

## Basic Concepts and Performance of High-Speed Protocols

T. Stock<sup>1</sup> and P. Tran-Gia

Institute of Computer Science  
University of Würzburg,  
Am Hubland,  
W-8700 Würzburg, Germany  
Tel.: +49 931/888-5510,  
Fax.: +49 931/888-4601,  
E-mail: {stock,trangia}@informatik.uni-wuerzburg.de .

**Keywords:** HSLAN & MAN access protocols, performance evaluation

In the last few years a large number of novel access protocols has been proposed for use in high-speed local and metropolitan area networks. On the one hand those protocols are dedicated for networks operating in gigabit-per-second transmission capacity local networks. On the other hand these protocols are also developed as access protocol to interface users of new high-performance services to broadband ISDN networks. Although normally, the protocol descriptions and operating modes are rather complex and spacious, a number of basic mechanisms can be identified as building blocks to design high-speed protocols. Examples of such mechanisms are 'distributed queueing', 'credit assignment', 'reservation procedures' etc. The main purpose of the paper is to give an overview on these protocol mechanisms and their influence on overall high-speed protocol performance. Further, modelling issues for this class of protocols will be addressed and examples for protocol models and their evaluation methods will be discussed.

### 1. Introduction

In the current standardization process of high-speed local and wide area networks a large number of protocol proposals have been presented in the literature. The diversity of these protocols can be observed from viewpoints of system structure, control mechanism and signalling schemes. On the one hand, various medium access protocols can be found as candidate for high-speed local and metropolitan area networks (HS-LANs and MANs).

---

<sup>1</sup>T. Stock is now with Alcatel STR, Zürich, Switzerland

On the other hand, to provide customer access to high-capacity wide area networks like ATM systems (asynchronous transfer mode), generic flow control (GFC) protocols have been proposed in the emerging standardization. Thus, a systematic treatment of these proposals, considering both architecture and performance aspects, is quite complex.

From performance analysis viewpoint various problems arise with this new class of high-speed protocols. Due to the large medium length and the high transmission capacity, packets appear shorter on the medium. An interesting measure to describe this phenomenon is the logical medium length. With  $C$  be the transmission capacity,  $l$  the medium length and  $c'$  the signal propagation speed, the logical medium length  $\Lambda$  is defined as

$$\Lambda = \frac{l}{c'} \cdot C \quad (1)$$

where  $l/c'$  is the latency of the medium. The measure  $\Lambda$  represents the number of bits (or rescaled in packets, cells, slots) which can be placed on a row on the medium. In a classical LAN environment with  $l = 2.5$  km,  $C = 10$  Mbps and  $c' = 0.67$  c we yield  $\Lambda$  in the range of a single packet's length. Here classical queueing models and analysis methods can still reasonably be applied [18]. In a high-speed system, e.g. with  $l = 100$  km,  $C = 100$  Mbps and  $c' = 0.67$  c the logical medium length increases by roughly three orders of magnitude. Thus  $\Lambda$  is significantly larger than the typical packet length.

Each attached station would observe another state process, since no time synchronization exists in the system. Therefore new decomposition methods must be developed to cope with this phenomenon, in order to obtain accurate analysis techniques. Due to new requirements in the performance evaluation process, such as delay variation and transfer jitter, analytical methods gain importance, since computer simulations will touch their limits sooner than in conventional systems.

In this paper we approach to present high-speed protocols in a more generic context. We first try to identify a list of mechanisms used as building blocks in the protocol design process. Subsequently we will take a few known protocols as a synthesis of these basic mechanisms and discuss them in more detail. To illustrate performance analyses of such high-speed protocols we will take as examples some DQDB-based protocols (distributed queue dual bus, for an overview see [14], for the standard see [7]) and the CRMA (cyclic reservation multiple access, see [15, 16]) mechanism and outline their performance models.

## 2. Basic Protocol Mechanisms and their Properties

In this section we will describe the basic protocol mechanisms, some of their basic performance properties and some modifications to overcome the characteristic disadvantages. These descriptions will be rather brief so we refer to the literature for a more detailed description.

## 2.1 Basic Architecture

We will consider ‘linear’ topologies in the sense that the transmission media leave the medium access unit into two directions. Two-dimensional topologies like Manhattan street networks are not covered.

The following topologies are mentioned quite often (see Fig. 1):

- ◇ **single directed ring**; perhaps the most basic variant. all nodes are connected to their neighbours via two unidirectional links, an incoming and an outgoing one. Should be one of the cheapest solutions since only one receiver and transmitter are needed.
- ◇ **dual ring**; two of the above ones melted together. The main characteristic is the fact that the second ring points into the opposite direction (therefore: dual, not double).
- ◇ **multiple rings/thorus**; several directed rings operating in parallel and all pointing into the same direction.<sup>2</sup>
- ◇ **folded bus**; since a single unidirectional bus segment would prevent some nodes from sending data to some other nodes, this topology may be considered as the basic bus structure for directed media. It consists of an outbound channel, which leads from the first to the last station and an inbound channel in the reverse direction. There are two variants: the first only provides a single transmitter on the outbound and single receiver on the inbound channel. The second adds a receiver on the outbound channel right in front of the transmitter thus allowing a node to sense the medium ahead of the transmitter.
- ◇ **doubly folded bus/double spiral**; similar to the folded bus including the two variants described above with the exception that outbound and inbound channel are connected through a link segment (see [22] for a description of the Expressnet topology). Therefore outbound and inbound channel point into the same direction so that the data passes the stations on both channels in the same order. The double spiral is topologically equivalent but omits the link segment.
- ◇ **dual bus**; similar to the dual ring configuration where one segment was cut off. Note that in contrast to the dual ring here there is no free choice on which bus segment to transmit data to a specific node.

Architectures which are closed, like all the ring topologies have to be implemented in an active way. This means that the overall topology is based on point-to-point links between the nodes which have to forward data explicitly. In contrast open architectures like all the other topologies mentioned above can be realized passively as well since they provide

---

<sup>2</sup>Similar parallel structures can be implemented using other topologies, too. But since they are not used very often we will not mention them in the following.

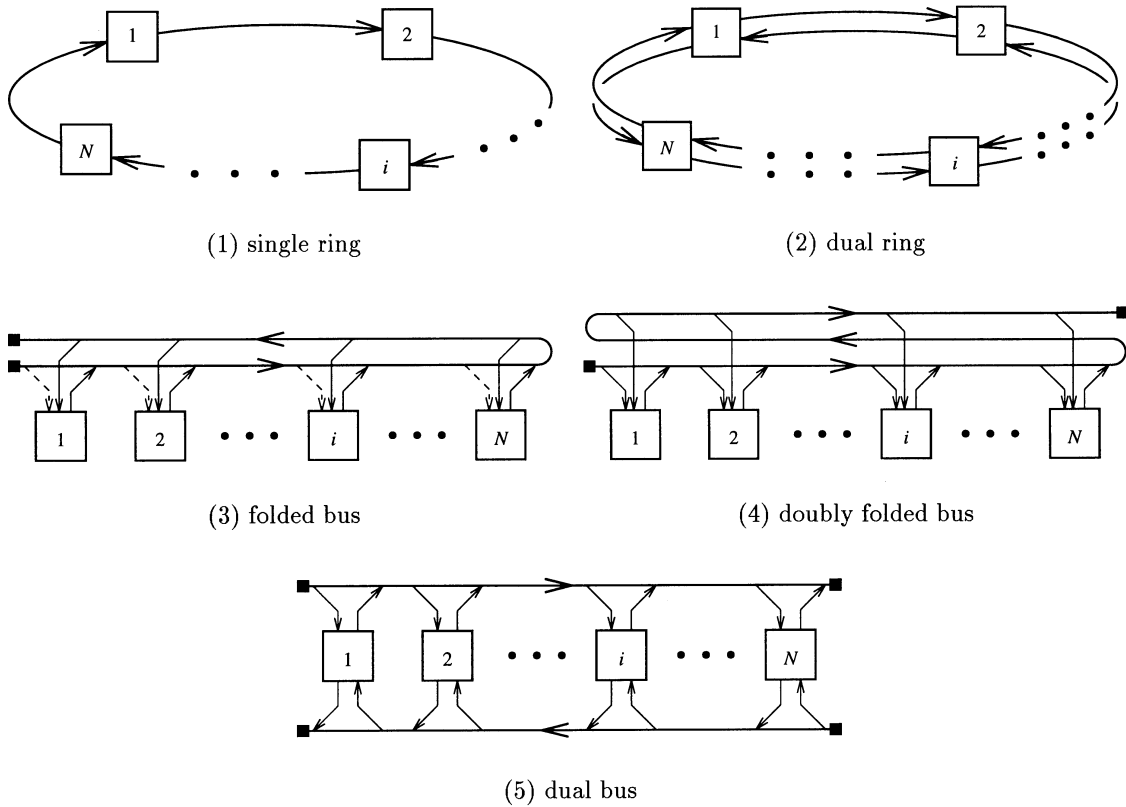


Figure 1: *Basic medium topologies*

an explicit termination at the end of the media. Of course this would require an access protocol that supports such an implementation, i.e. which is based on transmitters that are only capable of OR-writing.

## 2.2 Access Control

Access protocols will be described on the basis of a — single or dual — ring topology where possible. This is due to the cyclic characteristics of most of the protocols and makes comparing of the concepts easier.

### 2.2.1 Slot Format and Signalling Issues

The slots consist of two areas (see Fig. 2). The first one is a header, in the following called access control field (ACF), which forms a signalling channel. The second one is a field which contains addressing information and a payload area.

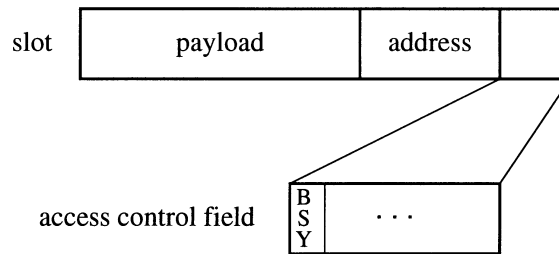


Figure 2: *Slot format*

### 2.2.2 Token Passing

Mainly there are two variants:

- ◊ the first is closely related to the classical token-ring protocol: A single symbol called token is rotating around the ring. A station that wants to transmit some data can take the token off the ring and send a certain amount of data segments before it has to put the token onto the ring again, so that the next station may gain access to the media. Several types of limits may be considered but the most simple one should be a fixed upper bound for the transmission window of each station.
- ◊ the second supports multiple tokens: Every empty slot is associated with such a token. This means that a station with some data pending for transmission can utilize the next free slot. This slot travels around the ring and will be cleared again by the sender who has to pass it unused to the next node.

The main disadvantages of these access mechanisms are their disability to support destination release properly thus providing slot reuse as well as the fact that per transmission cycle capacity equivalent to one ring latency is lost due to the rotating token(s). On the other hand they are fairly simple and robust against failures.

Examples of slotted token-based protocols are the Cambridge (fast) ring (cf. [4, 5]) and Fasnnet (cf. [10]).

### 2.2.3 Reservation

The class of reservation based protocols are strongly cyclic, too. Each cycle, which is governed by a master station called *scheduler*, is subdivided into a reservation and a transmission phase.

The main principle is as follows. The scheduler issues a reservation command which passes a stations attached to the ring. Each node that is touched by the command adds the number of segments it wants to transmit in the next cycle (reservation phase). After

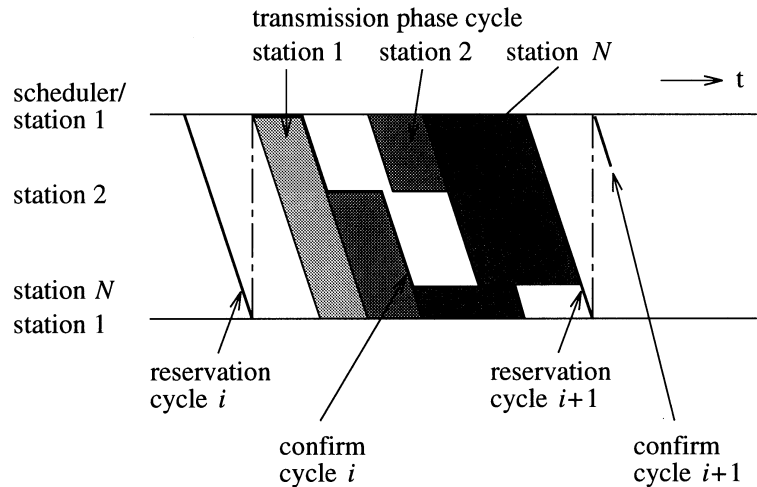


Figure 3: *Space-time diagram of a basic reservation mechanism*

these reservations have travelled around the ring, the scheduler determines how much data each station is allowed to transfer in the corresponding transmission phase. The stations are notified of their quantity by a confirm command also issued by the scheduler. In this simple variant the confirm command may be also be considered as a start signal for the next transmission cycle, in more elaborate version a separate start command may be necessary.

In the simplest case each station would hold the confirmation command similarly to the token mechanism until it has transferred its quantity in this cycle (see Fig. 3).

After starting the confirmation the scheduler waits for the duration of the transmission phase and then issues the next reservation.

This basic access mechanism still has the same disadvantage as the token concept described previously: one latency of transmission capacity is not used for data transfer while the reservation command rotates around the ring. Thus in improved versions the scheduler should issue the reservation command for the next cycle in advance so that in the ideal case the reservations for that cycle are known to the scheduler before the transmission phase of the current one ends.

One solution could be to issue reservations in relatively short intervals to accumulate a stock of reservations so there are some available when the current transmission phase ends. This is done in the CRMA-system (Cyclic Reservation Multiple Access, cf. [15, 16] and subsection 3.2). The problem with this modification is the fact that because of the potentially large number of reservations on the fly there may be a large accumulation of reservations in the scheduler. CRMA solves this by issuing a cancellation command when a certain level is reached. The cancellation command also passes each station and notifies them that all reservations not already confirmed are invalid and thus have to be

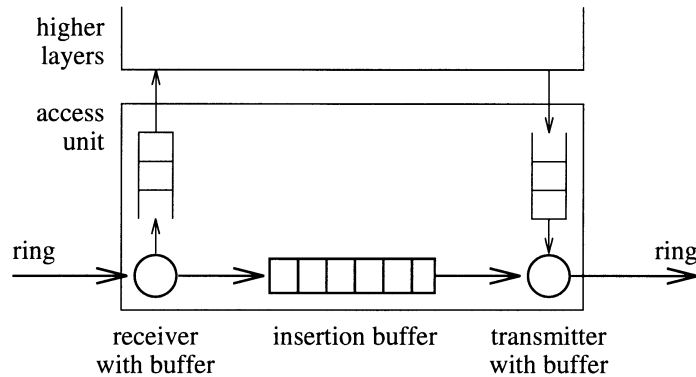


Figure 4: *Station with insertion buffer*

issued again. This mechanism is called backpressure.

Another solution as performed in the Cyclic Ring protocol (cf. [17]) where the reservation for the next cycle is issued one latency before the current transmission phase ends. In case the transmission cycle is shorter than one latency, the reservation command is issued immediately after the last confirmation command. Doing this way there is no unused capacity between successive cycles as long as the transmission phases are longer than one latency.

As described up to now all variants support transmission of longer packets in successive slots. This is not the case anymore when slot reuse is introduced. To implement this in an efficient way an end command has to be introduced. It is sent by the scheduler immediately after the confirm command and forwarded by the individual stations when they have transferred their amount of data. A further improvement can be achieved by embedding a so called *left count* into the slot stream which allows the scheduler to estimate the number of pending transmissions in the current cycle so that the next reservation can be issued properly.

#### 2.2.4 Buffer Insertion

Another protocol concept which primarily targets to a ring structure is buffer insertion (see [6] for SILK, an early example of a buffer insertion system). These systems implement a fixed sized buffer between receiver and transmitter of a station (see Fig 4). Incoming non-empty slots are accepted by the receiver and checked whether their target address is the station's one. In this case they are taken from the system and stored in the local reception buffer. This class of protocols therefore is based of destination release although source release would be possible, too. Otherwise they are appended to the data in the insertion buffer or transmitted immediately if it is empty, respectively. When there is nothing to send or forward empty slots are transmitted.

When a station wants to send some data it has to be distinguished whether the traffic on the ring or the local traffic are of higher priority. In the first case the station has to delay its transmission until the insertion buffer is empty. Once it has started transmitting a packet consisting of several slots it can not be interrupted. Therefore the insertion buffer must be big enough to hold the number of slots equivalent to one maximum sized packet. In the second case a station may transmit slots as long as it can be sure that there is no loss of data at the insertion buffer. There is no difference if the buffer is not larger than mentioned above but the station can send more at the same time if the insertion buffer is correspondingly larger.

The problem with both principles is that stations may starve as described in the following. Assume three subsequent stations, the first continuously sending data to the third and the second having a non-empty insertion buffer of minimal size. In this case the intermediate station never will be able to empty the insertion buffer and thus to transmit its data.

Therefore some additional signalling is required. Metaring (cf. [2]) accomplishes this by providing so-called SAT-signals (*SATisfied*) on a second counterrotating ring thus inducing some cyclic behaviour, again. These signals can be held by a particular station as long as it did not already transfer its quantity for the current cycle.

### 2.2.5 Credit Systems

Credit systems are rather closely related to reservation systems. The main difference between them is that credit systems try to avoid the delays caused by the initial reservation phase.

A cycle is again subdivided into two phases: the crediting and the transmission phase. At the start of a cycle each station gains a fixed amount of credits  $C_i$  which is not necessarily equal for all stations. It is now allowed to transmit up to  $C_i$  segments in the current cycle, each transmission decrementing the number of credits by one. Especially under light to moderate traffic conditions this enables a station to transmit its data without extensive delays, provided that there are enough credits available. At the end of the cycle remaining credits are canceled and the stations starts again with  $C_i$ .

The question is how to determine the end of a cycle, i.e. the time instant where a station's credits are consumed or there is no data to send. A centralized solution could work as mentioned above by embedding a counter variable into the slot stream. Each station would add the number of outstanding transfers for the current cycle to the counter and thus allow the scheduler to detect the end of the current cycle. The stations are notified of the end of the current and the start of the next cycle by a special *reset* signal which is sent by the scheduler.

Another fully distributed mechanism was introduced with the Orwell system (cf. [3, 1]). When a station cannot reuse a slot just released it marks the slot as a *trial* slot and enters its own address as target address. This slot is still accessible to other stations. But in case it returns unaltered to the station which marked it as trial this station detected the end of the cycle and thus issues a reset signal.



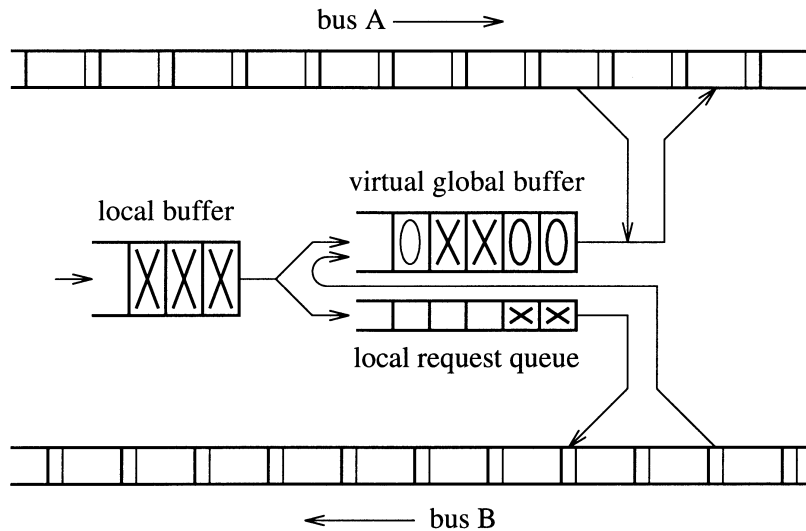


Figure 5: Data and signalling flow in a distributed queueing system

In the worst case there is still a loss of capacity equivalent to one latency. The probability for this situation to occur can significantly be reduced by coupling the trial signal to the source instead of the target address so that it can also be connected to busy slots.

### 2.2.6 Distributed Queueing

Distributed queueing protocols cannot be implemented on a single directed ring since they require a signalling channel pointing into the opposite direction of the payload channel. Most of them were introduced for a dual bus structure (see Fig. 2.1) but some can also be used on a dual ring. For an overview of distributed queueing see [14].

Considering data transfer on the upper and signalling on the lower bus the access procedure works as follows. Seen from a station, the main principles of distributed queueing are

- ◇ broadcast a request to all upstream stations for every slot required
- ◇ keep track of all access requests issued by downstream stations
- ◇ do not access the appropriate bus before all requests of downstream stations prior to its own are satisfied.

The data and signalling flow is depicted in Fig. 5. An incoming packet is segmented and stored in a local buffer. Later a certain number of segments are moved to the station's virtual global buffer. At the same time an equivalent number of requests has been generated and stored in the local request queue. In the virtual global queue the segments find

some requests of downstream stations in front of them which must be satisfied by leaving free passing slots unused on the upper bus. Downstream requests are again stored in the virtual queue behind the local segments.

Independently to this procedure the station transfers the requests in the request buffer by utilizing the access control field of the slots on the lower bus. The ACF can either provide single request bits thus allowing one request per slot or a larger request field allowing for the transfer of a bunch of requests in a single ACF.

Using this basic concept several variants can be distinguished. The simplest one forms the basis of the DQDB protocol where only one local segment may be queued in the virtual buffer of a station. Correspondingly only single requests can be issued on the signalling bus where they only provide single request bits. Besides the problem of the potentially large logical lengths of MAN systems, this fact further increases unfairness aspects of DQDB since it does not allow the upstream stations to build a up-to-date record of the downstream queueing situation.

It is possible to improve the situation by allowing to transfer full packets at once from the local to the virtual global buffer (cf. [13] presenting the DQDB/CB protocol). The success of this modification is questionable as will be shown in section 3.1. The main problem is that requests are still carried by single request bits in the access control field of the signalling bus.

This is overcome in the generalized distributed queueing concept which was introduced in DQMA (cf. [11]). Here the ACF provides an 8-bit request field allowing for up to 255 requests in one slot. Thus the upstream stations have a much more accurate picture of the downstream situation and therefore react faster on sudden changes in load.

Standard DQDB includes an additional mechanism. This is the so-called bandwidth balancing mechanism where each station has to leave every  $M$ -th — the bandwidth balancing modulus — passing free slot unused, i.e. may not remove the head of the queue segment in the virtual global buffer.

We want to note that systems which are based on the distributed queueing principle can implement destination-release and thus enable slot reuse. But since this can be done in different ways, which are not as straightforward as in the protocols described in the previous sections, we will not go into detail. Nevertheless it would allow for further classification.

### **2.2.7 Hybrid Protocols**

All previously described protocol concepts follow the same principle: they decide for one single basic mechanism and then try to overcome its drawbacks by introducing additional modifications.

Other proposals try to integrate more than one basic concept from the beginning so that a balanced performance is achieved. As an example we will briefly touch CRMA-II (Cyclic

reservation multiple access II, see [25]) which may be implemented on a single directed ring similarly to the reservation systems in 2.2.3.

Mainly CRMA-II is a synthesis of random access and reservation. In the beginning all stations can freely use every empty slot which is released by the destination, again. Embedded into this mechanism there is a reservation process governed by a central scheduler. By utilizing a reservation command which gathers the transmission requests for the next as well as the number of transmissions of all the stations in the current cycle the scheduler can use a confirm command to balance all active stations in the next cycle. The confirm command is dual to the reservation command and grants less requests to stations which were able to transfer more than others in the previous cycle than to those.

The way how this balance is calculated allows for several implementations. But even an algorithm which sums up the requests and cuts at a certain threshold can provide evenly distributed bandwidths among the active stations. As long as too many of them do not send at one time the potential cycle length will not grow excessively (see [21]).

The third mechanism which is optional in CRMA-II is buffer insertion. This may be implemented in cases where it is important to avoid the drawbacks of excessive segmentation. This modification fully integrates into the reservation mechanism mentioned above and thus does not require any additional changes so that stations with and without insertion buffer can operate together on the same system.

### **3. Performance Issues**

Due to the logical length of the media and the large number of on-the-fly unacknowledged data units, which are typical for high-speed systems, the performance investigations of those systems are more complex. In this section we will take two examples to illustrate the problems arised and the complexity of these analyses.

#### **3.1 Distributed Queueing**

In the following we will present some simulation based results for three protocols which are based on the concept of distributed queueing. They are DQDB with bandwidth balancing (see [7]), DQDB/CB with some slight modifications [9, 13] and generalized distributed queueing as first introduced with DQMA in [11]. All the protocols were implemented on the dual bus topology shown in Fig. 2.1.

For an approximate performance evaluation of the basic DQDB principle based on classical queueing analysis we refer to [19, 20, 23] where models for different arrival patterns as well as several priorities are considered. A more extensive comparison of this protocol class and other concepts is given in [21].

All three variants were based on a dual bus system of 100 km length with 16 equidistantly located stations. Transmission capacity was assumed at 140 Mbps, signal propagation

speed was chosen at 200,000 km/sec and the slot size at 424 bits (432 bits in the case of the DQMA derivate). These parameters result in a logical bus length of about 160 slots. The DQDB bandwidth balancing modulus  $M$  was chosen at 8, which is the default in [7].

The arrivals at the stations were given by Poisson streams. The batch size obeyed a bimodal distribution: 50 % of the packets were of maximum length of 80 slots, the other 50 % of the packets had a geometrically distributed number of slots around a mean value of 10 slots.

The amount 20 % of the total load was generated by station 8 the rest of 80 % arrived at the other stations. All target station indices appeared with equal probability. This means that the overall arrival rate matrix is symmetric. But as mentioned in the previous section the access protocol also is fully symmetric so that both directions of data transfer can be decomposed. This way we could restrict ourselves to downstream traffic on the upper bus and thus yield an upper triangular rate matrix.

The figures will show some of the main characteristics of the protocols discussed. This will be done mainly by observing the time from arrival of a packet until it is completely transferred to the bus. The corresponding interval is called  $T_{14}$ . In some of the figures we will also present some results for the subintervals of  $T_{14}$  that are induced by the time instant when the first segment of a packet enters the virtual global queue. They are denoted by  $T_{12}$  and  $T_{24}$ , respectively.<sup>3</sup> The delays are given in slots.

Fig. 6 compares the virtual service time a packet experiences when it is transferred at one of the stations 1 to 15. In general this measure increases along the bus since the load along the bus also increases. Thus downstream stations do not see as many free slots as the upstream ones and therefore it takes longer to transfer a single packet. It can be noticed that among all the stations the highly loaded one in the middle of the system shows some exceptional behaviour. The virtual service time of a packet at station 8 is significantly shorter than those at the neighbouring stations. This is due to the fact that station 8 issues more requests and therefore experiences a much smoother pattern of free and accessible slots on the data bus.

As expected the curve for the DQMA-derivate is the flattest since it allows the request for a whole packet to be transferred in a single request field. DQDB/CB shows the largest gradient resulting in a virtual service time which at the station 15 is four times as large as at the first.

Unfortunately the modifications in DQDB/CB (see previous section) — compared to DQDB with bandwidth balancing enabled — further privilege the upstream stations since a packet is transferred to the virtual global buffer earlier. So the unfair characteristics already present in DQDB with bandwidth balancing are further increased.

DQDB with bandwidth balancing more or less behaves as expected. It shows rather constant values up to station 8 significantly above those of DQMA and then grows between the curves for DQDB/CB and DQMA.

---

<sup>3</sup>The naming conventions of the intervals are due to the approximate analysis which was mentioned above and where one more time instant  $T_3$  played an important role.

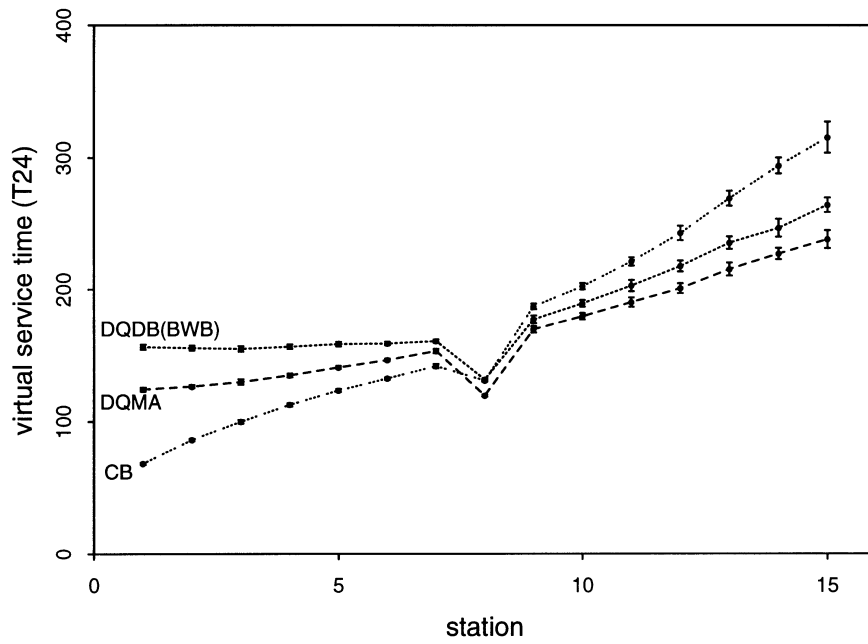


Figure 6: *Virtual service time*

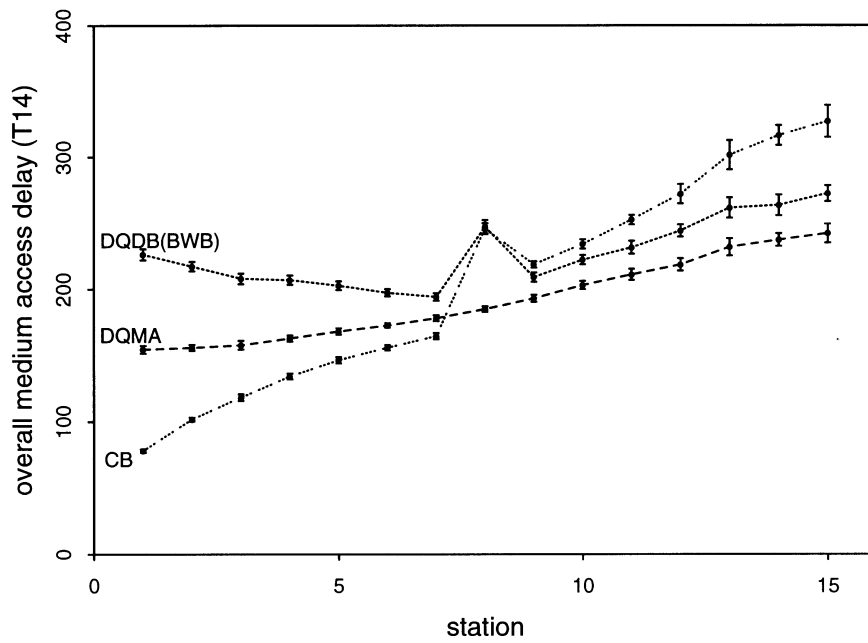


Figure 7: *Overall medium access delay*

Concerning the overall medium access delay  $T_{14}$  (see Fig. 7) it can be noticed that DQDB/CB and DQDB with bandwidth balancing have some problems integrating the highly loaded station 8. Again the gradient of the delay of DQDB/CB is excessively large. In contrast DQMA does not show any peaks at all and provides a very flat curve increasing from 160 slots at station 1 to 230 slots at the last active station thus providing the best-balanced behaviour among the distributed queueing protocols considered.

## 3.2 Cyclic Reservation Multiple Access Protocol

In this subsection we will take the Cyclic-Reservation Multiple-Access (CRMA) protocol as example to illustrate the complexity of a high-speed protocol model with multi-level of information units. The CRMA-protocol has been proposed recently as access scheme for high-speed LANs and MANs, especially in the network capacity region beyond 1 Gbit/sec. Detail description of an early version of this class of protocol, whose prototype has been shown at Telecom'91 (Geneva), can be found in [15] and [16].

We will outline the model and performance analysis as presented in [24]. The model contains two submodels operating on different levels of information units: one deals with packets, the other with slots or cells. To consider the efficiency of overload control mechanisms like the backpressure scheme in CRMA, it is necessary to know the entire state distributions as well as delay and transfer time distribution functions, rather than only the mean values and some higher moments. Thus, in [24] an approximate computational method was derived to obtain distribution functions of performance measures of interest.

The proposed CRMA protocol can basically be used in unidirectional folded-bus or dual-bus topologies. We will summarize only those CRMA properties and features which are relevant in the system modelling context. Since the basic operations are quite analogous for both unidirectional folded-bus and dual-bus structures, we restrict the following description on the folded-bus configuration, as depicted in Fig. 2.1. The headend, which has the function of the scheduler as discussed in the previous section, is a part of node 1 and controls the basic access mechanism (cyclic reservation) and the reservation cancelation (backpressure treatment). Each station is connected to the outbound bus segment, where packets are transmitted, and the inbound bus segment, where they are received. The system operates slot-wise; each slot consists of an access control field (ACF) and a segment field (SF) for payload data. The ACF consists of a busy/free-bit (B/F) to indicate if SF is empty and an access command with its arguments. Examples for the access command format can be found in [15, 16].

There exists a set of basic commands used to control bus access: the *reserve* and *start* commands for the basic access mechanism and the *confirm* and *reject* commands dedicated for the backpressure mechanism. Following the basic access mechanism the scheduler sends periodically *reserve* commands, each has a cycle number and a cycle length as arguments. The interreserve interval  $Q$  is set to a constant value ( $Q = \Theta$  slots). To reserve slots a node increases the cycle length by the number of required slots on the *reserve* command (R) passing by. For each reservation, a station is allowed to reserve slots for

only one packet. A packet consists of a number of  $X$  slots and can have a maximal size of  $X_{MAX}$  slots. Additionally, the station stores this number as the reserve length in accordance to the cycle number in a local reservation queue. When the *reserve* returns to the scheduler, it is ranged in a global reservation queue. This queue is maintained by the scheduler and operates in a FIFO order. To process a reserve cycle the scheduler sends a *start* command (S) consisting of the cycle number followed by as many empty slots as indicated by the cycle length. A node receiving a *start* will search in its local reservation queue for a entry with the current cycle number. The station will use as many empty slots as indicated by the reserve length.

To protect the scheduler queue from being overloaded a reservation-cancellation backpressure mechanism is introduced, aiming a minimization of the worst-case access delay by varying the length of the interreserve interval according to the traffic demands. It prolongs the interreserve intervals if a scheduler overload condition is detected. As overload indicator, the state  $U$  of the scheduler queue is used, i.e. the number of reserved slots waiting in the scheduler queue to be issued. For that purpose the scheduler checks  $U$ , after processing a returning *reserve*. If  $U$  exceeds an overload threshold  $L$ , the backpressure scheme becomes active. In this case no *reserve* is issued until the number of reserved slots is again equal to  $L$ . During this phase, all *reserve* commands already issued but still on the bus are cancelled.

The reservation-cancellation backpressure mechanism uses two additional access commands: the *confirm* command having the cycle number as argument and the *reject* command without any argument. Under normal load conditions, the scheduler issues a *confirm* to confirm the reservation after receiving a returning *reserve*, using the same cycle number as argument. If after the reception of a *reserve* the number of waiting slots in the scheduler queue exceeds the threshold  $L$ , the scheduler will work in the reservation-cancellation mode. The scheduler issues now a *reject*. Each station receiving a *reject* has to remove all unconfirmed reservations from its local reservation queue. The scheduler will return to the basic access mode when the scheduler queue size drops below the threshold  $L$ . Obviously, the first regular *reserve* will return at the scheduler when the scheduler queue becomes empty if the threshold is chosen as the bus latency  $\tau$ . Therefore  $\tau$  is a natural choice of the threshold  $L$  as discussed in [15].

The major purpose of the performance modelling as presented in [24] is to compute the distribution of the medium access delay and the transfer time of data packets. This is required to investigate fairness issues of the protocol and extreme values of the access delay and its jitter. Thus the performance evaluation has been done analytically using discrete-time methods, since simulations for the range of probability distribution functions required here might be intractable or lead to excessive computation times. Since the system operates slot-wise, analytical methods in time-discrete domain have been chosen.

The performance model was based on an observation of a test packet to be transmitted from a sending station  $i$  to a receiving station  $j$ . The breakdown of delays and transfer time into subintervals is illustrated in Fig. 8, where the following time instants are marked: (1) packet generation; (2) reservation instant; (3) reservation arrival at scheduler; (4) issuing time of cycle at scheduler; (5) issuing time of test packet at scheduler; (6) access time at

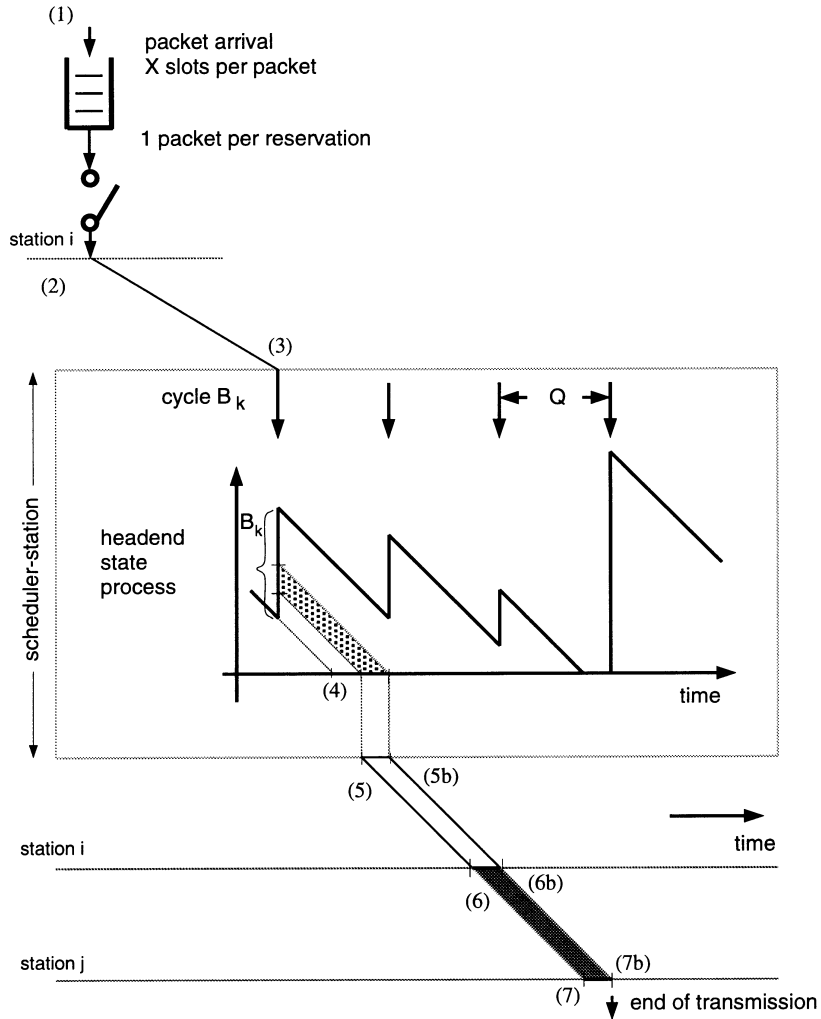


Figure 8: *Components of the medium access delay*

sender; (7) receiving time.

In denoting the time duration between two marked time epochs (x) and (y) with the random variable (r.v.)  $T_{xy}$ ,  $T_{12}$  is the prereservation delay and  $T_{34}$  the waiting time at the scheduler.  $T_{23}$  and  $T_{56}$  represent the propagation delays and form together the turnaround time  $\tau$  along the bus system. Clearly, the duration  $\tau$  depends on the bus configuration and the CRMA operation mode under consideration. For the folded bus configuration,  $\tau$  is just the bus latency, i.e. the propagation delay along the entire length of the bus. The medium access delay is  $T_{ACC} = T_{16}$ . Its part excluding the bus latency or propagation time is just  $T_{ACC}^* = T_{ACC} - \tau$  is more appropriate for comparisons of CRMA performance with different parameters, where the influence of the bus operation mode should be intentionally ignored. From a modelling viewpoint two model levels are considered to investigate the two major delay measures: the prereservation delay  $T_{12}$  and the scheduler delay  $T_{34}$ . The model for prereservation delay still takes packets as information units while in the



scheduler model the state process has to be described on slot level. To cope with this distinction a decomposition approach in conjunction with an independence approximation has been used (cf. [24]). Two submodels arose and have been analysed; a G/G/1 queue with state-dependent feedback and a M/G/1 queue with server vacation.

In the following we will discuss some numerical results out of the study in [24]. Two different types of attached stations are considered: i) stations under normal load conditions (random data traffic) and ii) saturated stations (e.g. stations during a file transfer phase, which can be in a quasi-stationary state). Under normal traffic conditions, the packet arrival process in a station  $i$  is assumed to be Poisson with rate  $\lambda_i$ . The packet size  $X$  can be arbitrarily distributed with mean  $EX$  and the maximal packet length is  $X_{MAX}$ . A saturated station is considered to have always a packet to send with the same packet size distribution.

A network with  $N = 16$  stations is observed, which are equidistantly located on a folded bus system. The slot length is chosen at 55 Bytes, consisting of 48 Bytes payload segment field, 5 Bytes wide area network header and 2 Bytes CRMA header. Packets are assumed to have a truncated Poisson distribution. The time axis in the diagrams is given in slot transmission time. In the following, a bus of capacity 1.2 Gbps with a length of 10km will be taken, corresponding to a bus latency of 273 slots.

Recall that the purpose of the reservation-cancellation backpressure mechanism is to protect the scheduler queue against overload. The main mechanism is to prolong the interreserve interval artificially in case of scheduler overload. Since a station is allowed to reserve only one packet per reservation command, this mechanism reduces the total load offered to the scheduler under overload conditions. However, since  $L$  is chosen to be equal to the bus latency  $\tau$ , the backpressure mechanism is expected to be active only under very high load levels or in the case where a few stations are sending packets under saturated conditions.

Fig. 9 shows the influence of the backpressure mechanism on the scheduler delay and its distribution function, where a comparison of CRMA with and without the reservation-cancellation backpressure mechanism is illustrated. A normal load level  $\rho = 0.5$  is chosen. In this Figure 95% confidence intervals obtained by simulation to validate the accuracy of the approximate analytical method are also depicted. The load-limiting function of the backpressure scheme can clearly be recognized. The distribution function of the scheduler delay is bounded by  $L - 1 - \Theta$ . This value corresponds to the case where after processing the last reserve command, exactly  $L - 1$  slots are in the scheduler queue.

The backpressure mechanism obviously diminishes the influence of a local overload situation, e.g. when a station is heavily overloaded or in saturation, on the global system performance. Due to the decreased reserve-rate the packets are delayed in the local queues instead of the scheduler. Thus the traffic is distributed among the stations and not concentrated at the scheduler. Therefore we expect in general a smaller access times for systems running with the backpressure mechanism than without, especially under heavy load.

Fig. 10 shows in summary the prereservation delay, the scheduler delay and the medium access delay of station 1, 8, and 16. A position-dependency of the access delay can be

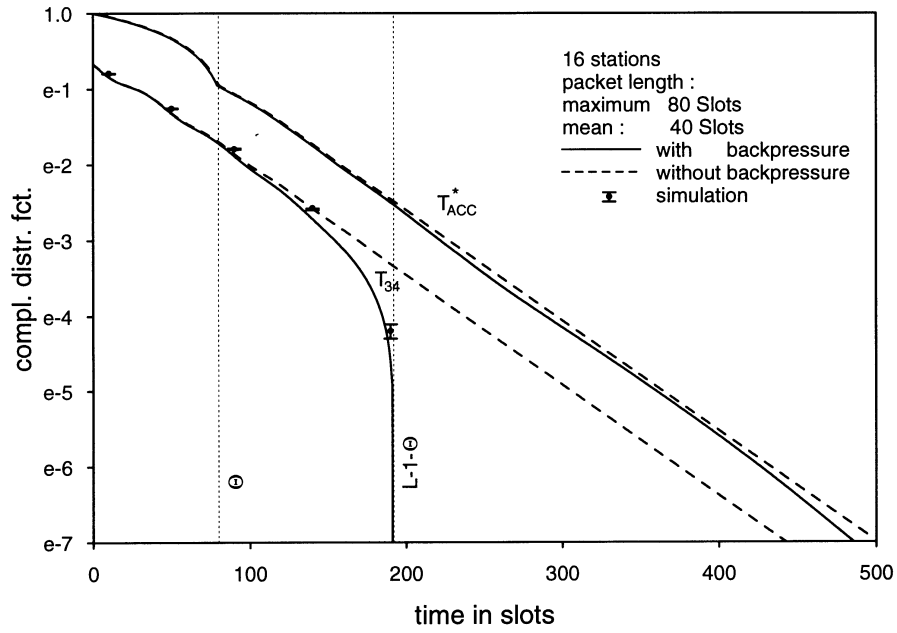


Figure 9: Comparison of scheduler delay with and without backpressure

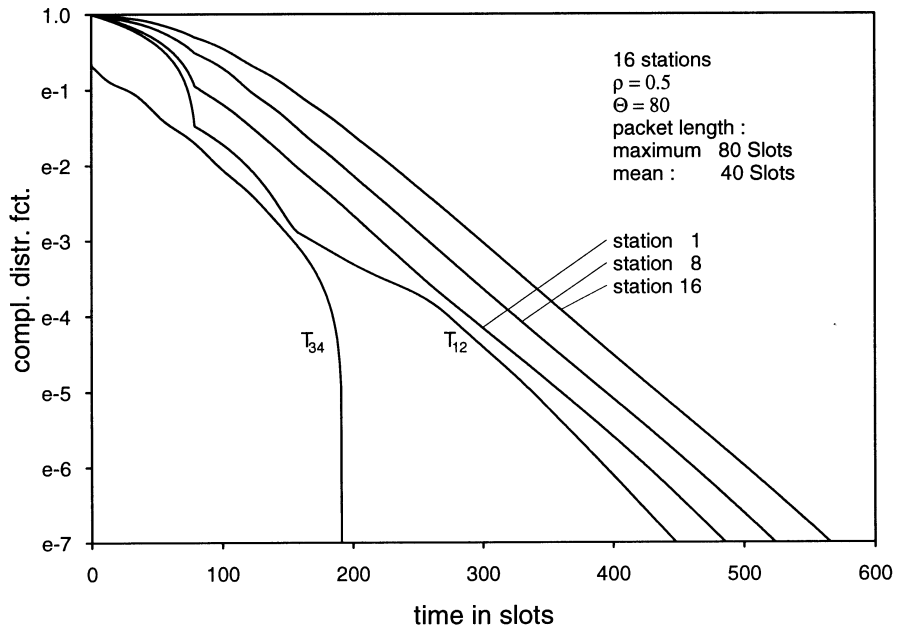


Figure 10: Delays observed in different stations

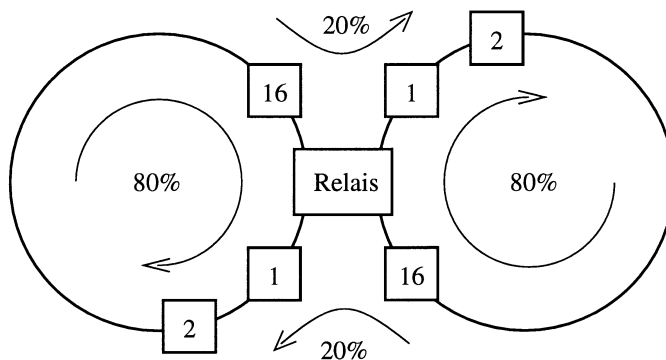


Figure 11: Configuration of bridged rings

recognized, which is caused by the different intra-cycle delays of the considered stations. The access delay depends on the sequence of the stations. In Fig. 10 the latency  $\tau$  (i.e. 273 slots for the configuration b) is subtracted, thus the *relative* differences of access delays seen by different stations are smaller than appeared.

## 4. Internetworking performance

Today a typical bridge between neighbouring LANs consists of an access unit, an error detection/correction-unit (and reassembly and packetizer unit, since e.g. the existing IEEE 802.x LANs and MANs support different maximum packet lengths). It may be very costly to stay with this structure when switching to high-speed environments. Furthermore this may be neither necessary nor useful since future systems tend to adapt to the ATM-principles of end-to-end flow control, end-to-end error detection/correction etc.

Therefore the bridge functionality should be reduced. Then obviously, some kind of common packet format is required. In the context of MAN and public ATM networks this means a common slot format. As a minimum requirement a common payload format should be introduced so that a complex assembly/reassembly unit can be omitted. Ideally, a common address space is also used, since this would help to avoid potentially large address translation tables. That way only the access protocol specific signalling data has to be converted.

A still remaining problem are the potentially large buffer requirements in the bridge. To give some first insight we observed a system of two rings interconnected by simple bridge as mentioned above. The parameters of the individual rings were identical to those of the dual bus system considered in subsection 3.1, except that 20 % of the traffic generated in one ring is destined for the other ring and therefore crosses the bridge located between stations 16 and 1 (see Fig. 11).

Fig. 12 compares the simulation results for two access mechanisms: a multiple token

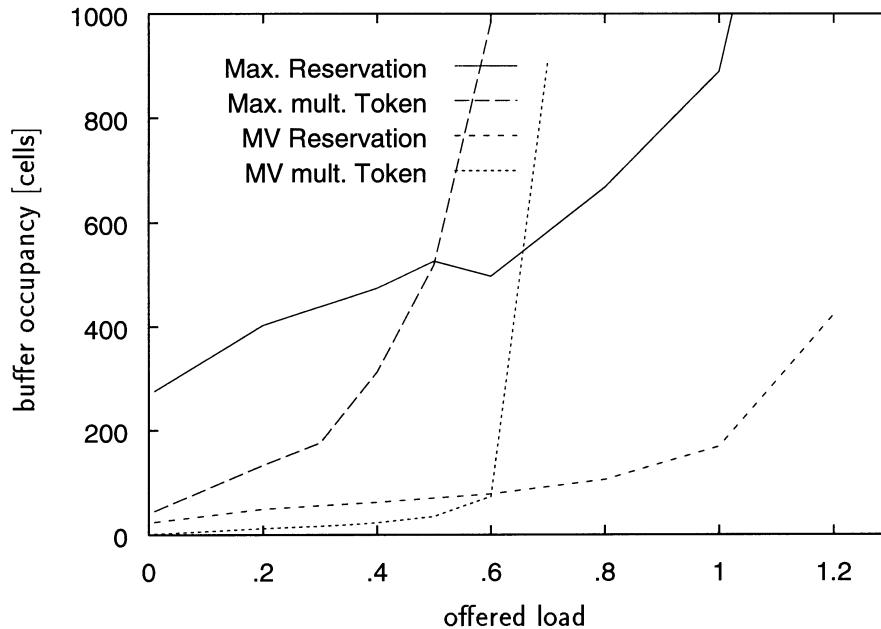


Figure 12: *Buffer occupancy in the bridge*

protocol (cf. subsection 2.2.2) and a reservation based method similar to Cyclic ring (see subsection 2.2.3).

Fig. 12 depicts the maximum (Max.) as well as the mean value (MV) of the buffer occupancy of the bridge during a simulation of symmetrical load (note that only one direction of transfer is observed). As long as it is stable the token based protocol provides significantly smaller mean and maximum values since even in a nearly empty system under the reservation mechanism the bridge typically stores around 25 cells on average.

The advantage of the reservation mechanism is that it supports cell reuse and thus allows for a much higher load than the token based variant. It is stable up to a load of about 120 %.

As a consequence it would be desirable to synthesize a protocol which supports immediate access at low load and equal distribution of bandwidth at high load as the reservation based mechanisms do. This is what e.g. CRMA-II tries to achieve.

## 5. Conclusion

In this paper we discussed basic concepts and performance issues of the class of high-speed protocol, such as proposed for HS-LANs, MANs or access schemes for B-ISDN. We approached to give a systematic introduction to basic protocol mechanisms in a more generic

context, where well-known protocols appeared as specific implementations of these mechanisms. To illustrate the performance evaluation methods used to investigate these protocols, two examples have been outlined: the DQDB protocol class, which is based on the distributed queueing principle, and the CRMA protocol, which represents a reservation-based case. Finally some aspects concerning internetworking and related performance investigation were mentioned.

## References

- [1] Adams J.L., *Orwell*, to be published in Special Issue of Computer Networks and ISDN Systems, Autumn, 1992.
- [2] Cidon I., Ofek Y., *Metaring — A Full-duplex Ring with Fairness and Spatial Reuse*, Proc. INFOCOM '90, San Francisco, 1990.
- [3] Falconer R.M., Adams J.L., *Orwell: A Protocol for an Integrated Service Local Network*, Advances in Local Area Networks, Kümmerle et. al. (Ed.), IEEE Press, 1987.
- [4] Hopper A., Williamson R.C., *Design and Use of an Integrated Cambridge Ring*, Advances in Local Area Networks, Kümmerle et al. (ed.), IEEE Press, 1987.
- [5] Hopper A., Needham R.M., *The Cambridge Fast Ring Networking System*, IEEE Trans. Comp., vol. 37, no. 10, Oct. 1988.
- [6] Huber D.E., Steinlin W., Wild P.J., *SILK: An Implementation of a Buffer Insertion Ring*, Advances in Local Area Networks, Kümmerle et al. (ed.), IEEE Press, 1987.
- [7] IEEE, *IEEE Standard 802.6-1990 — Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)*, IEEE, 1990.
- [8] Kaibel M., *Leistungsvergleich von Multi-Request-Methoden in DQDB-Protokollumgebungen*, Diploma thesis, institute of computer science, University of Würzburg, 1992, in german.
- [9] Kamal A.E., *Efficient Multi-Segment Message Transmission with Slot Reuse on DQDB*, Proc. INFOCOM '91, Miami, April 1991.
- [10] Limb J.O., Flores C., *Description of Fasnet — A Unidirectional Local Area Communication Network*, Advances in Local Area Networks, Kümmerle et al. (ed.), IEEE Press, 1987.
- [11] Müller H.R., Nassehi M.M., Wong J.W., Zurfluh E., Bux W., Zafiropulo P., *DQMA and CRMA: New Access Schemes for Gbit/s LANs and MANs*, Proc. IEEE INFOCOM '90, San Francisco, Ca, June 1990.
- [12] Mukherjee B., Banerjee S., *Alternative Strategies for Improving the Fairness in and an Analytical Model of DQDB Networks*, Research Report, Division of Computer Science, Department of Electrical Engineering and Computer Science, University of California, Davis, CSE-90-32, July 1990.

- [13] Mukherjee B., Kamal A.E., *Scheduling variable-length messages on slotted high-speed fiber optic LANs/MANs using the continuation-bit approach*, Proc. IEEE INFOCOM '91, Bal Harbor, FL, April 1991.
- [14] Mukherjee B., Bisdikian C., *A Journey Through the DQDB Literature*, Research Report RC 17016(#75478), IBM Research Division, November 1991.
- [15] Nassehi M.M., *CRMA: An Access Scheme for High-Speed LANs and MANs*, Proc. IEEE Supercom '90, Atlanta, Georgia, April 1990.
- [16] Nassehi M.M., *Cyclic-Reservation Multiple-Access Scheme for Gbit/s LAN's and MAN's Based on Dual-Bus Configuration*, Proc. EFOC/LAN '90, Munich, F.R.G., June 1990.
- [17] Pihlaja J., *Cyclic Ring*, Temporary working document, RACE project 1011, November 1990.
- [18] Schwartz M., *Telecommunication Networks*, Addison-Wesley, Reading, MA, 1987.
- [19] Stock T., Tran-Gia P., *A Discrete-Time Analysis of the DQDB Access Protocol*, Proc. ITS '90, Rio de Janeiro, September 1990.
- [20] Stock T., *Influences of Multiple Priorities on DQDB Protocol Performance*, Proc. 13th ITC, Copenhagen, Denmark, June 1991.
- [21] Stock T., *Schnelle lokale und regionale Netze — Konzepte und Fallstudien zur Leistungsbewertung*, Ph.D. thesis, Institute of Computer Science, University of Würzburg, 1992, *in german*.
- [22] Tobagi F.A., Borgonovo F., Fratta L. *Expressnet: A High-Performance Integrated-Services Local Area Network*, Advances in Local Area Networks, Kümmerle et al. (ed.), IEEE Press, 1987.
- [23] Tran-Gia P., Stock T., *Approximate Performance Analysis of the DQDB Access Protocol*, Proc. ITC Specialist Seminar, Adelaide, Sept. 1989, also in Computer Networks and ISDN Systems, 20 (1990) 231–240.
- [24] Tran-Gia P., Dittmann R., *A discrete-time analysis of the cyclic reservation multiple access protocol*, Performance Evaluation, 16, 1992, 185-200.
- [25] van As H.R., Lemppenau W.W., Zafiropulo P., Zurfluh E.A., *CRMA-II: A Gbit/s MAC Protocol for Ring and Bus Networks with Immediate Access Capability*, Proc. EFOC/LAN '91, London, UK, June 19–21, 1991.
- [26] Wilkes M.V., Wheeler D.J., *The Cambridge digital communication ring*, Proc. Local Area Commun. Symp., Boston, MA, May 1979.