Fairness Issues in a New Dual-Bus Network Technology

Csaba ANTAL, József MOLNÁR, Sándor MOLNÁR

High Speed Networks Laboratory, Department of Telecommunications and Telematics,

Technical University of Budapest, Sztoczek u. 2., H-1111 Budapest, Hungary

Abstract: Dynamic Synchronous Transfer Mode (DTM) is a new fast circuit switching networking technology. Performance analysis works showed that it could be a viable technology in the future. Its operation is based on multi-access dual fiber bus. In the case of multi-access protocols fairness is always an interesting question, and it has not been investigated for DTM protocol in detail. This paper discusses the fairness of known channel reallocation algorithms and proposes a new one, which is fair in all examined traffic scenarios. The work is based on simulation, which was carried out by a self-developed simulator.

Keywords: fairness, fast circuit switching, channel allocation

1 Introduction

Dynamic Synchronous Transfer Mode (DTM) [4,9] is an attempt to build the next generation of networking technologies on fast circuit switching basis. It is a new broadband network architecture developed at the Royal Institute of Technology in Stockholm (KTH). The technology is in the focus of several Swedish companies, because it can be the competitor to technologies like FDDI or Asynchronous Transfer Mode (ATM) [10,12] in special application areas.

FDDI is one of the most popular backbone networking technologies in the metropolitan area. It is based on dual fiber ring. DTM, which is also based on dual fiber can be an upgrade path of FDDI to higher speeds in the future.

ATM transfer capabilities define service classes, which are appropriate for several application types. However, ATM is not the most natural solution for applications with low delay and delay variance requirements due to its inherent cell (packet) switching characteristics. Applications with high bandwidth and low delay variance needs (e.g.: video on demand, video telephony) are more suited to the philosophy of circuit switched networks.

Performance analysis of DTM in the case of bus and fully connected mesh topology was investigated in [4]. It was based on a single distributed channel reallocation algorithm, which will be referred to as KTH algorithm. We described a new algorithm [1, 2], which can be used as an extension to the earlier method. In another paper [3] we proposed the extension of DTM to provide various ATM-like service classes.

DTM uses dual bus architecture so as with DQDB networks. The main disadvantage of DQDB, which prevented its wide acceptance, was unfair operation in the case of high network loads [8]. This similarity inspires that a sophisticated fairness analysis is to be done for DTM as well. This is the first attempt to do this work. As a first step, we investigate possible unfair

operations caused by wrong choice of channel reallocation algorithms. The effect of long buses and propagation delay are subject of future work. We considered average setup time and blocking probability as the most important performance characteristics of nodes, and used them to evaluate the fairness of DTM dual-bus.

We propose a new channel reallocation algorithm here. We compare it to the variants of three other techniques in three different network setups. Simulation study shows that the new algorithm and one of the three known algorithms are fair in all examined network configurations, while two of them are unfair.

The paper is structured as follows.

In Section 2 Dynamic Synchronous Transfer Mode is introduced. The proposed channel reallocation method and its known counterparts are also described in this section. In Section 3 the simulation model is shown. First part of this section discusses the model for traffic generation. In the second part of this section three network configurations are presented, which are used as the basis of our fairness investigations. In Section 4, the results of the simulation study are described where the variants of four algorithms are analysed in three different configurations. Finally, in Section 5, we conclude the paper.

2 Dynamic Synchronous Transfer Model

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. 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 1 shows the setup of a dual-bus network.

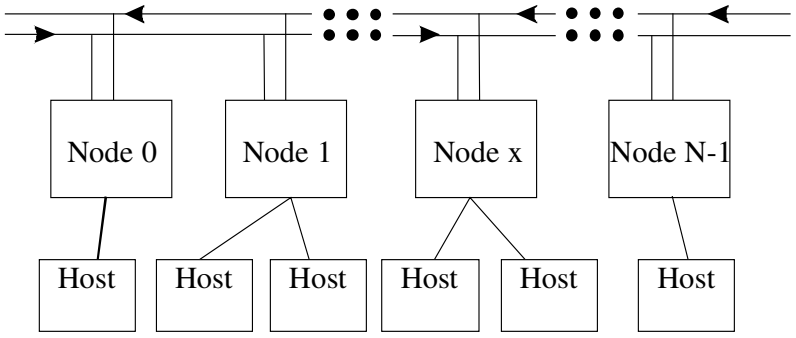

Figure 1: Structure of a DTM bus

The total communication channel on the physically shared medium is realised 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 ones. 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 are assigned to nodes. Each channel has exactly one owner at a time. If a node has the right to use a channel, then it has full control on it: it can setup 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 by the nodes but 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 reallocation.

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 [5]. A procedure for channel reallocation was proposed in [4], which is referred to as **KTH algorithm** in this paper. 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 for nodes, therefore it can happen that they are not up-to-date. If the signalling load of the DTM bus is high or the processing capacity of a node is overloaded, its status table can become outdated. The correct modelling of this overloaded situation is the subject of future research. Here we undertake the studying of the two extreme cases:

- status tables are up-to-date and nodes only try to get free channels from nodes that have them (KTH-S algorithm)
- status tables are useless and nodes try to get free channels from others without any apriori knowledge
  (KTH-CF, KTH-LR, KTH-RA)

2.1 KTH-S algorithm

In this algorithm we assumed that there is enough signalling bandwidth and processing power to keep status tables up-to-date.

The details of the operation are the followings:

There are two connection setup 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 setup 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. Now we finished the description of the connection setup procedure of KTH-S algorithm.

If nodes have correct status tables, the order of nodes during asking for channels is not relevant. If there are free channels in the

network, nodes are aware of them, so they can get them. Blocking occurs if there are no free channels in the system. The simulated KTH-S algorithm used logical ring, which will be described in Section 2.3, to determine the order of nodes asked for free channels. Any other order (closest first, random order) resulted the same results.

Retry limit 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.

If status tables become outdated, the sequence of nodes at channel request and the value of retry limit become important. We examined three possible algorithms regarding the order. KTH-CF algorithm was proposed in [4, 5]. We proposed KTH-LR in [1, 2]. KTH-RA is a new proposal to correct the unfair operation of the above two ones. The description of these algorithms can be found below.

2.2 KTH-CF algorithm

In the KTH-CF (Closest First) algorithm nodes ask for channels from the closest node first. The second closest node is the second in the sequence of channel requests, and so on. The fairness of this algorithm was investigated in [1]. It turned out that the algorithm was unfair because nodes at the middle part of the bus had less free channels in average. It resulted higher blocking probability or higher average connection setup time depending on the channel reallocation retry limit.

Because of this result this algorithm must be ignored, so we don't discuss further of its analysis.

2.3 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 take the closest neighbouring node along the ring first, then the second closest ring-neighbour and so on. The order can be determined based on

the same rule 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.

We constructed a ring 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 2.

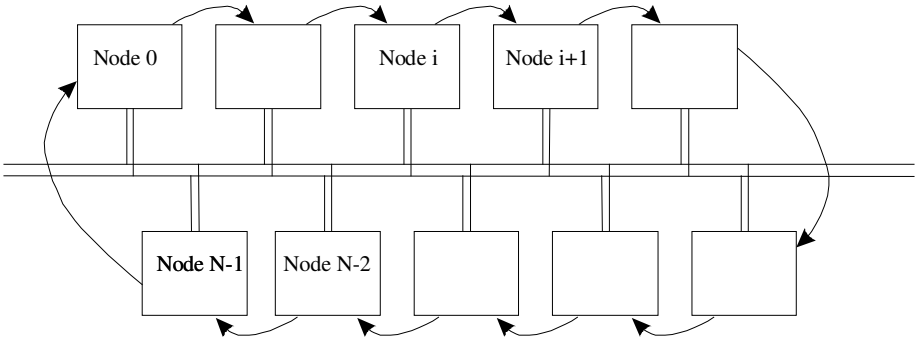

Figure 2: Logical ring structure

2.4 KTH-RA algorithm

Finally we propose a new algorithm to correct the unfair operation of DTM in the case of useless status tables. 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.

In the case of algorithms where nodes don't have status tables, the channel reallocation retry limit is a very important parameter. It was shown in [1] that the higher the retry limit was the lower the blocking probability and the higher the average connection setup time was. Fairness based on these characteristics is also retry limit dependent.

3 Configuration of the simulated networks

In this section first we describe the model of traffic generation, in other words the host model. The second part of the section discusses three network scenarios used for our simulation setups.

3.1 Traffic generation

The first question is what kind of model should be used for hosts, which are generating connections.

According to the present network traffic predictions a dominant part of future data services will generate World Wide Web traffic. The analysis of Web traffic showed that the

user-initiated TCP session arrival process could be well modelled by Poisson processes like in classical telephony [11]. However, the Poisson process cannot be used for modelling the arrival of WWW requests because it contains several non user-initiated requests. When a user calls out a page, the browser program generates a series of additional requests to download the images of the appearing page. Several studies suggest the use of long-tailed distributions such as Weibull or Pareto distributions for modelling the arrival process of WWW and for estimating the size of requested documents [6, 7].

Based on these studies, we use the following traffic model. The inter-arrival time X is modelled by a Weibull distribution given by the probability density function

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

where parameter β and parameter λ depend on the generated traffic profile. Analytical studies of the arrival process of WWW requests suggested the use of parameter $\beta = \frac{1}{3}$ [7]. With this value the mean of the inter-arrival time is

$$E(x) = \frac{6}{\lambda} \tag{2}$$

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

$$f(t) = \frac{\alpha k^{\alpha}}{t^{\alpha+1}} \tag{3}$$

where the parameter is chosen to be $\alpha = 1.9$. In the simulation, hosts generate one-channel connections, so the parameter k depends on the assumed mean size of the files to be transmitted.

The mean holding time T of a connection is

$$E(t) = \frac{\alpha}{\alpha - 1} k \tag{4}$$

All parameters were selected based on the analysis of measured WWW traffic [7].

In our model, all hosts generate bi-directional point-to-point connections and they require the bandwidth of one channel in both directions. A bi-directional connection at the DTM access control level is two unidirectional connections, where channels are reserved by the sender nodes. In other words, both the sender and the receiver of a point-to-point bi-directional connection are responsible for the reservation of channels on one of the unidirectional buses.

We recorded the blocking and setup time characteristics to the reserver node because we were interested in the characteristics at the DTM access control level. Due to this trick we were able to observe the characteristics of the switching node and the node of the server host.

3.2 Network Model

All of the investigated networks in different scenarios are based on a single dual-bus. In all of our setups, there are 100 nodes connected to the dual-bus. The simulated dual-bus has 1200 data channels in both direction. The distance between nodes is 10m so the total length of the bus is 1km. In all network scenarios to be described below a slightly overloaded network is simulated to make possible unfair operation obvious (see DQDB). The total load of the network is approximately 1.25 times higher than the network capacity.

Three different network scenarios are simulated. The first network scenario only considers the connections to hosts outside the simulated dual-bus (external). The other scenarios are based on client-server and peer-to-peer traffic between hosts in the same dual-bus.

The real traffic of a local area network can be modelled as the superposition of these traffic types. So the results gained from these scenarios can be used to estimate the operation of the algorithms in real circumstances.

The detailed description of the three network scenarios can be found below.

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 synchronised by a switching node, which is attached *to the end of both dual-buses*. The interest of this paper is a single dual-bus, therefore we modelled a connection to a host outside the simulated dual-bus by a connection to the switching node. Blocking probability and setup delay are higher in the case of a real external connection because we only considered the part of connection blocking and connection setup delay affected by the conversations between nodes on the observed dual-bus. The results are useful if we want 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.

We assume that hosts initiate bi-directional high level connections to the switching node. A bi-directional connection should be setup as two unidirectional connections at the DTM access control level. We replaced the backward direction of the connections, where the initiator is the receiver, 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:

- one 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.

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 the client-server traffic model, we assume 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 we replace the receiver initiated parts of the connections to a same but sender initiated connection between the same hosts, the following hosts are needed:

- one 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.

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.

4 Simulation Results

We developed an event-controlled simulator in C++ language to examine fairness and performance of the introduced algorithms. In this section we show the results of our fairness study in the above discussed traffic scenarios.

4.1 External Traffic

In the case of external traffic scenario, the offered load of the unidirectional buses can be seen in Figure 3. 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. If a channel reallocation algorithm turns out to be unfair in this load scenario, we should drop it.

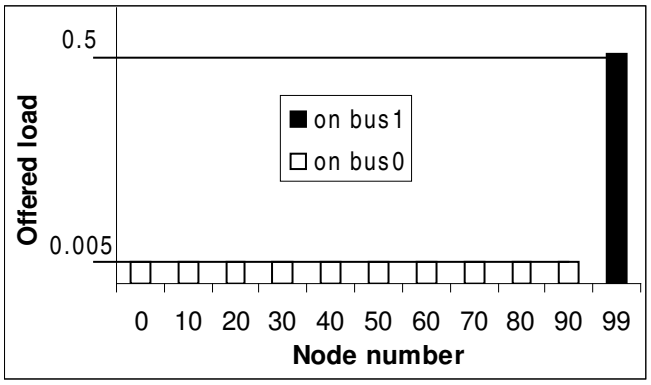

Figure 3: Offered load in the external traffic scenario

In Figure 4 the blocking probabilities and in Figure 5 the average setup times are compared at different algorithms. Blocking probability is the ratio of succeeded connections and requested connections. The numbers in the names of KTH-LR and KTH-RA algorithms are the channel reallocation retry limits. The algorithms are shown side by side in the figures. Nodes are represented according to their physical location on the *dual-bus*. In the case of algorithms using logical ring during channel reallocation (KTH-S, KTH-LR), it is important which nodes are neighbours on the logical ring. It is the reason why points of these algorithms are connected according to the logical ring.

We can see on Figure 4 and Figure 5 that KTH-S and all variations of KTH-LR and KTH-RA algorithms are fair from all points of view.

In the case of KTH-CF algorithm, it was shown in [1] that the ends of the buses cause unfair operation. The cause of the unfair characteristics is that nodes in the middle of the bus have less free channels in average.

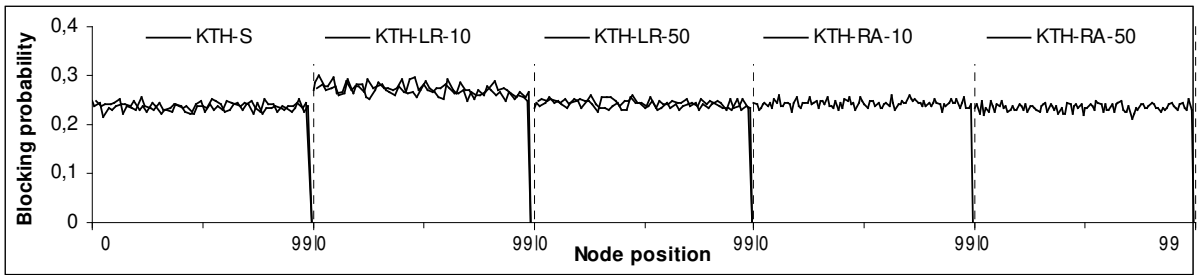

Figure 4: Blocking probability on bus 0 in the case of external traffic scenario

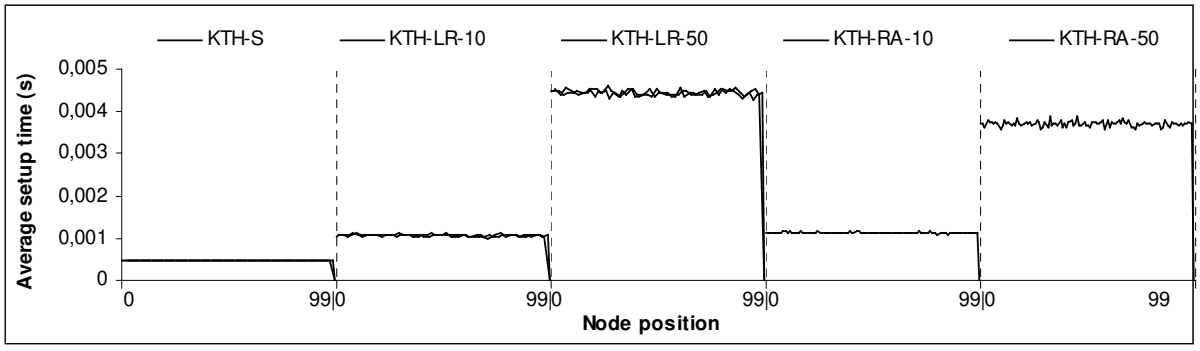

Figure 5: Average setup time on bus 0 in the case of external traffic scenario

This scenario is only the first step in the study because the shown unidirectional bus had balanced load. The other examples show how the network performs if nodes have different loads for a given unidirectional bus.

4.2 Client-Server Traffic

If we have a look at one of the unidirectional buses in the client-server setup, we 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 6 the load profile of the bus in the other direction is the same but mirrored.

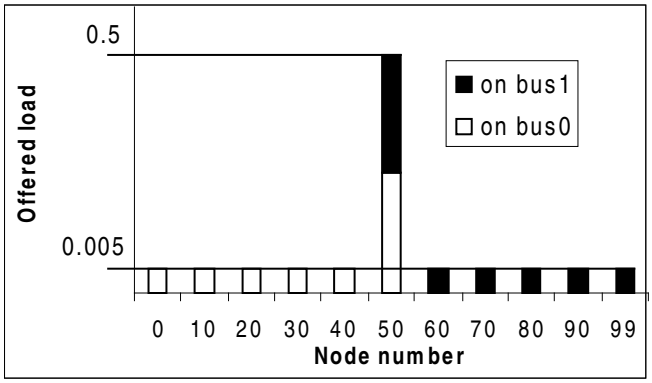

Figure 6: Offered load in the client-server traffic scenario

In Figure 7 the blocking probability of unidirectional connections directed to bus 0 could be seen node by node in the case of different algorithms. Figure 8 shows the average setup time of different nodes and different algorithms based on the traffic directed to bus 0.

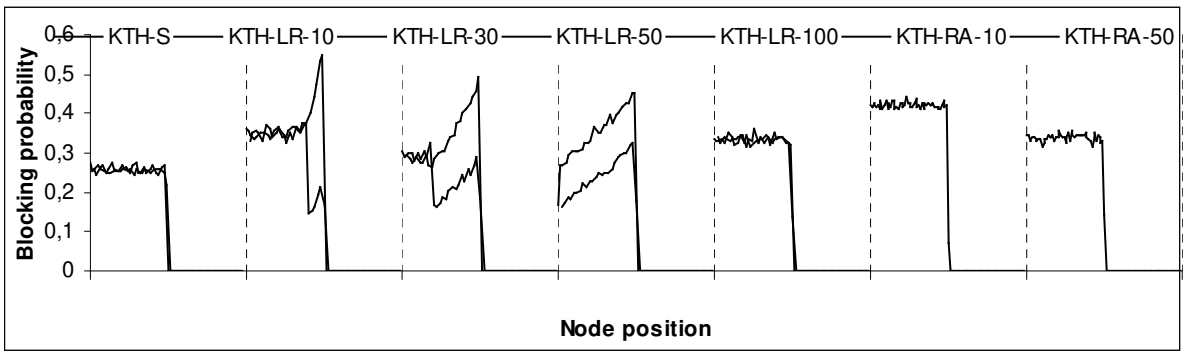

Figure 7: Blocking probability on bus 0 in the client-server scenario

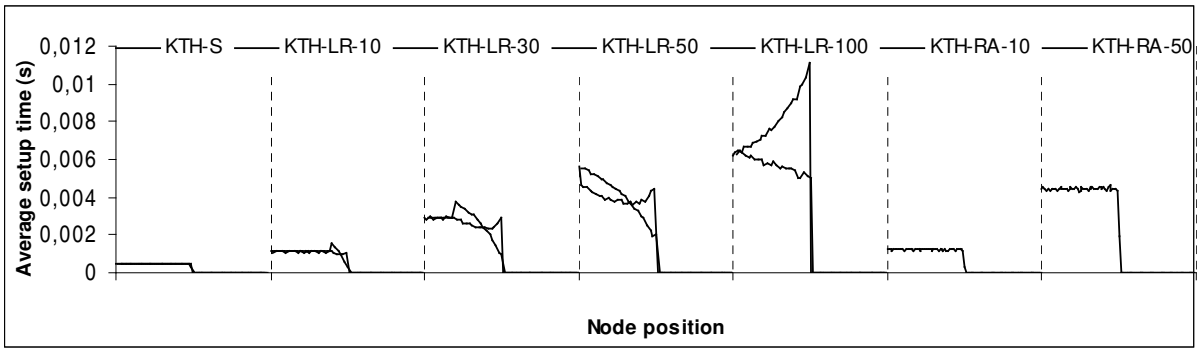

Figure 8: Average setup time on bus 0 in the client-server scenario

The characteristics of server node are hard to read in the figures, therefore we extracted it to Table 1.

Algorithm	KTH-S	KTH-LR-10	KTH-LR-30	KTH-LR-50	KTH-LR-99	KTH-RA-10	KTH-RA-50
Blocking probability	0.22	0.17	0.18	0.18	0.14	0.07	0.14
Average Setup Time (ms)	0.2	0.3	1.1	2.2	5.1	0.3	1.8

Table 1: Characteristics of the server node in the client-server traffic case

Though the curves of the algorithms differ, they are common in one property. Namely, the blocking probability and average setup time of the server node are lower than the characteristics of an average client.

Client nodes have the same characteristics when using KTH-S algorithm. Our prediction was true in this case: correct status tables yield to fair operation except the small difference between setting up a connection without channel request and setting it up with a very few requests. This small difference is the cause of the better characteristics of the server node. Because the node of the server has much higher intensity then client nodes, it collects most of the free channels. We will refer to this effect as "cache" property of the DTM protocol. Cache property decreases both setup time and blocking for very active nodes.

Apart from the better performance of the server node, the KTH-RA algorithm is also fair. The connection setup time and the blocking probability of the client nodes are the same.

KTH-LR algorithm is UNFAIR in this configuration. Either blocking probability or average setup time is different at clients with the same offered load in the case of all retry limits.

Although the detailed analysis of KTH-LR algorithm does not change the main conclusion that this algorithm is unfair, we try to explain the results.

The cache property can be applied here when analysing client nodes as well. The distance from the server node (very active node) and from the idle nodes affects the behaviour of the

client nodes. It is hard to understand the reasons behind the characteristics of nodes based in Figures 7 and 8 because they display nodes at their physical locations. This channel reallocation algorithms is based on a logical ring (ring was shown in Figure 2), therefore Figure 9 shows KTH-LR-30 algorithm in another view. In this figure nodes are displayed at their positions on the logical ring. There is a cut between node 0 and node 99, they are neighbours on the logical ring.

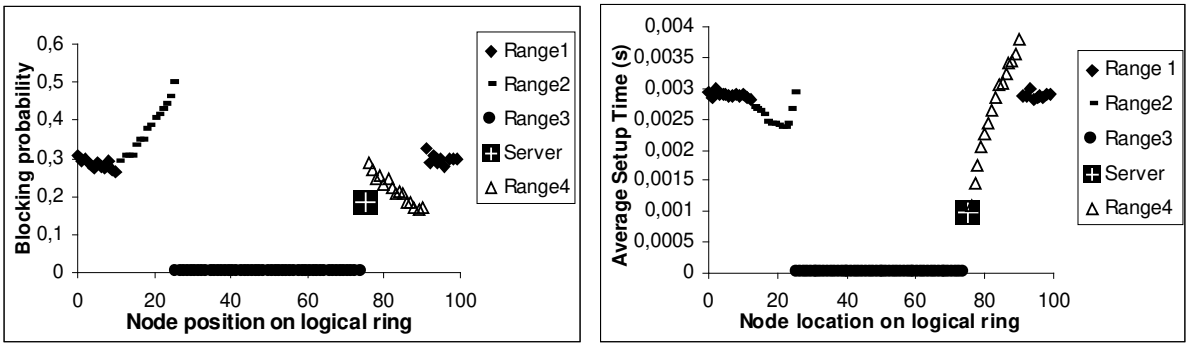

Figure 9: Characteristics of KTH-LR-30 algorithm based on the traffic directed to bus0

Based on this figure we can identify four ranges of client nodes 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)

Two nodes are far in this context if they can not change channels directly (because of the limited number of retries) and they are close if they can.

First let us see 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 setup 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. Starvation effect: The server (whose load is 50 times higher) asks channels from them. The traffic of the server is bursty (see section 3.1) therefore if a burst comes, it collects all the channels from its proximity. Conclusion: It is not likely that a client node in this range has free channels.
2. Proximity effect: Client nodes can ask the node of the server for channels. The server node has more free channels because of the bursty traffic and the heavier activity. Conclusion: It is likely that they can ask channels from the server.

The result can be seen on Figure 9. The closer a node is to the server the higher the blocking probability is due to the starvation effect. The farther a node is from the server the higher the average setup time is, because of the proximity effect.

In range 2, in the proximity of the idle nodes, the farther the nodes are from range 3 the lower the blocking probability is. The shape of the average setup 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 setup 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 at its active neighbours due to the starvation effect. At the border of range 4 and range 1, the node in range 4 has higher setup time due to the starvation effect (average setup time is based on successful connections) and lower blocking probability because of the proximity of 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.

In Figure 10 and 11 we can see the blocking probability and average setup time based on traffic of both buses.

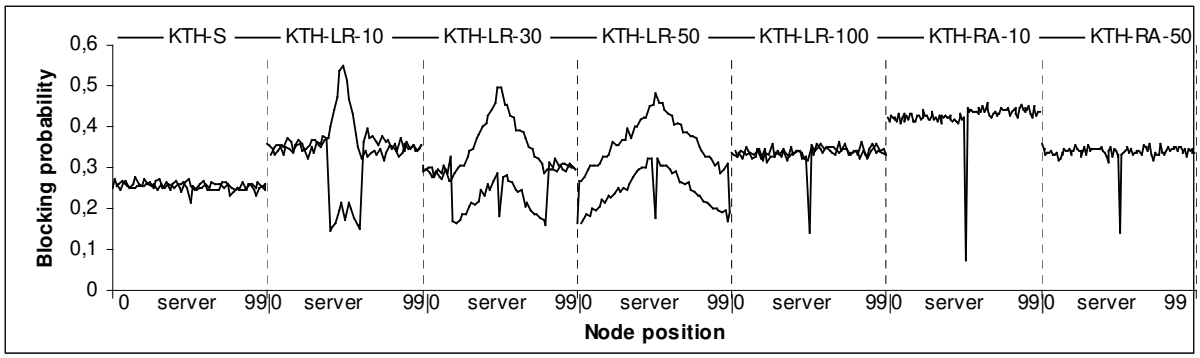

Figure 10: Blocking probability at the client-server scenario based on the traffic of buses in both direction

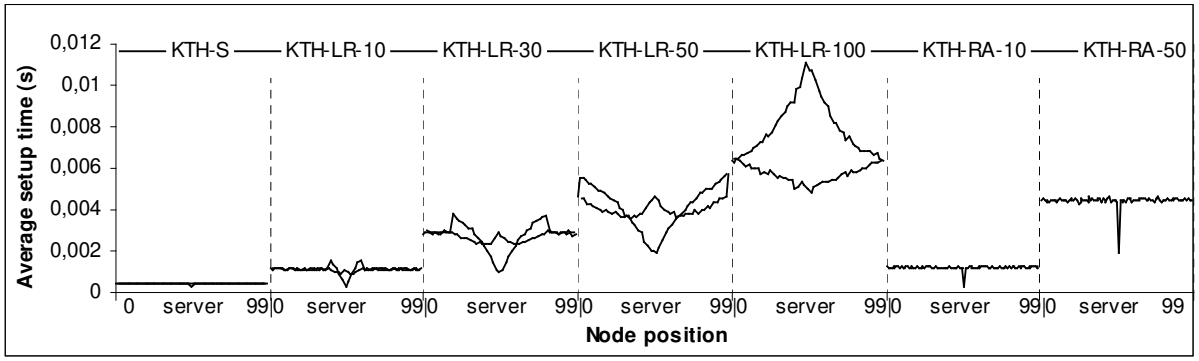

Figure 11: Average setup time at the client-server scenario based on the traffic of buses in both direction

It can be concluded from this traffic 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 as well due to the starvation and proximity effect, which is a very important drawback of

this algorithm. Another conclusion is that very active nodes (here the server) have lower blocking probability and setup time characteristics because of the "cache" effect. This advantageous property can be identified at all of the algorithms.

4.3 Peer-to-peer traffic

The next traffic scenario is the peer-to-peer traffic. The traffic profile of the unidirectional buses can be seen in Figure 12. If we consider only one of the buses the offered load of the nodes is proportional to the distance from the end of the bus.

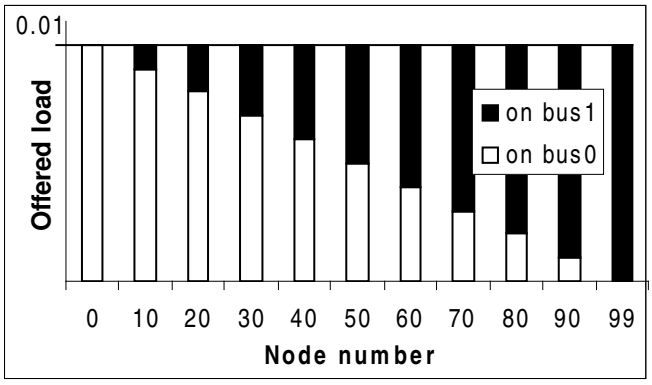

Figure 12: Offered load in the peer-to-peer traffic scenario

Figure 13 and 14 show the characteristics of nodes considering the traffic load directed to bus 0. The blocking probability of nodes based on the traffic directed to both buses can be seen in Figure 15.

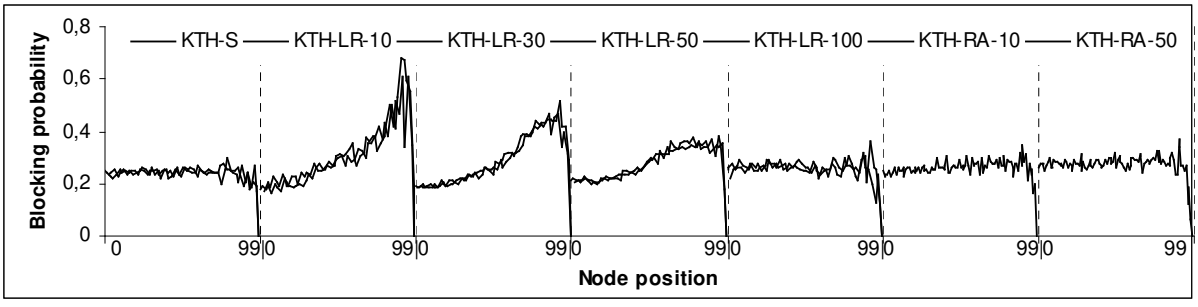

Figure 13: Blocking probability on bus 0 in the peer-to-peer scenario

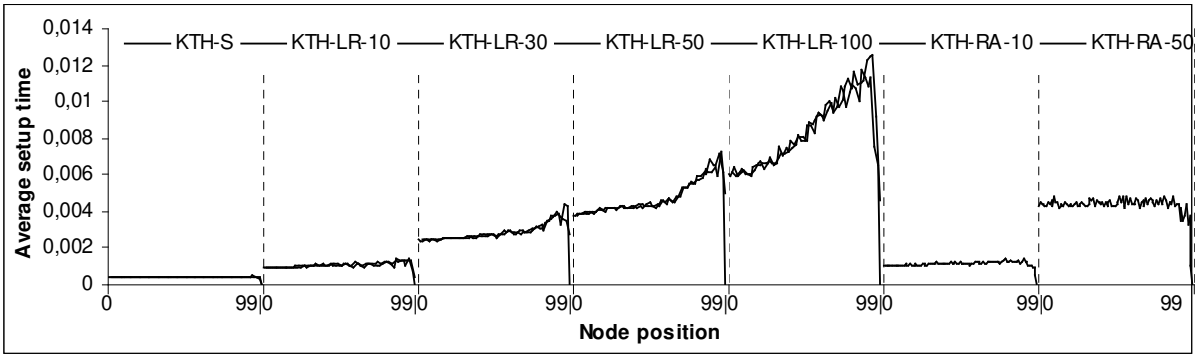

Figure 14: Average setup time on bus 0 in the peer-to-peer scenario

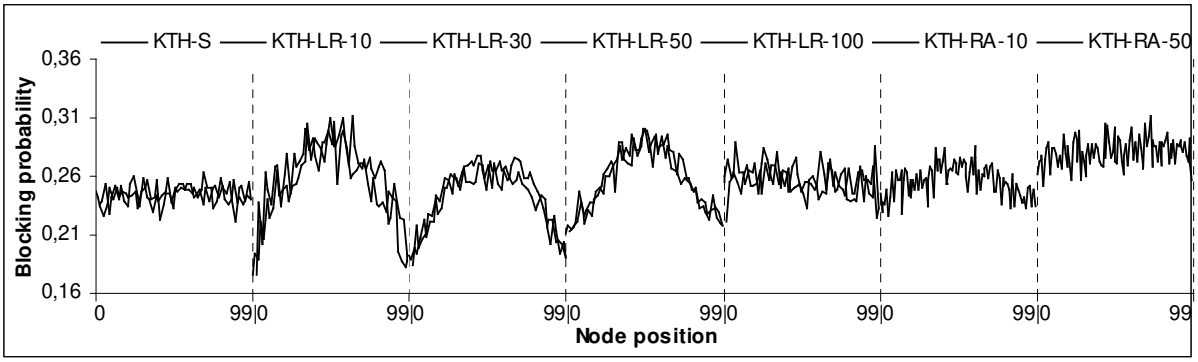

Figure 15: Blocking probability in the peer-to-peer scenario based on the traffic of both buses

It can be observed that if nodes use status tables then they have the same blocking and setup time. The variants of KTH-RA algorithm are nearly fair. If the channel reallocation retry limit is low the blocking is slightly lower at nodes at the end of the buses. This small difference between nodes can not be noticed in a real operating environment.

In the KTH-LR algorithm there is significant difference between nodes, so it is unfair in this traffic scenario too. The result of the starvation effect can be seen on the blocking probability and/or on the average setup time. In general it can be stated that the lower the offered load of a node directed to the observed bus is the higher the setup time and the blocking probability are. If the retry limit is low the starvation effect is more obvious on the blocking probability and at high limits it effects the setup time.

We can conclude that in peer-to-peer network scenario KTH-S and KTH-RA proved to be fair. All versions of KTH-LR algorithm were unfair in this scenario as well.

5. Conclusion

The paper examined four channel reallocation techniques. One of them, the KTH-RA algorithm, where nodes use random order during requesting channels from other nodes, was proposed here.

KTH-S channel reallocation algorithm proved to be fair in all investigated network scenario: nodes with the same offered load and traffic parameters had the same performance characteristics. Despite the fair characteristics, in the client-server network scenario, an interesting effect was observed. If a node was more active (it had to reserve channels more frequently) than all the others, it was able to reuse its channels before they were requested by other nodes. Due to this "cache" effect this active node had better characteristics than the others did.

Three of the examined algorithms did not use status tables to help channel reallocation. Only one of these techniques, the KTH-RA, turned out to be fair in all simulated network examples. Caching worked here as well providing better performance for the server node in the client-server scenario.

KTH-CF algorithm, which used "closest first" algorithm to determine the order of nodes, was proved to be unfair in [1] in the external traffic scenario. Therefore we did not discussed its analysis in other scenarios.

KTH-LR (closest first on logical ring) algorithm behaved fairly just in the case of external traffic scenario. In the client-server and peer-to-peer scenario it was proved to be unfair because active and passive nodes changed the characteristics of their neighbour nodes as well. Therefore nodes with the same load had different characteristics.

Algorithms without status tables can be seen as models of KTH algorithm in the case of overloaded control or processing capacity. That is the reason why we propose random order to select nodes during channel allocation, even for protocols where status tables help to improve the performance of the nodes.

This work did not examine the effect of long propagation delays to fairness. Possible slot reuse algorithms could also effect the fair operation of the network. These topics are subjects of future research.

6. Acknowledgements

The work is a part of the ”Performance Evaluation of Broadband Multimedia Networks” research project of the High Speed Network Laboratory (HSNLab), Department of Telecommunications and Telematics, Technical University of Budapest. We are grateful to the colleagues at HSNLab for their support. We also would like to thank Gábor Szabó for his work during the first phase of the development of the DTM simulation software.

7. References

1. Cs. Antal, J. Molnár; S. Molnár, G. Szabó "Performance Study of Distributed Channel Reallocation Techniques for a Fast Circuit Switched Network", submitted to Computer Communications Special Issue on the Stochastic Analysis and Optimisation of Communication Systems

2. Cs. Antal, S. Molnár, L. Györfi, G. Szabó, ”Resource Reallocation in a Communications Network”, patent application, May 1997, Sweden

3. Cs. Antal, ”Servicing Bursty Sources Efficiently via a Fast Circuit Switched System”, International Conference for Computer Communication '97, 19- 21 November 1997, Cannes, France

4. C. Bohm, M. Hidell, P. Landgren, L. Ramfelt, P. Sjödin, ”Fast Circuit Switching for the Next Generation of High Performance Networks”, IEEE J. on Selected Areas of Communications, Vol. 14., No. 2, February 1996

5. C. Bohm, P. Lindgren, L. Ramfelt P. Sjödin, ”Resource Reservation in DTM”, 1st IEEE Symposium on Global Data Networking, Cairo, Dec. 1993

6. 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

7. S. Deng, "Empirical model of WWW document arrivals at access link", IEEE International Conference on Communications, p. 1797-802 vol.3, ICC'96, 23-27 June 1996, Dallas, TX, USA

8. Kumar, L.N.; Douligeris, C. "Demand and service matching at heavy loads: a dynamic bandwidth control mechanism for DQDB MANs", IEEE Transactions on Communications, Vol: 44 Iss: 11 p. 1485-95

9. P. Lindgren, "A Multi-Channel Network Architecture Based on Fast Circuit Switching", Ph.D. thesis, ISRN KTH/IT/R-96/08-SE, Stockholm, May 1996

10. D. E. McDysan, D. L. Spohn, ”ATM, Theory and Application”, McGraw-Hill, 1994.

11. V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", IEEE/ACM Transaction on Networking, vol. 3, no. 3, pp. 226-244, 1995.

12. M. de Prycker, "Asynchronous Transfer Mode, Solution for Broadband ISDN", Ellis Horwood, 1991, Chester