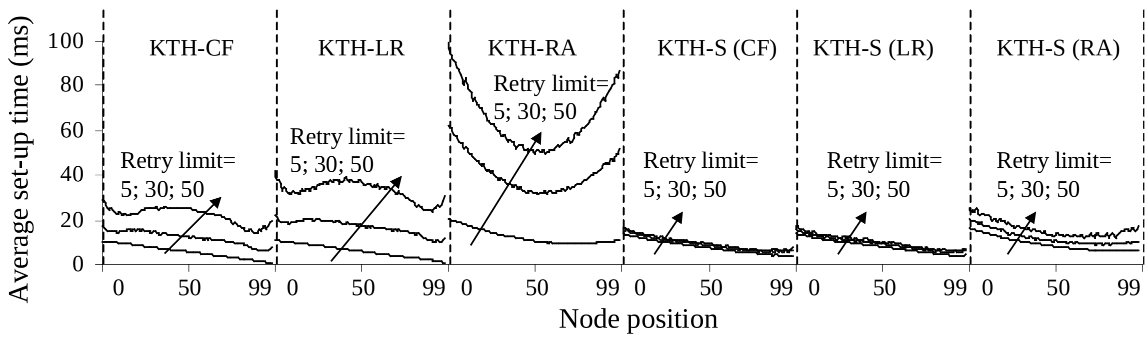


Figure 3.4.20 - Average set-up time, external model, long dual-bus, both directions

First, let us see the blocking probability. The fairness characteristics of algorithms without status table do not differ significantly from those in the case of short bus-length, so there is no need to explain the results again. However, unfairness of algorithms with status table needs explanation because it is the result of the long bus-length.

The next characteristic is connection set-up time, which consists of two parts:

- round-trip delay of the set-up message
- delay coming from slot requests

The first part depends on the location of the other party of the connection. As the switching node is at the end of the bus in our scenario, this factor is proportional to the distance measured from the end of the bus. As differences caused by the physical distance of the other party of the connection are usually respected by customers, the delay of set-up message and its acknowledgement is subtracted from the connection set-up time. The resulted new characteristic is referred to as average slot request time. Average slot request time is analyzed after the detailed evaluation of the blocking probability of KTH-S algorithms.

Detailed analysis of blocking probability of KTH-S algorithms

If status tables of nodes at KTH-S algorithm were up-to-date, each 1-slot connection would have been established or blocked with at most one slot request. Each node is allowed to ask any other, so there would be no difference between the blocking of nodes. In this case, however, status tables are outdated. To illustrate it Figure 3.4.21 displays the average number of slot requests in blocked calls. (It should be below 1 in optimal case.)

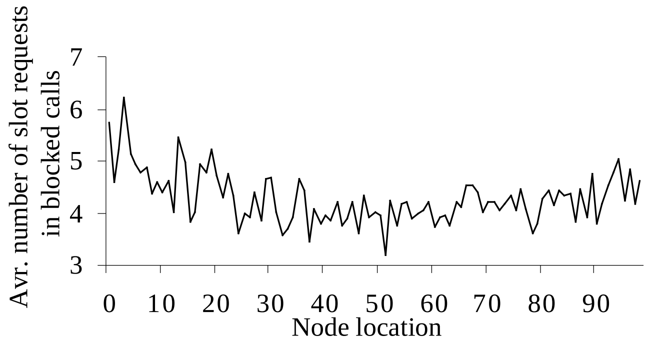


Figure 3.4.21 - Average number of slot requests in blocked calls, external model, long dual-bus, both directions (KTH-S-RA with retry limit of 30)

The average is above 3 for each node. It shows that there are many inconsistencies in status tables. The other interesting effect is that this value is higher for outer nodes. It signs that outer nodes have worse status tables, which can be explained with the so-called average distance.

- *d(i,k)* - the distance of node *i* and node *k*
- $D(i) = \sum_{k=0}^{99} d(i,k)$ - average distance of node *i*

D(i) decreases when walking from the outer part of the bus to the middle node. *D(i)* is minimal for the middle node. The consistence of status tables depends on the delay of status table update messages, which is proportional to *D(i)*. Unfairness is due to the propagation delay *of update messages*, which is independent of the request order. Therefore blocking probability of KTH-S algorithms is also independent of the request order (see Figure 3.4.21).

Detailed analysis of average slot request time

Average slot request time is shown in Figure 3.4.22 for each algorithm. Note that the scale of the vertical axis of algorithms with status table is 10 times lower than that of algorithms without status table is. Unfairness can be observed at each examined algorithm, but KTH-RA is the worst one both with and without status table.

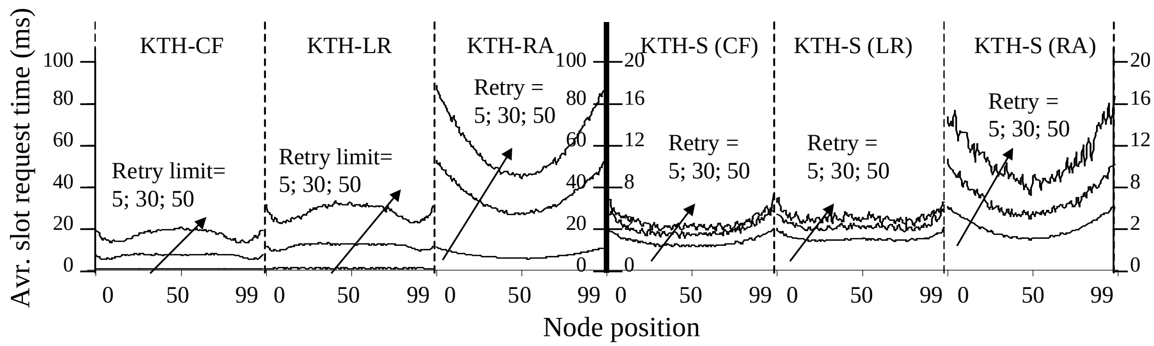


Figure 3.4.22 - Average slot request times, external model, long dual-bus, one direction

To explain the results of algorithms without status table, a new variable is defined:

$D_E(i) = \sum_{k \in E} d(i,k)$ - average distance of node *i* within its effective area *E* where effective area is the set of nodes from which node *i* is able to ask slots. *d(i,k)* is normalized so that *d(i,i+1)*=1, i.e. distance of neighbouring nodes is 1.

The effective area depends on the request order and the slot allocation retry limit. If retry limit is *r* at KTH-CF and KTH-LR algorithms there are only *r* nodes within the effective area. Every node is within the effective area of any node in the case of KTH-RA request order, according to the definition. $D_E(i)$ can be easily calculated for each request order and for retry limit of 5, 30 and 50. The result of the calculations is shown in Figure 3.4.23.

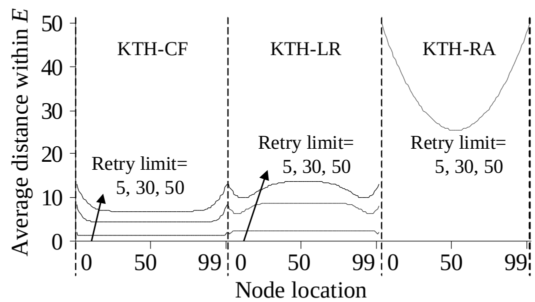


Figure 3.4.23 - Average distance within the effective area

At KTH-LR and KTH-RA, the calculated $D_E(i)$ and the simulated slot request time have the same characteristics, which shows that average distance within the area determines the slot request time.

The results of KTH-CF algorithm can be explained by the common effect of the average distance within the effective area and the uneven free slot distribution, which was already shown at short bus.

Differences in connection set-up times of nodes using KTH-S algorithms are due to the differences in the average distances. Because of the shorter average distances of middle nodes, their status tables are more accurate than those of outer nodes are. Better status tables yields to shorter connection set-up times.

Conclusion

Finally, Table 3.4.9 summarizes the fairness index of the examined algorithms concerning blocking probability and slot request time.

Algorithm	KTH-CF			KTH-LR			KTH-RA		
Retry limit	5	30	50	5	30	50	5	30	50
WITHOUT STATUS TABLE									
Slot request time	0.991	0.986	0.982	0.997	0.99	0.987	0.957	0.958	0.957
Blocking probability	0.995	0.988	0.992	0.998	0.999	0.999	0.999	0.999	0.998
WITH STATUS TABLE									
Slot request time	0.974	0.971	0.975	0.992	0.99	0.991	0.951	0.956	0.958
Blocking probability	0.996	0.998	0.997	0.998	0.998	0.998	0.997	0.996	0.996

Table 3.4.9 - Fairness index, external model, long dual-bus, both directions

Blocking probability is *fair* at each algorithm except KTH-CF without status table. Slot request time is *very unfair* at KTH-CF (with status table) and KTH-RA (with status table) and only *unfair* at both variants of KTH-LR algorithm and all algorithms without status tables.

As a conclusion it can be said, that the none of the algorithms is fully fair. The **best algorithm is the KTH-LR in this case**.

3.4.2.7 Long bus - Client-Server Load Profile

Configuration

The configuration parameters of the client-server load profile are displayed in Table 3.4.4. The only difference is that node-to-node distance is increased to 10 km from 10 m in this scenario.

Results

Simulation results are displayed in Figure 3.4.24, Figure 3.4.25 and Table 3.4.10.

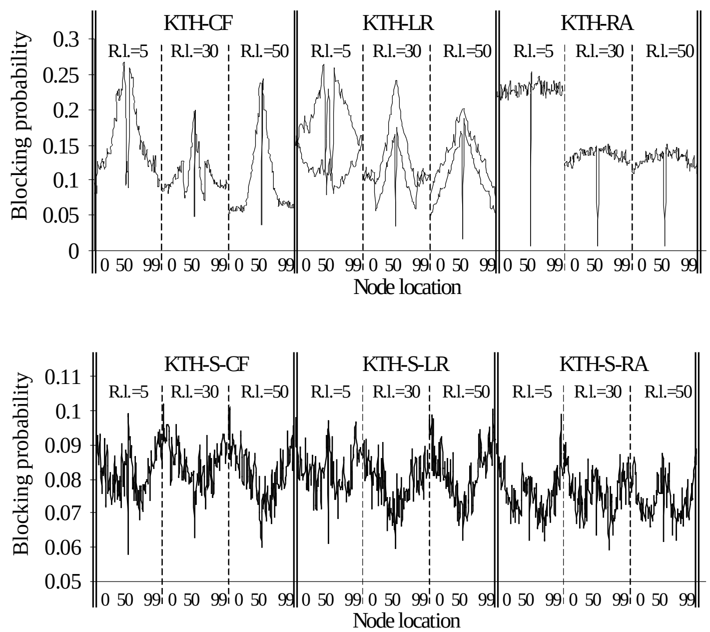


Figure 3.4.24 - Blocking probability, client-server model, long dual-bus, both directions

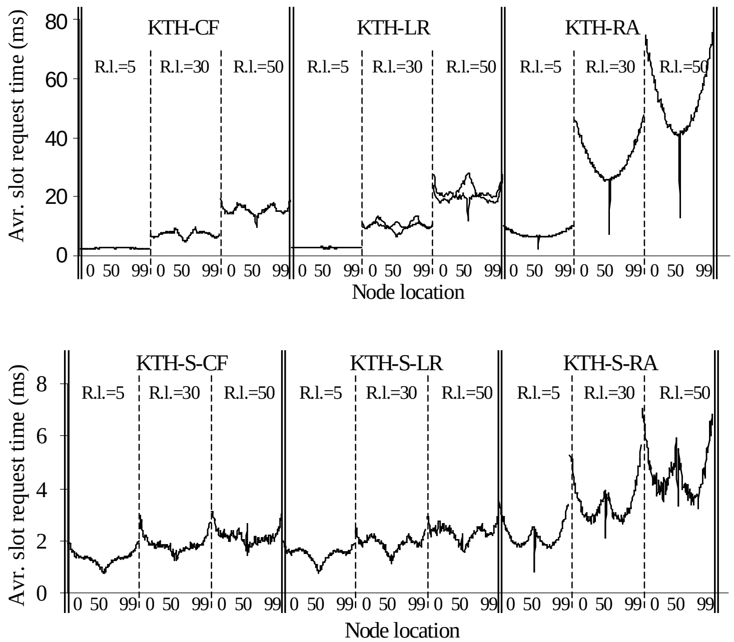


Figure 3.4.25 - Average slot request time, client-server model, long bus, both directions

Algorithm	KTH-CF			KTH-LR			KTH-RA		
Retry limit	5	30	50	5	30	50	5	30	50
WITHOUT STATUS TABLE									
Slot request time	0.968	0.967	0.99	0.967	0.978	0.984	0.962	0.959	0.96
Blocking probability	0.911	0.95	0.778	0.919	0.903	0.907	0.999	0.997	0.995
WITH STATUS TABLE									
Slot request time	0.965	0.973	0.981	0.976	0.982	0.984	0.96	0.962	0.964
Blocking probability	0.993	0.995	0.992	0.994	0.993	0.991	0.993	0.994	0.994

Table 3.4.10 - Fairness index, client-server model, long dual-bus, both directions

First, let us analyze **algorithms without status tables**. Blocking probability of algorithms did not change significantly due to the increased dual-bus length. Blocking of nodes at short and long bus-length is very similar (see Figure 3.4.24 and Figure 3.4.10). Characteristic of average slot request time (Figure 3.4.25) can be explained as the common effect of long bus-length and uneven load profile. The effects determining the shapes of the curves were discussed in previous sections. Figure 3.4.22 showed the effect of long bus and Figure 3.4.11 displayed the influence of active and passive nodes on the others.

Though performance of **KTH-S algorithms** is better, they are not fair at this network environment. Blocking probability is in the *unfair* category and slot request time is *unfair* or *very unfair*. There is no significant difference in the blocking probability of different request orders. Average slot request time, however, depends on slot request order (CF, LR or RA). Due to the higher average distance, KTH-RA has worse performance than the other algorithms have. Shape of the curves of the same algorithms with status tables and without them is very similar, because the main factor that forms the curves is different average distance of nodes (due to long bus).

Conclusion

There is **no winner** in this case, as (except blocking of KTH-RA without status table) there is **no fair algorithm**.

3.4.2.8 Long bus - Peer-to-peer Load Profile

Configuration

Finally, peer-to-peer load profile is checked again, this time with long dual-bus. Configuration of nodes and hosts are displayed in Table 3.4.7.

Results

Figure 3.4.26, Figure 3.4.27 and Table 3.4.11 show simulation results based on both directions of the dual-bus.

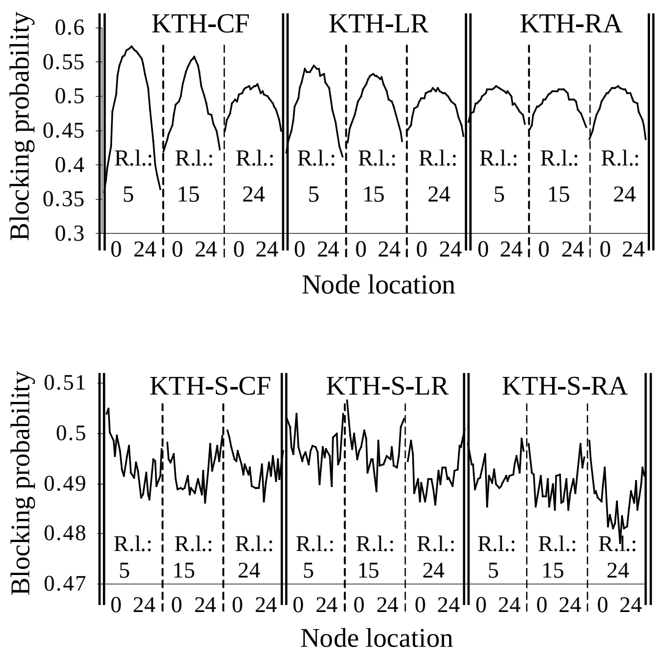


Figure 3.4.26 - Blocking probability, peer-to-peer model, long dual-bus, both directions

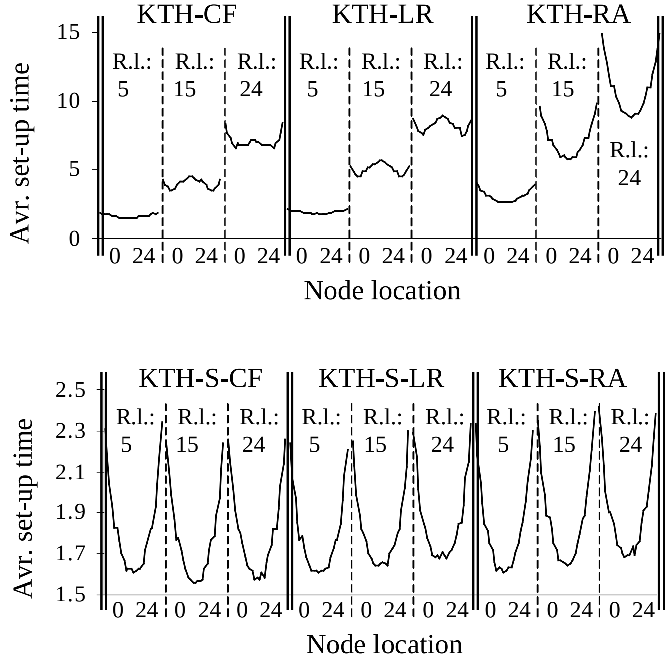


Figure 3.4.27 - Average set-up time, peer-to-peer model, long dual-bus, both directions

Algorithm	KTH-CF			KTH-LR			KTH-RA		
Retry limit	5	30	50	5	30	50	5	30	50
WITHOUT STATUS TABLE									
Set-up time	0.994	0.993	0.996	0.996	0.995	0.997	0.981	0.972	0.971
Blocking probability	0.977	0.993	0.998	0.993	0.996	0.998	0.999	0.999	0.998
WITH STATUS TABLE									
Set-up time	0.986	0.986	0.986	0.989	0.989	0.989	0.986	0.986	0.987
Blocking probability	1	1	1	1	1	1	1	1	1

Table 3.4.11 - Fairness index, peer-to-peer model, long dual-bus, both directions

According to fairness indices, each algorithm has fair blocking probability (except KTH-CF and KTH-LR with retry limit=5). In spite of good fairness indices, a small difference can be observed between blocking probability of nodes.

In the case of algorithms *with status table* blocking decreases when going to the middle of the bus. The unfairness of blocking probabilities is related to *D(i)*, because it is proportional to the average propagation delay of status table update messages. It does not depend on the algorithm, therefore blocking probability curves of different algorithms are similar.

At algorithms *without status table* blocking increases when going to the middle of the bus.

Among algorithms without status table, average set-up time is *fair* in the case of KTH-LR and *close to fair* at KTH-CF. It is *unfair* at any other algorithms. Average distances of nodes determine the set-up time when nodes do not use status tables. Set-up time of algorithms with status tables depends on the consistency of status tables, which is better in the middle of the dual-bus.

Conclusion

This load profile is not as big challenge for the network as client-server is. There is no *very unfair* algorithm. The only fair algorithm is the KTH-LR without status table in this scenario.

3.4.3 Study of Aggregate Performance Characteristics

Fairness evaluation of the algorithms has shown that the most challenging network load profile is the one based on client-server traffic. Therefore, this section is based on this type of network set-up. Characteristics of up-link connections (from a client to the server) and down-link connections (from the server to any client) are averaged separately. Characteristics of bi-directional connections are calculated from these averages.

Although only the fairness of different algorithms were discussed in Section 3.4.2, it was obvious from the figures that the number of allowed slot allocations (retry limit) and presence or absence of status tables significantly influence the performance of the network.

First, the evaluation of the effect of retry limit is described in Section 3.4.3.1. Different effects on the characteristics of the algorithms at fixed retry limit are shown in Section 3.4.3.2.

3.4.3.1 Effect of Retry Limit

Results presented in this subsection are based on the simulation of KTH-RA algorithm without status table. Though exact results belonging to other request orders differ, the characteristic of dependence on retry limit of those algorithms is the same.

Figure 3.4.28, Figure 3.4.29 and Figure 3.4.30 show the result of simulations. Figure 3.4.28 and 3.4.29 displays the characteristics of one of the directions of connections. Characteristics of client-to-server and server-to-client connections are presented in Figure 3.4.28 and Figure 3.4.29, respectively. Characteristics of bi-directional connections can be seen in Figure 3.4.30.

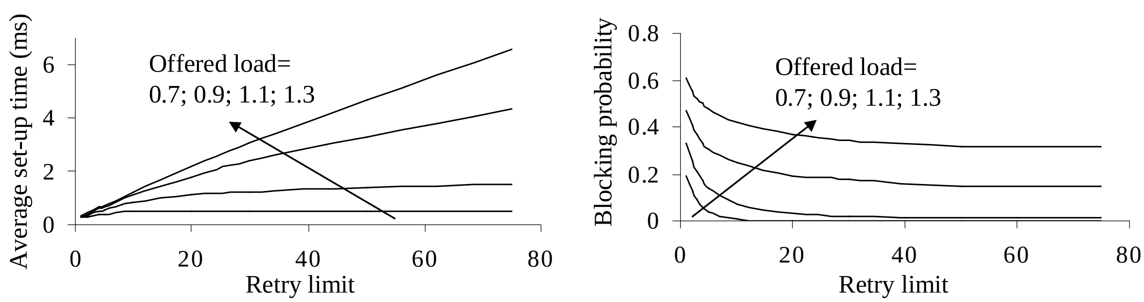


Figure 3.4.28- Effect of retry limit on connections from clients to server, KTH-RA algorithm, short bus-length, client-server profile, no status table

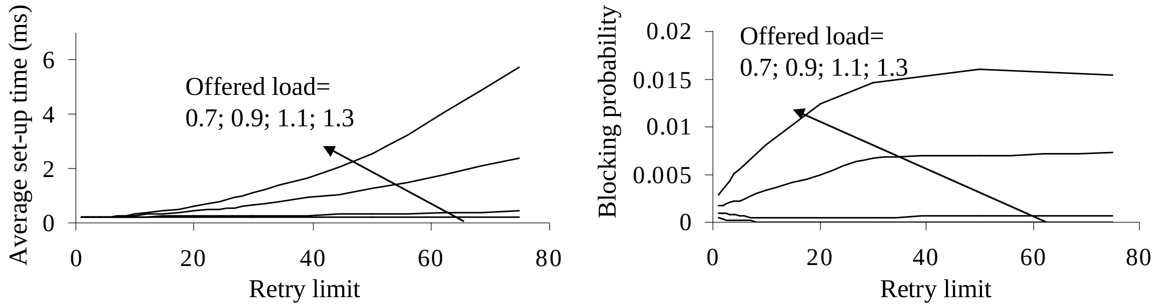


Figure 3.4.29 - Effect of retry limit on connections from server to clients, KTH-RA algorithm, short bus-length, client-server profile, no status table

Figure 3.4.28 and Figure 3.4.29 show an interesting effect. Blocking probability of server and client nodes is closer to each other when retry limit is higher, i.e. blocking of client nodes decreases and that of server node increases. It is interesting because at lower retry limits blocking of server nodes is lower despite of the fact that blocking of the whole system is higher. That is, ***low retry limit makes cache effect stronger***.

Set-up time of client nodes and that of server node move almost together. Both of them increase with increased retry limit.

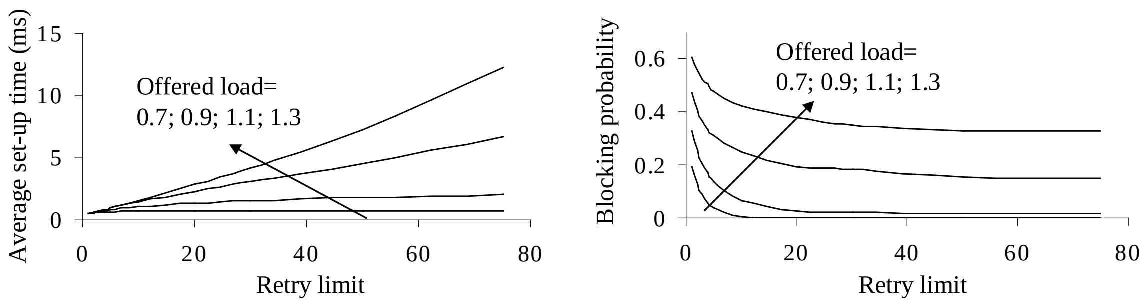


Figure 3.4.30 - Effect of retry limit on bi-directional connections, KTH-RA algorithm, short bus-length, client-server profile, no status table

Applying a retry limit has the opposite effect on average set-up time and blocking of bidirectional connections. If lower retry limit is applied then blocking probability increases and set-up time decreases. Figure 3.4.30 helps in finding the compromise between the two most important performance characteristics of the system. Set-up time of bidirectional connections was calculated as the sum of set-up times for both directions. A bi-directional connection is assumed to be blocked if any direction of the call is blocked.

Blocking probabilities of bi-directional connections decrease almost exponentially if the number of allowed slot allocation retrials increases. In the case of lower offered loads, the gradient of the blocking curves is bigger, in other words it increases faster. There is no blocking at the offered load level of 50% if retry limit is higher than 30.

The shape of the set-up time vs. retry limit curve depends on the load of the system. At low offered load (50-70%) the limit has a minor effect on set-up time. At higher offered loads (110%) it is closely proportional to retry limit.

The optimal operation of the system depends on the specific requirements. If set-up time is more important than throughput, a lower retry limit can be chosen. If keeping blocking on a low level is the highest priority, then a higher retry limit can be applied. As blocking converges fast to a value, it is advisable - in a general case - to chose retry limit to a value where blocking approached its minimum value. In Figure 3.4.30 it can be seen that when retry limit equals to 10 blocking is almost at its minimum value in all cases. Choosing a higher retry limit increases average set-up time and does not decrease blocking. Optimal limit is different for every offered load condition.

3.4.3.2 Performance at Fixed Retry Limit

In this section client-server load profile is used and retry limit is fixed to 10. KTH-LR and KTH-CF algorithms are simulated in the following circumstances:

- with and without status table
- with short (1 km) and long (100 km) bus
- with smooth (Poisson) and bursty (WWW) traffic
- with 5 different offered load settings between 50% and 130%

The task of this section is to evaluate the effect of listed parameters and to compare KTH-CF and KTH-LR algorithms.

Table 3.4.12 and Table 3.4.13 include average set-up time and blocking probability of bi-directional connections, respectively.

Set-up	Short Poisson				Short Bursty				Long Bursty			
Load	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA
50%	0.27	0.26	0.27	0.26	0.32	0.35	0.32	0.32	0.83	0.99	0.83	0.86
70%	0.27	0.27	0.27	0.27	0.44	0.47	0.37	0.36	0.98	1.36	0.89	0.95
90%	0.39	0.4	0.32	0.32	0.69	0.71	0.44	0.42	1.28	2.04	1	1.07
110%	0.81	0.75	0.44	0.44	0.91	0.91	0.5	0.47	1.57	2.6	1.16	1.2
130%	1	0.88	0.46	0.46	1.05	1.04	0.52	0.5	1.75	2.93	1.25	1.26

Table 3.4.12 - Avr. set-up time, bi-directional connections, client-server profile (in milliseconds)

Blocking	Short Poisson				Short Bursty				Long Bursty			
Load	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA	KTH-LR	KTH-RA	KTH-S-LR	KTH-S-RA
50%	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
70%	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
90%	0.00	0.01	0.00	0.00	0.07	0.08	0.02	0.02	0.07	0.08	0.02	0.02
110%	0.20	0.22	0.19	0.18	0.26	0.27	0.21	0.21	0.26	0.27	0.21	0.21
130%	0.42	0.45	0.41	0.41	0.43	0.46	0.41	0.41	0.44	0.47	0.42	0.41

Table 3.4.13 - Blocking probability, bi-directional connections, client-server profile

Status table

The effect of status table on performance is the most obvious conclusion of this section. In all examined settings algorithms with status table perform better than the same algorithms without tables. In other configurations - where short but frequent calls are sent to the bus and signaling bandwidth becomes the bottleneck - the performance of algorithms with status table degrade faster because maintaining status table requires extra control capacity [J2].

The biggest difference in average set-up time is 1.67 ms. That is, set-up time of KTH-RA at 130% load and long bus improved with 57% due to status table. The biggest difference in blocking probability is 0.06 in many cases, which is not as significant improvement as that of set-up time is.

The gain of status tables is bigger if offered load is higher as it can be seen mainly at set-up times. At higher offered loads there are less free slots in the system, so "guessing" is not as effective as it is at low system loads.

Bus-length

In Section 3.4.2.3, the effect of bus-length on fairness was studied. It was concluded that KTH-RA and KTH-S algorithms become unfair. Here, the performance of these algorithms is examined. Average set-up time increased obviously as inter-node distances are increased. It is interesting that though bus-length is 100 times more, set-up time increase at long bus is about 2 times of the value at short bus.

Blocking probability is almost independent of the bus-length. Only a very small increase can be observed at 130% offered load.

Burstiness

As it can be expected, performance characteristics of the network having bursty sources are worse than those of the network with smooth sources are. Both average set-up time and blocking probability decreased due to sources generating smoother traffic. The biggest improvement is 0.31 ms in average set-up time (KTH-RA without status table, 90% offered load) which is 43 % compared to the case of bursty source. The biggest improvement in blocking probability is 0.07 (KTH-RA without status table, 90% offered load).

Which request order?

In the examined circumstances the performance of logical ring and random request order algorithms is the same. There is only one exception: average set-up time of KTH-RA without status table is higher than that of KTH-LR. As both fairness and performance of KTH-RA algorithm is worse than those of KTH-LR at long buses, the usage of KTH-RA algorithm is only advised at short buses.

3.4.4 Conclusions on Set-up Time Slot Allocation Algorithms

3.4.4.1 Fairness

A comprehensive study has been performed to investigate the main environmental and algorithmic variables effecting the fair operation of nodes located on a DTM dual-bus. It has been found that

- the **order of slot requests** sent out during connection set-up
- the presence or absence of **status tables**
- and the **length** of the DTM bus

are the main factors causing unfairness.

In the case of unfair networks the performance difference of nodes depends on

- the **load** and **burstiness** of offered traffic
- and **upper limit on the number of slot requests** nodes are allowed to send out during a connection set-up (retry limit).

This thesis analyzes simulation results for the main variants of set-up-time slot allocation algorithms. Fairness of algorithms is studied in three different network configurations:

1. In “external” network configuration each node communicates with a dedicated node at the end of the dual-bus
2. In “client-server” network configuration each node initiates connections to the “server” node in the middle of the dual-bus.
3. Each node establishes connection with equal probability to any other node on the dual-bus in the case of “Peer-to-peer” model

Configurations were tested with short and long bus-length.

The summary of fairness results is shown in Table 3.4.14. Results in the cells are averaged from the 6 results (3 retry limits and two characteristics) presented in Sections 3.4.2.3-3.4.2.8.

	Short bus				Long bus					
	KTH-CF	KTH-LR	KTH-RA	KTH-S	KTH-CF	KTH-LR	KTH-RA	KTH-S-CF	KTH-S-LR	KTH-S-RA
External	unfair	fair	fair	fair	unfair	fair	very unfair	unfair	fair	very unfair
Client-server	very unfair	very unfair	fair	fair	very unfair	very unfair	very unfair	unfair	unfair	very unfair
Peer-to-peer	unfair	fair	fair	fair	unfair	fair	unfair	unfair	fair	unfair
Conclusion	unfair	unfair	fair	fair	unfair	unfair	very unfair	unfair	fair	very unfair

Table 3.4.14 - Summary of fairness study

Networks using **KTH-CF** channel allocation algorithm are *unfair.* The unfair operation is due to the following facts:

- In the case of short bus, nodes at the ends of the dual-bus are in a more favourable situation even in the case of external load model because they are asked less frequently for slots and consequently they have more free slots in average. These nodes have lower blocking probability and/or average connection set-up time depending on the reallocation retry limit.
- Active nodes influence the performance of their neighbours because of deterministic channel request order.
- In case of long dual-bus, the average distance from other nodes in the effective area[2] is bigger for nodes at the ends of the dual-bus than it is for ones in the middle. This effect mainly influences average slot request time, and it decreases the difference between middle and outer nodes. Due to this effect fairness of KTH-CF does not degrade significantly due to long distances.

[2] If node *i* can ask slots from node *j* then node *j* is in the effective area of node *i*.

Networks using **KTH-LR** channel allocation algorithm are *unfair* because two of the six examined configurations are *very unfair* and four ratings are *fair*. The main cause of unfairness is that active nodes influence the performance of nodes close to them. It is most obvious from client-server network load configuration.

KTH-LR is not very sensitive to the disturbing effect of longer distances. It is unfair in both short and long dual-bus, but in the case of *long bus* it has the best fairness measure among the algorithms. The relatively advantageous behaviour is due to the closely even average distance of nodes from other nodes within their effective area.

KTH-RA slot allocation algorithm is fair in the case of short bus-length. It is *very unfair* in the case of long bus-length. Though blocking probability is almost independent of inter-node distance, slot request time is distorted due to the uneven average distance of nodes from other nodes in the effective area. As in KTH-RA nodes ask randomly from any other nodes, each node is in the effected area of any node.

In the case of short bus-length, set-up-time channel allocation algorithms with status table, i.e. **KTH-S** algorithms, are *fair* independently of the request order.

In the case of long bus, the variants of this algorithm have different characteristics according to the request order. Closest first results in *very unfair* operation. Random request order is rated as *unfair*. In the case of closest first along the logical ring, the operation is *fair*.

The main reason of unfairness is:

- The further is a node from the middle of the bus, the longer is its average distance from other nodes (averaging over all nodes). The bigger is average distance the bigger is the delay of messages carrying free slot information, and thus the less consistent is the status table.

3.4.4.2 Aggregate Performance

It is shown in this section that algorithms with status table perform better than the same algorithms without tables with any parameter settings.

It is also shown that blocking probability is independent of the bus-length although, obviously, average set-up time increases as inter-node distances are increased.

It can be concluded from simulation results that performance characteristics in case sources generate bursty traffic are worse than they are at smooth sources. The biggest improvement is 0.31 ms in average set-up time (in the case of KTH-RA without status table, 90% offered load) which is 43% compared to the case of bursty source. The most significant improvement in blocking probability is 0.07 at the same algorithm and offered load.

The dependence of performance characteristics on the upper limit of the number of slot allocation retries specified for a connection is also studied in this section. It has been found that in the case of higher loads (i.e. bigger than 90%) the increasing upper limit increases the average connection set-up time and decreases the call blocking probability. The optimal retry limit based on the exact curves, which depends on the offered load of the system, and the demands on the network (blocking or set-up time is more important) were also determined.

3.5 Smoothing Algorithms

Based on the study of set-up time algorithms, two important observations were made in the previous section:

- Asymmetry in the characteristics of client-to-server and server-to-client directions at client-server network load profile can result in unfair operation.
- Algorithms without status table are not optimized, so the needed number of slot request retries for a connection is high. In this subsection, these properties are discussed in detail.

The first subsection is devoted to the detailed description of these properties because these are the motivations for new algorithms. The second subsection proposes two new algorithms to improve these properties. Then the effectiveness of proposed smoothing algorithms is shown by simulation. Finally, the conclusions of the section follow.

3.5.1 Motivation

3.5.1.1 Asymmetry

Based on the evaluation of set-up time slot allocation algorithms in Section 3.4, it is known that nodes initiating calls more often than the others have better performance characteristics. This effect is present at both algorithm types (i.e. with and without status table).

In a client-server network where clients have the same offered load (client-server load profile), this effect does not cause unfairness. Asymmetry, however, between the characteristics of client-to-server and server-to-client directions of a bi-directional connection always occurs.

In a different high level scenario, however, the same network load distribution can be obtained as the one used in the client-server network profile. E.g. suppose that there is an experienced user in the network (instead of the server), who uses resources from many computers, and all the others use the network less often (instead of clients). In this case the experienced user would have lower blocking probability and average connection set-up time than the others do. And it is *unfairness*!

Instead of displaying blocking probability and set-up time, in the following tables average number of slot request retries - needed for a successful connection establishment - are shown. Table 3.5.1 shows that value for KTH-RA algorithm without status table separately for the client-to-server, server-to-client directions and the average for all calls. The averages are calculated at different network loads. Table 3.5.2 shows the same values for KTH-S-RA algorithm. The parameters of nodes and hosts are the same as it was in Section 3.4.3.

Avr. # of retries	50%	70%	90%	110%	130%
Client-to-server	0.416	1.145	2.474	3.541	4.072
Server-to-client	0.009	0.033	0.121	0.26	0.389
All unidirectional	0.213	0.588	1.255	1.696	1.842

Table 3.5.1 - Asymmetry of server-to-client and client-to-server directions, KTH-RA-10 (without status table)

Avr. # of retries	50%	70%	90%	110%	130%
Client-to-server	0.193	0.438	0.73	0.905	0.969
Server-to-client	0.004	0.013	0.06	0.187	0.29
All unidirectional	0.099	0.226	0.395	0.547	0.629

Table 3.5.2 - Asymmetry of server-to client and client-to-server directions, KTH-S-RA-10 (with status table)

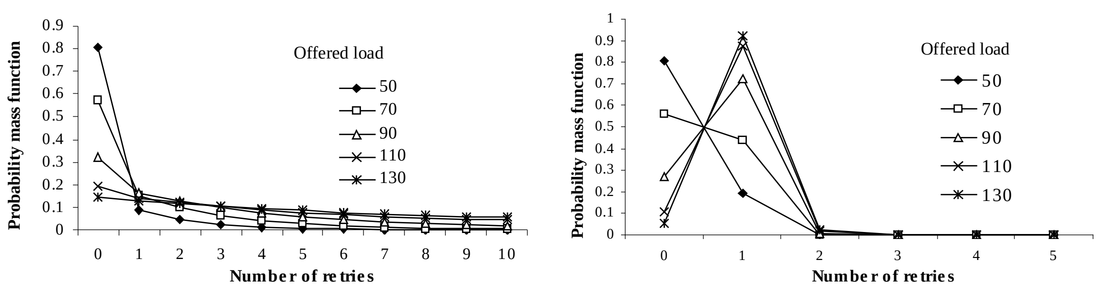
It can be seen that *due to status table* the average number of retries is less than 1 also at 130% offered load. *Without status table* an average client node needs to request slots from 4 other nodes to collect the slots for a uni-directional connection to the server.

The asymmetry is also reflected in these numbers. E.g. at 50% offered load, a 46 times more slot request are needed for client-to-server connection than for a server-to-client connection at both algorithms. Though this ratio is lower at higher offered load, it is significant there too.

As this phenomenon may be the cause of unfairness, an algorithm is needed that makes a balance between the characteristics of very active and not active nodes.

3.5.1.2 Too many slot requests

In addition to asymmetry a possible *improvement opportunity of algorithms without status table* can be seen in Table 3.5.1 and Table 3.5.2. Number of slot requests needed to set-up a client-to-server connection is very high even at moderate loads. It is 1.145 at 70% load, that means that an average node asks slots for a successful connection more than once. According to the simulation settings (Section 3.4.3.2) each connection required 1 slot, so 1.145 is a high number. The probability mass function of the same random variable (number of retries needed for a successful connection) can be seen for client-to-server connections in Figure 3.5.1.



KTH-RA (without status table) KTH-S-RA (with status table)

Figure 3.5.1 - Probability mass function of number of slot request retries during a successful connection set-up

It can be seen that algorithm with status table manage to collect the slot without slot request or with one slot request, as the probability of 2 or higher retries is very low. At the algorithm without status table, if the offered load is as low as 70% the probability that slots are needed from other node is 100-55=45%.

In the followings smoothing algorithms are proposed that are able to eliminate asymmetry and improve performance due to distributing free slots evenly (or unevenly) amongst nodes.

3.5.2 Description of Smoothing Algorithms

The goal of smoothing algorithms is to decrease (or eliminate) the need for slot allocation during call set-up and to balance the characteristics of active and passive nodes. To reach the goal smoothing algorithms are trying to distribute free slots amongst nodes according to their loads. I proposed BCA algorithm in [J2] and SSA algorithm in [C1] to fulfil this goal. This section describes both algorithms.

3.5.2.1 Background Channel Allocation Algorithm

Background Channel Allocation algorithm (**BCA algorithm**) [J2, P1], transfers slots between nodes in the background, independently of set-up requests coming from hosts. It is able to work parallel with any set-up-time algorithms.

In the algorithm nodes regularly exchange free channels with their direct neighbours along the logical ring.

The goal of the exchange *in the case of homogeneous network load* is to distribute free channels *evenly* amongst nodes. In order to achieve this goal, nodes check regularly if there is any difference between the number of local and neighbouring nodes' free channels. This process provides that neighbouring nodes have nearly the same number of free channels at any time instant, thus free channels are always distributed almost evenly amongst nodes.

This idea can be extended to a real algorithm, which considers the case of normal operation when the load is different at each node. A priority value for buses in both directions reflect the difference between nodes. That is, each node has a priority number for each bus, which depends on the traffic load sent to the given bus. Priorities can be constant or can change dynamically when adapting to the actual load of the network.

Exchange of free channels depends on the value of free channels and priorities. Node i initiates slot allocation with its ring neighbour - node $i+1$ - if expression

$$|(\text{free channels of node } i)*(\text{priority of node } i+1)-(\text{free channels of node } i+1)*(\text{priority of node } i)| \quad (3.5.1)$$

can be decreased by slot allocation. The amount of slots to be transferred is determined so as to minimize (3.5.1) and considering that only free channels can be transferred. Node i asks slots from node $i+1$ if the first term of expression (3.5.1) is below the value of the second term and it transfers slots if the first term is the higher one. That is, in the case of equal priorities, node i transfers one channel to node $i+1$ if its number of free channels is higher by 2 than the ones of node $i+1$.

Node i calculates expression (3.5.1) whenever a local connection was set up or released (number of local free channels changed).

If the priority of a node is equal to zero then it is left out from the ring. The next successive node is the exchange partner instead. For example if the priority of node $i+1$ is 0 for one of the buses then node $i+2$ is the partner of node i for the allocation of free slots on that bus.

BCA algorithm is based on the comparison of the amount of local and neighbouring free channels. This is why it requires a very small status table where nodes keep a record of free slots of direct neighbouring nodes on the ring. Nodes send administration messages to the first upstream neighbouring node along the ring after each change in the number of local free channels in order to provide information for maintaining up-to-date tables.

Priority defined above does not effect directly the amount of bandwidth available for a node. It is rather related to the possibility of setting up a channel without slot reallocation, independently of the bandwidth used. This definition of priority can be used for optimising the network utilization and channel set-up times.

Priorities can be dynamic and static as well. In the case of dynamic priorities, a traffic estimation procedure modifies the priority of the node. Estimators use parameters of previous connections (e.g.: amount of required bandwidth and interarrival times) to calculate the current priority. If the characteristics of the traffic are known effective estimators can be constructed. However, estimators can be built without preliminary information about the traffic. Dynamic priorities are not used in this examination, as offered load during a simulation run was static.

The other solution is to assign static priorities to nodes where priorities are changed at management level. In this case the basis of priority assignment can be the role of the node in the network or the price paid by the customer of the node.

If priority is based on the role of the node, we can assign higher priority to nodes connected to servers or to switching nodes, and lower priority to nodes connected to clients.

If priorities are proportional to charges paid by customers then it is a better solution to rewrite expression (3.5.1) so as those priorities are compared to the number of *all the channels* owned by nodes.

In this case priority is related to the bandwidth that can be used by the connections of the node without reallocation during set-up. If priority is high, many channels can be used without the additional delay of slot reallocation. If slots of the node are used by connections, then slot allocation is required at every new connection set-up. This kind of priority usage is appropriate for charged systems, because the customer who pays more can build up more connections without the delay of set-up-time slot allocation. There are significantly fewer channel allocations in this system compared to the one using the number of free channels for calculating function (3.5.1).

3.5.2.2 Set-up-time Smoothing Algorithm

Set-up-time Smoothing Algorithm (SSA) [C1] is the improved version of BCA. In BCA, slot exchange is performed always between the same nodes: only direct neighbours along a logical ring transfer slots between each other in the background. The advantage of this operation is that nodes only have to store status information about their ring neighbours. However, it has drawbacks too in real implementations. In certain cases, distribution of free channels may differ significantly from the priority distribution. If – for example – there are a few nodes with very low activity between nodes that have many free slots and nodes that have high priority, BCA does not transport free slots to high priority nodes.

In SSA, free slots can be exchanged between the parties of connections during connection set-up and release procedure. The rules of slot exchange are the same as it is in BCA: nodes transfer free slots if expression (3.5.1) can be decreased.

Exchange partners of BCA are always different, and SSA is aimed to improve algorithms without status tables. Therefore, nodes add additional information into DCP Announce and DCP Attach messages.

The SSA procedure during connection set-up is described in detail below:

1. The sender node sends the number of its free slots in the DCP Announce message.

2. The receiver compares the number of its and sender node's free slots.
3. a, If receiver node should send slots according to (3.5.1), it includes the number of slots to be transferred into the DCP Attach message, and initiates a slot transfer procedure.

   b, If sender node should send slots, the receiver puts the number of slots it asks from the receiver into the DCP Attach message. When the sender received the DCP Attach message, it initiates a slot transfer procedure with the receiver.

Note that definition of sender and receiver node is based on the data transmission roles (not on the transmission of control information). As connections are uni-directional at DCP level, one of the parties of the connection is always sender, and the others are receivers.

Operation of SSA algorithm can be described in the same way for connection release procedure.

The main differences between BCA and SSA are summarized in Table 3.5.3

	BCA	SSA
Who are slot request partners?	neighboring nodes along logical ring	parties of connections
When do slot requests occur?	any time	during connection set-up and release
How to send information about the number of free slots?	in separate messages	in modified connection set-up and release procedure

Table 3.5.3 – Differences between BCA and SSA algorithms

3.5.3 Simulation Results

BCA and SSA algorithms were proposed to balance performance characteristics of active and passive nodes and to improve performance of the network.

The primary goal of this subsection is to examine whether the above design goals of BCA and SSA algorithms are fulfilled or not. The effectiveness dependency on burstiness of traffic and bus-length is also investigated.

3.5.3.1 Short Bus, Bursty Traffic

First, the short bus and bursty traffic case is examined. Both smoothing algorithms perform best when using together with set-up-time slot allocation algorithms. As BCA uses logical ring during background allocation, it is applied together with KTH-LR. SSA is used with KTH-RA algorithm in the following study. Both KTH-LR and KTH-RA algorithms have *tuning* (Section 3.4.2.4), as it improves performance.

Performance

Performance of smoothing algorithms is simulated with the following configuration:

- Retry limit = 10
- client-server load profile
- priority settings : nodes with any activity have 1 as priority, idle nodes have 0 priority (separately for both directions)

Simulation results are displayed in Table 3.5.4, and the same data can be seen in Figure 3.5.2 and Figure 3.5.3.

	Blocking probability					Avr. connection set-up time (ms)				
Load	50%	70%	90%	110%	130%	50%	70%	90%	110%	130%
KTH-LR	0	0.004	0.072	0.258	0.434	0.321	0.44	0.686	0.913	1.047
KTH-LR+BCA	0	0	0.072	0.256	0.434	0.314	0.39	0.615	0.901	1.071
KTH-RA	0	0.005	0.077	0.272	0.455	0.347	0.471	0.706	0.915	1.039
KTH-RA+SSA	0	0	0.027	0.228	0.424	0.28	0.289	0.451	0.812	1.008
KTH-S-LR	0	0	0.018	0.209	0.411	0.324	0.369	0.437	0.498	0.521
KTH-S-LR+BCA	0	0	0.022	0.202	0.409	0.305	0.34	0.427	0.53	0.576

Table 3.5.4 - Performance of smoothing algorithms, short bus, bursty traffic, bi-directional connections

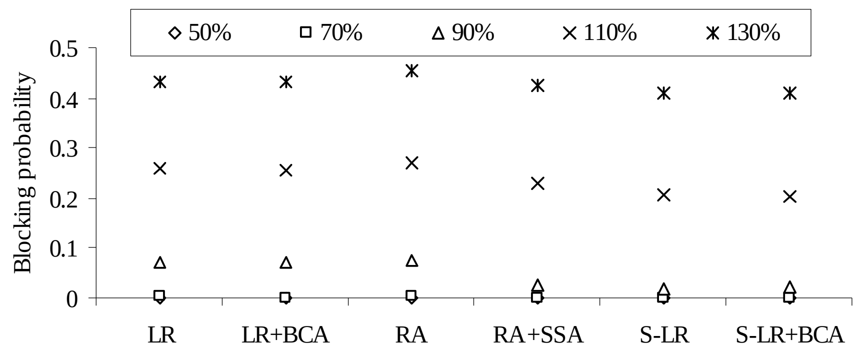


Figure 3.5.2 - Blocking probability of smoothing algorithms

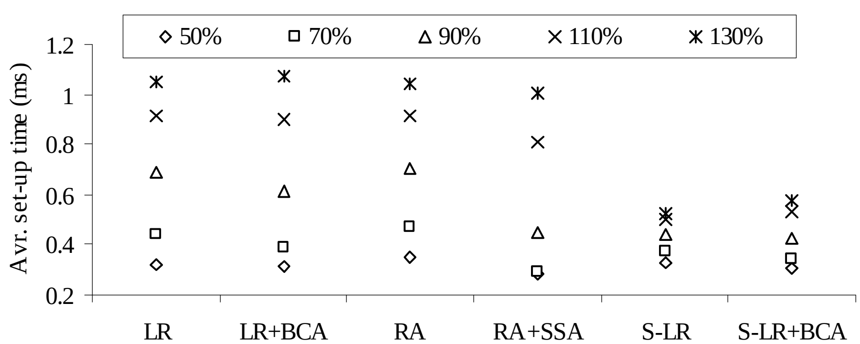


Figure 3.5.3 - Average set-up time of smoothing algorithms

Simulation results can be used to define the application area of smoothing algorithms. The most appropriate offered load range can be selected. The effect of status tables can be seen.

Figures show that smoothing does not improve the performance of algorithms with status table. Though a very small improvement can be noticed in set-up time at 50%, 70% and 90% load, at higher loads the performance of algorithms with status table degraded due to smoothing algorithms. The main performance gain of smoothing algorithms is that they could decrease the number of slot allocation retries needed for a successful connection. As figure 3.5.1 shows, the number of slot allocation retries is low when status tables are present, so there is no room for this kind of optimisation.

Improvement of set-up time of algorithms without status table is, however, significant if the network is not overloaded. Among KTH-LR, KTH-LR+BCA, KTH-RA and KTH-RA + SSA algorithms the last one is the best. SSA algorithm is most effective in 50%-100% offered load range. The highest improvement on set-up time is 0.25 ms at 90% load and KTH-RA + SSA algorithms. That is, SSA decreased average set-up time by 35 %. At 130% offered load, the gain of SSA algorithm is only 0.03 ms. Less effective ranges of smoothing algorithms can be explained with the followings:

- Above 100% load, "there is nothing to smooth", i.e. there are very few free slots in the system.
- Below 50% offered load, "there is nothing to optimize". i.e. there are many free slots in the system.

Blocking probability of the system was decreased due to SSA algorithm at each offered load, but its effect is small. The highest improvement is at 90% offered load, where blocking is lower with 0.05 (from 0.07 to 0.02).

The range between 50% and 100% offered load is the most important one for well-designed networks. If offered load is higher for longer time, the network is mis-dimensioned, and it is not able to provide efficient services. Lower offered load - for a long time - means that the network is over-dimensioned and the operator paid for unused bandwidth.

It is also interesting that at 50%, 70% and 90% offered loads KTH-RA with SSA have nearly the same performance as KTH-RA with status table.

Priority settings

In the previous section, performance was examined with fixed priority settings. Now, it will be shown how to find an optimal priority settings at client-server configuration taking into account the viewpoint of *symmetry and performance*.

Priorities of client nodes in this section are the same as they were before. Priority of server node, however, is varied between 50 and 0.05. The effect of priority settings on client--to-server connections, server-to-client connections and bi-directional connections can be seen in Figure 3.5.4 and Figure 3.5.5. The first item is KTH-RA algorithm without smoothing at each offered load. At KTH-RA + SSA algorithm, the ratio of server priority and client priority is displayed after the name of SSA algorithm.

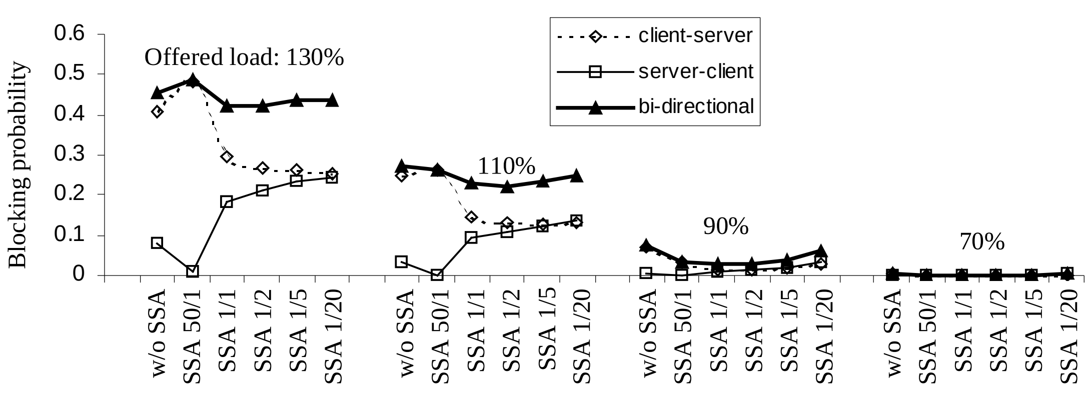


Figure 3.5.4 – Effect of priority settings on blocking probability

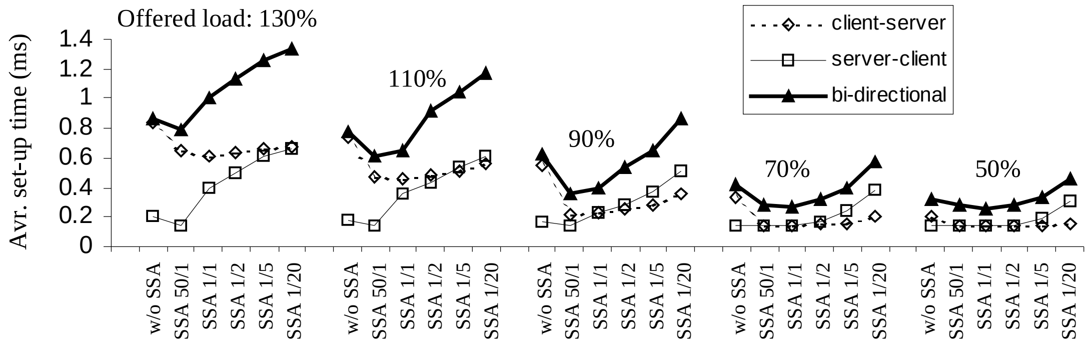


Figure 3.5.5 – Effect of priority settings on average set-up time

First, let us take the *viewpoint of symmetry vs. priority ratios*. Figure 3.5.4 shows that the lower is the priority of server the closer is blocking probability of client-server (uplink) and server-client (downlink) connections. Blocking of uplink connections decreases and blocking of downlink connections increases when the priority of server node decreases. Blocking probability of uplink connections, however, is always below that of downlink connections.

Average set-up time of uplink connections can exceed that of downlink connections at certain priority ratios. The crosspoint of uplink and downlink curves depend on the offered load of the network. At 130% load downlink/uplink ratio is 1/20, at 90% load it is 1/1. Priority setting of server node mainly influences the set-up time of downlink connections. Set-up time of uplink connections does not decrease in case the server has lower priority.

Both figures show that priority is an effective tool to balance the characteristics of nodes with different load of connections.

The next question to answer is *how performance of bi-directional connections depends on priority ratio*. Based on Figure 3.5.5, optimal priority is the one, which strengthen cache effect, i.e. the higher the priority of server the lower average set-up time of bidirectional connections is. Priority of server does not effect so significantly the blocking probability curve, except that blocking increases if the server has too high priority.

3.5.3.2 Effect of Long Bus and Less Bursty Sources

Section 3.5.4.2 analyzed SSA algorithm in detail. This section examines the effect of longer bus-length and less bursty traffic sources. Main configuration settings are the same as in the previous section:

- Retry limit = 10
- client-server load profile
- priority settings : nodes with any activity have 1 as priority, idle nodes have 0 priority (separately for both directions)

Long bus

Table 3.5.5 shows the performance of smoothing algorithms. The same conclusions can be drawn from this table as we obtained for short bus. Namely, smoothing does not improve significantly (rather degrade) the performance of slot allocation algorithms with status table. Set-up time of algorithms without status table, however, is decreased significantly in 50%-100% offered load range. SSA improved slightly blocking at almost every offered load. The largest improvement of set-up time is at 90% offered load: it is equal to 0.81 ms, which is 40% of the set-up time of KTH-RA algorithm. The largest improvement of blocking is 0.05 at 70% and 90% load.

Load	50%	70%	90%	110%	130%	50%	70%	90%	110%	130%
Algorithm	Blocking probability					Avr. slot request time				
KTH-LR	0	0.004	0.07	0.262	0.438	0.832	0.977	1.28	1.572	1.753
KTH-LR+BCA	0	0.007	0.075	0.252	0.441	0.822	0.885	1.194	1.54	1.764
KTH-RA	0	0.005	0.076	0.273	0.466	0.994	1.358	2.04	2.596	2.933
KTH-RA+SSA	0	0	0.025	0.227	0.427	0.786	0.808	1.23	2.325	2.89
KTH-S-LR	0	0	0.018	0.205	0.42	0.835	0.891	1.004	1.163	1.251
KTH-S-LR+BCA	0	0	0.019	0.203	0.423	0.81	0.842	0.99	1.196	1.315

Table 3.5.5 - Performance of smoothing algorithms, long bus, bursty traffic

Poisson traffic

The last topic in this section is the effect of burstiness on the effectiveness of smoothing algorithms. In Section 3.4.3 it was shown that Poisson traffic sources are more convenient for the network, as both performance characteristics improved compared to WWW traffic. Effectiveness of smoothing algorithms can be seen in Table 3.5.6 in the case of Poisson sources.

Load	50%	70%	90%	110%	130%	50%	70%	90%	110%	130%
Algorithm	Blocking probability					Avr. set-up time				
KTH-LR	0	0	0.005	0.202	0.419	0.266	0.274	0.388	0.813	0.999
KTH-LR+BCA	0	0	0.007	0.204	0.418	0.263	0.268	0.352	0.832	1.038
KTH-RA	0	0	0.006	0.219	0.451	0.263	0.272	0.398	0.745	0.878
KTH-RA+SSA	0	0	0.001	0.204	0.43	0.263	0.263	0.314	0.854	1.011
KTH-S-LR	0	0	0	0.186	0.413	0.265	0.271	0.319	0.438	0.463
KTH-S-LR+BCA	0	0	0	0.185	0.412	0.263	0.267	0.307	0.481	0.518

Table 3.5.6 - Performance of smoothing algorithms, short bus, Poisson traffic

Simulation results show that SSA algorithm is more effective at bursty traffic than here. SSA decreased blocking of KTH-RA without status table at any load. It also decreased set-up time below 100% offered load. When, however, offered load is above 100% SSA and BCA increased set-up time, so it is worth to switch off smoothing when the system is overloaded. The performance of algorithms with status table is not improved with smoothing algorithms.

3.5.4 Conclusion on Smoothing Algorithms

Simulation results presented in Section 3.5 showed that asymmetry of set-up-time slot allocation algorithms can be corrected with smoothing algorithms and proper priority settings. As set-up-time algorithms provide better service for active nodes, their priority should be set to a smaller value.

The exact dependence of asymmetry on priority values is also worked out. . Simulation results show that - in the case of client-server network configuration – blocking probabilities of the down-link (server-client) and up-link (client-server) connection are equal if the priority of the server is 1 and that of the clients is 20. The priority ratio, where average set-up times of up-link and down-link connections are symmetrical, depends on the offered load in the network.

E.g. at 130% offered load the symmetrical server/client priority ratio is 1/20; at 90% offered load it is 1/1.

Simulation also proved that adding smoothing algorithms to set-up time algorithms without status table improves the performance of the DTM dual-bus if offered load is between 50% and 100%.

It can be concluded that based on average set-up time, optimal priority is the one that strengthens cache effect, i.e. the higher the priority of the server is the lower the average set-up time of bi-directional connections is. Priority of server does not effect so significantly the blocking probability curve, except the case when the server has too high priority. In this special case blocking probability increases.

It is also shown that smoothing algorithms improve the performance of the system more significantly if sources generate bursty data.

Chapter IV: Message Level Characteristics of Multiplexing Methods

4.1 Introduction

DTM is an *integrated services* network with *512 kbps channel granularity* using *fast circuit switching*.

Due to its inherent circuit switched operation, resources have to be reserved prior to usage and they remain unused between bursts of information. Burst switching is only one of the solutions to utilize the channel between bursts. This chapter presents another solution: multiplexing that allows multiple sources to transmit data into the same DTM channel.

In addition to better utilizing the channel, multiplexing can also decrease granularity of DTM channels. This is important because the bandwidth of a DTM channel can change in relatively big - 512kbps - steps (64 bit slots within 125 microseconds long cycles), and a one-slot DTM channel has 512kbps capacity.

Two multiplexing methods are proposed in this chapter. Both of them support priority levels, which enables the definition of quality of service classes. Sources with high priority can transmit real-time traffic. Sources transmitting data communication traffic have low priority.

Though multiplexing can increase the utilization of network resources, it can also degrade service quality provided to users if the network is not dimensioned appropriately. A thorough analysis of the most important system characteristics is also presented for the proposed multiplexing methods in this chapter:

- For high priority sources - as they are assumed to have real-time behaviour- message delay and delay variation are the most important characteristics.
- Low priority sources - assumed to transmit data communication traffic - are sensitive to message loss and message delay. Variations in the delay are less important in this case. Loss of messages can be caused by buffer overflow in multiplexers, so buffer length is another relevant characteristics.

Consequently, it can be said that the distributions of two random variables are always important in a multiplexing system:

- *length of the queues (system content)*
- *queuing delay of messages (system time)*

System content and system time random variables expressed in the dissertation. System content is the number of messages in the server plus the messages in the queue. System time of a message is the time it spent in the queue plus the delay due to its service (in the server).

The multiplexing methods to be presented are analyzed with the means of discrete time queuing theory. The goal of the analysis is to obtain the probability distribution of the above characteristics. As the probability generating function (pgf) contains all information about the distribution, my goal is to derive the pgf of the system time of messages and that of the system content.

The chapter is structured as follows.

In the first part of Section 4.1 the basic assumptions of discrete-time queuing theory are introduced. Then the need for multiplexing is illustrated through a simple example, where the characteristics of the DTM channel serving a single source are analyzed. Finally, the section presents the proposed multiplexing methods.

In Section 4.2, three mathematical models are proposed for the first multiplexing method. The relation between the models and the detailed analysis of one of the models is also discussed. I obtained closed formulas for the pgfs of the discussed characteristics, for their first two moments and for the approximations of tail probability distributions. Results are illustrated with examples.

In Section 4.3, another multiplexing method is analyzed. Two models are presented and the solution of the models is cited from the literature.

Finally, the comparison of the multiplexing methods follows in Section 4.4.

4.1.1 Discrete Time Queuing Model

Before the analysis of the systems, the basics of the model used in discrete time queuing discipline are introduced [BrKi93].

The time axis in discrete time queuing systems is divided to fix length intervals, usually called slots. In DTM the word "slot" is reserved to the 64-bit long time slot of a cycle, so the slot of the discrete time queuing systems is referred to

- time-unit when generally speaking
- slot, cycle or frame when the time-unit is a slot, a cycle or a frame in DTM terminology (see definition in Section 2.2.2, Figure 2.2.4).

The main properties of the queuing model used in this document are the followings:

- When messages arrive they are stored in a buffer with infinite length.
- The length of a time-unit is normalized to 1, as usually in discrete time models.
- Message arrivals are assumed to take place at the end of the time-unit, because in the dissertation only the integer part of the system characteristics is examined.
- The service of a message that arrives in a time-unit starts soonest at the beginning of the next time-unit and lasts 1 time-unit.

Three types of variables are considered in the dissertation:

- system content (or system occupancy, queue length, buffer length)
- unfinished work
- system time (or waiting time, message delay)

Unfinished work is the time needed to empty the message queue. System time is the time between the completion of the service and the arrival of a message. System content is the number of messages in the message queue including messages under service.

4.1.2 Performance Parameters of a Queue Serving a Single Source

A simple system will be presented in this section to show the need for multiplexing methods: a source with real-time needs that is allowed to use the whole bandwidth of a DTM channel. In general, this channel consists of c slots in a cycle. If the number of messages sent to the queue in a cycle is a sequence of independent and identically distributed random variables, the GI-D-c discrete-time queuing model describes the operation. The short notation GI stands for general independent arrivals in a time-unit, D means that the service process is deterministic and c means that there are c servers in the system.

The complete analysis of the Gi-D-c queuing system can be found in e.g. [BrKi93]. The goal of this section is to show the possible gain of multiplexing. Therefore, only a simplified model is presented where the number of servers is 1 (Gi-D-1), or in other words the source transmits in a *one-slot DTM channel*.

The analysis of the system should start by the evolution equation for the queue length:

$$U_{k+1} = (U_k - 1)^+ + A_k \qquad (4.1.1)$$

where the operator $(\cdot)^+$ gives the value of the argument if it is greater than 0, otherwise it returns 0. A_k is the number of messages arrived during cycle *k*, and U_k is the system time at the beginning of cycle *k*. The arrival process is a batch Bernoulli process with general batch size distribution B [MB96]. In other words the sequence of $\{A_k\}$ is the sequence of independent and identically distributed random variables. In this case the sequence of $\{U_k\}$ forms a Markov chain, therefore the stability criterion for $\{U_k\}$ is $E\{A_k\} < 1$.

The definition of the probability generating function of a random variable X is

$$X(z) = \sum_{i=0}^{\infty} x(i) \cdot z^i = E\left\{z^X\right\} \qquad (4.1.2)$$

Where X(z) is the pgf of X, and $x(i)$ is its probability mass function $x(i) = Prob(X = i)$.

The pgf of system occupancy at random slot boundaries can be easily expressed from (4.1.1) and definition (4.1.2)

$$U(z) = \frac{(1 - A'(1))A(z)(z-1)}{z - A(z)} \qquad (4.1.3)$$

The pgf of the integer part of system times - $V(z)$ - can be written as

$$V(z) = \frac{(1 - A'(1))z(A(z)-1)}{A'(1)(z - A(z))} \qquad (4.1.4)$$

The mean values are the first derivatives of the generating functions at $z=1$ point:

$$U'(1) = A'(1) + \frac{A''(1)}{2(1 - A'(1))}; \quad V'(1) = 1 + \frac{A''(1)}{2A'(1)(1 - A'(1))} \quad \text{(cycles)} \qquad (4.1.5)$$

The variances can be expressed from the first and the second derivatives at $z=1$ point:

$$\mathrm{var}\{U\} = U''(1) - U'(1)^2 + U'(1) = \mathrm{var}(A) + \frac{A''(1)}{2(1-A'(1))} + \frac{A'''(1)}{3(1-A'(1))} - \frac{3A''(1)^2}{4(1-A'(1))^2} \quad (4.1.6)$$

$$\mathrm{var}\{V\} = \frac{A''(1)}{2A'(1)(1-A'(1))} + \frac{A'''(1)}{3A'(1)(1-A'(1))} - \frac{3A''(1)^2(1-2A'(1))}{4A'(1)^2(1-A'(1))^2} \quad (4.1.7)$$

The tail of the probability mass function can be approximated by an exponential distribution that belongs to the smallest positive real pole (which is greater than 1) of the generating function. The tail of the mass function can be expressed as (or approximated as) the sum of exponential distributions. Furthermore, it can be assumed that at the tail the exponential distribution belonging to the pole with the smallest absolute value is dominating. The tail probability of the system occupancy can be obtained using the results presented in [BSDP94, BrKi93].

$$P(U=n) = \frac{(1-A'(1))(1-z_0)}{1-A'(z_0)} z_0^{-1-n}; \qquad P(U>U_0) = -\frac{1-A'(1)}{1-A'(z_0)} z_0^{-1-U_0} \quad (4.1.8)$$

$$P(V=n) = \frac{(1-A'(1))(1-z_0)}{A'(1)(1-A'(z_0))} z_0^{-1-n} = \frac{P(U=n)}{A'(1)}; \quad P(V>V_0) = -\frac{1-A'(1)}{A'(1)(1-A'(z_0))} z_0^{-1-V_0} \quad (4.1.9)$$

where z_0 is the dominant pole of (4.1.3) and (4.1.4), the solution of equation $z = A(z)$.

4.1.2.1 Examples

Now, let we see three example distributions and calculate the mean values and the tail probabilities. Because there is a simple relationship between the system contents and the system time (see (4.1.5) and (4.1.9)), only the system time is displayed.

We assume in all of the examples that the sequence of the number of messages arrived in successive time-units is a sequence of independent and identically distributed random variables. The first example is a Poisson process, the second one is a batch Bernoulli process with constant batch size, and the last example is a batch Bernoulli process with uniform batch size distribution between two values.

Poisson arrival

The pgf of the poisson arrival process is $A(z) = e^{\lambda(z-1)}$, and its peakedness - i.e. the ratio of the variance and the mean value - is 1. Substituting into (4.1.5) and (4.1.9)

$$U'(1) = \lambda + \frac{\lambda^2}{2(1-\lambda)}; \qquad P(U>U_0) = -\frac{1-\lambda}{1-\lambda z_0} z_0^{-1-U_0} \quad (4.1.10)$$

Batch Bernoulli arrival with batch size *L*

The pgf of the arrival process is $A(z) = 1 - p + pz^L$, and the peakedness is $k = L - pL$

The expressions for the mean values and the tail distribution are

$$U'(1) = pL \cdot \left(1 + \frac{L - pL}{2(1 - pL)}\right); \qquad P(U > U_0) = -\frac{1 - pL}{1 - pLz_0^{L-1}} z_0^{-1-n} \tag{4.1.11}$$

Batch Bernoulli arrival with uniform batch size distribution between *L* and *K*

The pgf and the peakedness of the batch Bernoulli arrival process with uniform batch size distribution between *L* and *K* are

$$A(z) = 1 - \sum_{i=L}^{K} p + \sum_{i=L}^{K} pz^i \quad \text{and} \quad k = \frac{p\sum_{i=L}^{K} i^2}{p\sum_{i=L}^{K} i} - p\sum_{i=L}^{K} i \tag{4.1.12}$$

The characteristics of the queue length can be expressed as

$$U'(1) = p\sum_{i=L}^{K} i + \frac{p\sum_{i=L}^{K} i^2 - p\sum_{i=L}^{K} i}{2\left(1 - p\sum_{i=L}^{K} i\right)}; \qquad P(U > U_0) = -\frac{1 - p\sum_{i=L}^{K} i}{1 - p\sum_{i=L}^{K} iz_0^{i-1}} z_0^{-1-n} \tag{4.1.13}$$

Evaluation

Now, let us take concrete examples of the distributions described above. The peadkedness is also show next to the name of the distributions:

- Poisson process; $k = 1$
- Batch Bernoulli process with batch size 30; $k = 30 - \rho$ where ρ is the load of the queue.
- Batch Bernoulli process with uniformly distributed batch size between 30 and 90; from (4.1.12) the peakedness is $k = 64.4 - \rho$ where ρ is the load of the queue.

Based on the tail probabilities two important parameters can be calculated. Though our model assumes infinite buffers, it was shown in [BSDP94] that the probability $P(U > U_0)$ is a good estimation of the message loss probability of a finite queue with $U_0 + c$ size where c is the number of servers (in this case 1). The sizes of the buffers are dimensioned so that the message loss probability should be below a certain value. Figure 4.1.1 shows the required buffer size if the maximum message loss rate is 10^{-4}.

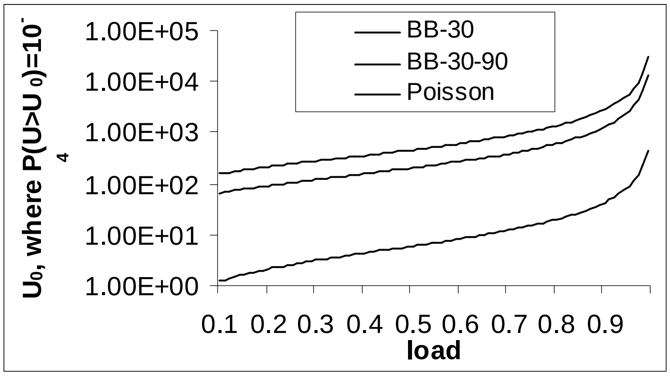


Figure 4.1.1 - Dimensioning the queue length

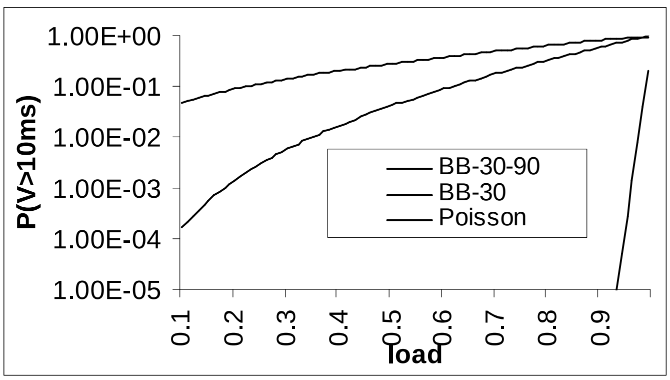


Figure 4.1.2 - Probability of too long delay

Figure 4.1.2 displays the ratio of messages having larger delay in the queue than 10 ms.

If the source is bursty and delay sensitive then its load should be limited in order to keep the delay below a certain level. In Figure4.1.2 it can be seen that in case the probability that the queuing delay is greater than 10 ms is specified to 10^{-3}, the maximum load limit for the batch Bernoulli source with batch size 30 is 0.35. That is, the utilization of the DTM channel is very low.

We can conclude that adding sources with flexible bandwidth and delay requirements to the same DTM channel can increase the utilization of the system.

4.1.3 Proposed Multiplexing Solutions

We learned from the previous section that the utilization of the DTM channel can be enhanced if we multiplex low priority (LP) sources with the high priority (HP) delay sensitive source. Two multiplexing solutions are proposed and analyzed in the dissertation, where several LP sources are multiplexed with a single HP source. In this section the common properties of the multiplexing methods will be summarized.

Both multiplexing methods leave the quality of the HP connection unchanged. This statement also involves that though LP sources can use the bandwidth when the HP source is listening, when the HP starts to transmit it does not have to wait for the channel to become available. If this switching from the LP sources to the HP source is not fast then the delay variation of the HP source increases.

To keep the delay variation at a low level, it is assumed in the multiplexing methods that HP and LP sources use the same DTM channel during the whole connection. No channel release and establishment is done when one source takes over the right of the transmission from another one. Due to this an additional addressing method is needed to distinguish between the transmitting sources. The addressing methods will be presented at the detailed description of the multiplexing solutions in the following sections.

The management of sources using the DTM channel is easier and the switching time between connections is shorter if sources are connected to the same node. The receiver hosts can reside at different nodes, but in this case a node has to listen to a common channel and to filter the data of its hosts. The topology restriction is shown in Figure 4.1.3.

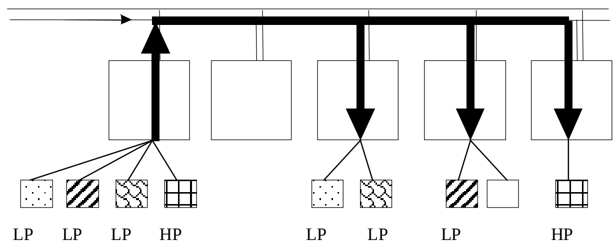


Figure 4.1.3 - Sources of a DTM channel are connected to the same node

A DTM channel can be shared in many ways. The first method to be discussed in Section 4.2 multiplexes LP sources using time division multiplexing. The second method - described in Section 4.3 - uses packet headers and trailers to separate LP sources from the others.

4.2 Time Division on Two Time Scales with Priorities Multiplexing Method

In this section, a new multiplexing method is described and analyzed, which was initially proposed in [C4], and further analyzed in [J1]. The section is structured as follows.

First, the introduction of the so-called “time division multiplexing on two time scales” multiplexing method is presented in Section 4.2.1. Then in Section 4.2.2, three queuing models will be proposed that describe the operation of low priority sources. In Section 4.2.3, 4.2.4 and 4.2.5 three parallel mathematical analyzes are given based on the presented models. The last subsection compares the presented models based on the mean values.

4.2.1 Description

In the *time division multiplexing on two time-scales with priorities* (TDM multiplexing) solution, low priority sources are multiplexed in the time domain:

M successive cycles of the DTM channel form a frame (see Figure 4.2.1). Each low priority source is allowed to transmit once in a frame. Since each low priority source has its own cycle, they do not share resources among themselves. The only high priority source belonging to the DTM channel is able to transmit messages in every cycles of the DTM channel; low priority sources can only use the M^{th} portion of the remaining bandwidth.

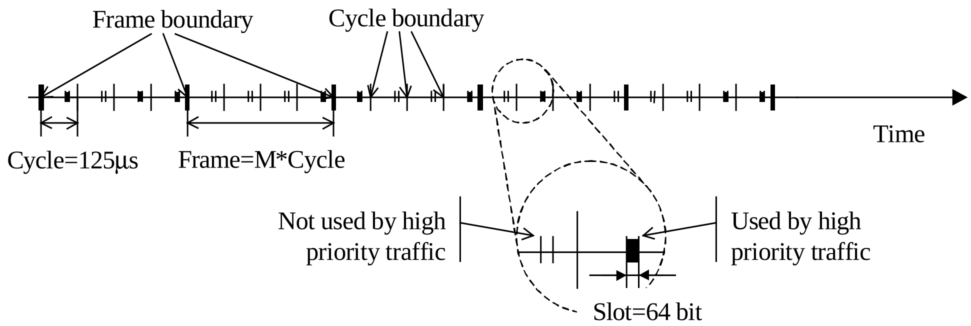


Figure 4.2.1 - Concept of Slot, Cycle and Frame

An addressing method is needed to distinguish between the transmitting sources. This involves two tasks:

- distinguishing the HP source from the LP sources
- distinguishing a LP source from other LP sources

The distinction between the HP source and LP sources is based on a priority bit assigned to each slot. If it is one then the slot contains HP data; else it carries LP data.

The receiver should also distinguish one low priority source from the others. As low priority sources are multiplexed with TDM, the location of the cycle within the frame identifies the LP source.

4.2.2 Models

Figure 4.2.2 shows the queuing model of the system. Each source has its own queue. Lines show in the figure when sources are allowed to transmit. The other signs on the time axis are explained in Figure 4.2.1. $U_{2,i,k}$ is the system content that belongs to low priority (priority 2) source i in cycle k and $A_{2,i,k}$ is the number of slots arrived to the low priority queue (priority 2) of source i in cycle k. $U_{1,k}$ is the system content that belongs to the high priority (priority 1) source in cycle k.

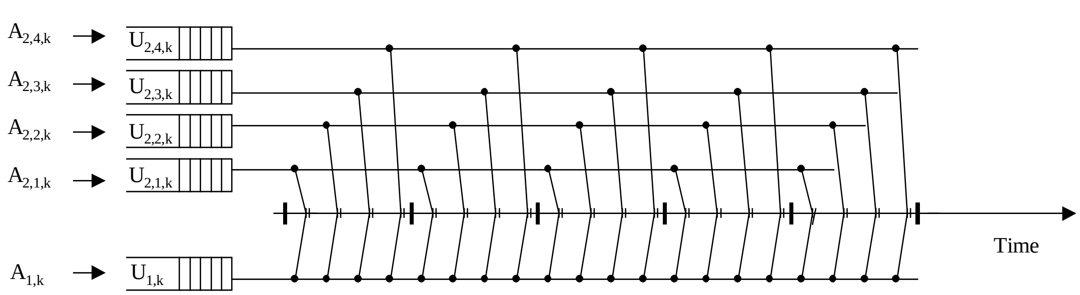


Figure 4.2.2 - Queues in the TDM solution

The operation of a given low priority source is independent of the other low priority sources. The characteristics of the high priority source are independent of any other source. Thus, it is enough to analyze one of the low priority sources, and the results can be applied to all of them. That is, the analysis of the system can be simplified to two sources as displayed in Figure 4.2.3.

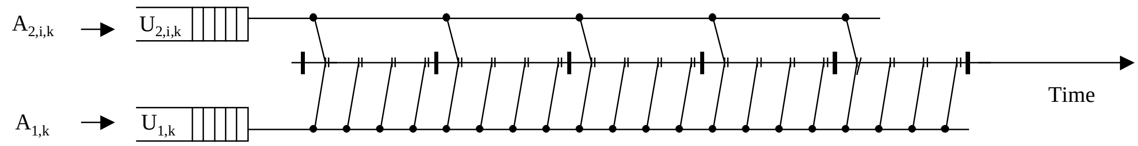


Figure 4.2.3 - Queuing model of the TDM solution

To specify the details of the model, the time scale of the analysis and the traffic model of the sources should be considered.

4.2.2.1 Times Scale and Traffic Model

The operation of the TDM solution can be modeled on three time scales:

- slot level, the length of a slot is ~100ns if the total bandwidth is 622Mbps
- cycle level, the length of a cycle is 125μs
- frame level, the length of a frame is 1ms if there are 8 cycles in a frame

The choice of the time scale depends on the operation of the queue.

High priority queue is served once in every cycle – it is operating at the cycle level. Therefore, its operation can not be described with a frame level model. Even though the slot level model provides better accuracy, it is based on a more accurate description of the sources, i.e. the complexity of the model increases.

Low priority queues are served once in every frame – they are operating at the frame level. That is, all three models are appropriate for the description of the queues. Both the accuracy and the complexity of the descriptions increase if smaller time scales are used. The frame level model is the least accurate and least complex, the slot level model is the most accurate and most complex.

The analysis of the slot level operation is too complex. The approaches are used in this work:

- Cycle level model – Cycle level model for both the low and the high priority queues
- Frame level model – Cycle level model for the high priority queue and frame level model for the low priority queues.

Frame and cycle level models are illustrated in Figure 4.2.4 and Figure 4.2.5.

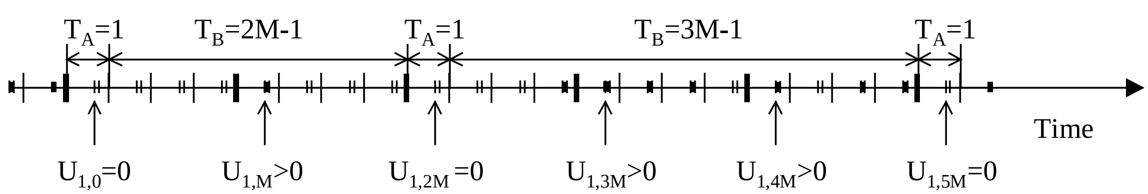


Figure 4.2.4 - Cycle level model

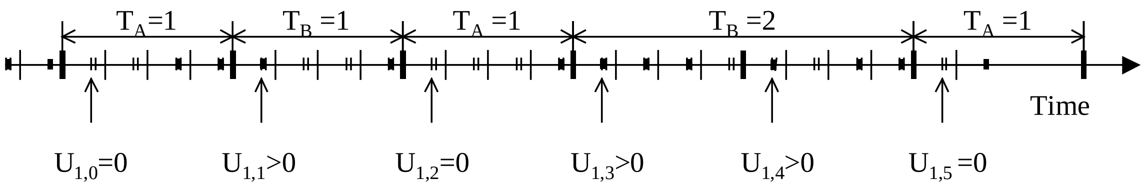


Figure 4.2.5 - Frame level model

Based on the description of the multiplexing system, two new random variables can be defined for each low priority source.

- length of the availability interval (or A-time) - T_A
- length of the blocking interval (or B-time) - T_B

The availability interval is the number of successive time-units (cycles or frames) when the output channel is open for the low priority source. Blocking interval is the number of successive time-units (cycles or frames) when the output channel is blocked for the source. Figures 4.2.4 and 4.2.5 display the availability and blocking times of the first low priority queue. Because low priority sources are multiplexed with time division multiplexing, the length of the availability times is always one time-unit (cycle or frame). Availability intervals occur when two events coincide:

- the high priority queue is empty
- the chosen low priority queue is allowed to transmit

The distributions of T_A and T_B random variables are important because two of the queuing models to be presented are based on these values instead of the distribution of the arrival process of the high priority source.

In the case of the *cycle level model*, the evolution equation for the system occupancy of the low priority queue is

$$U_{2,i,k+1} = \begin{cases} (U_{2,i,k} - (1-U_{1,k})^+)^+ + A_{2,i,k} & \text{if } k = Mn + i \text{ where } n \text{ is the frame number (integer)} \\ U_{2,i,k} + A_{2,i,k} & \text{otherwise} \end{cases} \quad (4.2.1)$$

with the notations of Figure 4.2.2.

In the case of the *frame level model*, the evolution equation for the length of the low priority queue is

$$U_{2,i,n+1} = (U_{2,i,n} - (1-U_{1,i,n})^+)^+ + A_{2,i,n} \quad (4.2.2)$$

where $U_{2,i,n}$ is the length of the low priority queue (priority 2) of source i in frame n and $A_{2,i,n}$ is the number of messages arrived to the low priority queue (priority 2) of source i in frame n. $U_{1,i,n}$ is the content of the high priority queue (priority 1) in the cycle of source i within frame n.

Each description has its advantages. The results obtained from the cycle level approach can be used when the TDM multiplexing method is compared to another technique in Section 4.4. Using the frame level description, two models (in Section 4.2.6) of the TDM multiplexing method can be compared.

The next factor to be considered is the arrival processes. Traffic models can be grouped to two main categories: independent and non-independent arrivals.

Independent arrival assumes that the sequence of the number of messages arrived in successive time-units is a sequence of independent and identically distributed random variables. Due to these properties the queue length and the number of arrived messages in a given time-unit (e.g.: cycle) are mutually independent random variables. That is, the queue length in a given cycle only depends on the arrivals in previous cycles that are independent of the actual arrival.

Non-independent arrival models are analyzed in e.g. in [BrKi93] assuming correlation in the arrival process. In those models the queue length and the arrival process are correlated that makes the description of the system more difficult.

4.2.2.2 Applied Models

Three models will be presented to describe the operation of the TDM multiplexing method. This section shortly introduces them. The detailed models are discussed in the following subsections (Sections 4.2.3, 4.2.4 and 4.2.5).

In the first model, which is called the *interrupted server model with uncorrelated interruptions* (later: **uncorrelated model**), it is assumed that the number of messages a low priority queue can serve in successive frames form a sequence of independent and identically distributed random variables. This criterion yields more restrictions on the arrival process of the high priority source. That is, the model can only be applied to systems where the high priority source can be characterized as a **Bernoulli source**. In the dissertation, only the **mean values** of the system content and the system time are discussed, but in the general case of **multi-slot channels**. The generating functions of these measures are expressed from the third accurate model for single-slot channels.

The advantage of this model is due to its simplicity:

- multi-slot channels can be analyzed
- the system time and system content can be expressed directly from the parameters of the arrival process

The weakness of this model is that

- there is a strict restriction on the arrival process of the high priority source
- only the frame level approach can be used, which is not accurate

The second model describes the operation of low priority sources with the GI-G-1 queuing model (later: **GI-G-1 model**). GI-G-1 is a basic discrete-time model with single server, infinite waiting room, independent arrivals and arbitrary service times [BrKi93, Hun83].

The service time – i.e. the time a low priority message spends at the first place in the queue – of low priority messages is upper bounded with the random variable that is the sum of an A-time and a B-time. A-time is deterministic and B-time is i.i.d. r.v. if the arrival process of the high priority source is independent. Therefore, substituting the real service time with the sum allows us to apply the GI-G-1 model.

The main weaknesses of the model are:

- it is only an approximation
- it can be applied to one-slot DTM channels
- in a general case the distribution of the blocking interval should be numerically calculated from the arrival process of the high priority source

Its advantages are:

- each source can have general independent arrival distribution
- can be applied to both cycle and frame scale models

The third model is based on the *interrupted server model using the distribution of the length of the availability and the blocking intervals of the output channel* [BrKi93, Bru84, Bru86] (later: **AB model**). In [BrKi93], the general model is analyzed where both A-times and B-times can have general independent distribution. In the case of TDM multiplexing method, a special case of this model can be used, so I was able to calculate the pdf of new system characteristics (system time, unfinished work) in addition to the ones already published elsewhere (system content).

The advantage of this model is its generality:

- the arrival process of the high priority traffic can be *general independent*
- the results can be applied to both frame and cycle level models
- the generating function, and thus also the moments (mean value and variance) and the tail distribution, can be obtained for the system contents, system time and unfinished work variables

The weakness of this model is that

- only single-slot channel can be analyzed
- in a general case the distribution of the blocking interval should be numerically calculated from the arrival process of the high priority source

4.2.3 Interrupted Server Model with Uncorrelated Interruptions

The first model is called *interrupted server model with uncorrelated interruptions*. It is assumed in the model that an independent identically distributed process interrupts the service of low priority queues.

The mean value analysis of a queuing system is described in [GyPa96], which is based on the assumption that the number of served messages per time-unit is independent and identically distributed random variable. This system has uncorrelated interruptions because the probability that no messages are served in a time-unit is independent from the past of the queue. Lemma 1 summarizes the conclusions taken from [GyPa96] without presenting the proof.

Lemma 1

Consider a system whose evolution equation has the form of

$$U_{n+1} = (U_n - C_n)^+ + A_n \tag{4.2.3}$$

where U_n is the system content in the n^{th} frame, A_n is the number of messages arrived during the n^{th} frame and C_n is the number of served messages in the n^{th} frame.

Assume that A_n and C_n are independent identically distributed random variables.

With the above assumptions the mean system content can be expressed as

$$E\{U\} = \frac{E\{A\} \cdot (1 - E\{A\}) + \text{var}\{A\}}{2 \cdot (E\{C\} - E\{A\})} \tag{4.2.4}$$

Now we should show when Lemma 1 can be applied to the description of the low priority queues in our system.

Let us generalize equation (4.2.2), which is the evolution equation of our system, to the case when the capacity of a channel can be more than 1 (denoted by c).

$$U_{2,i,n+1} = (U_{2,i,n} - (c - U_{1,i,n})^+)^+ + A_{2,i,n} \tag{4.2.5}$$

Equation (4.2.5) can be converted to the form of (4.2.3) if $(c - U_{1,i,n})^+$ is independent and identically distributed. It means that the $U_{1,i,n}$ random variable, which denotes the system content of the high priority queue in the i^{th} cycle of frame n, should also be independent and identically distributed. It only stands if **no queue builds up** in the high priority queue (messages are served during 1 cycle). That is, the number of high priority messages arrived in a cycle should be less than the capacity of the channel (c). A batch Bernoulli process with a batch-size less than c fulfils these requirements.

With these assumptions we can write that

$P(A_{1,i,n} > c) = P(U_{1,i,n} > c) = 0$.

Then follows that $E\{(c - U_{1,i,n})^+\} = E\{c - U_{1,i,n}\} = c - E\{U_1\} = c - E\{A_1\}$.

So the mean value of the system content is

$$E\{U_{2,i}\} = \frac{E\{A_{2,i}\}(1 - E\{A_{2,i}\}) + \text{var}\{A_{2,i}\}}{2(c - E\{A_1\} - E\{A_{2,i}\})} \tag{4.2.6}$$

for low priority source i. It should be noted that the unit of A_1 is message/cycle, while that of $A_{2,i}$ is message/frame.

Due to Little's theorem the system time is

$$E\{V_{2,i}\} = \frac{E\{U_{2,i}\}}{E\{A_{2,i}\}} = \frac{1 - E\{A_{2,i}\} + \text{var}\{A_{2,i}\} / E\{A_{2,i}\}}{2(c - E\{A_1\} - E\{A_{2,i}\})} \tag{4.2.7}$$

4.2.4 Approximation using the GI-G-1 queuing model

Let us recall the assumptions of the Gi-G-1 model.

- the arrival process characterising the source is general independent, i.e. the number of messages entering successive cycles are independent and identically distributed (non-negative integer) random variables
- the service time of each message is general independent
- there is one server in the system

In order to apply the Gi-G-1 to our system, we should show that the above three assumptions are true.

To fulfil the first and the third criteria, it should be assumed that the arrival process of the observed low priority source is general independent, and that the DTM channel consists of one slot in each cycle.

The second criterion, however, needs more attention. The service time should be defined as the time a message spends at the first place in the low priority queue plus the time spent in the server (which is the availability time: one cycle). This service time (T_S) is always less than the sum of a blocking interval (T_B) and an availability interval (T_A) because

- If the queue is not empty when the observed message arrives, the message reaches the first place in the queue right after an availability interval, and it leaves it when a blocking and an availability intervals have elapsed. $T_S = T_B + T_A$
- If the queue is empty when the observed message arrives, it reaches the first place immediately. It is there for the remaining part of the current blocking interval plus an availability interval. $T_S \le T_B + T_A$

Consequently, the service time is not an independent random variable. However, it can be replaced with random variable $S = T_B + T_A$, which is its upper bound. S is an i.i.d. random variable in our system because T_A is constant (1cycle long) and T_B is i.i.d. if the arrival process of the high priority source is i.i.d..

The next question is how to express S with know variables. The distribution of availability times is given (they are always 1 cycle long). The length of the blocking interval should be expressed from the arrival process of the high priority source. Now the derivation of these expressions is postponed, they are addressed in Section 4.2.5.

The pgf of the system time and system content for the Gi-G-1 system are described in [BrKi93]. Here only the final results, which are obtained in [BrKi93], are shown using the notations of the dissertation.

Generating function of the system content, which belongs to low priority source i and observed at random cycle boundaries ($U_{2,i}(z)$), has the following form:

$$U_{2,i}(z) = \frac{\left[1 - A_{2,i}^{'}(1) \cdot S^{'}(1)\right] \cdot (z-1) \cdot S(A_{2,i}(z))}{z - S(A_{2,i}(z))} \qquad (4.2.8)$$

Generating function of the integer part of the system time of low priority source i ($V_{2,i}(z)$) is

$$V_{2,i}(z) = \frac{\left[1 - A_{2,i}^{'}(1) \cdot S^{'}(1)\right] \cdot (z-1) \cdot S(z) \cdot \left[1 - A_{2,i}(S(z))\right]}{A_{2,i}^{'}(1) \cdot (1 - S(z)) \cdot \left[z - A_{2,i}(S(z))\right]} \qquad (4.2.9)$$

where $S(z)$ is the generating function of random variable S.

In the next section, a model in which the system characteristics can be computed accurately will be presented.

4.2.5 Interrupted Server Model Using the Distribution of the Length of the Availability and Blocking Intervals of the Server

4.2.5.1 Introduction

Applicability

The third model, which is called later as AB model, is also based on the distribution of the length of the availability and blocking intervals of the service process. The model of the TDM multiplexing method is a special case of the general model, which is presented in [BrKi93], because here the length of the availability interval is always 1 time-unit. In the TDM multiplexing method the independence of the B-times, which is due to the independence of the arrival process of the high priority source, allows the application of the AB model.

Background

The most general interrupted server model, which was based on the length of availability and blocking intervals and which assumed general independent A-times and B-times, was published in [BrKi93]. It derived the probability generating function of *system content* [Bru84, BrKi93]. Several other articles discussed system content in special cases, e.g. assuming Markovian server interruptions or geometric A-times [Tow80]. Delay characteristics, however, were studied only in very specific cases. Because there are only a few papers dealing with delay in the literature, a short overview about them is given below.

Delay was examined in connection with a special case of the model where both A-times and B-times were deterministic in many papers, e.g.: [SB91, SB92, RZ88]. This case occurs in STDMA systems, where several users share the link capacity so that they use one or more slots in a strictly periodic way.

The probability generating function of the delay of a randomly chosen packet was expressed for geometric A-time and general B-time distributions in [LaBr94].

[Sha81] also obtained the delay characteristics for Poisson arrivals using continuous-time models.

[RT89] discussed the delay of a system that is very similar to the multiplexing method described here. The paper analyzed priority-based TDMA schemes. Non-preemptive and preemptive resume scheduling disciplines were examined, and the pgf of the delay of messages was expressed.

The main difference between the TDMA systems in [RT89] and the TDM multiplexing method presented in the dissertation is that here several low priority sources are multiplexed also with TDM. *As a result, low priority sources and the high priority source operate at **different time scales**.*

In this section, it will be shown that the probability generating functions of system content and unfinished work can be obtained for the TDM multiplexing method.

Section outline

The remaining part of this section is structured as follows.

In Section 4.2.5.2, system content is analyzed. First, the probability generating functions is expressed based on [BrKi93]. Then the mean, variance and an approximation for the tail probability of the system content is derived.

In Section 4.2.5.3, the unfinished work of the low priority queue of the TDM system is analyzed. The generating function is expressed in closed form, which can be further analyzed to express other system characteristics.

In Section 4.2.5.4, the generating function of the system time of low priority sources is derived for the TDM system. Its mean, variance and an approximation for the tail probability distribution are also expressed.

The results of Sections 4.2.5.2-4 are based on the distribution of the length of availability and blocking intervals. If we know only the arrival process of the sources then a link should be found between B-time distribution and the arrival process of the high priority source. In Section 4.2.5.5, it is shown how to calculate the distribution of the length of the blocking intervals.

Finally, in Section 4.2.5.6, the applicability of results is illustrated with an example.

4.2.5.2 System Content

A general generating function formula for system content is derived in [BrKi93]. In this subsection a short proof is described for the pgf similar to that of [BrKi93] for the $T_A = 1$ case. Then the most important characteristics are expressed from the generating function.

Definitions and Notations

The final goal of the derivation of the generating function is to get an expression for the system content *in an arbitrarily chosen time-unit*, which is denoted by $U(z)$. As it shown later, however, conditional probabilities should be used during the derivation because the system content could be expressed under certain conditions.

Denoting conditional probabilities with standard notations would result in very long expression. Therefore, the conditions of conditional probabilities are shown in the indices. In order to avoid confusion, indices denoting the priority and the identity of the low priority source are omitted during the derivation.

The most important conditional probabilities are shown below:

$$P(U = u \mid \text{time - unit is in an A - time; } L_A = j, P = k) \equiv P(U_{A,j,k} = u)$$

$$U(z \mid \text{time - unit is in an A - time; } L_A = j, P = k) \equiv U_{A,j,k}(z)$$

where L_A is the length of the A-time and P is the position of the observed time-unit within the A-time. The conditioning, consequently, means that the observed time-unit is in an A-time, which has a length of $L_A=j$ (in our case j=1), and the time-unit is the k^{th} within the A-time.

A special notation is used for the $k=0$ case, which denotes the time-unit just before the first time-unit of an A-time (actually it is the last time-unit of a B-time). Because both the length of A-times and the length of B-times are independent random variables, the conditional probabilities belonging to $k=0$ are independent of the length of the A-time (L_A):

$$P(U=u \mid \text{time-unit is in an A-time; } L_A = j, P=0) = P(U=u \mid \text{time-unit is just before an A-time}) \equiv P(U_{A,0}=u)$$

$$U(z \mid \text{time-unit is in an A-time; } L_A = j, P=0) = U(z \mid \text{time-unit is just before an A-time}) \equiv U_{A,0}(z)$$

Another special notation is used, when the condition only specifies that the observed time-unit is in an A-time:

$$P(U=u \mid \text{time-unit is in an A-time}) \equiv P(\mathrm{U_A}=u)$$

$$U(z \mid \text{time-unit is in an A-time}) \equiv \mathrm{U_A}(z)$$

The corresponding notations are applied also for B-times:

$$P(U=u \mid \text{time-unit is in an B-time; } \mathrm{L_B = j, P = k}) \equiv P(\mathrm{U_{B,j,k}}=u)$$

$$U(z \mid \text{time-unit is in an B-time; } \mathrm{L_B = j, P = k}) \equiv \mathrm{U_{B,j,k}}(z)$$

where L_B is the length of the B-time and P is the position of the observed time-unit within the B-time. The conditioning, consequently, means that the observed time-unit is in a B-time, which has a length of $L_B=j$, and the time-unit is the k^{th} within the B-time.

The special notation for the $k=0$ case is:

$$P(U=u \mid \text{time-unit is in an B-time; } \mathrm{L_B = j, P = 0}) = P(U=u \mid \text{time-unit is just before a B-time}) \equiv P(\mathrm{U_{B,0}}=u)$$

$$U(z \mid \text{time-unit is in an B-time; } \mathrm{L_B = j, P = 0}) = U(z \mid \text{time-unit is just before a B-time}) \equiv \mathrm{U_{B,0}}(z)$$

The special notation, when the condition only specifies that the observed time-unit is in a B-time, is:

$$P(U=u \mid \text{time-unit is in a B-time}) \equiv P(U_B=u)$$

$$U(z \mid \text{time-unit is in a B-time}) \equiv U_B(z)$$

System time and unfinished work are denoted by V and W, respectively. During their derivation the same indexing is used, therefore the meaning of indices are summarized in Table 4.2.1.

Index	**Condition**
no index	Time-unit is arbitrarily chosen.
A,j,k	The observed time-unit belongs to an A-time, which is j units long. The position of the time-unit within the A-time is k.
$A,0$	The observed time-unit is just before an A-time.
A	The time-unit is arbitrarily chosen, but it belongs to an A-time.
B,j,k	The observed time-unit belongs to a B-time, which is j units long. The position of the time-unit within the B-time is k.
$B,0$	The observed time-unit is just before a B-time.
B	The time-unit is arbitrarily chosen, but it belongs to a B-time.

Table 4.2.1: Notations

$P_A(z)$ and $P_B(z)$ denote the generating function of the length of A-times and B-times, respectively.

Derivation of the generating function

As it is shown in Appendix A, we are able to express $U_{A,j,k}(z)$ and $U_{B,j,k}(z)$ generating functions at any valid j and k pairs. Based on the theorem of total probability the pgf of system time is the following, assuming that the arbitrarily chosen time-unit is in a B-time:

$$P(U_B = u) = \sum_{j=0}^{\infty}\sum_{k=1}^{j} P(U_{B,j,k} = u)\cdot P(K_B = k; J_B = j) \qquad (4.2.10)$$

where K_B is the position of the arbitrarily chosen time-unit within its B-time, and J_B is the length of the B-time that the chosen time-unit belongs to. The second factor in the argument of summation can be decomposed to known probabilities:

$$P(K_B = k; J_B = j) = P(K_B = k | J_B = j)\cdot P(J_B = j) \qquad (4.2.11)$$

The probability that the position of the arbitrarily chosen time-unit within its B-time is k $P(K_B = k | J_B = j)$ equals to $1/j$ if the length of the B-time is j. The reason is that the random variable is equally distributed on the [1,j] interval.

The probability that the length of the B-time of the randomly chosen time-unit equals to j is different from the distribution of an arbitrary B-time's length. It is proportional to the length of the interval and to the probability that length of an arbitrary B-time is j, which is denoted by $P(T_B = j)$. Note that we assume that we choose *from the time-units* according to a uniform distribution and *not from the B-times*. The formal proof for the distribution below can be found in [BrKi93, page 20] and in [Kle75].

$$P(J_B = j) = \frac{j\cdot P(T_B = j)}{E\{T_B\}} \qquad (4.2.12)$$

And now, (4.2.11) and (4.2.12) can be combined and substituted to (4.2.10):

$$P(U_B = u) = \frac{1}{E\{T_B\}}\sum_{j=0}^{\infty}\sum_{k=1}^{j} P(U_{B,j,k} = u)\cdot P(T_B = j) \qquad (4.2.13)$$

After z-transformation we obtain that

$$U_B(z) = \frac{1}{E\{T_B\}}\sum_{j=0}^{\infty}\sum_{k=1}^{j} U_{B,j,k}(z)\cdot P(T_B = j) \qquad (4.2.14)$$

As the length of all A-times is 1, the mass function and generating function of the system content in an arbitrarily chosen time-unit of an A-time is

$$P(U_A = u) = P(U_{A,1,1} = u) \text{ and } U_A(z) = U_{A,1,1}(z) \qquad (4.2.15)$$

From (4.2.14) and (4.2.15) the result for the pgf of the system content in an arbitrarily chosen time-unit can be expressed:

$$U(z) = \frac{1}{1+E\{T_B\}}U_A(z) + \frac{E\{T_B\}}{1+E\{T_B\}}U_B(z) \qquad (4.2.16)$$

where the weight of $U_A(z)$ is the fraction of time during which the output of the observed low priority queue is available and the weight of $U_B(z)$ is the fraction of time during which the output is blocked. The $E\{T_A\}=1$ condition is included in equation (4.2.16).

Now, we know how to express $U(z)$ from $U_{A,j,k}(z)$ and $U_{B,j,k}(z)$. The derivation of $U_{A,j,k}(z)$ and $U_{B,j,k}(z)$ can be found in Appendix A. The result is:

$$U_{B,j,k}(z)=\frac{(z-1)\cdot U_{A,1,0}(0)\cdot A(z)}{z-P_B(A(z))A(z)}\cdot A(z)^k \qquad \text{if } j\geq k \tag{4.2.17}$$

$$U_{A,1,1}(z)=\frac{A(z)(z-1)\cdot U_{A,1,0}(0)}{z-P_B(A(z))A(z)} \tag{4.2.18}$$

where $U_{A,1,0}(0)=1-(1+P_B'(1))A'(1)$

Now let us express $U_B(z)$ from (4.2.14) and (4.2.17)

$$U_B(z)=\frac{1}{E\{T_B\}}\sum_{j=0}^{\infty}\sum_{k=1}^{j}U_{B,0}(z)\cdot A(z)^k\cdot P(T_B=j)=\frac{U_{B,0}(z)\cdot A(z)\cdot(P_B(A(z))-1)}{E\{T_B\}\cdot(A(z)-1)} \tag{4.2.19}$$

$U(z)$ can be expressed from (4.2.18) and (4.2.19):

$$U(z)=\frac{U_{A,0}(0)A(z)(1-z)[1-A(z)P_B(A(z))]}{[1+P_B'(1)]\cdot[A(z)-1]\cdot[z-A(z)P_B(A(z))]} \tag{4.2.20}$$

We can see that the pgf of the B-time length $P_B(z)$ is always multiplied with z. It can be interpreted such that a random variable, which is the length of an A-time plus the length of a B-time, appears in expression (4.2.20). Therefore, we can rewrite (4.2.20) using the new random variable $S\equiv T_A+T_B=1+T_B$. The relation of generating functions is $S(z)\equiv P_A(z)\cdot P_B(z)=zP_B(z)$. Using again the indices indicating the priority and the identifier of the source, the following is the pgf of the system contents of a low priority source in an arbitrarily chosen time-unit:

$$U_{2,i}(z)=\frac{1-S'(1)A_{2,i}'(1)}{S'(1)}\cdot\frac{A_{2,i}(z)\cdot(1-z)\cdot[1-S(A_{2,i}(z))]}{[A_{2,i}(z)-1]\cdot[z-S(A_{2,i}(z))]} \tag{4.2.21}$$

Now, that the description of derivation of the pgf of system content is finished, I analyze this equation and express the most important system characteristics.

Mean and variance

The moments of the probability distribution can be obtained from the derivatives of the generating function in the $z = 1$ point.

The **mean value** of the system contents for any low priority source is

$$U'_{2,i}(1) = A'_{2,i}(1) + \frac{A'_{2,i}(1)\left(2P'_B(1) + P''_B(1)\right) + A''_{2,i}(1)\left(1 + P'_B(1)\right)^2}{2(1 + P'_B(1))\left(1 - A'_{2,i}(1)\cdot(1 + P'_B(1))\right)} \tag{4.2.22}$$

With other notations this expression can be written as

$$E\{U_{2,i}\} = E\{A_{2,i}\}\cdot\left(\frac{1}{2} + \frac{\frac{\text{var}\{A_{2,i}\}}{E\{A_{2,i}\}}E\{S\} + \frac{\text{var}\{S\}}{E\{S\}}}{2\left(1 - E\{S\}E\{A_{2,i}\}\right)}\right) = E\{A_{2,i}\}\cdot\left(\frac{1}{2} + \frac{k\{A_{2,i}\}E\{S\} + k\{S\}}{2\left(1 - E\{S\}E\{A_{2,i}\}\right)}\right) \tag{4.2.23}$$

where k is the peakedness of the argument. Peakedness is defined as $k\{X\} = \frac{\text{var}\{X\}}{E\{X\}}$. Other substitutions are: $E\{S\} = P'_A(1) + P'_B(1) = 1 + P'_B(1)$; $\text{var}\{S\} = \text{var}\{P_B\} = P''_B(1) + P'_B(1) - P'_B(1)^2$

The expression for the mean system content is relatively simple. It contains only the first two moments of the distribution of the B-time length random variable and the first two moments of the number of messages arrived from a low priority source during a time-unit.

When the system is overloaded, the queue is never empty, so the time between the departure of two messages from the queue is equal to S. The arrival intensity is always $A_{2,i}$ and there is one server in the system. So the stability criterion of the system is $E\{S\}E\{A_{2,i}\} < 1$. The same conclusion can also be drawn from expression (4.2.23), because the system content yields to infinity when $E\{S\}E\{A_{2,i}\}$ goes to 1.

The **variance** can be also obtained from equation (4.2.21).

$$\text{var}\{U_{2,i}\} = \text{var}\{A_{2,i}\} - \frac{2A'''_{2,i}(1) + 3A''_{2,i}(1)}{6A'_{2,i}(1)} - \frac{3A''_{2,i}(1)^2}{4A'_{2,i}(1)^2} + \frac{2o''' + 3o''}{6o'(1+o')} + \frac{3\left(o''\right)^2}{4\left(o'(1+o')\right)^2} \tag{4.2.24}$$

where $o(z) = 1 - A_{2,i}(z)\cdot P_B(A_{2,i}(z))$, and $o';o'';o'''$ are the first second and third order derivatives of $o(z)$ at z = 1 point, respectively. The proof of expression (4.2.24) can be found in Appendix B.

Tail distribution

The **tail of the probability mass function** of the system content is very important because it can be used to calculate the probability of having longer queue than a specified value. In many practical situations, the tail of the mass function has exponential distribution. Theorem 1 proves that this is the case for the system content of any low priority source if the conditions of stability are fulfilled. It also gives a formula for the tail of the mass function.

Theorem 1

If the stability criterion is fulfilled, the tail of the probability mass function of the system content of low priority source i can be expressed as

$$P(U_{2,i} = n) = \frac{\left(A'_{2,i}(1)\cdot(1 + P'_B(1)) - 1\right)\cdot A_{2,i}(z_0)(1 - z_0)^2}{(1 + P'_B(1))\cdot z_0\cdot(A_{2,i}(z_0) - 1)\cdot\left(1 - A'_{2,i}(z_0)\cdot\left(P_B(A_{2,i}(z_0)) + A_{2,i}(z_0)P'_B(A_{2,i}(z_0))\right)\right)} z_0^{-n} \tag{4.2.25}$$

where z_0 is real pole of generating function (4.2.25) with the smallest absolute value outside the unit circle.

Proof:

As $U_{2,i}$(z) is the generating function of the system content, it is analytic inside the complex unit circle, which also involves that the absolute values of its poles are greater than 1. In [BSDP94] it was shown that if the generating function X(z) of the integer valued random variable x has one positive real pole outside the unit circle and it has the form

$$X(z) = \frac{W(z)}{Y(z)} \tag{4.2.26}$$

where W(z) and Y(z) are polynomials then

$$Prob[x = n] \cong -c \cdot z_0^{-n-1} \text{ and } Prob[x > n] \cong \frac{c}{1 - z_0} \cdot z_0^{-n-1} \tag{4.2.27}$$

where $c = \frac{W(z_0)}{Y^{'}(z_0)}$ and z_0 is the positive real pole of $X(z)$ with the smallest modulus.

Now, let us introduce the notations $F(z) = A_{2,i}(z) - 1$; $G(z) = z - A_{2,i}(z) P_B(A_{2,i}(z))$, so the denominator of $U_{2,i}$(z) is $(1 + P_B^{'}(1)) F(z) G(z)$.

First we prove that $U_{2,i}$(z) has exactly one positive pole outside the unit circle. For this it is enough to show that F(z) has no positive real zero and G(z) has exactly one positive zero greater than 1, and at this value the numerator of $U_{2,i}$(z) can not be 0.

Since $F(z) = \sum_{k=0}^{\infty} P(A_{2,i} = k) z^k - 1$

$$F^{'}(z) = \sum_{k=1}^{\infty} P(A_{2,i} = k) \cdot k \cdot z^{k-1} > 0 \text{ if } z > 0. \tag{4.2.28}$$

Due to $F(1) = 0$ and $F^{'}(z) > 0$ $\forall z > 0$ the generating function $F(z)$ has no zero greater than 1.

Differentiating the other term G(z) we obtain

$$G^{'}(z) = 1 - A_{2,i}^{'}(z)\left(P_B(A_{2,i}(z)) + A_{2,i}(z) P_B^{'}(A_{2,i}(z))\right) \tag{4.2.29}$$

At z=1 it becomes $G^{'}(1) = 1 - A_{2,i}^{'}(1)\left(1 + P_B^{'}(1)\right)$ which is always greater than zero because the assumed stability condition for the system is $1 > A_{2,i}^{'}(1)\left(1 + P_B^{'}(1)\right)$. For the second derivative of G(z)

$$G^{''}(z) = (-1)\begin{bmatrix} A_{2,i}^{''}(z)\left(P_B(A_{2,i}(z)) + A_{2,i}(z) P_B^{'}(A_{2,i}(z))\right) + \\ A_{2,i}^{'}(z)^2 \left(2P_B^{'}(A_{2,i}(z)) + A_{2,i}(z) P_B^{''}(A_{2,i}(z))\right) \end{bmatrix} < 0, \ \forall i > 0 \tag{4.2.30}$$

because all values inside the brackets are positive $\forall z > 0$.

It means that $G'(z)$ becomes negative for sufficiently large z, and there is another zero of G(z) in addition to $z=1$. It also means that there is exactly one real-valued zero of the denominator of $U_{2,i}$(z). For this zero denoted by z_0 it holds that

$$z_0 = A_{2,i}(z_0)P_B(A_{2,i}(z_0)) \text{ and } z_0 > 1 \tag{4.2.31}$$

The numerator of $U_{2,i}$(z) at $z=z_0$ takes the value

$$\left(1-\left(1+P_B'(1)\right)A_{2,i}'(1)\right)A_{2,i}(z_0)(1-z_0)\left(1-A_{2,i}(z_0)P_B(A_{2,i}(z_0))\right) \tag{4.2.32}$$

which is strictly positive due to equation (4.2.31). Now we can state that z_0 is a real valued pole of $U_{2,i}$(z).

Now only parameter c is remained to compute in the approximation of tail probability. The derivative of the denominator of $U_{2,i}$(z) at $z=z_0$ is

$$\left(1+P_B'(1)\right)F(z)G'(z) \tag{4.2.33}$$

The numerator of $U_{2,i}$(z) at $z=z_0$ is

$$\left(A_{2,i}'(1)\cdot(1+P_B'(1))-1\right)\cdot A_{2,i}(z_0)(1-z_0)^2 \tag{4.2.34}$$

Now from equation (4.2.27) we can express (4.2.25), which is our result.

□

We obtained a simple exponential approximation for the tail distribution of the system content. If low priority sources transmit data traffic the tail probability distribution is the most important characteristics because it can be used to dimension the buffer size in order to keep the message loss probability below a certain level.

Now, we have the mean, the variance and the tail probability distribution for the system content. We can proceed to the analysis of other system characteristics.

4.2.5.3 Unfinished Work

The next random variable to study is unfinished work, which the time (number of cycles or frames) required to empty the queue. This measure is not as important in practice as system content and system time are, but it has a definite physical meaning: the time until the low priority queue contains messages after the time-instant when the arrival process is stopped. The results of this section are also used during the derivation of the generating function of the system content.

Definitions and Notations

The equilibrium generating function of unfinished work in an arbitrarily chosen time-unit (denoted by W(z)) is derived so as with system content, therefore the same indexing is used. That is, indices indicating the priority and the identity of the low priority sources are omitted until the final generating function is obtained to simplify the expressions. New indices are introduced, according to Table 4.2.1 to show the conditions, which apply to the generating functions. A few examples are shown below:

$$P(W = w | \text{time - unit is in an B - time}; L_B = j, P = k) \equiv P(W_{B,j,k} = w)$$

$$W(z | \text{time - unit is in an B - time}; L_B = j, P = k) \equiv W_{B,j,k}(z)$$

where L_B is the length of the B-time and P is the position of the observed time-unit within the B-time. The conditioning, consequently, means that the observed time-unit is in a B-time, which has a length of $L_B=j$, and the time-unit is the k^{th} within the B-time.

The special notation for the k=0 case is:

$$P(W = w | \text{time - unit is in a B - time}; L_B = j, P = 0) = P(W = w | \text{time - unit is just before a B - time}) \equiv P(W_{B,0} = w)$$

$$W(z | \text{time - unit is in a B - time}; L_B = j, P = 0) = W(z | \text{time - unit is just before a B - time}) \equiv W_{B,0}(z)$$

The special notation, when the condition only specifies that the observed time-unit is in a B-time, is:

$$P(W = w | \text{time - unit is in a B - time}) \equiv P(W_B = w)$$

$$W(z | \text{time - unit is in a B - time}) \equiv W_B(z)$$

The same definitions can be written for A-times.

Derivation of the generating function

First, the pgf of the conditional unfinished work need to be expressed assuming that the position of the time-unit and the length of the corresponding A-time or B-time are known. That is, first we are looking for the $W_{A,j,k}(z)$ and $W_{B,j,k}(z)$ conditional generating functions for all valid j and k.

After unconditioning is done in the same way as shown for system content, the unfinished work can be obtained for an arbitrarily chosen time-unit.

$$W_B(z) = \frac{1}{E\{T_B\}} \sum_{j=0}^{\infty} \sum_{k=1}^{j} W_{B,j,k}(z) \cdot P(T_B = j) \qquad (4.2.35)$$

$$W_A(z) = W_{A,1,1}(z) \qquad (4.2.36)$$

$$W(z) = \frac{1}{1+E\{T_B\}} W_A(z) + \frac{E\{T_B\}}{1+E\{T_B\}} W_B(z) \qquad (4.2.37)$$

The first step is to express the $W_{B,j,k}(z)$ generating functions.

$$W_{B,j,k} = \begin{cases} 0 & if \quad U_{B,j,k} = 0 \\ 1 + j - k + \sum_{i=2}^{U_{B,j,k}} (1 + T_{B,i}) & if \quad U_{B,j,k} \geq 1 \end{cases} \qquad (4.2.38)$$

The expression for the $U_{B,j,k} \geq 1$ case has two parts:

- The first part ($1+j-k$) means that the first message in the queue has to wait for *j-k* time-units (the remaining part of the current blocking period) until its service starts, and the service itself takes 1 additional time-unit.

- The second part - $\sum_{i=2}^{U_{B,j,k}}(1+T_{B,i})$ - means that the service of the next messages in the queue takes 1 whole blocking period (waiting for the service) and 1 time-unit (the service itself).

The generating function of $W_{B,j,k}$ from (4.2.38) is

$$W_{B,j,k}(z) = U_{B,j,k}(0) + \sum_{i=1}^{\infty} P(U_{B,j,k} = i)\cdot z^{1+j-k+\sum_{i=2}^{i}(1+T_{B,i})} \qquad (4.2.39)$$

After averaging over T_B we get

$$W_{B,j,k}(z) = U_{B,j,k}(0) + \frac{z^{1+j-k}}{zP_B(z)}\left(U_{B,j,k}(zP_B(z)) - U_{B,j,k}(0)\right) \qquad (4.2.40)$$

Using (A.2) we can convert (4.2.40) to

$$W_{B,j,k}(z) = U_{B,0}(0)\cdot A(0)^k + \frac{z^{1+j-k}}{zP_B(z)}\left(U_{B,0}(zP_B(z))\cdot A(z)^k - U_{B,0}(0)\cdot A(0)^k\right) \qquad (4.2.41)$$

Now we have reached our first goal: to express $W_{B,j,k}(z)$ generating functions.

The next step is to derive $W_A(z)$ and $W_B(z)$. Interestingly, both generating functions can be obtained from (4.2.41).

We can express $W_{B,0}(z)$ from (4.2.41) by substituting *k*=0 and averaging over *j*. The result is:

$$W_A(z) \equiv W_{A,1,1}(z) \equiv W_{B,0}(z) = U_{B,0}(zP_B(z)) \qquad (4.2.42)$$

After unconditioning (4.2.41) according to (4.2.35) we obtain $W_B(z)$:

$$W_B(z) = \frac{U_{B,0}(0)}{P_B^{'}(1)}\left(\frac{A(0)}{A(0)-1}\left(P_B(A(0))-1\right) - \frac{A(0)}{P_B(z)(A(0)-z)}\left(P_B(A(0)) - P_B(z)\right)\right) + \frac{U_{B,0}(zP_B(z))}{P_B^{'}(1)}\cdot\frac{A(zP_B(z))\cdot\left(P_B(A(zP_B(z))) - P_B(z)\right)}{P_B(z)\cdot(A(zP_B(z))-z)} \qquad (4.2.43)$$

The unfinished work in an arbitrarily chosen time-unit can be expressed from $W_A(z)$ and $W_B(z)$ according to (4.2.37).

$$W(z) = \frac{1}{1+P_B^{'}(1)}\left(U_{B,0}(0)A(0)\left(\frac{\left(P_B(A(0))-1\right)}{A(0)-1} - \frac{P_B(A(0)) - P_B(z)}{P_B(z)\left(A(0)-z\right)}\right) + U_{B,0}(zP_B(z))\left(\frac{A(zP_B(z))\cdot P_B\left(A(zP_B(z))\right) - P_B(z)z}{P_B(z)\cdot(A(zP_B(z))-z)}\right)\right) \qquad (4.2.44)$$

The final equation for $W(z)$ of any low priority queue is received after substituting (A.10) and (A.11):

$$W(z)=\frac{1-(1+P_B^{'}(1))A^{'}(1)}{1+P_B^{'}(1)}\left(\frac{\dfrac{A(0)}{P_B(A(0))}\left(\dfrac{P_B(A(0))-1}{A(0)-1}-\dfrac{P_B(A(0))-P_B(z)}{P_B(z)\big(A(0)-z\big)}\right)+}{\left(\dfrac{(1-zP_B(z))\cdot A(zP_B(z))}{P_B(z)\cdot(A(zP_B(z))-z)}\right)}\right) \tag{4.2.45}$$

The pgf of the unfinished work of a low priority queue in an arbitrarily chosen time-unit depends only on the pgf of B-time lengths and on the pgf of the arrival process of the low priority queue. We can see, however that the complex formula contains a constant, which has no physical meaning and which is difficult to calculate: $P_B(A(0))$. The mean, the variance and the tail probability of the unfinished work are not shown here because they would result in very long expressions. They can be computed so as with the corresponding measures of the system content. It can be seen from (4.2.45), however, that both the mean and the variance contain the unpleasant $P_B(A(0))$ constant.

4.2.5.4 System Time

In this subsection, the analysis of the system time of an arbitrarily chosen message is described. The system time in the dissertation includes queuing time and the time spent in the server. First, the derivation of the generating function is presented, then the derivation of the mean, the variance and the tail probability of the system time follow.

Definitions and Notations

The equilibrium generating function of system time in an arbitrarily chosen time-unit (denoted by V(z)) is derived so as with system content and unfinished work, therefore the same indexing is used. That is, indices indicating the priority and the identity of the low priority sources are omitted until the final generating function is obtained to simplify the expressions. New indices are introduced, according to Table 4.2.1 to show the conditions, which apply to the generating functions. Three examples are shown below:

$$V(z|\text{time - unit is in a B - time; } L_B = j, P=k)\equiv V_{B,j,k}(z)$$

where L_B is the length of the B-time and P is the position of the observed time-unit within the B-time. The conditioning, consequently, means that the observed time-unit is in a B-time, which has a length of $L_B=j$, and the time-unit is the k^{th} within the B-time.

The special notation for the k=0 case is:

$$V(z|\text{time - unit is in a B - time; } L_B = j, P=0)=V(z|\text{time - unit is just before a B - time})\equiv V_{B,0}(z)$$

The special notation, when the condition only specifies that the observed time-unit is in a B-time, is:

$$V(z|\text{time - unit is in a B - time})\equiv V_B(z)$$

Derivation of the generating function

The way of the derivation of the generating function is like at system content and unfinished work random variables. First, all $V_{A,j,k}(z)$ and $V_{B,j,k}(z)$ functions need to be expressed. Then

$V_A(z)$ and $V_B(z)$ can be calculated using the theorem of total probability. Finally $V(z)$ can be obtained using mean availability and mean blocking from $V_A(z)$ and $V_B(z)$.

The expression of the system time of the message arrived in the k^{th} time-unit of a B-time with length j has the following form:

$$V_{B,j,k} = j+1-k+\sum_{i=2}^{F}(T_{B,i}+1)+\sum_{i=1}^{U_{B,j,k-1}}(T_{B,i}+1) \tag{4.2.46}$$

where F is the ordinal number of the arbitrarily chosen message within the time-unit (in other words: the number of messages arrived in the same time-unit but not later than the arbitrary message (including the chosen one)); j is the length of the blocking interval that is going on when the message arrives; and $T_{B,i}$ is the length of an arbitrary blocking interval.

Equation (4.2.46) has three parts:

The first part ($j+1-k$) stands for the remaining part of the current B-time (j-k) plus the first A-time (1 time-unit).

The second part $\left(\sum_{i=2}^{F}(T_{B,i}+1)\right)$ is the service time (including the server interruptions) of the messages arrived in the same time-unit but before the arbitrarily chosen message (excluding the chosen one).

The third part $\left(\sum_{i=1}^{U_{B,j,k-1}}(T_{B,i}+1)\right)$ is the time needed to transmit the messages that were in the queue at the end of the preceding time-unit.

The generating function can be expressed from (4.2.46).

$$V_{B,j,k}(z) \equiv E\left\{ z^{\,j+1-k+\sum_{i=2}^{F}(T_{B,i}+1)+\sum_{i=1}^{U_{B,j,k-1}}(T_{B,i}+1)} \right\} \tag{4.2.47}$$

Averaging over T_B and F is straightforward:

$$V_{B,j,k}(z) = \frac{z^{\,j-k+1}F(zP_B(z))}{zP_B(z)}U_{B,j,k-1}(zP_B(z)) \tag{4.2.48}$$

Now let us express the unknown F(z) generating function. The $P(F=f)$ probability can be expressed in the following way using the total probability theorem:

$$P(F=f) = \sum_{n=1}^{\infty} P(F=f \mid N=n)\cdot P(N=n) \tag{4.2.49}$$

where N is the number of messages arrived in the same time-unit as the selected message and F is the position of the selected message within its time-unit.

The first factor in the summation can be expressed as

$$P(F = f \mid N = n) = \begin{cases} 0 & if \quad f > n \\ \dfrac{1}{n} & if \quad f \le n \end{cases} \tag{4.2.50}$$

because the random variable is equally distributed on the [1,*n*] interval if $n \geq f$.

The probability that the number of messages *in the time-unit of the randomly chosen message* equals to *n* is different from the distribution of the random variable that shows the number of messages arrived *in an arbitrarily chosen time-unit*. $P(N = n)$ is proportional to the number of messages arrived in the time-unit and to the probability that number of messages arrived in an arbitrary time-unit is *n*, which is already defined as $P(A = n)$. Note that it is assumed that we choose *from the messages* according to a uniform distribution and *not from the time-units*.

$$P(N = n) = \frac{n \cdot P(A = n)}{E\{A\}} \tag{4.2.51}$$

After combining (4.2.49), (4.2.50) and (4.2.51) we obtain

$$P(F = k) = \sum_{j=k}^{\infty} \frac{P(A = j)}{A^{'}(1)} \tag{4.2.52}$$

where *P(A=j)* is the pmf of the message arrived in an arbitrary time-unit belonging to the given low priority source and *A'(1)* is the mean of the same random variable. The generating function can be expressed from (4.2.52).

$$F(z) = \frac{z(A(z) - 1)}{(z - 1)A^{'}(1)} \tag{4.2.53}$$

The expression of $U_{B,j,k}(z)$ is already described at (4.2.17), so all factors in $V_{B,j,k}(z)$ are known. To keep the size of our expression short, we postpone the substitution of *F*(z) to a later point. As all expressions are know in (4.2.48), we can proceed to the derivation of the system time of the arbitrarily chosen message, which arrived in a B-time.

$V_B(z)$ can be expressed using the theorem of total probability and equation (A.2).

$$V_B(z) = \frac{F(zP_B(z)) \cdot U_{B,0}(zP_B(z))}{P_B(z) \cdot P_B^{'}(1)} \cdot \sum_{j=0}^{\infty} \sum_{k=1}^{j} z^{j-k} \cdot A(zP_B(z))^{k-1} \cdot P(T_B = j) \tag{4.2.54}$$

After the summation we obtain a closed form formula for $V_B(z)$

$$V_B(z) = \frac{F(zP_B(z)) \cdot U_{B,0}(zP_B(z))}{P_B^{'}(1)P_B(z)} \cdot \frac{P_B(A(zP_B(z))) - P_B(z)}{A(zP_B(z)) - z} \tag{4.2.55}$$

The next task is to work out the generating function $V_A(z)$. An expression can be written for the system time of in the first time-unit of the A-time (it is equivalent to: any time-unit of an A-time):

$$V_{A,1,1} = (W_{A,0} - 1)^{+} + \sum_{j=1}^{F} (1 + T_{B,j}) \tag{4.2.56}$$

The first term of equation (4.2.56) means the time needed to transmit messages that were in the system in the preceding time-unit. The second term is the service time (including the blocking interval) of the messages arrived in the same time-unit but not later than the arbitrarily chosen message (including itself).

To obtain the first term of (4.2.56) first $W_{A,0}(z)$ need to be expressed. The basis of its derivation is equation (4.2.41).

$$W_{A,0}(z) = \sum_{j=0}^{\infty} W_{B,j,j}(z) \cdot P(T_B = j) = \\ = \sum_{j=0}^{\infty} \left[\frac{(P_B(z)-1) \cdot U_{B,0}(0)}{P_B(z)} \cdot A(0)^j + \frac{U_{B,0}(zP_B(z))}{P_B(z)} \cdot A(z)^j \right] \cdot P(T_B = j) \tag{4.2.57}$$

Where we used again the theorem of total probability and the fact that the time-unit just before an A-time is the last time-unit of a B-time. So we made an unconditioning of the length of the B-times to obtain $W_{A,0}(z)$. The result after some calculations is

$$W_{A,0}(z) = \frac{P_B(A(zP_B(z))) \cdot U_{B,0}(zP_B(z))}{P_B(z)} + \frac{P_B(A(0)) \cdot U_{B,0}(0) \cdot (P_B(z)-1)}{P_B(z)} \tag{4.2.58}$$

Finally, (4.2.56) and (4.2.58) can be used to express $V_A(z)$:

$$V_A(z) = V_{A,1}(z) = \frac{F(zP_B(z))}{P_B(z)z} \left(U_{B,0}(zP_B(z)) \cdot P_B(A(zP_B(z))) + (zP_B(z)-1) \cdot U_{A,0}(0) \right) \tag{4.2.59}$$

Now, the final probability generating function $V(z)$ can be written as the weighted sum of $V_A(z)$ and $V_B(z)$, where the weights are the mean availability and blocking of the output channel, respectively.

$$V(z) = \frac{1}{1+P_B'(1)} V_A(z) + \frac{P_B'(z)}{1+P_B'(1)} V_B(z) \tag{4.2.60}$$

Using equations (4.2.53), (4.2.55), (4.2.59) and (4.2.60) the unknown pgf is obtained. Using the indices referring to the priority and the identity of the low priority source, it is

$$V_{2,i}(z) = \frac{1-(1+P_B'(1)) \cdot A_{2,i}'(1)}{(1+P_B'(1)) \cdot A_{2,i}'(1)} \cdot \frac{z(A_{2,i}(zP_B(z))-1)}{z - A_{2,i}(zP_B(z))} \tag{4.2.61}$$

We can see that the obtained generating function is very simple. It contains a constant (which is not a function of z), which contains the mean of the B-time length and the mean of the messages arrived from a low priority source during one time-unit. The second part of the pgf contains a nested function, where the argument of the pgf of the low priority arrival process $A_{2,i}(z)$ contains the product of the pgf of B-time length $P_B(z)$ and the pgf of A-time length (i.e.: $P_A(z) = z$). It means that the $S = T_A + T_B$ random variable appears in our equations again. Based on the generating function, a relatively simple mean, variance and tail probability approximation can be expected. The expression of the system time using random variable S is:

$$V_{2,i}(z) = \frac{1 - S'(1) \cdot A'_{2,i}(1)}{S'(1) \cdot A'_{2,i}(1)} \cdot \frac{z(A_{2,i}(S(z)) - 1)}{z - A_{2,i}(S(z))} \tag{4.2.62}$$

Mean and variance

Now that the generating function of the system time of an arbitrarily chosen message is given, its mean value can also be expressed as

$$V'_{2,i}(1) = 1 + \frac{A'_{2,i}(1) \cdot \left(P''_B(1) + 2P'_B(1)\right) - A''_{2,i}(1) \cdot \left(1 + P'_B(1)\right)^2}{2A'_{2,i}(1) \cdot (1 + P'_B(1)) \cdot \left(1 - A'_{2,i}(1)(1 + P'_B(1))\right)} = \frac{U'_{2,i}(1)}{A'_{2,i}(1)} \tag{4.2.63}$$

Note that the relation between (4.2.22) and (4.2.63) gives Little's result [e.g.: Kle75] because

$$E\{V_{2,i}\} = \frac{E\{U_{2,i}\}}{E\{A_{2,i}\}}$$

It is stated during the derivation of the system content that it is expressed for an arbitrarily chosen time-unit. It can be shown that it is statistically equivalent to the system content in the time-unit of an arbitrarily chosen message. That is why the Little theorem applies.

The mean value can be written using S instead of T_A and T_B.

$$E\{V_{2,i}\} = \frac{1}{2} + \frac{k\{A_{2,i}\}E\{S\} + k\{S\}}{2\left(1 - E\{S\}E\{A_{2,i}\}\right)} \tag{4.2.64}$$

The obtained mean value is also very simple, it contains only the first two moments of the distribution of the length of the B-time, and the first two moments of the distribution of the number of messages arrived from a low priority source during a time-unit. When the system is overloaded, the mean of the system time tends quickly to infinity.

Equation (4.2.61) can also be used to calculate higher moments of the distribution like the variance. With abbreviated notations the variance of the system time is

$$\mathrm{var}\{V_{2,i}\} = \frac{3r'' + 2r'''}{6r'(1 - r')} + \frac{3\left(r''\right)^2 (1 - 2r')}{4\left(r'(1 - r')\right)^2} \tag{4.2.65}$$

where $r(z) = A_{2,i}(z \cdot P_B(z)) - 1$ and $r'; r''; r'''$ are the derivatives of $r(z)$ in $z = 1$ point.

The derivation of equation (4.2.65) is based on Appendix B.

Tail distribution

The last characteristic expressed from the generating function of the system time is the tail probability. This characteristic is especially important if one wants to know the probability that a message should wait longer than a specified value.

Theorem 2 proves that the exponential form is a good approximation, and shows the formula for the calculation.

Theorem 2

Assuming the system has a stochastic equilibrium, the tail probability distribution of the system time can be expressed from equation (4.2.61) in the following form:

$$P(V_{2,i} = n) = -\frac{\left(1-(1+P_B^{'}(1))\cdot A_{2,i}^{'}(1)\right)(1-z_V)}{(1+P_B^{'}(1))\cdot A_{2,i}^{'}(1)\cdot\left(A_{2,i}^{'}(z_V P_B(z_V))\cdot(P_B(z_V)+z_V P_B^{'}(z_V))-1\right)} z_V^{-n} \qquad (4.2.66)$$

where z_V is the (real) pole of generating function (4.2.64), which is one of the solutions of equation $z = A_{2,i}(zP_B(z))$.

Proof:

First, it should be shown that $V_{2,i}(z)$ has only one positive real pole, which is greater than 1. The proof is based on the fact that denominator of $V_{2,i}(z)$ has one zero in addition to the $z = 1$ point. (Note that $z = 1$ is not a pole of $V_{2,i}(z)$ because this function is analytic inside the unit circle.) Let us introduce the notation

$$G(z) = z - A_{2,i}(zP_B(z)) = z - A_{2,i}(S(z)) \qquad (4.2.67)$$

Then $G^{'}(z) = 1 - A_{2,i}^{'}(S(z))\cdot S^{'}(z)$ and $G^{'}(1) = 1 - A_{2,i}^{'}(1)\cdot S^{'}(1)$. As $1 > A_{2,i}^{'}(1)\left(1+P_B^{'}(1)\right)$ is the assumed stability condition of the system, $G^{'}(1) > 0$ for every stable system. It can also be seen that

$$G^{''}(z) = -\left(A_{2,i}^{''}(S(z))\cdot S^{'}(z)^2 + A_{2,i}^{'}(S(z))\cdot S^{''}(z)\right) < 0 \text{ for all } z > 0 \qquad (4.2.68)$$

Consequently, $G^{'}(z)$ becomes negative for sufficiently large z. This means that $G(z)$ has one positive real zero besides $z = 1$ point, which is greater than 1. ($z = 1$ is not a pole of $V_{2,i}(z)$). Thus, $V_{2,i}(z)$ has exactly one real-valued pole, which is greater than 1. Therefore, expression (4.2.27) can be applied.

□

We obtained again a simple exponentially decaying approximation for the tail probability.

4.2.5.5 Distribution of the Length of Blocking Intervals

In the previous sections the key characteristics of the TDM multiplexing method are derived using the AB model. The connection between the model for the TDM multiplexing method and the queuing model hase not been discussed in detail yet. The link between the models is the calculation of the probability mass function (pmf) or probability generating function (pgf) of B-times in the queuing model using the arrival process of the high priority messages - $A_{1,k}$ - in the model of the multiplexing method. The distribution of the length of the blocking intervals also depends on the number of the sub-channels in a DTM channel (denoted by *M*).

First, the probability mass function of the length of the blocking intervals are expressed with the conditional probability $P(U_{iM} = 0 \mid U_0 = 0)$ where *M* is the number of cycles in a frame, *iM* is the number of cycles elapsed from the zero time instant and *U* is the system content of the high priority source.

Then it is shown how to calculate the probability mass function of this conditional probability and therefore the pmf of the length of the blocking intervals.

Expressing the distribution of the blocking intervals with a conditional probability

In order to avoid notational overhead the indices, which refer to the priority of the high priority source, are omitted.

Cycle level model

In the cycle level model the B-time is defined as the *number of cycles* between two consecutive cycles when a given low priority source is allowed to transmit a message (regarding the time division multiplexing among low priority sources) and when the high priority queue is empty. Due to the first condition the B-time must be *Mi-1* cycles long (where i is a positive integer). The actual value of i depends on the second condition, namely the length of the high priority queue in the observed time-points. So only the $P(T_B = Mi - 1)$ probabilities should be expressed, any other length has zero probability.

For the sake of simplicity let the zero time instant of the time axis be a cycle, which the given low priority source belongs to and where the length of the high priority queue is zero ($U_0 = 0$).

Using the definition, the B-time length can be expressed using the length of the high priority queue in different time instants:

$$P(T_B = Mi - 1) \equiv b(Mi - 1) = P(U_{iM} = 0, U_{(i-1)M} \neq 0, ..., U_M \neq 0 \mid U_0 = 0)\,. \qquad (4.2.69)$$

That is, the length of a B-time is equal to *Mi-1*, assuming that the cycle just before the B-time is the zero time instant where the high priority queue is empty, if and only if the high priority queue is not empty at time instants Mj for all j=1,2 …,i-1 and it is empty at time instant iM. What is the length of the high priority queue in other cycles than jM in the (1,iM) interval is irrelevant for the given low priority source.

Now the goal is to transform the equation to a computable form. The next theorem helps with it.

Theorem 3

Let us denote the conditional probability $P(U_{iM} = 0 \mid U_0 = 0)$ by $x_M(i)$ where i is the number of frame-times elapsed since the zero time instant and M is the number of cycles in a frame. For all $i > 0$ and $M > 1$ the following recursive formula can be applied to compute the probability mass function of the length of the B-times:

$$b(Mi - 1) = x_M(i) - \sum_{j=1}^{i-1} b(Mj - 1) \cdot x_M(i - j) \qquad (4.2.70)$$

Proof

The probability $b(Mi - 1)$ can be decomposed to two terms.

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_M}\neq \boldsymbol{0} \mid U_0=0)=$$
$$=\sum_{i=0}^{\infty} P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_M}=\boldsymbol{i} \mid U_0=0) - P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_M}=\boldsymbol{0} \mid U_0=0) \quad (4.2.71)$$
$$=P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{2M}\neq 0 \mid U_0=0) - P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_M}=\boldsymbol{0} \mid U_0=0)$$

The first term of the expression can be decomposed in the same way to two terms:

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_{2M}}\neq \boldsymbol{0} \mid U_0=0)=$$
$$=P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{3M}\neq 0 \mid U_0=0) - P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, \boldsymbol{U_{2M}}=\boldsymbol{0} \mid U_0=0) \quad (4.2.72)$$

The decomposition of the first term can be done *i*-1 times. Finally these expressions are obtained:

$$P(U_{iM}=0, \boldsymbol{U_{(i-1)M}}\neq \boldsymbol{0} \mid U_0=0) = P(U_{iM}=0 \mid U_0=0) - P(U_{iM}=0, \boldsymbol{U_{(i-1)M}}=\boldsymbol{0} \mid U_0=0) \quad (4.2.73)$$

$$P(U_{iM}=0 \mid U_0=0) \equiv x_M(i) \quad (4.2.74)$$

The second term of the above expressions can also be decomposed to two parts. Let us begin with the second part of (4.2.71):

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_M=0 \mid U_0=0)=$$
$$=P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{2M}\neq 0 \mid U_0=0, U_M=0)\cdot P(U_M=0 \mid U_0=0)= \quad (4.2.75)$$
$$=P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{2M}\neq 0 \mid U_M=0)\cdot P(U_M=0 \mid U_0=0)$$

Where the second step is due to the independence of B-times, which is due to memoryless property of the arrival process (the number of messages arrived in a cycle is an i.i.d. r.v.).

The factors of the product in (4.2.75) can be written using other notations.

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{2M}\neq 0 \mid U_M=0) = b(M(i-1)-1) \quad (4.2.76)$$

$$P(U_M=0 \mid U_0=0) = x_M(1) \quad (!4.2.76)$$

The second term of the other expressions can be transformed in the same way.

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{2M}=0 \mid U_0=0) = b(M(i-2)-1)\cdot x_M(2) \quad (4.2.78)$$

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_{3M}=0 \mid U_0=0) = b(M(i-3)-1)\cdot x_M(3) \quad (4.2.79)$$

...

Using the above equations

$$P(U_{iM}=0, U_{(i-1)M}\neq 0, \ldots, U_M\neq 0 \mid U_0=0) = x_M(i) - \sum_{j=1}^{i-1} b(Mj-1)\cdot x_M(i-j) \quad (4.2.80)$$

Which is the proof of (4.2.70).

□

So the derivation of the algorithm to calculate the distribution of the length of B-times in the cycle level model based on $P(U_{iM}=0\,|\,U_0=0)$ conditional probabilities is finished.

Frame Level Model

In the frame level model, the B-time is defined as the *number of frames* between two consecutive frames when a given low priority source is allowed to transmit (regarding the time division multiplexing among low priority sources) and when the high priority queue is empty. According to this definition the length of the B-time can be any non-negative integer (including zero).

The $P(T_B=i)$ mass function can be expressed so as with the cycle level model using the $x_M(i)$ conditional probabilities.

The frame level version of (4.2.70) is

$$b(i-1)=x_M(i)-\sum_{j=1}^{i-1} b(j-1)\cdot x_M(i-j) \qquad (4.2.81)$$

This statement can be proved in the same way as the formula for the cycle level model.

Calculating the used conditional probability

The $x_M(i)$ conditional probability is independent of the model of the multiplexing method (cycle or frame level), so the same applies to both models.

New notations are needed in this section:

$u_k(l)$ is the conditional probability that the system contains l messages in the k^{th} cycle given that it was zero in the zero time point. Formally $u_k(l)=P(U_k=l\,|\,U_0=0)$ for any cycle k and for all $l \geq 0$.

From the definition it follows that $u_k(l)\big|_{k=iM;l=0} \equiv x_M(i)$ and $u_k(l)\big|_{k=0;l=0} \equiv x_M(0)=1$ and $u_k(l)\big|_{k=0;l\neq 0}=0$.

$a(j)$ is the probability that j messages arrived from the high priority source during one cycle.

A formula can be obtained for $u_k(l)$ using the evolution equation of the high priority queue (4.1.1) .

$$u_{k+1}(l)=u_k(0)\cdot a(l)+\sum_{j=0}^{l} u_k(j+1)\cdot a(l-j) \qquad (4.2.82)$$

The formula and the initial $u_0(l)$ values for all $l \geq 0$ are known, so the $u_k(l)$ distribution can be determined numerically for any $k \geq 0$ and $l \geq 0$. It also means that $x_M(i)=u_{iM}(0)$ can be calculated.

Using (4.2.70) and (4.2.82) at the cycle level model or (4.2.81) and (4.2.82) at the frame level model, the probability distribution function of length of the B-times can be computed.

Expressing the generating function of B-times with the z-transform of the $x_M(i)$ conditional probability

The pgf of the B-times can be expressed as

Cycle level model:
$$P_B(z) = \frac{X_M(z^M)}{z \cdot \left(1 + X_M(z^M)\right)} \tag{4.2.83}$$

Frame level model:
$$P_B(z) = \frac{X_M(z)}{z \cdot \left(1 + X_M(z)\right)} \tag{4.2.84}$$

where $X_M(z) = \sum_{i=1}^{\infty} x_M(i) \cdot z^i$ is the z-transform of the $x_M(i)$ conditional probability.

For the proof of (4.2.83) is based on the definition of z-transformation and on (4.2.70).

$$P_B(z) \equiv \sum_{k=0}^{\infty} b(k) \cdot z^k = \sum_{i=1}^{\infty} b(iM-1) z^{iM-1} = \sum_{i=1}^{\infty} x_M(i) \cdot z^{iM-1} - \sum_{i=1}^{\infty} \sum_{j=1}^{i-1} b(Mj-1) \cdot x_M(i-j) \cdot z^{iM-1} =$$
$$= z^{-1} \cdot X_M(z^M) - \sum_{j=1}^{\infty} b(Mj-1) \cdot z^{Mj-1} \sum_{i=j+1}^{\infty} x_M(i-j) \cdot z^{(i-j)M} = z^{-1} \cdot X_M(z^M) - P_B(z) \cdot X_M(z^M)$$

(4.2.83) can be easily expressed from this equation.

The proof of the expression for the frame level model can be done in the same way, using (4.2.81)

$$P_B(z) \equiv \sum_{i=1}^{\infty} b(i-1) \cdot z^{i-1} = \sum_{i=1}^{\infty} x_M(i) \cdot z^{i-1} - \sum_{i=1}^{\infty} \sum_{j=1}^{i-1} b(j-1) \cdot x_M(i-j) \cdot z^{i-1} =$$
$$= z^{-1} \cdot X_M(z) - \sum_{j=1}^{\infty} b(j-1) \cdot z^{j-1} \sum_{i=j+1}^{\infty} x_M(i-j) \cdot z^{(i-j)} = z^{-1} \cdot X_M(z) - P_B(z) \cdot X_M(z)$$

4.2.5.6 An Example

The following simple example assumes that the high priority source can be characterized with a Bernoulli arrival process with generating function $A_1(z) = 1 - p + pz$, and all of the low priority sources have Batch Bernoulli arrival process with batch-size 30 and pgf. $A_{2,i}(z) = 1 - q + qz^{30}$. It is also assumed that the load coming from low priority sources equals to the load coming from the high priority source. That is, if M denotes the number of subchannels (and number of low priority sources):

$$A_1'(1) = p = M \cdot A_{2,i}'(1) = 30qM$$

As at most 1 slot arrives from the high priority source during a cycle and the output capacity is 1 slot in each cycle, no queue builds up at the high priority source. In other words, the system content has also Bernoulli distribution. Due to the independence of system contents in successive cycles, it can be written that

$$x_M(i) = P(U_{Mi} = 0 \mid U_0 = 0) = P(U_{Mi} = 0) = A_1(0) = 1 - p \qquad \text{if } i>0 \qquad (4.2.85)$$

The same conclusion can be drawn from equation (4.2.82). Substituting (4.2.85) to (4.2.70) a recursive formula is obtained:

$$b(Mi-1) = p \cdot b(M(i-1)-1) \qquad (4.2.86)$$

From equation (4.2.86) the pmf and the pgf of the B-times are

$$b(Mi-1) = (1-p)p^{i-1}; \qquad P_B(z) = z^{M-1}\frac{1-p}{1-pz^M} \qquad (4.2.87)$$

$$E[S] = \frac{M}{1-p}; \ \mathrm{var}\{S\} = \frac{M^2 p}{(1-p)^2} \qquad (4.2.88)$$

So if the high priority source can be characterized with a Bernoulli process, the generating function for the system content and the system time of a given low priority source can be expressed directly from the arrival distributions.

$$U_{2,i}(z) = \frac{\left[1 - p - MA'_{2,i}(1)\right] \cdot A_{2,i}(z) \cdot (1-z) \cdot \left[1 - A_{2,i}(z)^M\right]}{M \cdot \left[A_{2,i}(z) - 1\right] \cdot \left[z\left(1 - pA_{2,i}(z)^M\right) - A_{2,i}(z)^M (1-p)\right]} \qquad (4.2.89)$$

$$V_{2,i}(z) = \frac{z\left(1 - p - M \cdot A'_{2,i}(1)\right)}{M \cdot A'_{2,i}(1)} \cdot \frac{A_{2,i}\left(z^M \dfrac{1-p}{1-pz^M}\right) - 1}{z - A_{2,i}\left(z^M \dfrac{1-p}{1-pz^M}\right)} \qquad (4.2.90)$$

The mean values of the examined variables can be found below.

$$E\{U_{2,i}\} = E\{A_{2,i}\}\left(M\frac{k\{A_{2,i}\} + E\{A_1\}}{2\left(1 - E\{A_1\} - ME\{A_{2,i}\}\right)} + \frac{1}{2}\right); \qquad E\{V_{2,i}\} = \frac{E\{U_{2,i}\}}{E\{A_{2,i}\}} \qquad (4.2.91)$$

$E\{U_{2,i}\}$ is the mean system content of a given low priority source. $E\{U_{2,i}\}$ should be multiplied by M to obtain the sum of queue length for all low priority sources.

Figure 4.2.6 and Figure 4.2.7 show the sum of mean queue lengths and the mean of the system time for low priority sources, respectively.

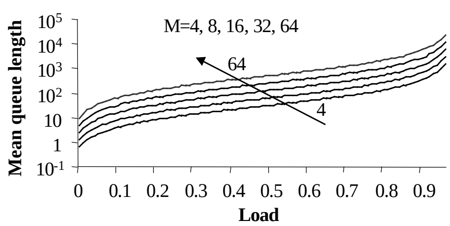


Figure 4.2.6 - Sum of mean system contents for low priority sources

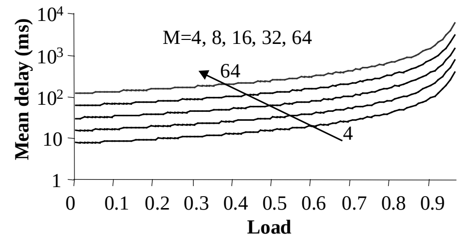


Figure 4.2.7 - Mean of the system time of any low priority source

Based on the tail probabilities two important parameters can be calculated:

- the limit of the queue length which is exceeded with a small probability (e.g.: 10^{-4} or 10^{-6})
- the probability that the delay is greater than a critical value (e.g.: 100 ms)

Though our model assumes infinite buffers, it was shown in [BSDP94] that the probability $P(U > U_0)$ is a good estimation of the message loss probability of a finite queue with $U_0 + c$ size where c is the number of servers (in this case 1). The size of the buffers is dimensioned so that the message loss probability should be below a certain value. Figure 4.2.8 and Figure 4.2.9 show the required buffer size if the maximum message loss rate is 10^{-4} and 10^{-6}, respectively.

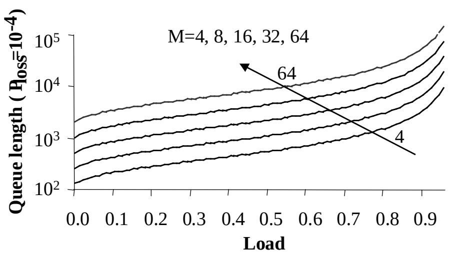


Figure 4.2.8 - Buffer size dimensioning for message loss rate of 10^{-4}

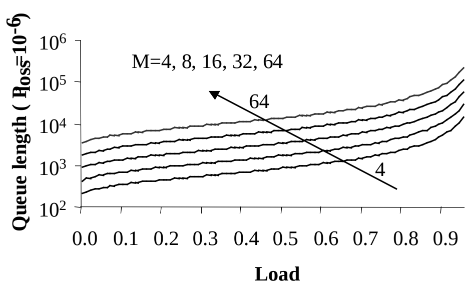


Figure 4.2.9 - Buffer size dimensioning for message loss rate of 10^{-6}

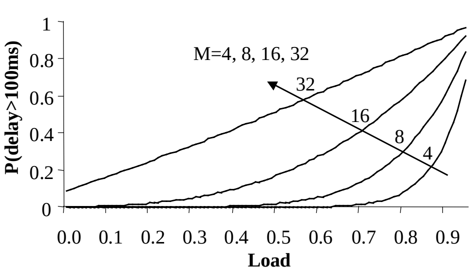


Figure 4.2.10 - Probability of having longer message delay than 100ms

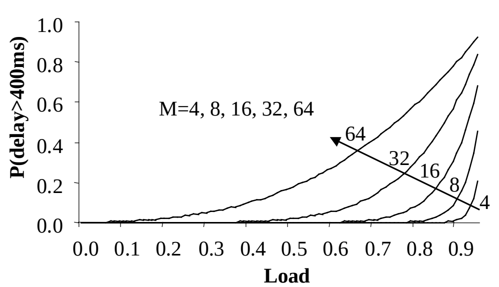


Figure 4.2.11 - Probability of having longer message delay than 400ms

In Figure 4.2.10 and Figure 4.2.11, the probabilities that the queuing delay is greater than 100 ms and 400 ms can be seen, respectively.

4.2.6 Connection between the models

Three models were presented in the previous sections:

- uncorrelated model (Section 4.2.3)
- GI-G-1 model (Section 4.2.4)
- AB model (Section 4.2.5)

The uncorrelated model is not accurate because it described the operation of low priority queues at frame level.

The GI-G-1 model also approximated the real system because it substituted the service time of low priority queues with the sum of one A-time and one B-time. This approximate service time can be described either in frame level or in cycle level. We only use the cycle level approach in the comparisons below.

The AB model is an exact description of the queuing system if the cycle level approach is applied. The AB model is identical with the uncorrelated model if we use the frame level approximation and if we assume that the high priority source can be characterized with a Bernoulli process.

In the next section, the cycle level and the frame level AB models is compared first. Then the cycle level GI-G-1 and the cycle level AB models are compared. Finally, the links between the frame level AB model and the uncorrelated model are described.

4.2.6.1 AB model at frame level vs. AB model at cycle level

First, let us have a look at the generating function of system content (equation (4.2.21)) and that of system time (equation (4.2.62)). Both generating functions could be interpreted both at the cycle level and at the frame level. In the two cases, however, the definition of $A_{2,i}(z)$ and $S(z)$ are different.

In the cycle level approach $A_{2,i,cycle}(z)$ is the number of messages arrived **in one cycle** and $S_{cycle}(z)$ is expressed **in cycles**.

In the frame level approach $A_{2,i,frame}(z)$ is the number of messages arrived **in one frame** and $S_{frame}(z)$ is expressed **in frames**.

Assuming that $A_{2,i,cycle}$ is an independent and identically distributed random variable, which need to be assumed at the AB model, $A_{2,i,frame}$ can be expressed as

$$A_{2,i,frame} = \sum_{k=1}^{M} A_{2,i,cycle} \qquad (4.2.92)$$

where M is the number of cycles in a frame. Using the above expression the link between the generating functions, mean values and variances are

$$A_{2,i,frame}(z) = \left(A_{2,i,cycle}(z)\right)^M \; ; \; E\{A_{2,i,frame}\} = ME\{A_{2,i,cycle}\}; \; \mathrm{var}\{A_{2,i,frame}\} = M\,\mathrm{var}\{A_{2,i,cycle}\} \qquad (4.2.93)$$

The link between the $S_{frame}(z)$ and $S_{cycle}(z)$ can be obtained from (4.2.83) and (4.2.84)

$$S_{cycle} = S_{frame}(z^M) \qquad (4.2.94)$$

where M is again the number of cycles in a frame. The connection between the means and the variances expressed from (4.2.94) is

$$E\{S_{cycle}\} = ME\{S_{frame}\};\ \mathrm{var}\{S_{cycle}\} = M^2\,\mathrm{var}\{S_{frame}\} \tag{4.2.95}$$

After some calculation the links between generating functions of the system content and system time can also be expressed. The substitutions are straightforward and the generating functions provide no real conclusions. Therefore, the comparison is presented based on the mean values of system content and system time.

Using (4.2.93) and (4.2.95) the expression for the system contents are

$$E\{U_{2,i,cycle}\} = \frac{E\{A_{2,i,frame}\}}{M}\cdot\left(\frac{1}{2} + \frac{k\{A_{2,i,frame}\}ME\{S_{frame}\} + Mk\{S_{frame}\}}{2(1 - E\{S_{frame}\}E\{A_{2,i,frame}\})}\right) \tag{4.2.96}$$

$$E\{U_{2,i,frame}\} = E\{U_{2,i,cycle}\} + \frac{M-1}{2}\cdot E\{A_{2,i,cycle}\} \tag{4.2.97}$$

System content is measured in messages, so the unit of the cycle level and the frame level model is the same. According to equation (4.2.97), the mean frame level system content is always bigger than its cycle level counterpart. The difference is proportional to M and to the mean of $A_{2,i,cycle}$.

$$E\{V_{2,i,frame}\} = \frac{1}{M}\left(E\{V_{2,i,cycle}\} + \frac{M-1}{2}\right) \tag{4.2.98}$$

Cycle level system time is measured in cycles and frame level system time is in frames. So to make the time-unit transformation the cycle level result should be divided by M. Consequently, what (4.2.98) shows is that the frame level system time is always larger by $\frac{M-1}{2}$ cycles than the cycle level system content. That is, the inaccuracy of the frame level approach is $\frac{M-1}{2}$ cycles.

4.2.6.3 Gi-G-1 vs. AB model

To compare the Gi-G-1 model and the AB model we can compare the generating functions of the system contents ((4.2.8) and (4.2.21)) and the generating functions of the system times ((4.2.9) and (4.2.62)):

$$U_{2,i,Gi-G-1}(z) = \frac{S'(1)\cdot S(A_{2,i}(z))}{1 - S(A_{2,i}(z))}\cdot\frac{1 - A_{2,i}(z)}{A_{2,i}(z)}\cdot U_{2,i,AB}(z) \tag{4.2.99}$$

$$V_{2,i,Gi-G-1}(z) = \frac{S'(1)\cdot S(z)}{1 - S(z)}\cdot\frac{1-z}{z}\cdot V_{2,i,AB}(z) \tag{4.2.100}$$

Using Lemma 1 in Appendix B, the connection between the mean values can be also expressed as

$$E\{U_{2,i,GG1}\} = E\{U_{2,i,AB}\} + \frac{E\{A_{2,i}\}}{2} \cdot \left(E\{S\} - 1 - k\{S\}\right) \qquad (4.2.101)$$

$$E\{V_{2,i,GG1}\} = E\{V_{2,i,AB}\} + \frac{E\{S\} - 1 - k\{S\}}{2} \qquad (4.2.102)$$

The relation between the mean values show that the system content of the GI-G-1 model can be expressed from that of the AB model by adding a term that is proportional to the mean value of $A_{2,i}$, to the mean and variance of S.

On the other hand, the inaccuracy of the system time of the GI-G-1 model depends only on the mean and the variance of S.

An interesting example is when the high priority source can be characterized with a Bernoulli process like in Section 4.2.5.6. In this case, using (4.2.88) we obtain that

$$E\{U_{2,i,GG1}\} = E\{U_{2,i,AB}\} + \frac{E\{A_{2,i}\}}{2} \cdot (M - 1) \text{ and } E\{V_{2,i,GG1}\} = E\{V_{2,i,AB}\} + \frac{M - 1}{2} \qquad (4.2.103)$$

Comparing (4.2.103) with (4.2.97) and (4.2.98), it can be seen that the inaccuracy of the GI-G-1 model and the uncorrelated model is the same if the high priority source can be characterized with a Bernoulli process.

4.2.6.3 Uncorrelated vs. frame level AB model

If the arrival process is a Bernoulli process with generating function $A_1(z) = 1 - p + pz$ then the generating function of the blocking intervals can be expressed by $A_1(z)$. In this case the length of the blocking intervals has exponential distribution. The probability mass function and the generating function of the length of the B-times are

$$b(i) = (1-p)p^i; \qquad P_B(z) = \frac{1 - A_1'(1)}{1 - A_1'(1)z} \qquad (4.2.104)$$

Using the results of the AB model combined with (4.2.104) the generating function of the system content and the system can be obtained:

$$U_{2,i}(z) = \frac{\left(1 - A_1'(1) - A_{2,i}'(1)\right) \cdot A_{2,i}(z) \cdot (1 - z)}{z - A_{2,i}'(1) - zA_1'(1)A_{2,i}'(1) + A_1'(1)A_{2,i}'(1)} \qquad (4.2.105)$$

$$V_{2,i}(z) = \frac{z(1 - A_1'(1) - A_{2,i}'(1))}{A_{2,i}'(1)} \cdot \frac{A_{2,i}\left(\frac{z - A_1'(1)z}{1 - A_1'(1)z}\right) - 1}{z - A_{2,i}\left(\frac{z - A_1'(1)z}{1 - A_1'(1)z}\right)} \qquad (4.2.106)$$

In these expressions only the parameters of the arrival processes are used.

The mean values can be obtained now in two ways: by derivating of (4.2.105) and (4.2.106) or by substituting the variance and the mean value of the S variable into (4.2.23) and (4.2.64).

From either direction the mean values of the system content and the system time are

$$E\{U_{2,i}\} = \frac{\operatorname{var}\{A_{2,i}\} + E\{A_{2,i}\}(1 - E\{A_{2,i}\})}{2(1 - E\{A_1\} - E\{A_{2,i}\})}; \quad E\{V_2\} = \frac{E\{U_{2,i}\}}{E\{A_{2,i}\}} \qquad (4.2.107)$$

These equations are identical to single server versions of equations (4.2.6) and (4.2.7), which are obtained from the uncorrelated model. The frame level AB model is identical to the uncorrelated model when the high priority source can be characterized with a Bernoulli process. Therefore, the identical results obtained from the two models show that *the results of the calculations are correct*.

4.2.6.4 Comparison when high priority source transmits Bernoulli process

Concluding the comparisons we can say that if the high priority source transmits according to a Bernoulli process, then the inaccuracies of the frame level AB model, the uncorrelated model and the GI-G-1 model are the same regarding the mean values of the system content and system time. Figure 4.2.12 shows the difference between the accurate cycle level AB model and the others when there are 8 cycles in a frame and average load of the high priority queue is 0.35 message in a cycle.

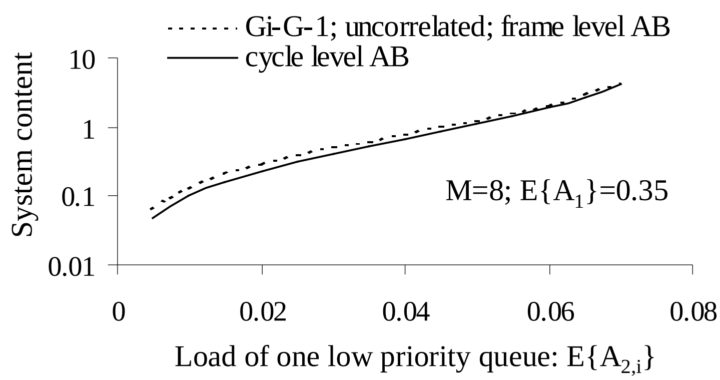


Figure 4.2.12 - Comparison of the cycle level AB model with the inaccurate models

Figure 4.2.12 shows that the closer the load of the low priority queue to its saturation point is (0.08125), the smaller the relative difference between the cycle level AB model and the inaccurate models is.

4.3 Packet switched multiplexing with Priorities

4.3.1 Description

An alternative multiplexing method is the so-called packet switched multiplexing with priorities. In this system, low priority sources share a common queue, which is served when the high priority queue is empty at the arrival of the slot of the corresponding DTM channel. Messages in the low priority queue are served according to the first come first served rule. The operation of the packet switched system is determined by the following rules:

Receivers distinguish between LP and HP sources so as with the TDM multiplexing method: a *priority bit* shows whether the slot carries message from a low priority source or from a high priority source.

The distinction between LP sources, however, is different. Connections of the transmitting low priority sources are identified by additional bits in the message header. These bits produce overhead for both the transmission capacity and the processing capacity. Therefore, the number of parallel connections could not be high.

A scenario can be easily imagined where the above described *packet switched multiplexing* can be applied: The DTM network is used as a transmission network for ATM networks. Hosts generating LP traffic are ATM switches. UBR and ABR ATM connections are set up within LP DTM connections. Real-time HP ATM connections are set up within HP DTM connections.

4.3.2 Models

The queuing model of the packet switched multiplexing method can be constructed from the description, which is shown in Figure 4.3.1.

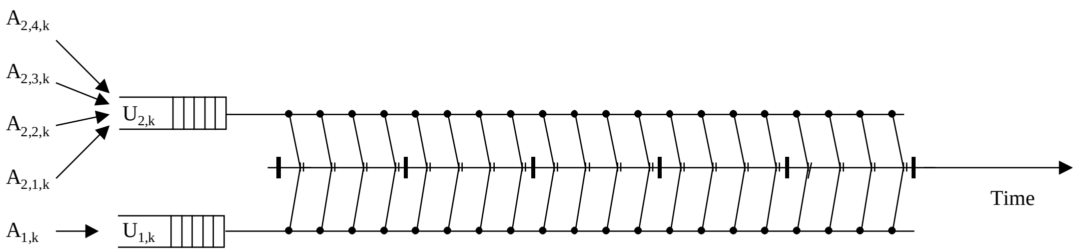


Figure 4.3.1 – Queues in the packet switched multiplexing method

The evolution equation of the high priority queue is independent of the low priority source, so that of a single source can be used – see (4.1.1).

$$U_{1,k+1} = (U_{1,k} - 1)^+ + A_{1,k} \qquad (4.3.1)$$

where $U_{1,k}$ is the system content of the high priority queue (priority 1) in cycle k and $A_{1,k}$ is the number of slots arrived to the high priority queue (priority 1) in cycle k.

The evolution equation of the low priority queue is

$$U_{2,k+1} = (U_{2,k} - (1 - U_{1,k})^+)^+ + \sum_{i=1}^{N} A_{2,i,k} \qquad (4.3.2)$$

where $U_{2,k}$ is the system content of the common queue of the low priority (priority 2) sources in cycle k and $A_{2,i,k}$ is the number of messages arrived to the low priority queue (priority 2) from source i in cycle k.

To simplify the expressions, a new notation is introduced for the sum of the arrivals from all low priority sources in a cycle.

$$\sum_{i=1}^{N} A_{2,i,k} = A_{2,c,k} \quad \text{and in equilibrium} \quad \sum_{i=1}^{N} A_{2,i} = A_{2,c} \qquad (4.3.3)$$

The next two subsections present two models for the packet switched multiplexing method.

The first one is the *interrupted server model with uncorrelated interruptions*. The advantages and drawbacks of the uncorrelated model are the same as that of the corresponding model of the TDM multiplexing method, which is presented in Section 4.2.3. This simple model is presented here to show a case where the results of the TDM method can be analytically compared to the packet switched method.

The second one, the *direct priority queuing model* is a more general description of the queue. It is presented and analyzed in [LB98] for multi-server queues. In the dissertation the single-server case will be described. The model is more general than the interrupted server model is because here the high priority source can have a general independent distribution.

4.3.3 Interrupted Server Model with Uncorrelated Interruptions

Using the interrupted server model the mean value of the system content and the system time can be obtained very quickly from (4.2.3). The multi-server version of equation (4.3.2) can be converted to the form of (4.2.3) if $(c-U_{1,k})^{+}$ is independent and identically distributed, where c is the number slots belonging to the corresponding DTM channel is a cycle. That is, the $U_{1,k}$ random variable should also be independent and identically distributed. It holds if $A_{1,k}$ is a batch Bernoulli process with batch-size less than c. So the expressions for the packet switched multiplexing method can be calculated so as with the time division multiplexing (see Section 4.2.3). The mean value of the system content of the common queue and the system time of messages of the low priority sources are

$$E\{U_2\}=\frac{E\{A_{2,c}\}(1-E\{A_{2,c}\})+\mathrm{var}\{A_{2,c}\}}{2(c-E\{A_1\}-E\{A_{2,c}\})} \tag{4.3.4}$$

$$E\{V_2\}=\frac{E\{U_2\}}{E\{A_{2,c}\}}=\frac{1-E\{A_{2,c}\}+\mathrm{var}\{A_{2,c}\}/E\{A_{2,c}\}}{2(c-E\{A_1\}-E\{A_{2,c}\})} \tag{4.3.5}$$

4.3.4 Direct Priority Queue Model

The direct priority queue model gives the full description of the queue in the case of general independent sources and multi-slot channels. In this section, a short derivation of the generating function of the system content is presented for single-slot channels based on the derivation presented in [LB98]. The generating function of the system time and the mean of both random variables will be also described. For a more detailed analysis of the multi-server case see [LB98].

4.3.4.1 System Content

The most straightforward way to obtain the generating function of the system content is to express the two dimensional generating function based on the single-server case when c=1 in equation (4.3.2). The first step is to interpret the meaning of the evolution equations:

$$P(U_{1,k+1}=i;U_{2,c,k+1}=j)=\begin{cases} P(U_{1,k+1}=A_{1,k};U_{2,c,k+1}=A_{2,c,k}) & \text{if}\quad \mathrm{U}_{1,\mathrm{k}}=0 \text{ and } \mathrm{U}_{2,\mathrm{c},\mathrm{k}}=0 \\ P(U_{1,k+1}=A_{1,k};U_{2,c,k+1}=U_{2,c,k}-1+A_{2,c,k}) & \text{if}\quad \mathrm{U}_{1,\mathrm{k}}=0 \text{ and } \mathrm{U}_{2,\mathrm{c},\mathrm{k}}\neq 0 \\ P(U_{1,k+1}=U_{1,k+1}-1+A_{1,k};U_{2,c,k+1}=U_{2,c,k}+A_{2,c,k}) & \text{if}\quad \mathrm{U}_{1,\mathrm{k}}\neq 0 \end{cases} \tag{4.3.6}$$

The definition of the two-dimensional generating function of the system content is

$$U_k(z_1, z_2) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P(U_{1,k} = i; U_{2,k} = j)\cdot z_1^i \cdot z_2^j \qquad (4.3.7)$$

After some algebraic manipulations the equilibrium generating function of the system content for the one-server case can be expressed as

$$U(z_1, z_2) = \frac{A_1(z_1)\cdot A_{2,c}(z_2)}{(z_1 - A_1(z_1)\cdot A_{2,c}(z_2))\cdot z_2}\left[z_1 \cdot \left(U(0,0)\cdot(z_2 - 1) + U(0, z_2)\right) - z_2 \cdot U(0, z_2)\right] \qquad (4.3.8)$$

Using the notation of $x(z_2) = \frac{U(0, z_2)}{U(0,0)\cdot(z_2 - 1) + U(0, z_2)}$ and $K(z_2) = \frac{U(0,0)\cdot(z_2 - 1)}{z_2 \cdot (1 - x(z_2))}$ we can write that

$$U(z_1, z_2) = \frac{A_1(z_1)\cdot A_{2,c}(z_2)\cdot K(z_2)}{z_1 - A_1(z_1)\cdot A_{2,c}(z_2)}\cdot(z_1 - x(z_2)) \qquad (4.3.8)$$

As the generating function $U(z_1, z_2)$ is a analytic inside the unit circle, the denominator should vanish, which defines the function $x(z_2)$. That is, $x(z_2)$ is the solution of the equation

$$x(z_2) = A_1(x(z_2))\cdot A_2(z_2) \qquad (4.3.9)$$

The derivatives of $x(z)$ at z=1 point can be determined from equation (4.3.9). Furthermore, $U(0,0) = (1 - A_{2,c}^{'}(1))(1 - A_1^{'}(1))$ because the $U(0,0)$ is the probability that the system is empty. Now the mean value can be expressed with some calculations

$$U_2^{'}(1) = \frac{E\{A_{2,c}\}}{2} + \frac{\mathrm{var}\{A_{2,c}\} + E\{A_{2,c}\}\cdot \frac{\mathrm{var}\{A_1\}}{1 - E\{A_1\}}}{2\cdot\left(1 - E\{A_1\} - E\{A_{2,c}\}\right)} \qquad (4.3.10)$$

The variance and the tail probabilities are analyzed in [LB98].

4.3.4.2 System Time

The pgf of the system time from [LB98] applied to the single-server case is

$$V_2(z) = z\frac{\left(1 - A_1^{'}(1) - A_{2,c}^{'}(1)\right)\cdot A_1(\omega(z))\cdot\left(A_{2,c}(\omega(z)) - 1\right)}{A_{2,c}^{'}(1)\left(\omega(z) - A_1(\omega(z))A_{2,c}(\omega(z))\right)} = \frac{\left(1 - A_1^{'}(1) - A_{2,c}^{'}(1)\right)\cdot\left(A_{2,c}(\omega(z)) - 1\right)}{A_{2,c}^{'}(1)\cdot\left(z - A_{2,c}(\omega(z))\right)} \qquad (4.3.11)$$

where $\omega(z)$ is the solution of equation

$$\omega(z) = zA_1(\omega(z)) \qquad (4.3.12)$$

To calculate the mean of the system time $\omega^{'}(1)$ and $\omega^{''}(1)$ should be known. By deriving both sides of (4.3.12), substituting z=1, and using that $\omega(1) = 1$ the following expressions are obtained

$$\omega'(1) = \frac{1}{1 - A_1'(1)}; \quad \omega''(1) = \frac{\mathrm{var}\{A_1\} + A_1'(1)(1 - A_1'(1))}{(1 - A_1'(1))^3} \tag{4.3.13}$$

Using (4.3.13) after some calculations the mean of the system time can be expressed.

$$V_2'(1) = \frac{\dfrac{\mathrm{var}\{A_1\}}{1 - A_1'(1)} + k\{A_{2,c}\}}{2\left(1 - A_1'(1) - A_{2,c}'(1)\right)} + \frac{1}{2} = \frac{U_2'(1)}{A_{2,c}'(1)} \tag{4.3.14}$$

For the detailed derivation of the pgf and the analysis of the variance and the tail probabilities see [LB98].

4.3.5 Connection of the Models

Now the results obtained in the previous sections can be compared. In this section the mean values obtained from the direct priority queue model will be expressed for the case when the high priority source transmits according to Bernoulli process. Then the comparison of the mean values received from the direct priority queue model and the results obtained from the uncorrelated interruption model follows.

The length of the high priority queue is an independent identically distributed random variable if the arriving messages are served immediately in the first cycle after the arrival. If the arrival process is a Bernoulli process with generating function $A_1(z) = 1 - p + pz$ then the necessary parameters to calculate the mean values of the system content and system time from (4.3.10) and (4.3.14) are the followings:

$$\mathrm{var}\{A_1\} = E\{A_1\} - E\{A_1\}^2 \tag{4.3.15}$$

Using (4.3.15) the mean values are obtained for the direct priority queue model with Bernoulli high priority sources

$$E\{U_2\} = \frac{E\{A_{2,c}\}(1 - E\{A_{2,c}\}) + \mathrm{var}\{A_{2,c}\}}{2(1 - E\{A_1\} - E\{A_{2,c}\})}; \quad E\{V_2\} = \frac{E\{U_2\}}{E\{A_{2,c}\}} \tag{4.3.16}$$

As expression (4.3.16) is identical with the equations in (4.3.4) and (4.3.5). That is, we obtained the same results from both models when the low priority source can be characterized with a Bernoulli process.

4.4 Comparison of the Multiplexing Solutions

In this section, the multiplexing solutions analyzed in the dissertation will be compared. It is assumed that there are N low priority sources connected the DTM channel, and they have the same arrival processes. Furthermore, the TDM multiplexing system is assumed to have M cycles in a frame where $M \geq N$.

A general analysis would require numerical calculation. In the special case, however, when the high priority source generates messages according to a Bernoulli process, the comparison becomes simpler. Because of this simplicity the dissertation compares the systems in this case. First, the mean characteristics of the TDM multiplexing method will be expressed then those of the packet switched multiplexing follow.

4.4.1 TDM multiplexing

The accurate *cycle level AB model* of the TDM multiplexing method is used for the comparison.

If the high priority source can be characterized with a Bernoulli arrival process with generating function $A_1(z) = 1 - p + pz$, then we can use the results of Section 4.2.5.6. Based on (4.2.91) the mean of the system content and the system time is

$$E\{U_{2,i,TDM}\} = E\{A_{2,i}\}\left(M\,\frac{k\{A_{2,i}\} + 1 - ME\{A_{2,i}\}}{2(1 - E\{A_1\} - ME\{A_{2,i}\})} - \frac{(M-1)}{2} \right) \qquad (4.4.1)$$

$$E\{V_{2,i,TDM}\} = M\,\frac{k\{A_{2,i}\} + 1 - ME\{A_{2,i}\}}{2(1 - E\{A_1\} - ME\{A_{2,i}\})} - \frac{(M-1)}{2}\,;\; E\{V_{2,i}\} = \frac{E\{U_{2,i}\}}{E\{A_{2,i}\}} \qquad (4.4.2)$$

As there are N low priority sources connected to the DTM channel, the whole mean system content is N times more than that of one source.

$$E\{U_{2,TDM}\} = NE\{U_{2,i,TDM}\} = NE\{A_{2,i}\}\left(M\,\frac{k\{A_{2,i}\} + E\{A_1\}}{2(1 - E\{A_1\} - ME\{A_{2,i}\})} + \frac{1}{2} \right) \qquad (4.4.3)$$

The mean of the system time for an arbitrary message is the same as that of a given message.

$$E\{V_{2,TDM}\} = E\{V_{2,i,TDM}\} \qquad (4.4.4)$$

where in the index TDM stands for the TDM multiplexing method.

4.4.2 Packet switched multiplexing

Because there are N independent low priority sources with the same characteristics, the variance and the mean value of the sum of the arrivals can be expressed with the descriptor of a single source.

$$E\{A_{2,c}\} = N \cdot E\{A_{2,i}\} \text{ and } \operatorname{var}\{A_{2,c}\} = N \cdot \operatorname{var}\{A_{2,i}\} \qquad (4.4.5)$$

The mean values of the system characteristics can be written now using (4.4.5) and (4.3.16)

$$E\{U_{2,PS}\} = NE\{A_{2,i}\}\left(\frac{k\{A_{2,i}\} + E\{A_1\}}{2(1 - E\{A_1\} - NE\{A_{2,i}\})} + \frac{1}{2} \right) \qquad (4.4.6)$$

$$E\{V_{2,PS}\} = \frac{E\{U_2\}}{NE\{A_{2,i}\}} = \frac{k\{A_{2,i}\} + E\{A_1\}}{2(1 - E\{A_1\} - NE\{A_{2,i}\})} + \frac{1}{2} \qquad (4.4.7)$$

The following relation can be noticed between the characteristics of the systems:

$$E\{V_{2,TDM}\} - \frac{1}{2} = \left(E\{V_{2,PS}\} - \frac{1}{2} \right) \cdot \frac{1 - E\{A_1\} - NE\{A_{2,i}\}}{1 - E\{A_1\} - ME\{A_{2,i}\}} \cdot M \qquad (4.4.8)$$

$$E\{U_{2,TDM}\} - \frac{NE\{A_{2,i}\}}{2} = \left(E\{U_{2,PS}\} - \frac{NE\{A_{2,i}\}}{2}\right) \cdot \frac{1 - E\{A_1\} - NE\{A_{2,i}\}}{1 - E\{A_1\} - ME\{A_{2,i}\}} \cdot M \qquad (4.4.9)$$

To express the relation between the multiplexing methods I defined the gain function below:

$$G = \frac{E\{U_{2,TDM}\} - \frac{NE\{A_{2,i}\}}{2}}{E\{U_{2,PS}\} - \frac{NE\{A_{2,i}\}}{2}} = \frac{E\{V_{2,TDM}\} - \frac{1}{2}}{E\{V_{2,PS}\} - \frac{1}{2}} \qquad (4.4.10)$$

From (4.4.8) and (4.4.10) and due to $E\{A_{2,i}\} < \frac{1}{M}$

$$G = M \cdot \frac{1 - E\{A_1\} - NE\{A_2\}}{1 - E\{A_1\} - ME\{A_2\}} = \left(1 + \frac{(M-N) \cdot E\{A_2\}}{1 - E\{A_1\} - ME\{A_2\}}\right) \cdot M < M + \frac{M-N}{1 - E\{A_1\} - ME\{A_2\}} \qquad (4.4.11)$$

That is, the gain function is the product of two factors:

- the first is M
- the second is the ratio of the free capacity in the systems and the free capacity of the system with M sources

The second factor is always greater or equal to 1 because $M \geq N$. It expresses an additional advantage of the packet switched solution, namely if there are less sources in the system than the maximum in the TDM solution there are unused low priority subchannels. The larger the difference between M and N is (the more unused low priority subchannels are in the TDM solution) the bigger the second factor is.

The advantages of the packet switched solution are summarized here:

- If all subchannels are used in the TDM method ($N=M$), the gain function equals to M. It means that with the TDM method messages are delayed nearly M times longer.
- There is no upper limit on the number of multiplexed sources
- and the bandwidth allocated by the LP connection can be any value within the capacity limits of the DTM channel.

The above comparison emphasized the superior properties of the packet switched system as [SW81]. Despite of the analytical results the packet switched multiplexing method has also drawbacks:

- Its implementation is more difficult because of the complex receiving algorithm and the fast operation of the network. Only a small fraction of DTM data channels can be used for multiple service classes. Otherwise, the receivers will not be able to process the control information (i.e. the headers of low priority messages) in the data slots.
- The headers of low priority messages mean larger overhead for the transmission.

4.5 Conclusions on Multiplexing Methods

Two multiplexing methods were presented in this chapter to increase the utilization of a DTM channel and support two priority classes. In both methods, multiple flexible sources with low priority (referred to as LP - low priority - source) are multiplexed with a delay sensitive source with high priority (referred to as HP -high priority - source) in a DTM channel.

The prioritized time division on two time scales multiplexing method (TDM method) has the following properties:

- The HP source is allowed to transmit in all time-slot of the DTM channel.
- LP sources share the remaining bandwidth using time division multiplexing: M successive cycles form a frame. A LP source is allowed to transmit in one cycle of the frame if it is not used by the HP source.

The other multiplexing method, *packet switched multiplexing*, works as follows:

- low priority sources generate packets with start and end delimiters; receivers differentiate between low priority sources based on these delimiters
- low priority sources share the same queue, which is only served if the buffer of the high priority source is empty

It was shown for the **TDM method** that the message delay of low priority sources and the length of low priority queues could be modeled with three different models:

- GI-G-1 model
- AB model
- Uncorrelated model

The comparison and analysis of the models was discussed in detail. The AB model using the distribution of the availability (A-times) and blocking periods (B-times) was based on the most general assumptions about the high priority source, and it provided the most general and exact results. In the dissertation, closed form expressions were obtained for the probability generating function of system content, system time and unfinished work of the low priority sources for the AB model.

The results were checked in two ways:

- It was proved that the mean values of system content and system time calculated from generating functions with the AB model fulfilled Little's theorem.
- It was also proved that the most general model AB model gave the same results as those of un-correlated model when the same assumptions about the high priority source were used.

New results were presented related to the AB model, which is based on the distribution of A-times and B-times:

- Closed form expressions were obtained for the generating functions of *unfinished work* and *system time* (message delay) for the case when $T_A = 1$ and T_B had general independent distribution.
- The most important characteristics – i.e. *mean values, variances and tail probability* – of the system content and system time were calculated from the generating functions.

Two known models were applied to the **packet switched multiplexing** method:

- Uncorrelated Server Interruptions

- Direct Priority Queue

Both models gave the same results when using most restrictive assumptions.

Finally, the performance of the TDM and the packet switched multiplexing methods was compared through a simple example. It was shown that the packet switched method outperforms the TDM technique regarding both queue length and message delay.

A note was also made that the mathematical models did not consider the drawbacks of the packet switched solution, i.e. the implementation complexity and the overhead of message headers.

Chapter V: Conclusions

5.1 Summary

The body of the dissertation was divided into three parts. The description of the *DTM protocol*, *the simulation study* about the fairness and aggregate performance of slot allocation methods, and the *mathematical analysis* of different multiplexing methods in a DTM channel were presented in different chapters.

In the first part, an overview about the DTM architecture was presented. It included a general classification of media access protocols with the positioning of DTM. The detailed explanation of the operation of the basic DTM protocol can also be found in this chapter. The enhancements of DTM were also outlined, including slot reuse, wavelength division multiplexing and interoperation with IP protocol. Finally, performance studies available in the literature were summarized.

The second part of the dissertation was dedicated to the simulation study of call level characteristics. The modeling and implementation issues of the simulation software, the network model of the simulations, and the simulation results were discussed in detail. The main emphasis was on fairness analysis of different slot allocation methods and on the performance study of slot allocation methods based on smoothing algorithms. Several conditions were examined in the simulations, for example:

- three different network scenarios: 1. external connections through a switching node at the end of the bus; 2. client-server connections with a server in the middle of the bus; 3. peer-to-peer connections, i.e. everybody is connected to everybody
- two different source models: WWW source model and Poisson model
- two different bus-length: 100 m inter-node distance and 10 km inter-node distance
- several slot allocation algorithms with several parameter settings

The mathematical analysis of message level characteristics of multiplexing methods was placed in the third main chapter. Two multiplexing methods were analyzed with the means of discrete time queuing theory. For the TDM multiplexing method three models were presented. The packet switched method was analyzed using two different models. The main differences between these five models were:

- application area: arrival process of the high priority source, rate of the DTM channel (one-slot channel, or multirate channel)
- accuracy of results: approximate and exact models
- time scale of the models in the TDM method: cycle and frame level

Finally, the TDM method was compared to the packet switching method. It was shown that the TDM method has longer delay and larger queues.

The main contributions of the dissertation are:

- performance analysis of different slot allocation methods with a great emphasis on fairness
- recommendations for new slot allocation methods and their performance analysis

- modeling of different multiplexing methods
- queuing analysis of the TDM multiplexing method with the model, which is based on the distribution of the availability and blocking intervals

5.2 Application Areas

Slot allocation methods are very important when *burst switching* is used. Analysis of message delays and buffer lengths within a successfully established DTM connection are relevant for *multiplexing methods*. Therefore, results of the dissertation can be applied to both performance improvement methods described in the Introduction.

Based on the analysis and comparison of slot allocation algorithms the operation of DTM network nodes can be designed. Dimensioning of slot allocation parameters - like retry limit, priority - and selecting the best application area of slot allocation algorithms - bus-length, traffic type - are also possible based on the results. The main novelty of performance analysis results is that other performance studies of DTM networks regard slot allocation algorithms using status tables, and here the emphasis is on algorithms without tables.

During the evaluation of different variants of the basic KTH slot allocation algorithm the main emphasis was on fairness. Although fairness is one of the most important characteristics of media access protocols, it has not been studied for DTM except my contributions. It was shown that even algorithms using *status tables* could become unfair in case of big inter-node distances. The study also emphasized that the *request order* during slot requests influences significantly the fairness of DTM networks.

New algorithms proposed in the dissertation managed to break down the inherent cache-like behavior of DTM networks without major performance loss. The aggregate performance of algorithms operating without status tables was also improved.

Although queuing analysis of multiplexing methods was presented in the context of DTM networks, results are more general and can be applied to any TDM system. Besides the analysis of the multiplexing methods, the discussion and comparison of more models help to better understand each model. New results in connection with the multiplexing methods are applicable to the calculation of e.g. message loss vs. buffer length characteristics. Dimensioning of network elements can be done based on the results discussed here. A good understanding how the high priority source, the number of multiplexed sources, and buffer length effect system time and unfinished work of low priority sources can be obtained based on the results of the analysis.

DTM is a new networking technology. As the first products are released in 1999 [Netins, Dynarc] ongoing research has significant impact on future products and standardization process.

5.3 Future directions

There are several areas to be studied as a continuation of the simulation study of DTM networks:

- The most important application area of the DTM is interconnection of IP networks. DTM is based on resource reservation but the traffic demand of IP traffic changes dynamically. Further work is needed to merge call level results in existing performance evaluations and IP traffic models, in order to obtain good *dimensioning parameters for resource management*.
- The analysis of *priority settings of smoothing algorithms* was studied in specific cases. A more general investigation is to be done.
- The performance study of the dissertation is based on one-slot unicast DTM connections. Therefore, further simulations are needed to study *multi-rate and multicast connections*.
- This work is restricted to the simulation of a DTM dual-bus. The study of *routing mechanisms* in a network containing several connected DTM dual-buses is another direction of future research.
- Fairness of media access protocols using *slot reuse and wavelength division multiplexing* attracted great attention at other networking technologies. Fairness studies of these enhancements of DTM are also important research areas.

Although the queuing analysis of multiplexing methods involved many different models, the application area of the models can be further extended using *correlated models*, which are more suitable for several source types.

Combination of different *multiplexing methods* is also an interesting field of future studies.

References

[AH93] B. Ahlgren and M. Hidell, "The DTM Host/Network Interface - Hardware, Software Design and Protocol Layering", *In Proc. 6th MultiG Workshop*, Kista, Sweden, May 1993.

[AL97] G. Anastasi, L. Lenzini “Performance Evaluation of a MetaRing MAC Protocol Carrying Asynchronous Traffic”, *Journal of High Speed Networks*, Vol 6, No 1, 1997.

[ALM97] G. Anastasi, L. Lenzini, B. Meini, “Performance Evaluation of a Worst Case Model of the MetaRing MAC Protocol with Global Fairness”, *Performance Evaluation*, Vol. 29, N. 2, March 1997, pp. 127 - 151.

[ALS94] H.R. van As, W.W. Lemppenau, H.R. Schindler, P. Zafiropoulo, “CRMA-II: A MAC protocol for ring-based Gb/s LANs and MANs” *Computer Networks and ISDN Systems* 26 (1994) 831-840

[Ams89] S. R. Amstutz, "Burst Switching - An Update", *IEEE Communications Magazine*, Vol. 27, No. 9, pp. 50-57, Sep. 1989.

[APL97] G. Anastasi, M. La Porta, L. Lenzini, “Delay analysis of a worst-case model of the MetaRing MAC protocol with local fairness” *Computer Communications* 20 (1997) pp. 671-680

[ATM96] ATM Forum, *“Traffic Management Specification Version 4.0”*, April, 1996

[Bac96] C. Bach; “Fairness Issues of the MAC protocol in the Gbit/s range”, ATM, Networks and LANs, *Proceedings of the European Conference on Networks and Optical Communications* (NOC’96), Amsterdam, Netherlands, 1996 pp. 30-37

[BHL94] C. Bohm, M. Hidell and P. Lindgren, "DTM - A True Broadband Concept", *In Proc. 6th ERCIM Workshops*, Stockholm, Sweden, pp. 159-168, June 1994.

[BL95] C. Bohm and P. Lindgren, "Distributed Synchronous Switching for Multi Channel Systems", In Proc. *IEEE International Workshop on Broadband Switching Systems*, Poznan, Poland, pp. 177-185, Apr. 1995.

[BLR93] C. Bohm, P. Lindgren, L. Ramfelt and P. Sjödin, "Resource Reservation in DTM", In Proc. *First IEEE Symposium on Global Data Networking*, Cairo, Egypt, pp 191-197, Dec. 13-15, 1993.

[BLR94] C. Bohm, P. Lindgren, L. Ramfelt and P. Sjödin, "The DTM Gigabit Network", *Journal of High-Speed Networks*, Vol. 3, No. 2, pp. 109-126, 1994.

[BLR96] C. Bohm, P. Lindgren, L. Ramfelt and P. Sjödin, "Fast Circuit Switching for the Next Generation of High Performance Networks", *IEEE Journal on Selected Areas in Communication*, Vol. 14, No. 2, pp. 298-305, Feb. 1996.

[Boh94] C. Bohm, "*The DTM Protocol- Design and Implementation"*, Licentiate Thesis, Royal Institute of Technology, ISRN KTH/IT/R-94/05-SE, Stockholm, Sweden, Mar. 1994.

[Boh96] C. Bohm, *"Circuit Switching for High Performance Integrated Services Networks"*, PhD Thesis, Royal Institute of Technology, ISRN KTH/IT/R-96/10-SE, Stockholm, Sweden, June 1996.

[Bra90] C.A. Brackett., "Dense Wavelength Division Multiplexing Networks: Principles and Applications", *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 6, pp 948-964, Aug. 1990.

[BrKi93] H. Bruneel, B. G. Kim; *Discrete-Time Models for Communications Systems Including ATM*; Kluwer Academic Publishers, Dordrecht, 1993

[Bru84] H. Bruneel, “Analysis of Discrete-time Buffers with One Single Output Channel, Subjected to a General Interruption Process”, *Proc Performance ’84*, Paris, Dec. 1984, pp. 103-115

[Bru86] H. Bruneel, “A general Treatment of Discrete-time Buffers with one Randomly Interrupted Output Line”, *European Journal of Operational Research*, 1986, vol. 27, no. 1, pp. 67-81

[BSDP94] H. Bruneel, B. Steyaert, E. Desmet, G. H. Petit; “Analytic derivation of tail probabilities for queue lengths and waiting times in ATM multiserver queues”, *European Journal of Operational Research* 76, pp. 563-572, 1994, North-Holland

[Cal91] R. Calvo, “An Architectural Overview of FDDI-II”, *Interop91*

[CB96] M. E. Crovella, A. Bestavros, “Self-Similarity in the World Wide Web Traffic: Evidence and Possible Causes”, *Proc. SIGMETRICS ’96*, Philadelphia, May 23-26, 1996

[CCO92] J. Chen, I. Cidon, Y. Ofek, “A local fairness algorithm for the MetaRing and its performance study” *Proc. IEEE GLOBECOM’92*, Orlando, December 1992

[CN98a] C.-J. Chang, A. A. Nilsson. “Analytical Model for DTM Access Nodes”, *Technical Report*, Center for Advanced Computing and Communication, North Carolina State University, Raleigh, TR-98/11

[CN98b] C.-J. Chang, A. A. Nilsson. “Performance Evaluation of DTM Access Nodes”, *Technical Report*, Center for Advanced Computing and Communication, North Carolina State University, Raleigh, TR-98/05

[CO90] I. Cidon, Y. Ofek, “Metaring – A Full-duplex Ring with Fairness and Spatial Reuse”, *Proc. IEEE Infocom’90*, San Francisco, CA, June 1990, pp.969-981

[DBL93] A. Danthine et al., "QoS Enhancements and the New Transport Services", *Local Area Network Interconnection* pp. 1-22, edited by R. O. Onvural and A. Nilsson, Plenum Press, 1993

[Den96] S. Deng, "Empirical model of WWW document arrivals at access link", *IEEE International Conference on Communications*, pp. 1797-802 vol.3, ICC'96, 23-27 June 1996, Dallas, TX, USA

[DGL90] N. R. Dono, P. E. Green, K. Liu, R. Ramaswami and F. F. Tong, "A Wavelength Division Multiple Access Network for Computer Communication", *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 6, pp. 983- 994, Aug. 1990.

[Dynarc] Dynarc Co. home page, http://www.dynarc.se

[FM96] N. Finn and T. Mason, "ATM LAN Emulation," *IEEE Communications Magazine*, June 1996, pp. 96-100.

[GH85] D.Gross, C. M. Harris, *"Fundamentals of Queuing Theory"*, Second Edition, John Wiley & Sons, 1985

[GHP92] L. Gauffin, L. Håkansson and B. Pehrson, "Multi-gigabit networking based on DTM," *Computer Networks and ISDN Systems*, Vol. 24, No. 2, pp. 119- 130, 1992.

[Gir90] A. Girard, "*Routing and Dimensioning in Circuit-Switched Networks*", Addison-Wesley Publishing Co., 1990

[Gre96] D. Greenfield "An Ethernet Path to Voice and Data Harmony", *Data Communications*, Jan. 1996, pp. 94-96

[GyPa96] L. Györfi, I. Páli, *"Tömegkiszolgálás informatikai rendszerekben"*, Műegyetemi Könyvkiadó, 1996, Budapest, id:55024

[Hag85a] E. Hagersten, "Duper - En kommunicationsmekanism", XT/DU 85 073, *Internal report*, Ericsson, Oct. 16, 1985.

[Hag85b] E. Hagersten. "DUR - Duper-realisering av ett distribuerat telefonsystem", XT/DU 86 024, *Internal report*, Ericsson, Oct. 9, 1985.

[Hag86] E. Hagersten, "En kort beskrivning av Duper-protokollet" XT/DU 86 022, *Internal report*, Ericsson, Feb. 17, 1986.

[Hey98] P. Heywood, "Circuit Switching, Scandinavian-Style", *Data Communications*, Nov. 21 1998

[Hid96] M. Hidell, "*Supporting Internet Protocols in a High Performance Real-Time Network*", Licentiate Thesis, Royal Institute of Technology, ISRN KTH/IT/ R-96/07- SE, Stockholm, Sweden, Apr. 1996.

[Hol98] M. Holmlund, "IP traffic in a DTM network" *Net Insight white paper*, 1998, http://www.netinsight.se/pdf/ipodintro.pdf

[HSNLab] home page of High Speed Networks Laboratory, Department of Telematics and Telecommunications, Technical University of Budapest, http://hsnlab.ttt.bme.hu

[HU90] J. Hsieh, D. R. Ucci, "Modeling and Analysis of Burst Switching for Integrated Voice/Data Services", *IEEE International Conference on Communications*, ICC'90, 1990, pp.1186-1190

[Hun83] J. J. Hunter, "*Mathematical Techniques of Applied Probability, vol. 2, Discrete Time Models: Techniques and Applications*", Academic Press, 1983

[IEEE97] *IEEE Personal Communications*, Special Issue, IMT-2000: Standards Efforts of the ITU, M. H. Callendar, Ed., vol. 4, no. 4, Aug. 1997

[IEEE98] *IEEE Transactions on Vehicular Technology*, Special Issue on the Universal Mobile Telecommunication System (UMTS), P. Jung and Z. Zvonar, ed., vol 47., no. 4, Nov. 1998

[IHK90] K. Imai, T. Honda, H. Kasahara, T. Ito; "ATMR: Ring Architecture for broadband networks", *IEEE GLOBECOM'90*, pp. 1734-1738

[IIK94] K. Imai. T. Ito, H. Kasahara, N. Morita, "ATMR: Asynchronous Transfer Mode Ring Protocol", *Computer Networks and ISDN Systems* 26 (1994), pp. 785-798

[Jai91] R. Jain, “*The Art of computer systems performance analysis:techniques for experimental design, measurement, simulation and modeling”*, John Wiley and Sons Inc., New york, 1991

[JCH84] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems", *DEC Research Report* TR-301, September 1984, http://www.cis.ohio-state.edu/~jain/papers/fairness.htm

[JR86] R. Jain and S. Routhier, "Packet Trains - Measurements and a New Model for Computer Network Traffic", *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, pp. 984-995, Sep. 1986.

[JSAC90] Special Issue on Dense WDM Networks, *IEEE Journal on Selected Areas in Communication*s, Vol. 8, No. 6, Aug. 1990.

[KA97] P. Karlsson, A. Arvidsson, “The Characteristics of WWW Traffic and the Relevance to ATM”, *Technical Document* COST257(97)21, COST 257 Management Commitee Meeting, Helsinki, Finland, May 1997

[Kah98] L. Kahn, “Introduction to DTM”, *Net Insight white paper*, 1998, http://www.netinsight.se/pdf/dtmintro.pdf

[Kar93] G. Karlsson, "Stockholm Gigabit Network", In Proc. *First IEEE Symposium on Global Data Networking*, Cairo, Egypt, pp. 7-14, Dec. 1993.

[Kle75] L. Kleinrock, “*Queuing Systems. Vol. I:Theory*”. Wiley-Interscience, New York, 1975

[KWH94] A. E. Kamal, J. W. Wong, H. S. Hassanein, “An algorithm for slot reuse in DQDB networks with erasure nodes”, *Computer Networks and ISDN Systems* 26, 1994, pp. 695-710

[LaBr94] K. Laevens, H. Bruneel, “Performance Analysis of a Discrete-time Queue with a Single Server Subjected to Random Interruptions”, *Book of Abstracts ORBEL 8*, Brussels, January 1994, pp. 37-38

[LB94] P. Lindgren and C. Bohm, "Fast Connection Establishment in the DTM Gigabit Network", In Proc. *5th IFIP Conference on High Performance Networking*, Grenoble, France, pp. 283-294, June 27- July 1, 1994.

[LB98] K. Laevens, H. Bruneel; “Discrete-Time Multiserver Queues with Priorities”, *Performance Evaluation* 33 (4), 1998, pp. 249-275

[Lin94] P. Lindgren, "*Host Interfacing and Connection Management in the DTM Gigabit Network*", Licentiate Thesis, Royal Institute of Technology, ISRN KTH/IT/R-94/04- SE, Stockholm, Sweden, Mar. 1994.

[Lin96] P. Lindgren, “*A Multi-Channel Network Architecture Based on Fast Circuit Switching*”, Ph.D. Thesis, Royal Institute of Technology, ISRN KTH/IT/R-96/08-SE, Stockholm, Sweden, May 1996

[LK91] A. M. Law, W. D. Kelton, *“Simulation Modeling & Analysis”*, McGraw-Hill International Editions, 1991

[Mar94] P. Martini, “Connection Oriented Data Service in DQDB”, *Computer Networks and ISDN Systems* 26 ,1994, pp. 679-694

[Mar97] B. G. Marchent, "Third Generation Mobile Systems for the Support of Mobile Multimedia based on ATM Transport", *5th IFIP Workshop*, 21st to 23rd July 1997

[MB96] S. Molnár, S. Blaabjerg, "Correlations in ATM Cell Streams Exposed to Cell Delay Variation", *Journal on Communication*, Vol. 47, Jan.-Febr., 1996, pp. 52-56

[Min96] D. Minoli, et al, "*LAN, ATM, and LAN emulation technologies*," Artech House Inc., 1996.

[MM88] B. N. W. Ma, J. W. Mark, "Performance Analysis of Burst Switching for Integrated Voice/Data Services", *IEEE Trans. Communications*, vol. 36, no. 3, pp.282-297, March 1988

[Mol98] J. Molnár, "*Development of a Simulation Tool for Dynamic Synchronous Transfer Mode Networks*", M.Sc. Thesis, Technical University of Budapest, Department of Telecommunications and Telematics, May 1998

[MPOA] ATM Forum, "*Multiprotocol over ATM Specification v. 1.0*", ftp://ftp.atmforum.com/atm/documents/letter-ballot/af-mpoa-0087.000.pdf

[MS94] D. E. McDysan, D. L. Spohn, "*ATM, Theory and Application*", McGraw-Hill, 1994.

[MS97] M. A. Marshan, F. Saluta, "Fairness and efficiency of slot reuse algorithms in DQDB MANS", *Computer Communications* 20, 1997, pp. 1323-1341

[NetIns] Net Insight Co. home page, http://www.netinsight.se

[NHRP] IETF draft, "*NBMA Next Hop Resolution Protocol (NHRP)*", 02/06/1998, ftp://www.ietf.org/internet-drafts/draft-ietf-rolc-nhrp-14.txt

[Ofe94] Y. Ofek, "Overview of the MetaRing architecture", *Computer Networks and ISDN Systems* 26 ,1994, pp. 817-829

[Ore88] P. O'Reilly, "The case for circuit switching in future wide bandwidth networks", In Proc. of *IEEE ICC*, Philadelphia, USA, pp. 899-904, June 1988.

[PF95] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, pp. 226-244, June 1995.

[PPG92] B. Pehrson, S.Pink and P. Gunningberg, "MultiG- a research program on distributed multimedia applications and gigabit networks", *IEEE Network Magazine*, Vol. 6, No. 1, pp. 26-35, Jan. 1992.

[PRL92a] B. Pehrson, F. Reichert, P. Lindgren, C. Bohm, L. Håkansson, L. Elmstedt, P. Knorring and L. Gauffin. "The Design of the MultiG Network". In Proc. *IFIP TC6 Workshop on Broadband Communications*, Estoril, Portugal, Jan. 1992

[PRL92b] B. Pehrson, F. Reichert, P. Lindgren, C. Bohm, L. Håkansson, L. Elmstedt and L. Gauffin. "The Swedish Optical Fiber Network". *In Proc. OFC '92*, San Jose, USA, pp. 109-110, Feb. 1992.

[Pry91] M. de Prycker, "*Asynchronous Transfer Mode, Solution for Broadband ISDN*", Ellis Horwood, 1991, Chichester

[PS93] S. Pink, P. Sjödin, "The DTM Multicast Channel Protocol", Proc. of the *6th MultiG Workshop*, Stockholm, Sweden, 1993

[Rah93] M. Rahnema, "Overview of the GSM System and Protocol Architecture", *IEEE Communications Magazine*, Vol. 31, No. 4, pp. 92-100, Apr. 1993

[Ram96a] L. Ramfelt, “*Architectures for High Performance Optical Networks*”, Ph.D. Thesis, Royal Institute of Technology, ISRN KTH/IT/R-95/22-SE, Stockholm, Sweden, Febr. 1996

[Ram96b] L. Ramfelt, “Performance Analysis of Slot Management in DTM Networks”, *Draft paper*, 1996

[Res96] IETP, “Resource Reservation Protocol - Version1 Functional Specification”, *Internet Draft*, Nov. 1996, ftp://ftp.ietf.org/internet-drafts/draft-ietf-rsvp-spec-14.txt

[rfc1483] RFC 1483, "*Multiprotocol Encaptulation over ATM Adaptation Layer 5*," July 1993, ftp://ftp.isi.edu/in-notes/rfc1483.txt

[rfc1577] RFC 1577, "*Classical IP and ARP over ATM*," 1/20/94, ftp://ftp.isi.edu/in-notes/rfc1577.txt

[rfc1987] RFC 1987, *’Ipsilon's General Switch Management Protocol Specification Version 1.1’*, August 1996. ftp://ftp.isi.edu/in-notes/rfc1987.txt

[rfc2068] RFC 2068, "*Hipertext Transfer Protocol – HTTP/1.1*" Jan. 1997, ftp://ftp.isi.edu/in-notes/rfc2068.txt

[rfc2098] RFC 2098, "*Toshiba's Router Architecture Extensions for ATM: Overview*", 02/04/1997, ftp://ftp.isi.edu/in-notes/rfc2098.txt

[rfc2105] RFC 2105, "*Cisco Sytems' tag switching architecture overview*," IETF RFC 2105, Feb. 1997. ftp://ftp.isi.edu/in-notes/rfc2105.txt

[RT89] I. Rubin, Z-H. Tsai, “Message Delay Analysis of Multiclass Priority TDMA, FDMA and Discrete-Time Queuing Systems”, *IEEE Transactions on Information Theory*, Vol. 35, No. 3, May 1989, pp. 637-647

[RW96] I. Rubin, H.T. Wu; “Performance Analysis and Design of CQBT Algorithm for a Ring Network with Spatial Reuse”, *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, august 1996, pp. 649-659

[RZ88] I. Rubin, Z. Zhang, “Message Delay Analysis for TDMA Schemes Using Contiguous-slot Assignments”, Proc. *ICC’88*, Philadelphia, June 1988, pp. 418-422 and in *IEEE Transactions on Communications*, Vol 40 4, pp. 730-737

[SB91] B. Steyaert, H. Bruneel, “A General Analysis of the Packet Delay in TDMA Channels with Contiguous-slot Assignments”, Proc. *ICC ’91*, Denver, June 1991, pp. 1539-1543

[SB92] B. Steyaert, H. Bruneel, “Exact Message Delay Derivation for TDMA Schemes with Multiple Contiguous Output and General Independent Arrivals”, *AEU (International Journal of Electronics and Communications)*, 1993, vol. 47, No. 2, pp. 77-84

[Sha81] J. G. Shanthikumar, “On the Buffer Behaviour with Poisson Arrivals, Priority Service, and Random Server Interruptions”, *IEEE Trans. Comput.*, C30, 19981, pp.781-786

[SM98] Andrew Schmidt, Daniel Minoli, "*Multiprotocol over ATM Building State of the Art ATM Intranets Utilizing RSVP, NHRP, LANE, Flow Switching, and WWW Technology*," Prentice Hall, 1998.

[SW81] D.R. Smith, W. Whitt, "Resource Sharing for Efficiecy in Traffic Systems", *Bell System Tech. J.*, 60 (1981) , pp..39-55

[Tan89] A. Tanenbaum, *"Computer Networks"*, Prentice-Hall International Ltd., UK

[Tow80] D. Towsley, "The Analysis of a Statistical Multiplexer with Nonindependent Arrivals and Errors" *IEEE Trans. Communications*, 1980, COM-28, pp. 65-72

[Vic97] N. Vicari, "Measurement and Modeling of WWW-Sessions", *Technical Document* COST257TD(97)52, 1997

[whp99a] *Dynarc white paper*, "Distributed Switching and Routing", 1999, http://www.dynarc.se/pdf/wp_ip_over_dtm.pdf

[whp99b] *Dynarc white paper*, "Business class VPNs", 1999, http://www.dynarc.se/pdf/wp_vpn.pdf

[Wou96] IETF draft, "*ARIS: Aggregate Route-Based IP Switching*", 03/26/1997, ftp://www.ietf.org/internet-drafts/draft-viswanathan-aris-overview-00.txt

[WR97] H.T. Wu, I. Rubin; "Performance analysis and design of an access algorithm for an ATM ring network with spatial reuse", Computer Communications 20 (1997), pp. 848-862

Publication of New Results

Journals and Book Sections

[J1] Cs. Antal; J. Bíró „Analysis of a Time Division Multiplexing Method with Priorities", 10-page extended abstract, IFIP WG7.3 *Performance '99*, 15-17 October 1999, Istanbul, Turkey, proceedings will appear in book format at CRC Press

[J2] Cs. Antal, J. Molnár; S. Molnár, G. Szabó "Performance Study of Distributed Channel Reallocation Techniques for a Fast Circuit Switched Network", *Computer Communications* Vol. 21 Number 17, November 1998, Special Issue on the Stochastic Analysis and Optimal Management of Communication Systems, pp. 1597-1609

[J3] Cs. Antal, K. Szarkowicz, "Introduction to Asynchronous Transfer Mode", *Telecommunications Magazine* (Hungarian), July 1996, Budapest

[J4] Cs. Antal, I. Cselényi, "Distributed multimedia applications: telework and teleuniversity" *Telecommunications Magazine* (Hungarian), Nov. 1995, Budapest,

Conferences and Workshops

[C1] Cs. Antal, S. Molnár: Smoothing Algorithms for DTM Networks, submitted to *Networking 2000 conference*, May 14-19, 2000, Paris, France

[C2] Cs. Antal, S. Molnár, " Fairness of a DTM Dual-bus with Large Inter-node Distances", submitted to *8th International Conference on Telecommunication Systems, Modeling and Analysis*; March 9-12, 2000, Nashville, Tennessee, USA

[C3] Cs. Antal, J. Molnár, S. Molnár, "Fairness Issues in a New Dual-Bus Networking Technology", *7th IFIP/ICCC Conference on Information Networks*, Aveiro, Portugal, 1998 June

[C4] Cs. Antal, "Servicing Bursty Sources Efficiently via a Fast Circuit Switched System", *International Conference for Computer Communication '97*, 19- 21 November 1997, Cannes, France

[C5] Cs. Antal, M. Boda, G. Gordos, T. Henk, G. Szabó, "Comparison of High Speed Networking Technologies", tutorial; *7th Joint European Networking Conference*, May 1996, Budapest, Hungary

[C6] Cs. Antal, T. Henk, G. Szabó, "High Speed Networking", *Seminar on Telecommunication Systems and Services*, organized by HKTCC and Dept. for Microwave Tel. TUB; April 1996; Balatonfüred, Hungary

[C7] Cs. Antal, J. Molnár, S. Molnár, "Fairness Issues in DTM Networks", *High Speed Networks'98, International Workshop*, Balatonfüred, Hungary, 1998 May

Inventions

[P1] Cs. Antal, S. Molnár, L. Györfi, G. Szabó, ”Resource Reallocation in a Communications Network”, *patent application*, May 1997

[P2] Cs. Antal, J. Molnár, "Fair Channel Allocation Protocol for DTM Networks", *patent application*, March 1998

Appendices

Appendix A

This appendix derives the generating functions $U_{A,j,k}(z)$ and $U_{B,j,k}(z)$.

The relation between successive time-units of blocking intervals is easy to express because during blocking intervals the output channel is permanently blocked.

$$U_{B,j,k} = U_{B,j,k-1} + A_k \quad \text{if } j \geq k \tag{A.1}$$

A_k is independent of $U_{B,k}$, and it is an i.i.d. random variable. Therefore, an expression can be obtained between the time-unit just before a B-time and the one k units later. After z-transformation it is:

$$U_{B,j,k}(z) = U_{B,j,0}(z) \cdot A(z)^k \quad \text{if } j \geq k \tag{A.2}$$

During availability times the output channel is permanently available, thus the relation between system content in the time-unit just before an A-time and the one in the A-time is

$$U_{A,1,1} = (U_{A,1,0} - 1)^+ + A \tag{A.3}$$

After z-transformation the following relation is obtained:

$$U_{A,1,1}(z) = \frac{A(z)}{z}\left(U_{A,1,0}(z) + (z-1) \cdot U_{A,1,0}(0)\right) \tag{A.4}$$

Now we can use that the time-unit just before a B-time is the same as the last time-unit of an A-time, and vice versa.

$$P(U_{A,1,0} = u) = \sum_{j=0}^{\infty} P(U_{B,j,j} = u) \cdot P(T_B = j) \tag{A.5}$$

$$P(U_{B,j,0} = u) = P(U_{A,1,1} = u) \tag{A.6}$$

As we can see from (A.6) $P(U_{B,j,0} = u)$ is independent of j, and also the length of the A-time is always 1. To further simplify our notations the following variables are defined:

$U_{A,1,0} \equiv U_{A,0}$; $U_{A,1,0}(z) \equiv U_{A,0}(z)$; $U_{B,j,0} \equiv U_{B,0}$; $U_{B,j,0}(z) \equiv U_{B,0}(z)$

Using (A.2) and (A.5) we obtain

$$U_{A,0}(z) = U_{B,0}(z) \cdot P_B(A(z)) \tag{A.7}$$

and using (A.4) and (A.6) the resulting equation is

$$U_{B,0}(z) = \frac{A(z)}{z}\left(U_{A,0}(z) + (z-1) \cdot U_{A,0}(0)\right) \tag{A.8}$$

$U_{A,0}(z)$ and $U_{B,0}(z)$ can be expressed solving the above equations. The result is

$$U_{A,0}(z) = \frac{(z-1)U_{A,0}(0)\,A(z)\,P_B(A(z))}{z - P_B(A(z))\,A(z)} \tag{A.9}$$

$$U_{B,0}(z) = \frac{(z-1)U_{A,0}(0)\,A(z)}{z - P_B(A(z))\,A(z)} \tag{A.10}$$

From $U_{B,0}(1) = 1$ condition the unknown constant $U_{A,0}(0)$ can be expressed:

$$U_{A,0}(0) = 1 - (1 + P_B^{'}(1))\,A^{'}(1) \tag{A.11}$$

Finally, the unknown generating functions $U_{A,j,k}(z)$ and $U_{B,j,k}(z)$ can be expressed:

$$U_{B,j,k}(z) = \frac{(z-1)\cdot U_{A,1,0}(0)\cdot A(z)}{z - P_B(A(z))\,A(z)} \cdot A(z)^k \quad \text{if } j \ge k \tag{A.12}$$

$$U_{A,1,1}(z) = \frac{A(z)(z-1)\cdot U_{A,1,0}(0)}{z - P_B(A(z))\,A(z)} \tag{A.13}$$

□

Appendix B

Lemma1:

If $F(x) = C(x) * \prod_{i=1}^{n} \frac{N_i(x)}{D_i(x)}$ then where $F(1) = 1; C(1) \ne 0; N_i(1) = 0; D_i(1) = 0; N_i^{'}(1) \ne 0; D_i^{'}(1) \ne 0$ then the following values can be obtained:

$$F^{'}(x) = F(x)\cdot\left(\frac{C^{'}(x)}{C(x)} + \sum_{i=1}^{n}\left(\frac{N_i^{'}(x)}{N_i(x)} - \frac{D_i^{'}(x)}{D_i(x)}\right)\right) \tag{B.1}$$

$$F^{'}(1) = \frac{C^{'}(1)}{C(1)} + \frac{1}{2}\sum_{i=1}^{n}\left(\frac{N_i^{''}(1)}{N_i^{'}(1)} - \frac{D_i^{''}(1)}{D_i^{'}(1)}\right) \tag{B.2}$$

$$\begin{aligned} F^{''}(x) = F(x)\cdot&\left(\frac{C^{'}(x)}{C(x)} + \sum_{i=1}^{n}\left(\frac{N_i^{'}(x)}{N_i(x)} - \frac{D_i^{'}(x)}{D_i(x)}\right)\right)^2 + \\ &+ F(x)\cdot\left(\frac{C^{''}(x)\cdot C(x) - C^{'}(x)^2}{C(x)^2} + \sum_{i=1}^{n}\left(\frac{N_i^{''}(x)\cdot N_i(x) - N_i^{'}(x)^2}{N_i(x)^2} - \frac{D_i^{''}(x)\cdot D_i(x) - D_i^{'}(x)^2}{D_i(x)^2}\right)\right) \end{aligned} \tag{B.3}$$

$$\mathrm{var}\{F\} = F''(1) + F'(1) - F'(1)^2 = \frac{C'(1) + C''(1)}{C(1)} - \frac{C'(1)^2}{C(1)^2} +$$
$$+ \frac{1}{2}\sum_{i=1}^{n}\left(\frac{N_i''(1)}{N_i'(1)} - \frac{D_i''(1)}{D_i'(1)}\right) + \frac{1}{3}\sum_{i=1}^{n}\left(\frac{N_i'''(1)}{N_i'(1)} - \frac{D_i'''(1)}{D_i'(1)}\right) + \frac{3}{4}\sum_{i=1}^{n}\left(\left(\frac{N_i''(1)}{N_i'(1)}\right)^2 - \left(\frac{D_i''(1)}{D_i'(1)}\right)^2\right) \quad \text{(B.4)}$$

Proof: Using L'Hopital's rule repeatedly.

Lemma 2:

The mean and the variance of the system content of low priority queues in Section 4.2.5.2 are

$$E\{U_2\} = A_2'(1) - \frac{A_2''(1)}{2A_2'(1)} + \frac{b(1)''}{2b'(1)(1 + b'(1))} \quad \text{(B.5)}$$

$$\mathrm{var}\{U_2\} = \mathrm{var}\{A_2\} - \frac{2A_2'''(1) + 3A_2''(1)}{6A_2'(1)} - \frac{3A_2''(1)^2}{4A_2'(1)^2} + \frac{2b(1)''' + 3b(1)''}{6b'(1)(1 + b'(1))} + \frac{3b''(1)^2}{4\left(b'(1)(1 + b'(1))\right)^2} \quad \text{(B.6)}$$

where $b(z) = 1 - A_2(z) \cdot P_B(A_2(z))$.

Proof:

The system content can be described as $U_2(z) = N \cdot A_2(z)\frac{a(z) \cdot b(z)}{c(z) \cdot d(z)}$ where

$N = \frac{1 - (1 + P_B'(1)) \cdot A_2'(1)}{1 + P_B'(1)}$; $\quad a(z) = 1 - z$; $\quad b(z) = 1 - A_2(z) \cdot P_B(A_2(z))$;

$c(z) = 1 - A_2(z)$; $\quad d(z) = z - A_2(z) \cdot P_B(A_2(z))$

The derivatives of *a(z)* and *c(z)* can be obtained easily. The derivatives of *d(z)* can be expressed with those of *b(z)*:

$$d'(1) = 1 + b'(1); \quad d''(1) = b''(1); \quad d'''(1) = b'''(1) \quad \text{(B.7)}$$

The derivatives of *b(z)* are

$$b'(1) = -A_2'(1)(1 + P_B'(1)) \quad b''(1) = A_2''(1) \cdot (1 + P_B'(1)) + A_2'(1)^2 \cdot (P_B''(1) + 2P_B'(1)) \quad \text{(B.8)}$$

$$b'''(1) = A_2'''(1) \cdot (1 + P_B'(1)) + 3A_2'(1) \cdot A_2''(1) \cdot (P_B''(1) + 2P_B'(1)) + A_2'(1)^3 \cdot (P_B'''(1) + 3P_B''(1)) \quad \text{(B.9)}$$

Applying (B.2), (B.4), (B.7), (B.8) and (B.9) one obtains (B.5) and (B.6).

Lemma 3:

The mean and the variance of the system time of low priority sources in Section 4.2.5.4 are

$$V_2'(1) = 1 + \frac{a''(1)}{2a'(1) \cdot (1 - a'(1))} \quad \text{(B.7)}$$

$$\mathrm{var}\{V_2\} = \frac{a''(1)}{2a'(1)\cdot(1-a'(1))} + \frac{a'''(1)}{3a'(1)\cdot(1-a'(1))} + \frac{3a''(1)\cdot(1-2a'(1))}{4a'(1)^2\cdot(1-a'(1))^2} \qquad \text{(B.8)}$$

Proof:

The system time can be written as $V_2(z) = N\cdot z\cdot\frac{a(z)}{b(z)}$ where

where $N = \frac{1-(1+P_B'(1))\cdot A_2'(1)}{(1+P_B'(1))\cdot A_2'(1)}$; $a(z) = A_2(z\cdot P_B(z))-1$; $b(z) = z - A_2(z\cdot P_B(z))$

The derivatives of *b(z)* can be expressed from those of *a(z)*:

$b'(1) = 1-a'(1)$; $b''(1) = -a''(1)$ and $b'''(1) = -a'''(1)$

The derivatives of *a(z)* are

$a'(1) = A_2'(1)\cdot(1+P_B'(1))$; $a''(1) = A_2''(1)\cdot(1+P_B'(1))^2 + A_2'(1)\cdot(P_B''(1)+2P_B'(1))$;

$a'''(1) = A_2'''(1)\cdot(1+P_B'(1))^3 + 3A_2''(1)\cdot(1+P_B'(1))\cdot(P_B''(1)+2P_B'(1)) + A_2'(1)\cdot(P_B'''(1)+3P_B''(1))$

Using (B.2) and (B.4) the mean and the variance of the system time of low priority sources can be expressed.