TWENTENET:
a LAN with message priorities, design and performance considerations.

I.G. Niemegeers and C.A. Vissers

Twente University of Technology
7500 AE Enschede, The Netherlands

ABSTRACT

This paper discusses design and performance aspects of Twentenet, one of the few implemented LANs which offers a service based on message priorities. The medium access mechanism uses the CSMA/CD principle, however with a deterministic collision resolution method. These characteristics make Twentenet suitable for real-time applications, as well as a mixture of real-time and non real-time applications. The general system structure is introduced followed by a detailed description of the priority access method. The performance of the system is shown for various traffic conditions and distributions of message priorities. The effect of system parameters, such as transmission rate, message length, cable length, and retry limits, is indicated.

1. INTRODUCTION

The literature on Local Area Networks (LANs) is abundant. Many systems have been built and even more systems have been proposed for various applications using various technologies. Franta and Chlamtac give a good survey of the work that has been done in this area [FRAN 81]. Twentenet is yet another LAN which, however, has the peculiarity that it is one of the few implemented systems with message priorities.

The project was started at Twente University of Technology in 1979 [TILB 80]. The aim of the project was to design and build a LAN that would be suitable for applications in a technical university environment where real-time applications coexist with non real-time applications in the various laboratories and educational and administrative departments. The project also had to serve as an educational tool for giving students hands on experience in design and implementation of LANs.

The system tries to reconcile the diverging requirements of the strongly disciplined process control environment with the the strongly undisciplined educational and experimental environment. In process control, for example, real time constraints demand deterministic access rules and reliability requirements favour the absence of active repeaters in the medium. In the educational and experimental environment, systems have diverging traffic characteristics, are switched on-line and off-line arbitrarily, and network reconfiguration is rather rule then exception implying that status information concerning the network configuration in each station should be avoided.

The nominal traffic load was assumed to be low, however, when the load becomes critical, urgent requests should still be served with relatively short delays.

Out of such requirements resulted the Twentenet which we will now describe and analyse. Let us proceed by first introducing the general structure, next by giving more details about its components especially the access method, and finally by discussing some performance aspects.

2. GENERAL STRUCTURE OF TWENTENET

The topology of Twentenet is a bus structure. The medium is a passive (i.e. without active repeaters) coaxial cable and is operated in baseband mode. The bus length is a parameter of the access mechanism which is loaded during power-on. This allows a drastic improvement in real time performance at short(er) bus length. The maximum length is two kilometres. The network is designed to operate at transmission rates of up to 16 Mb/sec. The address space allows up to 255 connected stations. The message length is variable from zero words to 128K words of 16 bits.

Twentenet uses message-based priorities. Herein Twentenet is rather unique. Due to those priorities Twentenet is a candidate for real-time as well as non-real-time applications. As far as we know, message-based priorities on bidirectional bus systems have been considered in the literature by Tobagi [TOBA 82], Chlamtac and Franta [CHLA 80] and Iida et al. [IIDA 80]. A reported implementation of these schemes is an Ethernet-like network with a priority function in [IIDA 80].

The access mechanism of Twentenet is based on a priority CSMA/CD principle and works according to the following rules:

-If a ready station senses the bus idle the station transmits its message.

-In case two or more ready stations sense the bus idle, a collision of messages may occur. On collision, contention is first resolved on the basis of message priorities: the message with the highest priority gets access first.

-However, collision may occur again when there are two or more messages with the same highest priority. On repeated collision, contention is resolved among the highest priority messages on the basis of a number which is unique for each station. For this number usually, but not necessarily, the station's address is chosen.

During arbitration new messages with yet higher priority are favoured by the access mechanism. Also the carrier sense mechanism keeps working, which implies that a station will refrain from taking access when another station is already using the bus.

The architectural structure of Twentenet follows the principles of the ISO 'Open Systems Interconnection Reference Model' (OSI-RM) [ISO 80]. The Medium and the Physical Service provide an elementary 'connectionless' broadcast service. The Data Link provides a connectionless service which can be parameterized for several different behaviours. The message priority is a parameter of the Data Link service primitives.

The implementation structure for these two layers is shown in Figure 1:

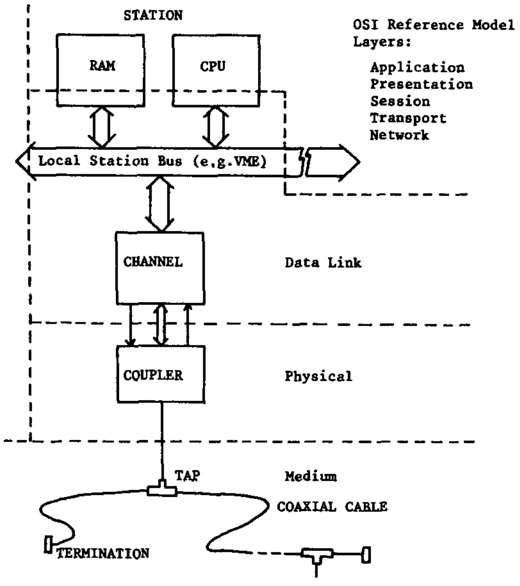


Fig. 1 General Structure of Twentenet

-The Coupler implements the Physical Layer in the OSI-RM. It performs message delimiting and a number of error detection functions which are derived from the logical and physical representation of the message.

-The Channel implements most of the Data Link Layer in the OSI-RM. The channel contains the access mechanism, provides the parallel to serial transformations beween the bit serial interface with the Coupler, and operates the 16 bit parallel interface with the Random Access Memory. The channel also performs message format control, error handling, and status reporting.

-The Random Access Memory, CPU, and System Bus implement the remaining part of the Data Link, and provide the interface to the Network Layer. These components, which, of course, can be arbitrarily extended, are also intended to implement (parts of) the Network and higher layers.

In the channel the send and receive operations function in parallel. The channel consists of four parts (Figure 2):

-The DMA-unit communicates via the local system bus with the CPU and RAM. It consists of the DMA/Format-controller, DMA-sender and DMA-receiver.

-The Transfer-Out unit requests access to the bus and converts a 16 bit word to a serial bit flow and passes that flow on to the coupler.

-The Access mechanism monitors the bus and grants access to the Transfer-Out function.

-The Transfer-In unit assembles 16 bit words which are passed to the DMA-receiver.

In terms of the proposed IEEE P802 Local Network Standard [IEEE 82] the DMA-unit is a part of the LLC sublayer and the Transfer-Out, Access mechanism and Transfer-in implement the MAC sublayer.

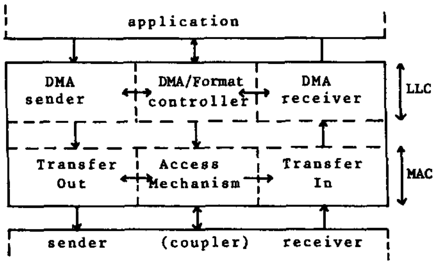


Fig.2 The channel structure

3. THE PHYSICAL LAYER.

Although the coupler is a sophsticated design [KOHL 82] which deserves a more elaborate treatment, the scope of this paper allows to mention only its main features.

The line signal representation uses a Partial

Response 4 (PR4) code. Characteristic for this linecode is its spectrum in which a DC component is absent, maxima occur at 0.5N and 1.5N, and zeroes at N and 2N, where N is the Nyquist frequency (Figure 3). The absence of the DC component allows to filter out the lower 100 KHz of the bandwidth, eliminating most of the external interferences and problems with differences in ground potentials (and implied safety risks). The dips in the spectrum are used to locate the preamble signal and the synchronizing, so called, 'beep-signal'. Precoding in the channel makes the bit recovery simple. Redundancy in the PR-4 code allows to detect bit errors on a logical basis in addition to signal quality detect mechanisms.

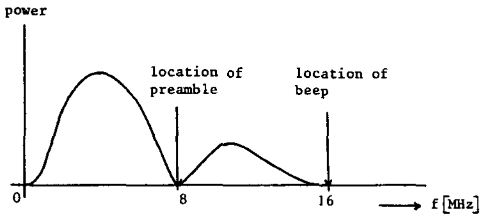


Fig. 3 Spectrum of Twentenet signal at 16 Mbaud

The power spectrum allows to defer the use of active repeaters to a larger distance than when less efficient coding schemes, e.g. Manchester Biphase, would have been used. In this way a transmission rate of 16 Mbps over a repeaterless distance of 2 km has been achieved.

The coupler, like the channel, consists of an independent send and receive part. This implies that the Twentenet components could also be used in a ring topology.

4. THE ACCESS MECHANISM

First we will discuss the normal operation of the access mechanism.

4.1. Access request.

In normal operation when the CPU wants to send a message, it notifies the DMA-unit. Thereupon the DMA-unit fetches (the beginning of) the message from RAM and activates the Transfer-out which in turn passes an access request to the access controller, which forms the heart of the access mechanism. When the access controller acquires the bus, an access grant signal is passed to the Transfer-out allowing it to send.

The access controller can be in one of three modes: the free mode (F-mode), the priority mode (P-mode) and the address- arbitration mode (Ai-modes). The access controller remains in its current mode when a message occupies the bus, and automatically returns to the F-mode, regardless of the current mode, when this message terminates normally or when the bus remains free for a certain period of time. Successive collisions cause the access controller to make transitions between modes. In case of low loads, the access controller will be mostly in the free mode wherein the bus is free or occupied by one message.

4.2. Free mode.

In F-mode a ready station has to wait until the bus becomes free before its Transfer-out gets permission to send.

If two or more stations (almost) simultaneously acquire access to the bus, a collision will occur. Each coupler can detect collision and notifies the access mechanism. The access mechanism orders the Transfer-out to stop the transmission and notifies the coupler to put a beep signal on the bus. The same procedure is followed when the coupler detects data mutilation or a beep on the bus. A collision, therefore, will initiate a concert of beep signals on the bus .This beep concert has two functions: collision enforcement and station synchronization. Every station will detect collision and all access controllers will make a transition to the P-mode.

4.3. Priority mode.

Upon entering the P-mode, a slot counter is started that indicates timeslots of equal duration according to Figure 4. During the first slot, the beep slot, the access mechanism does not respond to incoming signals in order to prevent an everlasting beep concert. The next four slots correspond to the four message priorities going from highest to lowest priority. At the beginning of a priority timeslot, indicating a priority that is equal to or lower than the station's message priority, the Transfer-out is granted access. When the coupler detects a message the counter is stopped and thus indicates a priority which is lower than or equal to the priority of the message that currently occupies the bus. Due to this, lower priorities are prohibited to interfere with the higher priorities.

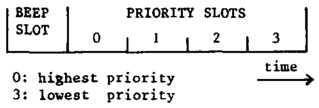


Fig. 4 P-mode Timeslot

We will refer to this as the semipreemptive protocol function in accordance with [TOBA 82]. Semipreemptive means that a station, which wants to send a message with a higher priority than the priority indicated by the slot counter, may send the message as long as no carrier is detected on the bus.

In the P-mode, however, two messages can collide again. In this case the address arbitration mode is entered with a contending priority equal to the priority indicated by the slot counter at the moment of collision, although the priorities of the colliding messages do not have to be equal due to semipreemption.

4.4. Address arbitration mode.

If collision occurs in the P-mode (or in a priority slot during an Ai-mode as discussed below), then address arbitration will resolve this collision. The address arbitration mode consists of maximum 4 successive phases, which we call A0-, A1-A2-, and A3-mode. As a result always one station will be granted access.

On collision in the P-mode the access controller in every station makes a transition to the A0-mode and the contending priority is stored. By doing so every station can decide whether it is allowed to contend for bus access or not: only the colliding stations with the present priority level and those stations that generate a higher priority message during the contention period, may participate in the contention during address arbitration. This design follows from the semipreemptive protocol requirement: in order to implement semipreemption in an orderly way, priority timeslots must precede the address timeslots (in Figure 5 called 'MAi=' timeslots) in the Ai-modes. These priority slots may only be used by messages with a higher priority than the contending priority. If in the priority slots collision occurs again, the contending priority is made equal to the priority associated with that slot and the access controller in every station moves to the A0-mode again. Some of the priority slots in the Ai-mode cannot be used, because those slots correspond with the stored contending priority or an even lower priority. Therefore those superfluous slots are left out (Figure 5).

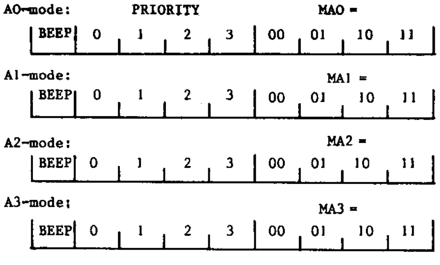


Fig. 5 Address-mode timeslots

In each of the Ai-modes the access controller counts four extra timeslots, which we call address slots, in addition to the priority slots. These address slots are defined as follows: the unique 8 bit number (MA) that is assigned to each station, and for which we will take the station's address, is sliced into four parts: MA=MA3/MA2/MA1/MA0, wherein MA0 contains the two least significant bits. These bits are interpreted as a priority level (0 through 3) in much the same way as the message priority.

Whenever the access controller indicates the address slice that corresponds to the current Ai-mode, the station is enabled to send. Note that this station must be allowed to participate in the contention on the basis of its message priority. Note also that because of the semipreemption a station with a message priority higher than the one that is currently contending in the address arbitration, may also start transmission in the MAi slots provided the bus is sensed idle.

If collision occurs again during the MA0-slots the access controller moves to the A1-mode. However, only those stations that were granted access in the A0-mode are allowed to participate in the A1 contention. Therefore, since the sliced addresses represent some form of arbitrariness, the chance to move to a next address mode will decrease, and only in the worst case situation the access controllers will move to the A3-mode. Due to the fact that the addresses are unique, finally only one station will seize the bus.

4.5. Preemptive operation.

In certain real-time applications it might be desirable that very urgent (i.e. high priority) requests be able to interrupt low priority messages of long duration. The Twentenet access mechanism has this option available. The DMA controller can detect the priority of the ongoing transmission and can deliberately generate a beep signal to interrupt it. Hereafter the access arbitration mechanism is started in all stations to give access to the highest priority request (which is not necessarily the interrupting request).

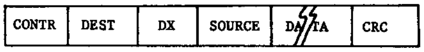
4.6 Complexity and correctness

Although the procedure for access arbitration may seem complex it is rather systematic and modular in nature, and thus allows a simple implementation. The access controller is modeled by a 12-state automaton, supported by two counters and three filters. Its hardware implementation takes only 21 flip-flops. The access mechanism has been extensively verified to check correct operation under all possible error conditions.

5. THE DMA-UNIT

Data transfer between the channel and its user is based on DMA. A format-controller which converts the data from memory to the correct transmission format and vice versa is present next to the DMA-function (Figure 6). Received data is only transferred to memory under conditions stored in the DMA-unit e.g. 'the destination address equals the station's address', or 'it is a broadcast message', or 'the message is transmitted by a specific station'. Broadcast messages can also be suppressed.

Both DMA-sender and receiver have a 16 word FIFO-buffer at their disposal to allow for sufficient access time to the local system bus.



CONTR : 8 bit control field (with message priority)
DEST : 8 bit destination address
DX : 8 bit spare address field

SOURCE: 8 bit source address
DATA : a multiple of 17 bit data words
CRC : 16 bit cyclic redundancy check

Fig. 6 Format Data Link Protocol Data Unit

The DMA-unit is also capable of inserting dummy words between the actual data words during transmission. This is used as a flow control mechanism whereby stations that cannot meet a high datarate, are also able to use Twentenet. The dummy words are marked as such by setting a special bit added to each word, and deleted upon receipt.

The DMA-unit further monitors status information such as the number of arbitrations in which a message participates. This number is limited by a retry-parameter.

Because the first word of each message contains the message priority the DMA-unit can detect the priority that currently occupies the bus, which must be known before the controller can decide to use preemption.

6. IMPLEMENTATION

The channel has been completely designed with the use of a formal description technique based on synchronous finite state machines [VRIE 82]. The total number of memory elements, therefore, can be simply calculated and ranges less than 1000 memory elements.

Prototypes of the channel have been built functioning at 1Kbps, 1Mbps, and 12Mbps. The current 12 Mbps version was done in terms of MSI. This implementation has been tested extensively and yielded only one error caused by a typing errr. An implementation in terms of VLSI is in preparation. The VLSI gate count equivalent is on the order of 13,000.

Prototypes of the coupler have been built functioning at 1Kbps, 1Mbps, and 16Mbps. The 16Mbps version is now in test phase.

7. PERFORMANCE MODELING

A simulation model has been developed to investigate the performance of Twentenet for all possible configurations [COST 83]. It assumes an error free system and considers only the non-preemptive operation of the access mechanism. A computationally efficient yet detailed model was designed based on the observation that the model can act as an omniscient controller, able to predict the outcome of bus contention. The station's access mechanisms are not simulated individually. Instead a global non distributed MAC Service Provider model is simulated. A regenerative technique is used for data analysis. Where available 95% confidence intervals of performance measures are given.

An analytical model [NIEM 83] was also developed using the observation that if one considers the pairs (message priority, station address) as a priority level the access mechanism behaves as a single server queueing system with a non work conserving head-of-the-line priority queueing discipline. The model estimates the extensions to the service times caused by arbitration in a heuristic way. With respect to the simulation model there are some additional assumptions:

-no semi-preemption, and

-no retry limit, hence no message loss.

At high loads therefore we expect the analytical model to show higher queueing delays.

A single server no overhead priority queueing system with four message priority levels and FCFS handling of all messages of given priority regardless of station address, was used as a reference. To a certain extent this reference represents an ideal priority access-mechanism.

In all cases the workload is modeled as an infinite source generating messages with arbitrary length distribution and Poisson arrival process.

8. PERFORMANCE RESULTS

Our aim was to investigate the effect of system parameters, configuration and workload on message transfer delay and system throughput. A standard network configuration (Table 1) was used.

Table 1 The Standard Configuration

number of stations	50
addresses	sequential (0-49)
spatial distribution	uniform
cablelength	1000 m
propagation speed	5 nsec/m
transmission rate	10 Mbps

The influence of the following parameters was investigated: retry limit, transmission rate, cable length, number of stations, station addresses, message length and message priority. Let us examine the effect of each of these.

8.1. Retry limit.

Table 2 displays for four values of the retry limit system throughput, average transfer delay and arbitration overhead. The results are for the standard configuration, a 4 Mbps uniform load and message priority 0. We observe that increasing the retry limit results in a throughput improvement, however, with diminishing returns. Indeed the overhead becomes high when the retry limit reaches 16. This causes a rapid increase in the mean transfer delay making a retry limit of 4 acceptable (at the cost of some messages not being transferred) and a limit larger than 16 questionable.

Table 2 Effect of the retry limit

- standard network configuration (Table 1)
- uniform traffic pattern, one message priority (0)
- offered load = 4 Mbit/sec.
- message length: 1024 bits

	Retry limit			
	2	4	16	32
effective throughput [Mbit/sec]	3.72	3.81	3.90	3.94
arbitration overhead [% of system capacity]	6.5	9.6	17.4	21.7
mean transfer delay [usec]	188	231	402	570

8.2. Physical layer transmission rate and message length.

The effect of the transmission rate on the efficiency of the access mechanism was examined. Figures 7 and 8 display for 16 and 10 Mbps total and effective bus utilization (relative throughput) as a function of the relative offered load.

The general behaviour of CSMA/CD access mechanisms is confirmed. Everything else remaining the same, an increase of the transmission rate lowers the ratio of message duration to end-to-end propagation delay causing the access mechanism to be less efficient because the ratio of acquisition interval to contention interval decreases [TOBA 80]. Figure 7 shows indeed that for 10 Mbps saturation occurs at a load of about 80% of the bus capacity while at 16 Mbps this point is already reached near 60%. If we look at the effective throughput the point becomes even clearer. Due to the retry limit for which in Figure 8 a low value of 4 was selected, the maximum effective throughput reaches about 45% (4.5 Mbps) of the bus capacity at 10 Mbps and only 35% (yet 5.6 Mbps) at 16 Mbps.

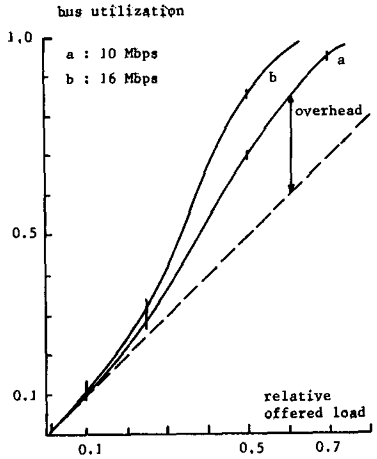


Fig. 7 Effect of transmission rate on bus utilization

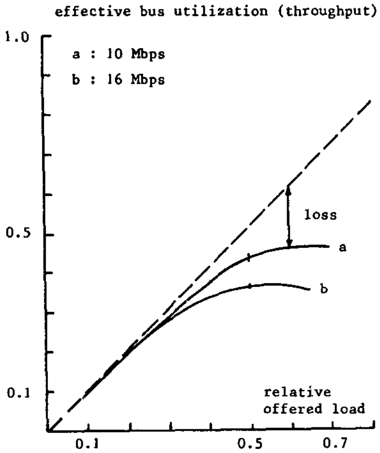


Fig. 8 Effect of transmission rate on relative throughput

Message length has two effects. An increase in message length (at the same transfer rate) improves the efficiency of the protocol because the ratio of the number of protocol control information bits to service data unit bits decreases. A second gain in efficiency is due to the same CSMA/CD property which caused the transmission rate effect. These effects show up in Figure 9 where the normalized mean transfer delay for an average length of 10240 bits increases much slower as a fuction of load than for 1024 bits.

8.3. Cable length.

Cable length determines the end-to-end propagation delay and the slot duration. In case of collision the smaller this value is the sooner arbitration results in a successfull transmission. This is demonstrated by Figure 10 which for lengths of 2, 1 and 0.5 km shows that for a given mean transfer delay (quality of service) a much higher throughput can be reached at shorter lengths.

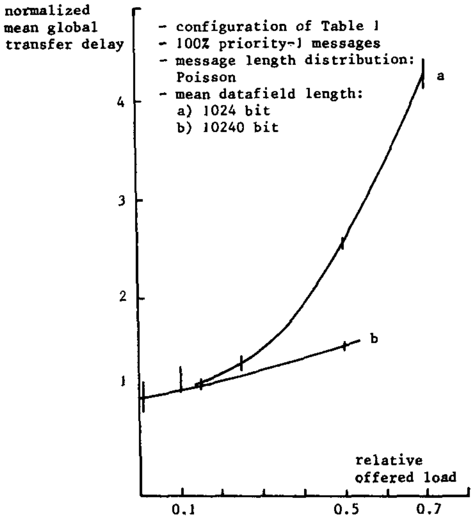


Fig. 9 Effect of message length

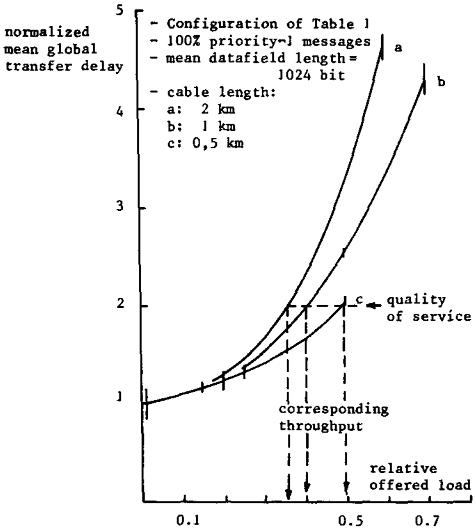


Fig. 10 Effect of cablelength

8.4. Number of stations.

Configurations with 10, 100 and 200 stations were simulated. The total offered load was kept constant. No significant effect was found on the mean transfer delay averaged over all stations, i.e. 95% confidence intervals were overlapping. However the mean transfer delays experienced by the individual stations were affected, since station addresses play a role in resolving contention between equal priorities.

8.5. Station addresses.

Station addresses are used to resolve contention between messages of equal priority, hence a station's performance is strongly influenced by its address when the load is high. This unfairness, which is inherent in priority mechanisms is illustrated by Figure 11. The mean transfer delay is shown as a function of the offered load for the best address, the worst

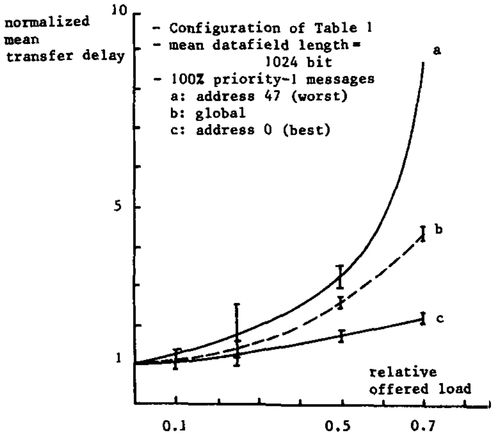


Fig. 11 Effect of station addresses

address and averaged over all stations. The station with the best address is virtually unaffected by the load while the station with the worst address suffers large delays and a high message loss-rate at high offered load.

8.6. Message priority.

The effect of message priorities was studied for two types of load.

Let us first consider a load consisting of a large amount of non-time critical traffic mixed with a small amount of real-time traffic, e.g. short alarm or control messages. This is represented by a pattern in which 90% of the messages are long (mean 1024 bits) low priority ones and 10% are short (mean 96 bits) high priority ones. In terms of offered system load the high priority messages account for about 1% and the low priority ones for 99%. In Figure 12 the mean transfer delay is plotted against the offered load. The results of the simulation-, analytical- and reference model are shown. We observe that Twentenet indeed has the behaviour of a single server priority queueing system with all curves shifted to the left with respect to this reference, which is due to the arbitration overhead. At high loads the overhead becomes more important causing the amount of shift to grow accordingly. This is most clearly shown by the analytical model, which at high loads predicts higher delays than the simulation model. This is due to the assumption of an an infinite retry limit (no loss) and the fact that the simulation results only reflect successful messages. The analytical results are thus the limiting case for infinite retries.

Note that at high loads (e.g. 6 Mbps, yielding a net throughput of 645 priority 0 messages/sec), the critical high priority traffic does not suffer any message loss, since arbitrations are quickly won in the P-mode of the access mechanism or through semi-preemption. The noncritical low priority traffic on the other hand is affected by message loss. This is shown in Figure 12 by the discrepancy between the analytical results (no loss) and the simulation results for priority 1.

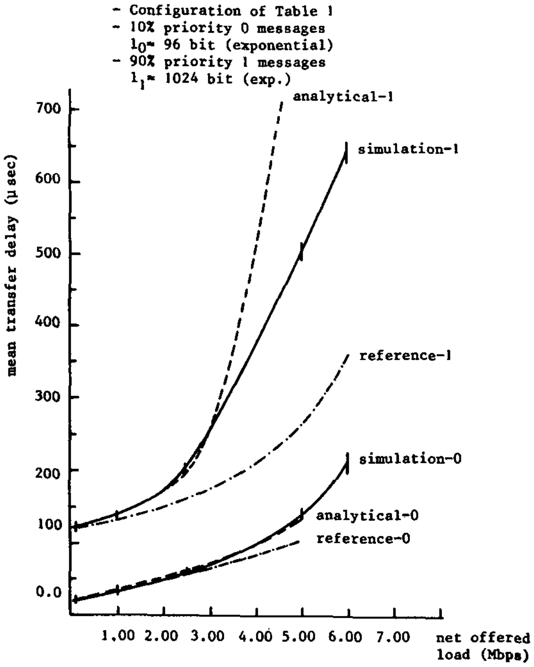


Fig. 12 The use of two priorities

A second case was considered in which all four message priorities were used. In this case 10% of the messages have priority 0 (highest), 20% priority 1, 30% priority 2, and 40% priority 3 (lowest). The mean values of the corresponding message lengths are 96, 1024, 1024, and 10240 bits. This traffic pattern could correspond to an application environment in which the quality of service requirements break down into a number (here: 4) of classes with different constraints for the transfer delay. We assume that time critical messages, which will be assigned higher priorities, will in general be shorter than less time critical messages, which are assigned the lower priorities and which represent the bulk of the traffic (88% in this case). Figure 13 shows for the four message priorities the mean transfer delay vs. the offered load curves obtained from the analytical model. Also shown are the results of the reference system. The three higher priorities do not suffer much under a high system load, this at the expense of the lower priority messages which experience rapidly increasing delays. Moreover, when we compare with figure 14, which shows the same results for the simulation model, we see that for the lower priority a lot of messages are lost at high loads. The three highest priorities exhibit the same behaviour in the analytical and simulation model, indicating that no loss occurs at these levels. The increase in transfer delay of the highest three priorities is almost solely due to waiting for the end of transmission of low prioritiy messages occupying the bus.

9. CONCLUSION

The aim of the Twentenet project was to design and build a LAN which would support real-time as well as non-real-time applications in a university

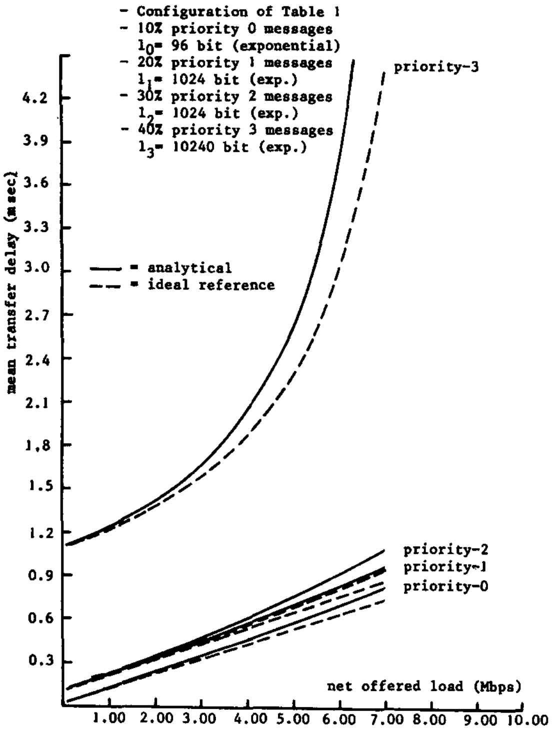


Fig. 13 The use of four priorities - Analytical results

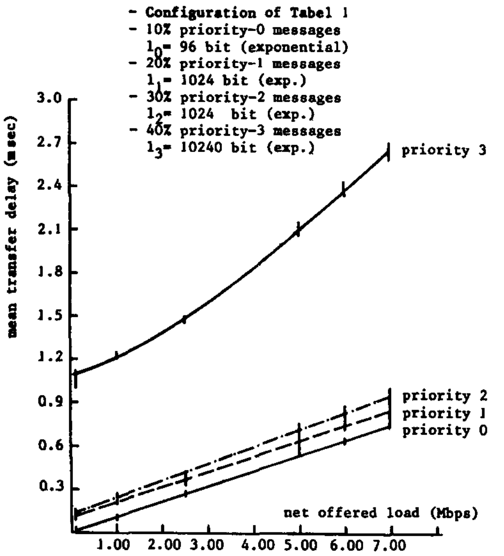


Fig. 14 The use of four priorities - Simulation results

environment. This aim has been largely achieved by choosing a CSMA/CD type access method with message priority collision resolution, and by using a repeaterless physical layer, which accomplishes a high bit-rate (16 Mbps) over a large distance (2 Km), and detects collisions without having to recur to a DC component in the line signal.

Although the system is intended for low nominal workloads, it was shown that it yields reasonably high throughput values (on the order of 5 Mbps). More important is that the message priority mechanism indeed guarantees a good quality of service to urgent requests when the system load becomes high. As in any priority service system, this is achieved at the expense of the low priority requests , which suffer higher delays and loss. It is the responsibility of higher layer protocol functions to handle this loss.

Currently a 16Mbps prototype is near completion, succeeding earlier 1Kbps and 1 Mbps versions.

10. REFERENCES

[CHLA 80] Chlamtac I., Franta W.R., Message Based Priority Access to Local Networks, Comp. Comm., Vol. 3, No.2, Apr. 1980.

[FRAN 81] Franta W.R., Chlamtac I., Local Networks, Lexington Books, 1981, Lexington, Massachusetts.

[IIDA 80] Iida I., Ishizuka M., Yasuda Y., Random Access Packet Switched Local Computer Network with Priority Function, Proc. Natl. Telecomm. Conference, Houston, Dec. 1980

[IEEE 82] IEEE Project 802, Local Network Standards, Draft C IEEE Standard 802, May 1982.

[ISO 80] ISO/TC 97/SC 16, Data Processing Open Systems Interconnection-Basic Reference Model, Draft Proposal, ISO/DP7498, 1982.

[KOHL 82] Kohlmann R.H.M., The Physical Layer of the Twentenet LAN, Rep.1251-82-12, Twente Univ. of Technology, Enschede, The Netherlands, Oct. 1982.

[MEER 81] Meer R. van der, Design and Realization of the Physical Communication Part of the CID Local Network, Rep.1251-81-04, Twente Univ. of Technology, Enschede, June 1981.

[NIEM 83] Niemegeers I.G., An Analytical Model for the Twentenet Access Mechanism, Rep.1251-83-13, Twente Univ. of Technology, Enschede, The Netherlands, 1983.

[OOST 82] Ooster J. van 't, Niemegeers I.G., Daemen J., TWENTENET: A LAN with Message Based Priorities, Rep.1251-82-09 Twente Univ. of Technology, Enschede, June 1982.

[OOST 83] Ooster J. van 't, Performance Analysis of Twentenet, Rep.1251-83-03, Twente Univ. of Technology, Enschede, The Netherlands, June 1983.

[TILB 80] Tilburgs P.K., Vissers C.A., Access Mechanism for Real Time Serial Bus Systems, Report submitted to IEEE P802, Twente Univ. of Technology, Enschede, Nov. 1980.

[TOBA 80] Tobagi F.A., Hunt V.B., Performance Analysis of Carrier Sense Multiple Access with Collision Detection, Comp. Netw., Vol.4, 1980.

[TOBA 82] Tobagi F.A., Carrier Sense Multiple Access with Message Based Priority Functions, IEEE Trans. on Com., Vol. COM-30, pp. 185-200, Jan. 1982.

[VRIE 82] de Vries D., Data Link Protocol voor het Twentenet, Rep.INF/IPS/005, Twente Univ. of Techn., Enschede, Jan. 1982.