Bergenhem, C. and M. Jonsson, "Analysis problems in a spatial reuse ring network with a simple clocking strategy," Technical Report IDE - 0253, School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University, Sweden, 2002.

Analysis problems in a spatial reuse ring network with a simple clocking strategy

Technical report no. IDE253
Halmstad University, 2002-09-20
Carl Bergenhem and Magnus Jonsson

{carl.bergenhem, magnus.jonsson}@ide.hh.se, Computers and Communications lab, Halmstad University, Halmstad, Sweden. Phone: +46-35-167100 Fax: +46-35-120348

Abstract

This technical report describes the problems associated with analysis of a spatial reuse ring network with a clock strategy also used in previously studied ring networks. The aim for the analysis was to find the worst case deterministic throughput. Some results are achieved and are presented. A worst case bound is found and this can be used for analysing feasibility of message connection sets. However, the bound is pessimistic and does not utilise the bandwidth reuse features of the network. This result does not meet the expectations that where foreseen because the medium access method inherently supports distributed global deadline scheduling. It is concluded that the current clock strategy of the network is the cause of the problems. It is also concluded that the analysis is complicated by the clocking strategy and that by introducing small changes in the network (will be presented in a separate paper), the analysis of the real-time performance will be much simpler. This will lead to a network with less pessimistic assumptions and hence better support for real-time traffic.

1 Introduction

This report discusses analysis problems in a control channel based fibre-ribbon pipeline ring network that is used together with a medium access method called two cycle medium access protocol (TCMA). The novel medium access protocol that uses the deadline information of individual packets, queued for sending in each node, to make decisions, in a master node, about who gets to send is described briefly. The previously presented network topology, the control channel based fibre ribbon pipeline ring (CC-FPR) network, is in depth presented in [1]. The protocol is further described in [2].

The proposed medium access protocol provides the user with a service for sending best effort messages, which are globally deadline scheduled. The global deadline scheduling is a mechanism that is built into the medium access protocol. No further software in upper layers is required for this service. Other networks may have upper layer protocols added to them to give them better characteristics for real-time traffic, but it is hard to achieve fine deadline granularity by using upper layer protocols. This feature in TCMA is indication that the network may be used to send traffic in an earliest deadline first (EDF) fashion, supporting hard guarantees by using a schedulability analysis.

Real-time services in the form of best effort messages, as mentioned above, guarantee seeking messages, and real-time virtual channels (RTVC) are supported for single destination, multicast and broadcast transmission by the network. There is also a service for non real time messages. The network also provides services for parallel and distributed computer systems such as short messages, barrier synchronisation, and global reduction. Support for reliable transmission service (flow control and packet acknowledgement) is also provided as an intrinsic part of the network. These services are all possible, but this report will focus on the analysis of the distributed global deadline scheduling of packets. More information about other services can be found in [3].

The network with the proposed protocol is best suited for LANs and SANs (system area networks) where the number of nodes and network length is relatively small. This is important since the propagation delay adversely affects the medium access protocol. Examples of suitable applications are embedded systems (e.g., for use as an interconnection network in a radar signal processing system) and cluster computing.

The physical ring network is divided into three rings or channels (see Figure 1). For each fibre ribbon link, eight fibres carry data, one fibre is used to clock the data, byte by byte, and one is used for the control channel. Access is divided into slots like in an ordinary TDMA (Time Division Multiple Access) network. The control channel ring is dedicated for bit-serial transmission of control packets, which are used for the arbitration of data transmission in each slot. The clock

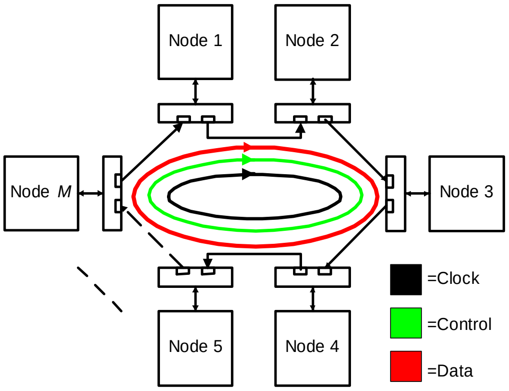


Figure 1: A Control Channel based Fiber Ribbon Pipeline Ring network.

signal on the dedicated clock fibre, which is used to clock data, also clocks each bit in the control packets. Separate and dedicated clock- and control fibres simplify the transceiver hardware implementation in that no clock recovery circuitry is needed [5]. The control channel is also used for the implementation of low-level support for barrier-synchronisation, global reduction, and reliable transmission.

The ring can dynamically (for each slot) be partitioned into segments to obtain a pipeline optical ring network [2]. Several transmissions can be performed simultaneously through spatial bandwidth reuse, thus achieving an aggregated throughput higher than the single-link bit rate (see Figure 2 for an example). Even simultaneous multicast transmissions are possible as long as multicast segments do not overlap. Although simultaneous transmissions are possible in the network because of spatial reuse, each node can only transmit one packet at a time.

A disadvantage with the CC-FPR protocol presented in [1] is that a node only considers the time constraints of packets that are queued in it, and not in downstream nodes. As an example (see Figure 2), Node 1 decides that it will send and books Links 1 and 2, regardless of what Node 2 may have to send. This means that packets with very tight deadlines may miss their deadlines. The novel network presented here does not suffer from this problem.

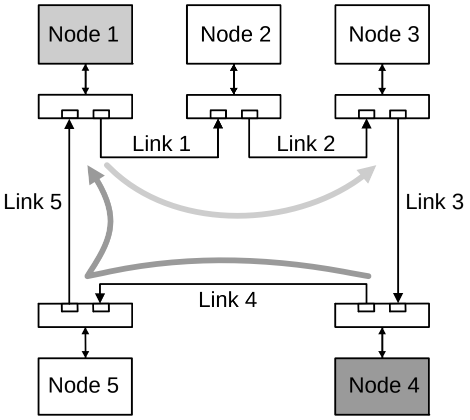


Figure 2: Example where Node 1 sends a single-destination packet to Node 3, and Node 4 sends a multicast packet to Node 5 and Node 1.

The clock strategy that is assumed for this and previous papers cause some major difficulties when analysing the worst case. Very briefly, the difficulties occur because traffic parameters affect the performance of the network. The problem is caused by the fact that the clocking strategy requires that the network clock signal is terminated at the master and therefore that no transmissions may take place across the master. This will be further explained below.

The rest of the paper is organised as follows. Section 2 briefly presents the novel medium access protocol. Section 3 presents the schedulability analysis and shown what problems have to be solved and their solutions. Consequences of the solutions are discussed. Section 4 presents the conclusions of the report and gives a recommendation for a better solution. Appendix 1 shows an example of the global queue anomaly in the network.

2 Two-cycle medium access protocol

The basic idea of the two cycle method is as follows. First, one request for transmission from each node in the network is collected into the node that is currently master. The list of requests in the master is sorted according to priority. This list reflects the state of the messages waiting to be sent in the network. It can be seen as a global queue of messages that is uppdated during each slot, although as restriction is that there is only one request per node in the system. The master then decides which request(s) can be granted and then notifies all nodes of the outcome of the request. Due to the pipelining possibility of the network, several requests may be granted at once depending on the requests themselves and if the schedulability analysis framework allows for this. Further details of the medium access protocol are found in [2]. For the follow discussions it is assumed that access to the network is divided into slots. Slots are organised into cycles. Each cycle consists of one slot for each node (*N* slots), so that each node is master once during a cycle.

The idea behind the medium access protocol is to enable distributed global deadline scheduling through the construction of a global queue in each slot. Thus in each instant the most urgent message in the network would be sent. This would enable earliest deadline scheduling of messages and the capability to provide interesting user services.

3 Schedulability analysis of TCMA

This section will discuss offline schedulability analysis of periodic messages, referred to as message connections. Online schedulability analysis is put out of the scope of the current discussion, but can be easily implemented. For the discussion it is assumed that a "central" node has information about message connections in all nodes and has the responsibility to devise a feasible schedule and distribute it before runtime.

The distinction between online and offline is how message connections can be accepted/declined and taken out of the network. With offline schedulability analysis all analysis of connections must be handled before the system is in runtime. When in runtime no modifications to the set of connections may take place. However with online schedulability analysis new connections may arrive at any time also after the system is in runtime. Online schedulability analysis can be further categorised by where the analysis of new connections take place. They may take place either in a central node that has complete knowledge of all connections or in a distributed fashion where all nodes have equal knowledge and responsibility.

The goal of the discussion is to show what problems arise when analysing the network. The aim of the research was to provide sufficient evidence that that studied ring network protocol can be regarded as an implementation of the earliest deadline first (EDF) policy and therefore, with a given heuristic, hard guarantees can be given to traffic within a connection. It will be shown that the worst case determinism is very pessimistic and hardly of any use.

The approach is to regard the network's function to be similar to uni-processor schedulability analysis. This suggests how the network should be transformed.

The clock anomaly, spatial reuse and other problems of the network and their consequence are explained in Section 3.1. Section 3.2 explains which solutions and constraints that are applied for the transformation to be valid. Section 3.3 presents the consequences of the constraints. The section also presents a worst case feasibility test similar to the EDF feasibility test. Worst case scenarios are discussed in section 3.4. Section 3.5 presents the final discussion.

3.1 The "problems" encountered

The problems found in the network are attributed to data dependence, i.e., the performance depends on message parameters, and that spatial reuse may cause priority inversion. Priority inversion in processor scheduling occurs when a higher priority task is blocked by a lower priority task. In the network a similar effect concerns messages.

The network has a dedicated fibre that contains the clock signal. The signal is generated at the current master node, travels *N* nodes and arrives back at the master. Nodes may transmit messages at most $N - 1$ hops since transmitting back to itself is not meaningful.

The clocking strategy of the network comes from previous research [4]. In this network, no feedback is required back at the master node. Therefore the nodes may be designed to take advantage of this. To simplify the design of the clock synchronisation and data channel in each node, the clock signal coming in to the master does not pass through to the outgoing clock signal. The same restriction applies for the data channel. The result is that no node can transmit past the master resulting in an anomaly that is dependent on the destination of the message queued for transmission, and therefore difficult to account for without statistics of the messages. See Figure 3 for such an example. This is the root of the

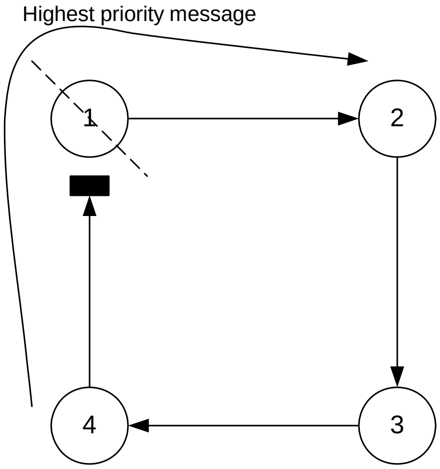


Figure 3: The break in the clock strategy. Node 1 is master. The clock signal is broken off before the master. Node 4 has a highest priority message destined to node 2 that cannot be sent during this slot.

		Nbr. Hops			
		4	3	2	1
	i+0	ok	ok	ok	ok
	i+1	no	no	no	ok
	i+2	no	no	ok	ok
Master	*i*+3	no	ok	ok	ok
	i+4	ok	ok	ok	ok

For N=5

Table 1: Shows node *i*'s opportunities for transmission for varying number of hops to the destination and with different nodes as master, i.e., for different location for the clock anomaly. E.g. *i*+0 implies that node *i* is master. As can be seen, in the worst case a message has only two opportunities per cycle irrespective of the

Item	Problem	Comment
Clock anomaly	priority inv, non determinism	Unsuitable clock strategy for MAC with feedback
TR	PI	See below
SR	PI	Strategy to increase performance by reuse
Global req. queue limit	PI	Because of wish to have SR
Feasibility requirement	PI, non determinism	Because of wish to have SR

Table 2: Table of problems. SR=Spatial reuse, TR=Temporal reuse, PI=Priority inversion, MAC=Medium Access Control

clocking strategy problem, also referred to as the clock anomaly. The new medium access protocol, TCMA, does require feedback and so the old clocking strategy is not suitable. In Figure 3 the problem is illustrated. In the current slot Node 4 has the highest priority message in the system. However the message may not be sent in the current slot because the clock is interrupted at Node 1. Therefore Node 4 has to wait until the next slot when Node 2 is the master and the clock break occurs there instead.

The role as master is cycled equally around the ring, so which message that is affected by the anomaly also depends on the distance to the master in the current slot. The effect of the anomaly is that messages can sometimes not be transmitted. In the current version of TCMA, the system will instead try to transmit a message possibly of lower priority and the result, seen from the higher messages immediate point of view, is priority inversion. Thus the anomaly indirectly causes priority inversion. The clock anomaly is depicted in Table 1. The general formula for the ratio between the number of possible opportunities to send and the total number of opportunities is as follows:

$$\frac{N(N-1)-\sum_{i=1}^{N-2} i}{N(N-1)} \tag{1}$$

As can be seen in Table 1, there are 14 possibilities out of 20 total. Equation 1 can be simplified into:

$$\frac{N+2}{2N} \tag{2}$$

Bandwidth reuse is a function of the network, which enables multiple simultaneous transmissions of messages, providing that the required network segments do not overlap. Reuse is present on two levels, on the cycle level (temporal reuse) and on slot level (spatial reuse). With the former, several transmissions may be possible during one cycle because each message may be at most one slot in length. With the latter, transmissions on separate segments may take place simultaneously during the same slot (see Figure 2). Spatial reuse makes analysis of the network difficult since the degree of reuse depends on the message source and destination and is not known more than statistically. Further more reuse on both levels may also imply priority inversion since in some situations the highest priority message cannot be sent. More on this in the following sections.

Arbitration with the TCMA protocol is based on collecting requests for transmission from the nodes in each slot. Each request is the highest priority message in each node that is feasible concerning the clock anomaly. All the requests are collected in the current master node, which also has the task to appropriately grant a request(s) based on priority and feasibility. In the current version of the TCMA protocol, the request that each node makes must be the highest priority and feasible message. Also, if the highest priority message in a node is unfeasible during the following slot then it will not be requested. Instead, the node will request to send the next highest priority feasible message and so on. This is a source for priority inversion, since a lower priority but feasible message would be requested instead of the higher priority message. It occurs because of the clock anomaly. The idea behind this is to increase performance.

It is also easy to show that the network is prone to a priority inversion anomaly caused by the fact that the global queue composed of (only) N independently feasible requests only gives a limited scope of the network state. The reason for limiting the number of requests is the limited size of the request packet in the arbitration method [2]. An example of this priority inversion anomaly is shown in Appendix 1.

The overall problem with the network seems to be priority inversion and the lack of determinism in performance. The latter problem stems from the fact that the degree of spatial reuse is dependent on the parameters of the transmitted message and therefore can only (without high restrictions on the traffic behaviour and/or a much more complex analysis) be accounted for statistically, e.g. an average case. As stated previous, the former problem, priority inversion, can be regarded at slot level and at cycle level. For priority inversion occurring at slot level, it must be determined weather this will affect the message at cycle level, remembering that deadline is enumerated in cycles. It will be concluded in the following chapter that priority inversion at slot level can be accepted in some cases. The lack of determinism is tackled by setting a more pessimistic worst case bound for performance. An overview of the "problems" in the network can be seen in Table 2.

3.2 Solutions

This section present solutions of how the network can be regarded and constrained to support determinism. Basically, the main problems that have to be addressed are: The global queue function of the network and reuse of network capacity, which functions on two levels.

When the master collects requests for transmission from each node, the nodes respond with the highest priority feasible message. As seen in the previous section, this is a source for priority inversion. A simple solution is for the node to always request its highest priority message only, even if this message is unfeasible. Requests in the global request queue are sorted by priority, etc. If the highest priority message in the global queue is unfeasible in the current slot, then all other message will be blocked. In this way priority inversion is avoided. If the highest priority message in the global queue can be sent in the current slot, then the next highest priority message may also be granted permission to be sent. If this message is feasible then the search continues, in priority order, in the global queue. If it is not feasible then the search ends. Thus, there is no spatial reuse at slot level, unless the highest priority message is feasible, and therefore no priority inversions at slot level either.

Restriction on the MAC	Maximum network throughput
No TR on cycle level, no SR on slot level	$1/N$
TR on cycle level, no SR on slot level	N/N
TR on cycle level, SR on slot level	N^2/N

Table 3: TR = Temporal Reuse, SR = Spatial reuse. The throughput also depends on traffic parameters.

The other problem with the global request queue concerns the limited scope of the state of the network that only *N* requests can give. This problem is closely associated with spatial reuse on slot level, since it occurs when granting several requests during one slot (see also Appendix 1: Example of global queue anomaly). The simple solution is to only grant the highest request, i.e. that only one request per slot is granted and only one packet can be sent at most.

Priority inversion because of (temporal) reuse at cycle level can happen in the proposed network if e.g. the highest priority message in a node is unfeasible because of the clock anomaly. With reuse at the cycle level the next message is considered as a request and so forth, until the highest priority feasible message is found. This action saves network capacity but implies priority inversion. A solution to the problem is to require the highest priority message to always be sent ahead of any other message. This also assumes that each node always sends its highest priority message as a request, regardless of current feasibility. The solution implies that there will be empty slots where no transmissions take place because the highest priority message is unfeasible. Thus, as stated above, there will be no reuse at cycle level, unless the highest priority message is feasible, and therefore no priority inversions at cycle level either. This means that if the highest priority message if feasible, then we can make use of reuse and consider other feasible messages from the global request queue also during the rest of the cycle. The solution discussed is rather pessimistic because it implies that some slots will be empty and not reused.

a:

Slot	0	1	2	3	4	5	6	7	8	9	10	11
Master	0	1	2	3	0	1	2	3	0	1	2	3
Cycle	0	0	0	0	1	1	1	1	2	2	2	2
Max dist 0	3	1	2	3	3	1	2	3	3	1	2	3
Max dist 1	3	3	1	2	3	3	1	2	3	3	1	2
Max dist 2	2	3	3	1	2	3	3	1	2	3	3	1
Max dist 3	1	2	3	3	1	2	3	3	1	2	3	3
MC00	🟨								🟨			
MC01				🟨								
MC10		🟨										
MC20			🟨									
MC30							🟨					

b:

Slot	0	1	2	3	4	5	6	7	8	9	10	11
Master	0	1	2	3	0	1	2	3	0	1	2	3
Cycle	0	0	0	0	1	1	1	1	2	2	2	2
Max dist 0	3	1	2	3	3	1	2	3	3	1	2	3
Max dist 1	3	3	1	2	3	3	1	2	3	3	1	2
Max dist 2	2	3	3	1	2	3	3	1	2	3	3	1
Max dist 3	1	2	3	3	1	2	3	3	1	2	3	3
MC00	🟨								🟨			
MC01				🟨								
MC10						🟨						
MC20							🟨					
MC30								🟨				

Table 4: Shows that the blocking time, e.g. for MC30, is not increased because we allow (temporal) reuse on cycle level. Max dist. *i* is the maximum distance that node *i* may send during that slot. MC*iy* is message connection *y* in node *i*. Table 5: In case a, PI is allowed and therefore messages from node 2 and 3 in between the highest priority messages from node 1. In case b, PI is not allowed which implies that nothing may be transmitted at Y because the message from node 1 is infeasible in these slots. Notice also that the response time (in cycles) for these highest priority messages are the same regardless of PI or not.

It is easy to show that the maximum blocking time of a highest priority message is the same regardless of weather or not transmissions take place in the slots in between, see Table 4 and 5. Further, it is easy to show that a highest priority message will always be sent at some point during the cycle providing that the network is not overloaded. This suggests that temporal reuse does not affect the highest priority message. However, this is not further considered.

If message deadlines are enumerated in whole cycles, and temporal reuse is not allowed, then the problem of priority inversion at cycle level is overcome. This is because at most the highest priority message would be sent during a cycle. However, this is a pessimistic assumption.

The worst case is assumed that the message will be sent at the end of the cycle. This constraint is not too serious since we also assume that the periods of the messages are all greater than the cycle length. E.g. assume a slot time of 2 μs, a network of 16 nodes and 1 slot per node per cycle. This gives a cycle length and minimum message period of 32 μs. A message can arrive at a node during any time during the cycle but the deadline always has to be enumerated in whole cycles. Thus, regarded in this way, priority inversion is not a problem.

3.3 Constraints that must be applied and their consequences

From the discussion in Section 3.1 and 3.2, we now present the guidelines for the schedulability analysis. The schedulability analysis assumes the worst case of no case spatial at all even if it can appear at run time. This makes the analysis pessimistic in that the network is utilised poorly. How pessimistic the analysis is in practice will be discussed later.

EDF requires that message's priority change according to their relative deadline. This function is already implemented in the network because in each slot it is always the highest priority message in the network that is transmitted. However, as was explained previously, the highest priority message may not be feasible. The previous sections also explained how to overcome / disregard these problems.

The goal for this analysis is to be able to model the network according to the same framework laid out for rate monotonic and earliest deadline [6]. For this to be possible there are a number of requirements. One of these is that it must be provable that the worst-case response time of a message, measured in cycles, is highest when all messages have the same release time, at time zero. This is true only if deadlines are enumerated in whole cycles and no priority inversion is permitted.

So far we have discussed the effects of the limitations that constraints the network will have on the performance. Now we will consider the effects of different traffic patterns. The worst case scenario is when all traffic is generated in one single node and that all the traffic is destined $N-1$ hops. If temporal reuse is allowed on cycle level, then the maximum usable capacity of the network is only $2 / N$. This is because, assuming that all nodes generate traffic with equal parameters ($N - 1$ hops), there will always be two opportunities per cycle to transmit a message destined $N - 1$ hops, see Table 1. If temporal reuse is not allowed then the maximum usable capacity of the network is only $1 / N$. Transmission of the message, destined $N - 1$ hops, is always possible when the node is master (once per cycle). The latter bound will be used when performing to schedulability analysis. See also Table 3.

The restrictions to the protocol discussed in this section are sufficient to fit the network protocol within the EDF framework for uni-processors. In the TCMA network the upper bound is motivated slightly differently. Thus the schedulability of message set M can be proved with Equation 3 below. M is a set of all message connections in all nodes. $M = \bigcup_{\forall i}\bigcup_{\forall y} M_{iy}$, where y is a message connections in node i. Each connection is composed of three parameters associated to it: $M_{iy} = (P_{iy}, e_{iy}, D_{iy})$, where P_{iy} is the period of connection y with source i. e_{iy} is the message size in slots of connection y with source i. D_{iy} is the destination of connection y with source i.

$$\sum_{\forall i}\sum_{\forall y}\frac{e_{iy}}{P_{iy}} \le {}^{1}\!/_{N} \qquad (3)$$

The schedulability test is sufficient but not necessary. This means that if the test yields a positive answer, then the message set is schedulable. However, a negative answer does not imply that the set of message connections are unschedulable. The restrictions make the schedulability test rather pessimistic and if reuse is allowed more messages could use the network although these cannot be guaranteed with the current model.

Assumption about the traffic	Max network throughput
All traffic $N - 1$ hops, from one node	${}^{2}\!/_{N}$
All traffic $N - 1$ hops, from all nodes	${}^{N}\!/_{N}$
All traffic average number hops, from all nodes	${}^{2N}\!/_{N}$
All traffic 1 hop, from all nodes	${}^{N^2}\!/_{N}$

Table 5, Note: For the first case, the node can send two messages per cycle, because it has two opportunities per cycle, see also table 1. The above quotes for throughput are when there are no restrictions to the MAC-method. Throughput is quoted in slots per cycle, assuming that there are *N* slots in a cycle. The throughput quoted is the total throughput of the network, not the throughput perceived by the individual node.

3.4 Discussion of the worst case scenario

There are two aspects of the worst case throughput in the network. First, there is the question of what assumptions are made about traffic at the nodes. This is important since network access availability is greatly affected by the destination of the traffic because of the clock anomaly. Secondly, the performance of the network depends on what restrictions are applied to the medium access scheme. Various restrictions to the scheme must be applied to overcome problems such as priority inversion. Table 3 and 4 present these different assumptions and their consequences for the throughput.

As can be seen in Table 4, when all required restrictions are in place, it leads to low utilisation of the network. Table 4 shows how greatly the traffic pattern affects performance.

3.5 Discussion

To be able to use a uniprocessor schedulability analysis framework with the network at hand, we need to apply several constraints to the network, as discussed in previous sections. The problem with network schedulability analysis is that any message in any node does not have access to any destination node in any time slot, if analysis is regarded at slot granularity. This is a very serious problem and is “masked” by regarding analysis at cycle granularity. This way each node is guaranteed an opportunity to transmit unrestricted during each cycle. A uniprocessor scheduling assumption is that the maximum response time of tasks occur when the release time of the tasks are all 0. This is not the case in the network without the cycle granularity constraint.

However, the constraints to the network imply that only a small part of the network capacity can be scheduled. The worst case capacity is $1/N$ per cycle and therefore EDF-scheduling, under these assumptions can not do any better than round-robbin scheduling where a node gets to send when it is master.

4 Conclusions

The network, as it is currently implemented, makes real time analysis complicated and overly pessimistic. Therefore we propose that the network be reimplemented in such a way that many of the “problems” of the current implementation are removed. The most important problem to get rid of is the clock anomaly. A different clock strategy is needed where there are no restrictions on sending any message during any slot, any number of hops.

Acknowledgement

This work is part of M-NET, a project financed by SSF (Swedish Foundation for Strategic Research) through ARTES (A Real-Time network of graduate Education in Sweden).

References

[1] M. Jonsson, “Two fibre-ribbon ring networks for parallel and distributed computing systems,” *Optical Engineering,* vol. 37, no. 12, pp. 3196-3204, Dec. 1998.

[2] C. Bergenhem, M. Jonsson, and J. Olsson, "Fiber-ribbon Pipeline Ring Network with Distributed Global Deadline Scheduling and Deterministic User Services," *Proc. of the Workshop on Optical Networks held in conjunction with International Conference on Parallel Processing 2001,* Valencia, Spain, Sept. 3-7, 2001.

[3] Jonsson, M., C. Bergenhem, and J. Olsson, "Fiber-ribbon ring network with services for parallel processing and distributed real-time systems," *Proc. ISCA 12th International Conference on Parallel and Distributed Computing Systems (PDCS-99)*, Fort Lauderdale, FL, USA, Aug. 18-20, 1999, pp. 94-101.

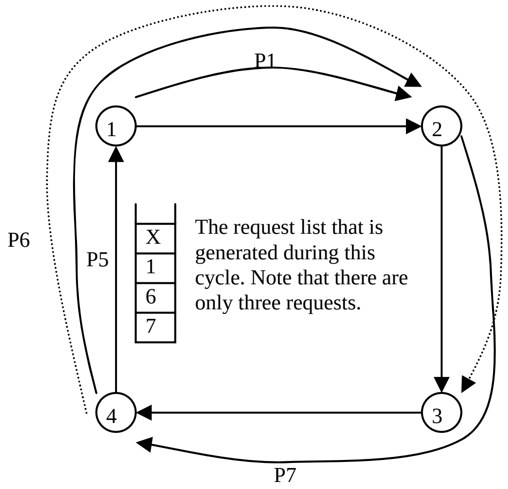
[4] M. Jonsson, “Two fibre-ribbon ring networks for parallel and distributed computing systems,” *Optical Engineering,* vol. 37, no. 12, pp. 3196-3204, Dec. 1998.

[5] C. Bergenhem, “A demonstrator for a CC-FPR network,” *Techical report 0010, School of Information Science, Computer and Electrical Engineering (IDE), Halmstad University, Sweden,* Feb. 2000. In Swedish.

[6] C. L. Liu and J. W. Layland “Scheduling algorithms for multiprogramming in a Hard-Real-Time Environment,” *Journal of the ACM*, 20(1):46-61, 1973

5 Appendix 1: The global queue anomaly

This is an example of a priority inversion anomaly that occurs because of the limitations of the global queue and because of the strategy for granting requests. The priority of the message is noted above the arc that denotes the source and destination of the message. Higher number implies higher priority and a more important message. Thus the most urgent message has priority number 7 and is sent from node 2 to node 4.


P1
1
2
3
4
P6
P5
X
1
6
7
The request list that is generated during this cycle. Note that there are only three requests.
P7

This is the distribution packet that all nodes receive at the end of the arbitration phase. '1' and '0' in the respective node's field denote that the request was granted or not granted respectively.

Distribution list:	1	1	0	0
Node id:	1	2	3	4

- The node with the requests decide that the P6 message could not be sent because P7 has higher priority and has overlapping segments.
- The next request in the list, P1, is granted instead.
- If node 4 had requested P5 instead of P6, then P5 would have been feasible together with P7. This is not possible since there may only be one request per node.
- Instead the result is that P5 is "priority inversed" by P1.
- It is arguable that it may have been better to have P5 be sent before P6 than P1 before P6, but both cases are still priority inversion because a higher priority message is blocked by a lower priority message.
- The only solution in this case is to only grant P7 the right to send.
- The anomaly is caused by the request list not having "complete picture" of the network and because of the spatial reuse strategy.