Using a more complex analytical comparison, we found out that the ISO implementation will outperform the TMS, and this difference will increase with the number of stacking and transmitting stations. Simulating the token ring network produces the same result. Because of very small differences between implementations and thus necessity to simplify very difficult analytical problem, in mathematical evaluation we considered one concrete situation on the ring, but only simulation could give us the right answer on the problem. Other question is, which result is most costly. Of course, computer simulation is a more expensive way to find a better implementation, on the other hand, it is often closer to reality. In this example the mathematical cost-effective evaluation shows differences between implementations, but by simulation we determine how high these differences are. Another problem occurs, when the simulation model only weakly corresponds to the real implementation. In this situation the simulation can fail because of behavior differences in the real implementation. From this perspective, simulation models closely tied to the implementation scheme, although more complex, can provide on advantage in pre-implementation simulation and simplified automatic system implementation [7]. Such simulations can select the best approach from different implementation proposals or evaluate the behavior of the system, when the implementation alternatives are dynamically changed based on actual network parameters. Similar media access control proposals are in [8], [9], [10].

References

[1] TMS380 Adapter Chip set User's Guide, SPWU001D, Revision F, Texas Instruments 1988.

[2] ISO Informatiom processing systems – Local area networks – part 5: Token ring access method and physical layer specification, DP 8802/5 1986.

[3] W.Bux, "Local Area Subnetworks: A Performance Comparison", *IEEE Transactions on Communications,* Vol. COM-29, No. 10, 1981.

[4] U.Killat, H.A.Muscate, B.Wolfinger, "A Performance Analysis of The ISO 802.5 Token Ring Protocol", *Philips J. Res,* No. 43, 1988.

[5] Hammond J. L.,O'Reilly P.J.P., "Performance Analysis of Local Computer Networks", Addison-Wesley, 1986.

[6] Mak et all, "Design of IO for System Token Ring", *Proc. Dni Novej Techniky Tesla VUST,* Elektrotechnicka Spolocnost Praha 1990. (in Czech).

[7] A.S.Krishnakumar, B.Krishnamurthy, K.K.Sabnani, "Translation of formal protocol specification into VLSI designs", in *Protocol Specification, testing and verification, VII,* Elsevier Science Publishers B.V., North Holland, May 1987, pp. 375-390.

[8] H. Durrett, "A Milestone in Network Systems' History: Hyperchannel-DX", *Proc. Nexus XIX Spring conference, Minneapolis,* April 1988.

[9] Gerla Mario, Guang-Shing Wang, Rodrigues Paulo, "Buzz-Net:A Hybrid Token/Random Access LAN", *IEEE Journal of Selected Areas in Communications,* Vol. SAC-5, No. 6, July 1987.

[10] Amit Bhargava, James F. Kurose, Don Towsley, "A Hybrid Media Access Protocol for High Speed Networks", *IEEE Journal of Selected Areas in Communications,* Vol. SAC-6, No. 6, July 1988.

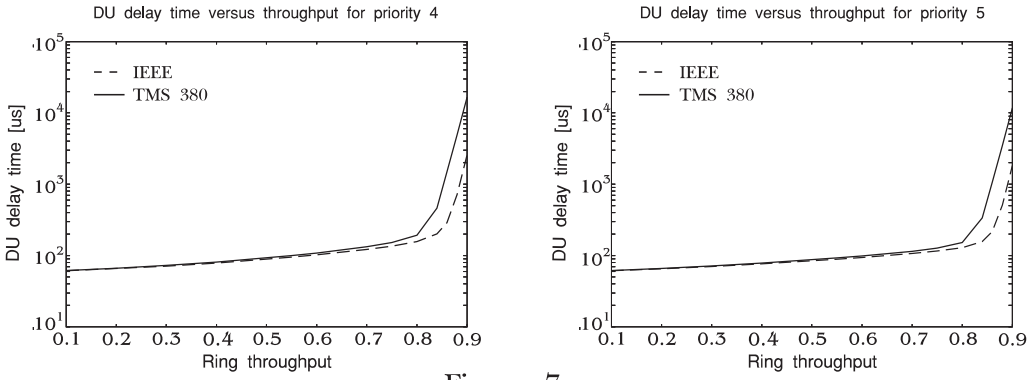

Figure 7

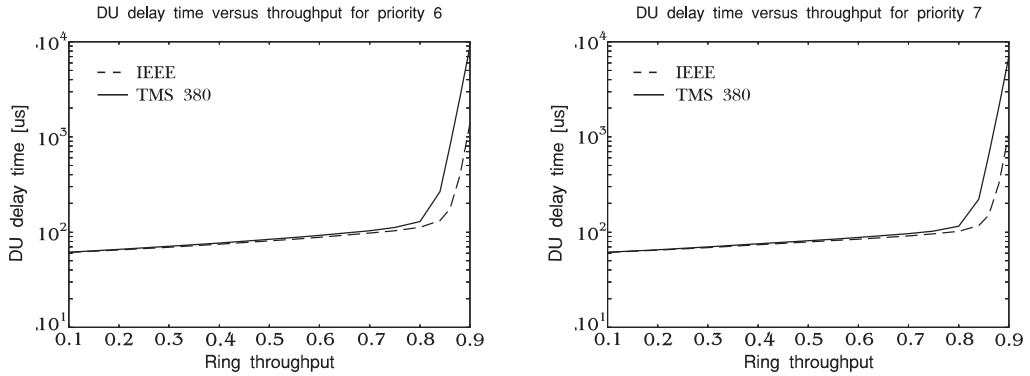

Figure 8

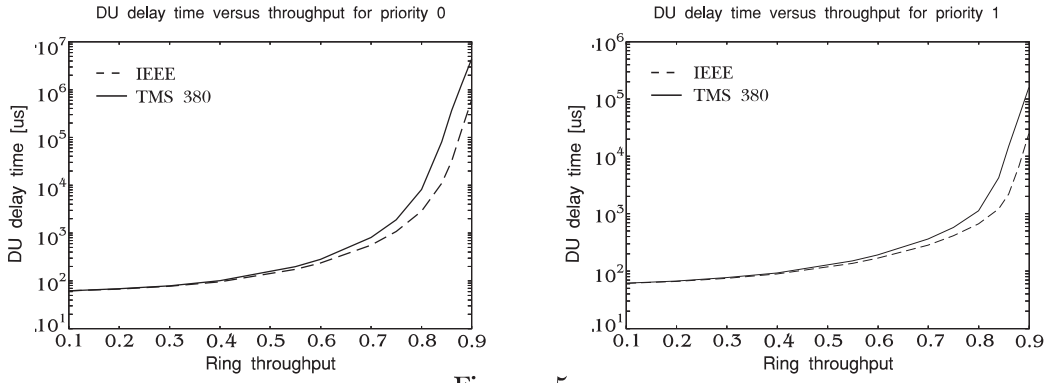

Figure 5

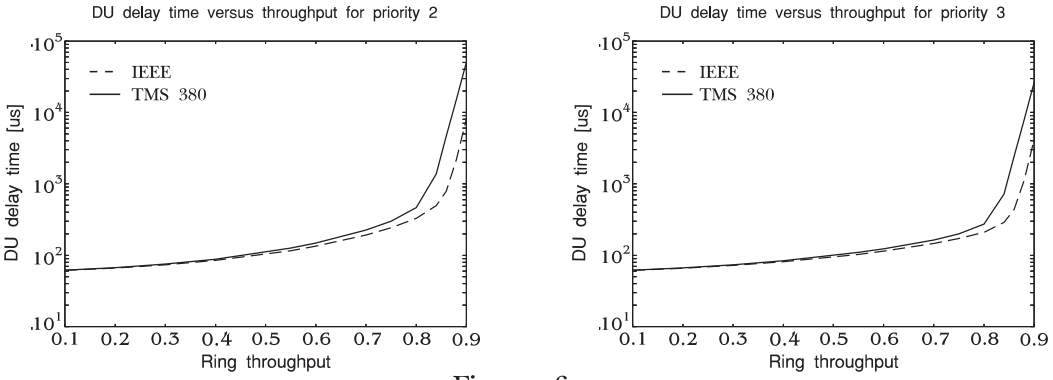

Figure 6

Table 7 – priority 6			
ring	*delay time* $[\mu s]$		*ratio*
load	*TMS 380*	*ISO*	*ISO/TMS*
0.8	129.3	112.5	87%
0.84	265.7	132.3	50%
0.86	802.0	175.8	22%
0.88	2614.0	394.4	15%
0.9	8912.9	1382.2	16%

Table 8 – priority 7			
ring	*delay time* $[\mu s]$		*ratio*
load	*TMS 380*	*ISO*	*ISO/TMS*
0.8	115.5	102.3	89%
0.84	220.9	117.3	53%
0.86	643.6	150.5	23%
0.88	2068.8	322.8	15.6%
0.9	7012.6	1090.2	15.5%

6 Conclusions

We have showed four different comparisons between two token ring local area network priority implementations. Using the first simple look at the problem it would appear that the TMS implementation should give better performance than the ISO implementation because of smaller delay in the token's priority decreasing procedure. The next comparison used network utilization at the higher priorities, where the stacking station increases the ring delay; here we derive the opposite result from the previous case, because in this case we evaluate the whole network – as oposite to the simplified comparison, where we considered the behavior of only one station. Our goal was to show how the first simple solution to the problem resulted in the wrong deduction and can lead to bad implementation choices.

Comparing difference of TRT_{tms} and TRT_{iso} we received a value of $136T$ which is comparable with the frame transmission time. Thus token capturing and then frame transmitting in the TMS implementation are delayed by approximately one frame. From this we can consider about 50% degradation of the throughput for the TMS implementation with comparison to ISO. The simulation results show a 35% ratio at 84% ring utilization which is not a negligible value. Such a high difference follows from the higher number of stacking and transmitting stations on the ring. Differences between analytical and simulation outputs result from the fact that analytical evaluation don't consider about possibility to transmit frame by more than $p = 9$ stations during one token rotation.

Table 1 – priority 0

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0.1	61.8	61.6	99%
0.2	68.2	67.3	98%
0.3	78.9	76.9	97%
0.4	100.4	95.5	95%
0.55	197.5	173.2	87%
0.6	281.1	233.9	83%
0.7	797.5	5.55.0	70%
0.75	1881.7	1068.6	57%
0.8	8072.4	2849.1	35%
0.84	80139.0	10840.0	14%
0.86	368740.0	31636.0	9%
0.88	1296900.0	142490.0	11%
0.9	4393700.0	611310.0	14%

Table 2 – priority 1

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0.75	577.4	415.7	72%
0.8	1133.0	670.1	59%
0.84	4271.4	1227.7	29%
0.8	15095.0	2208.3	15%
0.88	48848.0	6743.5	14%
0.9	160900.0	25139.0	16%

Table 3 – priority 2

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0.8	464.6	327.4	71%
0.84	1372.2	493.3	36%
o.86	4641.2	788.5	17%
0.88	15152.0	2177.1	14%
0.9	50380.0	8082.1	16%

Table 4 – priority 3

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0.8	273.0	209.8	77%
0.84	716.4	287.9	40%
0.86	2381.2	432.1	18%
0.88	7848.3	1119.3	14%
0.9	26242.0	4150.7	16%

Table 5 – priority 4

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0.8	192.9	157.0	81%
0.84	461.3	201.2	44%
0.86	1488.1	287.6	19%
0.88	4933.1	715.5	15%
0.9	16627.0	2591.2	16%

Table 6 – priority 5

ring load	*delay time* $[\mu s]$		*ratio*
	TMS 380	*ISO*	*ISO/TMS*
0,8	152.1	128.8	85%
0.84	335.6	157.7	47%
0.86	1052.1	216.1	21%
0.88	3467.2	510.5	15%
0.9	11724.0	1824.0	16%

4 Mbit/sec ring frequency; 16 bit addresses (SA – source address, DA – destination address). Each station transmitted DUs on all of 8 (0 ... 7) priority levels. All traffic was equally distributed among all priorities and all stations: each station contributed to the aggregate traffic by 1/10; each priority in the station contributed to station traffic by 1/8. Traffic was generated using a Poisson DU interarrival time distribution. A fixed DU length of 8 bytes resulted in a total frame length of 184 bits. This length is higher than total ring latency – 60 bits for the stated ring configuration. Each transmitting station addresses frames to its physically opposite station. Exhaustive type of station service was selected. Separate simulation runs were run long enough to achieve a minimum $\pm 2\%$ relative precision of measured values. Thirteen simulations were run with increasing ring utilization for both types of priority function implementation.

5.3 Simulation results

Figures 5 to 8 show graphical representation of the simulation results. Graphs evaluate the dependency of the total DU delay time on the ring utilization for each priority level. Dashed and full curves represent the DU delays of the ISO and TMS implementations respectively. The x axis – the ring throughput – uses a linear scale while the y axis – DU delay – uses an exponent scale. We use this scale because of huge differences between implementations at high ring throughput rates. Curves for different priorities have similar shapes and differ in the DU waiting time values. These values decrease for higher priorities because the higher priority DUs are transmitted as first. For each priority, the DU delay increases slowly for low ring throughput because less stations use the token and thus the waiting time for the token is smaller. As throughput increases, during one token rotation on the ring, many stations capture the token and transmit data, which increases the waiting time for the token.

For priority 0, the ring throughput – where the DU waiting time difference is not small – is about 0.5, while for priority 7, this throughput is about 0.8; at these points the differences between the two implementations becomes larger. This is caused by Priority Control State Machine delay in the TMS implementation which delays each transferred frame by $9.5T$. Tables 1 to 8 contain DU's waiting times for concrete ring loads. The first column is ring load and next two columns are DU waiting times for TMS and ISO in microseconds. The last column contains the ratio between ISO and DU waiting times calculated in percent. While for 50% ring utilization the differences are small, for 90% ring utilization the ratios are about 15%!

In the previous intuitive comparisons we could predict only small differences in network performance because of a simplified evaluation. Let's compare analytical evaluation with the results of the simulation. Use the values from the simulation for computing the result of equation (1) for worst case:

number of stacking stations	$k = 7$
number of transmitting stations	$p = 9$

We use equation (1) because the transmission time of the frame with the length of 184 bits comply with condition for this equation.

$$TRT_{tms} - TRT_{iso} = [9.5(k + p) - 16]T$$

$$[9.5(7 + 9) - 16]T = 136T$$

5 Comparison of the two priority system implementations and performance evaluation using simulation

Analytical analysis of the ring gives us some knowledge about operational characteristics of both implementations. As a token ring is a stochastic system, queuing theory can be used to obtain a more precise analysis of system behavior. In this case it is a difficult problem, because of the very small differences between implementations. Therefore we are making comparisons using a discrete event simulation model.

5.1 Simulation model description

To intercept small differences in the priority mechanism implementations, we have developed simulation model which describes the token ring behavior with the precision of one bit period. The model implemented by SIMSCRIPT II.5 simulation language uses events planning as the tool for expression of time dependencies. These events are as follows: An arrival of the token into a station, the beginning of data frame transmission from station, an end of data frame transmission from station, an arrival of data frame to station and an arrival of DU into station's output buffer. The events operate over static and dynamic data structures describing network configuration and its status. Each station is represented by permanent entity with attribute *token holding timer* THT and three sets for implementation of the output priority buffer for DU and stacks for *old* and *new* priorities. For each priority level the permanent entity is used for definition of packet inter-arrival time, packet length, and destination address distribution with the list of stations using that priority level. The *message generation* event – planned for each station and priority level – creates DUs with parameters (length, destination address) according to required distribution and plans itself again after the time given by interarrival time distribution. The illusion of the token (frame) rotation is achieved by planning of a *token (frame) arrival* event giving it next station address (and a pointer to the message) as parameters. *Token arrival* event checks weather DU with priority higher or equal to the token priority exists in the station's buffer; if such DU exists, *token arrival* event plans *frame arrival* event for next station. If no DU is waiting for transmission, the *token arrival* event plans the same event for next station. Moreover, it lowers priority of the ring if this is necessary or changes priority reservation field of the token through the station's request. The *frame arrival* event changes priority reservation field of the frame if needed and plans *frame arrival* event for next station. If the *frame arrival* event occurs on the source station, instead of *frame arrival* event the *token arrival* event for next station is planed. *Frame arrival* event occurred on the destination station collects the DUs statistics.

Ring configuration and monitoring actions were not considered for reasons of simplicity. During simulation runs the program collects statistics of ring utilization by each priority and for whole network. Mean and maximal waiting times for token and mean and maximal transfer times for DUs on each priority level are computed. Comparative simulations for both priority mechanism implementations were run on Sun-4 workstation.

5.2 Simulation conditions

The simulation of the token ring network had the following parameters: Ten stations with 1 bit station latency; 375 meter interstation distance; approximation of signal velocity to speed of light;

t_{x_tms}	is	9.5-bit token delay caused by priority decrease in station B.
t_{fm_tms}	is	analogous to t_{fm_iso}.
t_{nt_tms}	is	9.5 token delay caused by Priority Control State Machine at each of p transmitting stations, when the token is released after finishing frame transmission.

As

$$t_{fm\_tms} = \begin{cases} t_{fx}, & \text{for } t_{fx} \geq (t_{rl} + t_{ma} + kt_{x\_tms}); \\ t_{rl} + t_{ma} + kt_{x\_tms}, & \text{for } t_{fx} \leq (t_{rl} + t_{ma} + kt_{x\_tms}) \end{cases}$$

the final result for TRT_{tms} is:

$$TRT_{tms} = \begin{cases} t_{rl} + pt_{fx} + pt_{nt\_tms} + kt_{x\_tms}, & \text{for } t_{fx} \geq (t_{rl} + t_{ma} + kt_{x\_tms}); \\ (p+1)t_{rl} + pt_{ma} + (p+1)kt_{x\_tms} + pt_{nt\_tms}, & \text{for } t_{fx} \leq (t_{rl} + t_{ma} + kt_{x\_tms}) \end{cases}$$

Compare now the results of both implementations. We use 3 conditions:

1. for $t_{fx} \leq t_{rl} + t_{ma}$

$$TRT_{tms} - TRT_{iso} = [(p+1)t_{rl} + pt_{ma} + (p+1)kt_{x\_tms} + pt_{nt\_tms}] - [(p+1)t_{rl} + pt_{ma} + t_{x\_iso}] =$$

$$= (p+1)kt_{x\_tms} + pt_{nt\_tms} - t_{x\_iso} = 9.5[p(k+1) + k - 1]T$$

2. for $t_{rl} + t_{ma} \leq t_{fx} \leq t_{rl} + t_{ma} + kt_{x\_tms}$

$$TRT_{tms} - TRT_{iso} = [(p+1)t_{rl} + pt_{ma} + (p+1)kt_{x\_tms} + pt_{nt\_tms}] - (t_{rl} + pt_{fx} + t_{x\_iso}) =$$

$$= p[t_{rl} + t_{ma} + 9.5(k+1)T - t_{fx}] + 9.5kT - 16T$$

3. for $t_{fx} \geq t_{rl} + t_{ma} + kt_{x\_tms}$

$$TRT_{tms} - TRT_{iso} = (t_{rl} + pt_{fx} + pt_{nt\_tms} + kt_{x\_tms}) - (t_{rl} + pt_{fx} + t_{x\_iso}) =$$

$$= kt_{x\_tms} - t_{x\_iso} = 9.5kT + 9.5pT - 16T = [9.5(k+p) - 16]T \qquad (1)$$

As $p \geq 1$ and $k \geq 1$, for the first and third conditions the difference of the token rotation time between TMS and ISO implementations is at minimum $19T$ and $3T$ respectively and with growing p or k the difference is also growing. For the second condition, the difference alters from $3T$ to $19T$ depending on the number of transmitting and stacking stations and on the ratio between token ring latency and frame length. Using this comparison we find that the ISO implementation is better, even if the differences are small and could be neglected for small numbers of stacking and transmitting stations. For larger numbers of these stations, the differences would be even greater, in favor of the ISO implementation.

4.2 Analytical evaluation

In this section we calculate token rotation time (TRT) in the following model situation: Station A has requested a priority increase from n to m by writing reservation bits into B's data frame. Station B has increased ring priority to m by releasing the new token; furthermore station A has transmitted its DUs at priority m and thereafter releases the token at priority m. Finally, station B changes the ring priority back to n. TRT is the time period from when B releases the m priority token to the time when B releases the token decreased to priority n.

Consider the situation that during one m priority token rotation there are p stations successively capturing the token and transmitting data frames. There are k stacking stations on the ring. For simplicity presume that all p transmitting stations transmit data frames for the same time period, t_{fx}.

- for the ISO_DP priority function implementation

$$TRT_{iso} = t_{rl} + pt_{fm\_iso} + t_{x\_iso}$$

where

TRT_{iso}	is	token rotation time as specified above.
t_{rl}	is	total ring latency.
t_{fm_iso}	is	time period from A's start of data frame transmission at priority m to A's completion of priority m token transmission.
t_{x_iso}	is	16-bit token delay caused by priority decrease at station B.

and

$$t_{fm\_iso} = \begin{cases} t_{fx}, & \text{for } t_{fx} \geq (t_{rl} + t_{ma}); \\ t_{rl} + t_{ma}, & \text{for } t_{fx} \leq (t_{rl} + t_{ma}) \end{cases}$$

(the first and second cases in this equation occur when station A recognizes its own address before and after the end of its data frame transmission respectively)

where

t_{fx}	is	transmission time of the frames transmitted by station A.
t_{ma}	is	time from station A receiving the first bit of A's first data frame until A recognizes its own address in the frame source address field.

which finally results in:

$$TRT_{iso} = \begin{cases} t_{rl} + pt_{fx} + t_{x\_iso}, & \text{for } t_{fx} \geq (t_{rl} + t_{ma}); \\ (p+1)t_{rl} + pt_{ma} + t_{x\_iso}, & \text{for } t_{fx} \leq (t_{rl} + t_{ma}) \end{cases}$$

- for the TMS 380 priority function implementation

$$TRT_{tms} = t_{rl} + p(t_{fm\_tms} + t_{nt\_tms}) + kt_{x\_tms}$$

where

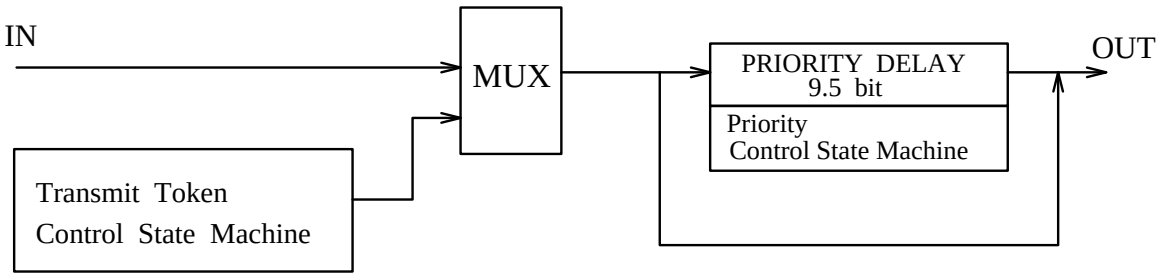

Figure 4. Priority machine in TMS 380

3.2 TMS 380 20 implementation of priority system

The TMS 380 Token Ring LAN adapter chipset includes the TMS 38020 chip, which performs hardware based LAN protocol functions. A Priority Control State Machine is dedicated to all ring priority management in the TMS 38020. After frame transmission, the Transmit Token Control State Machine releases the new token and sends it to the Priority Control State Machine (See Figure 4), which makes priority and reservation settings if necessary. Note that the Priority Control State Machine employs a 9.5 bit delay during every token transmission, even if the token doesn't change. If the station is requested for the ring priority increasing, the priority control state machine is enabled, computes the new token priority and inserts the 9.5 bit delay register into the ring path. It remains enabled until the stacking station has to decrease (or change) the new priority token. The stacking station's priority control state machine checks each incoming token for agreement with the top of the new priority stack; when this happens, the token's priority and reservation bits are modified (See Figure 3b). Note that all traffic passing through stacking stations is delayed by 9.5 ring periods.

4 Performance comparison

4.1 Comparison of the priority system implementation characteristics

Compare priority decreasing methods of the TMS and ISO implementations by examining Figures 3a and 3b. For the ISO implementation, the changed token is delayed by 16 bits with respect to the previous, while the TMS implementation releases the new token with a 9.5 bit delay. The station which has requested this new token needs to use it as soon as possible; if the network uses TMS priority machines, this station receives it earlier than in the ISO implementation. Shorter token delivery time decreases data transfer time. Thus it appears that TMS implementation of the priority machine is better in performance comparison with the ISO implementation. Similar consideration shows that if some stacking stations are active on a token ring network built with TMS priority machines, and all stations use greater priorities than the last priority increase on the ring, this situation degenerates into a token ring network with a reduced number of priorities, with some stations experiencing an increased delay. Prior performance evaluation work [3], [4], [5] shows the decrease of network performance with the increase in station delay. This implies that the ISO 8802/5 implementation outperforms the TMS 380 20 implementation, in contrast to the preceding result.

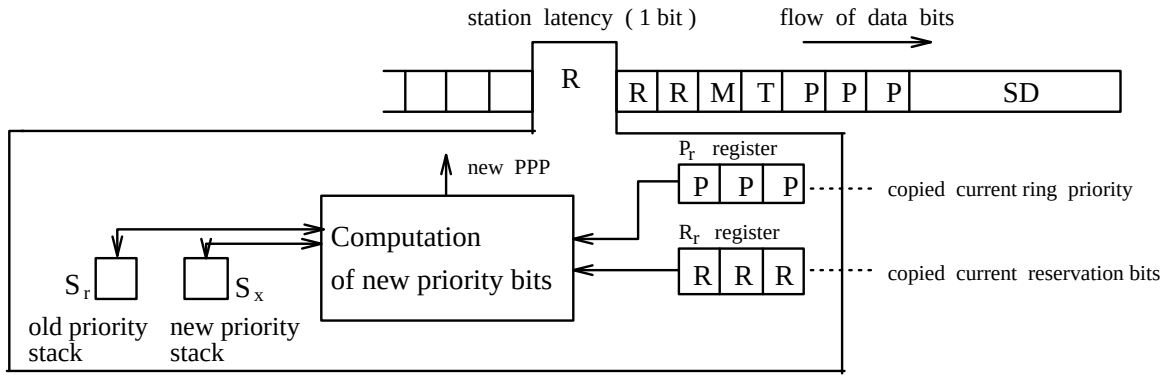

Figure 2

.

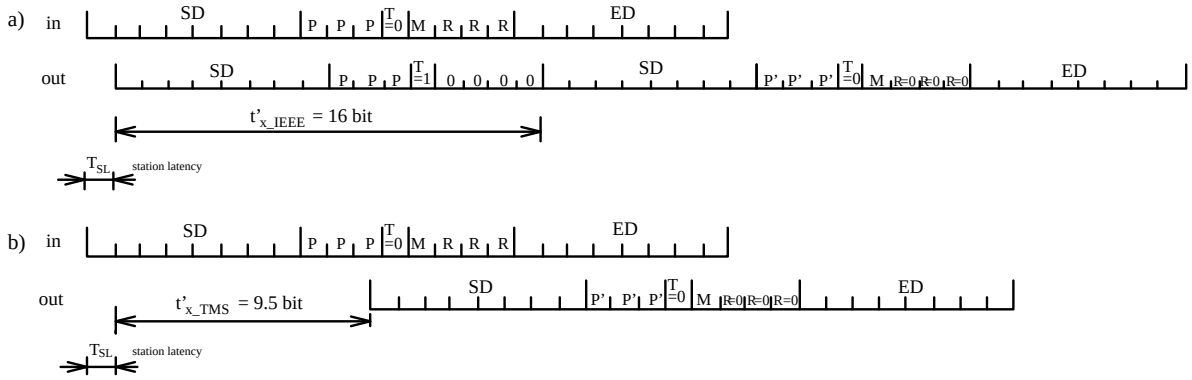

Figure 3. Priority decreasing action a) ISO 8802/5 implementation

b) TMS implementation

3.1 Priority system implementation according to ISO 8802/5

This section describes only the scheme for decreasing the token's priority, as gaining the token was described in the previous section. ISO 8802/5 defines priority decreasing by the use of an Operational Finite State Machine as follows. When the token passes the stacking station and a ring priority decrease is needed, the station changes the T bit of the token's access control field from 0 to 1, and transmits the frame *to nobody*. After the T bit transmission, the station transmits zeros until it completes receiving token reservation bits from upstream, then computes the new priority and does all necessary priority stack operations. After this, the station releases the new token (See Figure 3a). In Figure 3a, T_{sl} is the 1-bit station latency delay. The new token is 16 bits delayed compared with the normal token processing time. The *to nobody* frame will be removed from the ring by the station which captures the new token (which is transmitted immediately after the *to nobody* frame). This station then begins stripping its data frames. As the *to nobody* frame precedes all the station's data frames, it will be stripped first. An example of ISO based token ring network adapter implementation is in [6].

PPP	token priority (0-7)	ED	ending delimiter
T	=0 token =1 frame	FC	frame control
		DA	destination address
M	monitor bit	SA	source address
RRR	priority reservation (0-7)	INFO	information bits
SD	starting delimiter	FCS	frame check sequence
AC	access control field	FS	frame status

Figure 1

2 ISO 8802/5 Token Ring MAC and priority mechanism

The token ring local area network uses a circulating token for media access control (MAC) [2]. A station, which wants to transmit a frame on the ring, must wait for the token; after receiving it, the station transmits prepared data and then sends the token downstream on the ring (formats of frames and tokens are in Figure 1). After receiving the token, the frame transmission is done by changing *T bit* in the *AC field* of the token from 0 to 1 – thus changing token to the *start of frame sequence* . By assigning priority to each data unit (DU) and to the token, the network transfers data requiring shorter travel time as first by following rules. After capturing the token, a station can transfer only those DUs which have a priority higher or equal to the priority of the arriving token. Let A denotes the station with the highest-priority DUs queued for sending. This station can restrict other stations from sending low-priority data to decrease its medium access time as follows. It reserves the highest priority token by placing its request in the priority reservation bits of the frame currently passing the station. This frame is transmitted by other station; let B denotes it. After finish of sending this frame, station B transmits the token with the requested higher priority and stores the old and new priorities in its *old* and *new* priority stacks – thus becoming the *stacking station* . Station A then captures the token, transmits its high-priority data and re-sends the token with the same priority as it received. The token traveling around the ring now still has the new, increased priority. The *stacking station* B recognizes the token as the one with the priority equal to the top of the *new* priority stack and changes the priority of the token to the old one by poping the top of the *old* priority stack. Thus the *stacking station* makes its decision about changing token priority by current token priority, tops of the stacks and request for changing priority, which can be based on reservation bits or a station's self-request. Note that ring priority is specified by current token priority.

3 Two implementation of the priority function

In this section we describe two priority implementations which differ in their method of decreasing priority. To decrease ring priority it is necessary to have the token's priority, reservation bits, and the tops of stacks. In the time the station takes reservation bits to be copied into stations' R_r register (See Figure 2) and perform computation of the new token priority, the priority bits originally received have already been transmitted and are unavailable for modification (See Figure 2). There are two ways to solve this problem. The first is to transmit a new token after the new value of the priority bits is calculated, while changing the old token to the idle frame. The second approach inserts special delay register into the token path thus allowing the necessary calculations and priority bit modifications to occur while the token's access control field is still internal to the station. In the next sections we will describe both approaches and their effects on the total token ring parameters.

A Comparison of Two Implementations of the Token Ring Priority Function

Ivan Fellner, Ivan Racko, Milos Racek, Karol Fabian
Institute of Automation and Communication
Slovak Academy of Sciences
Czechoslovakia

Darrell Long*
Computer & Information Sciences
University of California, Santa Cruz

Abstract

System performance is strongly influenced by the quality of its implementation. We describe four ways of evaluating the performance of two implementations of the priority mechanism used in the Token Ring local area network. We use various methods for comparing the small priority machine's differences and show, how a simplified comparison can lead to bad results. With connection to the evaluation complexity, we also consider the simulation cost of the implementation process.

1 Introduction

A network communication standard does not specify all implementation details, and thus provides the adaptor designer some freedom without sacrificing compatibility. The designer should use this freedom to develop an adapter that provides the best performance. It is thus necessary to analyze the performance characteristics of proposed solutions, using both mathematical descriptions and simulation models.

As an example of LAN adaptor implementation, we will introduce two solutions for the Token Ring Medium Access Control priority mechanism. One implementation directly follows the priority mechanism description in the ISO DP 8802/5 standard [2]. The other implementation is incorporated in the Texas Instruments TMS 380 chip set [1]. We analyze the impact of each of the two priority mechanism implementations on the same token ring parameters. Section 2 of this paper briefly describes the token ring medium access control (MAC) method and its priority mechanism. Section 3 explains the two different priority function implementations. Section 4 compares both implementations with respect to their characteristics and analytical evaluation. Section 5 describes a simulation model used to evaluate the basic characteristics of both implementations and analyses the simulation results.

*Supported in part by faculty research funds from the University of California, Santa Cruz.