PERFORMANCE EVALUATION OF ISOLATED AND INTERCONNECTED TOKEN BUS LOCAL AREA NETWORKS

Daniel A. Menascé
Leonardo Lellis P. Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
22453, Rio de Janeiro, RJ Brazil

Abstract

The token bus based local area network, REDPUC, designed and implemented at the Pontíficia Universidade Católica do Rio de Janeiro is briefly described. Analytic models are presented, which allow one to obtain an approximation for the average packet delay, as well as exact upper and lower bounds for the same performance measure. A performance evaluation of interconnected local networks is also given.

1. Introduction

In 1982, the Computer Networks Group of the Pontíficia Universidade Católica do Rio de Janeiro - PUC/RJ - (Catholic University at Rio de Janeiro) started the design and implementation of the local area network (LAN), called REDPUC (an acronym for PUC's Network, in Portuguese). One of the design goals was to come up with a LAN architecture suitable for real time process control. Therefore, reliability and high availability were important requirements. Another equally important requirement was the need to use a medium access protocol which could guarantee a bounded access time, defined as the time interval since the availability of a message for transmission at a network station and the beginning of its transmission. Therefore, contention based protocols were immediately excluded.

The group decided to design its own protocol using a token bus based architecture. The resulting protocol bears some similarities with one of the IEEE's Project 802 Standards [IEEE 82]. A detailed description of the protocol and of the hardware architecture used to implement our local network can be found in [SOARES 83]. A prototype REDPUC has been operational since October 1983.

This paper presents a performance evaluation of REDPUC, as well as an analytic model which allows one to obtain performance measures for a set of interconnected REDPUCs. Section two presents a brief description of the Bus Access Protocol. Section three gives lower and upper bounds for the average packet delay, as well as an approximation for the average packet delay. Numerical results are discussed and comparisons with simulation results are presented. Finally, section four introduces the model for interconnected local networks.

2. Bus Access Protocol Description

REDPUC is a general purpose token bus local network designed and implemented at the Pontíficia Universidade Católica do Rio de Janeiro (PUC/RJ). Each station is connected to the network via a Bus Interfaces Unit (BIU). The BIU implements the Physical Level Protocol (Level 1) as well as the Bus Access Protocol (Level 2).

The Physical Level Protocol provides Manchester encoded half-duplex bit transmission and reception.

The Bus Access Protocol (BAP) will be briefly described in this section so that the reader may understand the analytic model that follows in the remaining sections. A more detailed description may be found in [SOARES 83].

The BAP implements a virtual ring among the set of stations. In each BIU there is a Cycle Control List (CCL) which indicates the logical order in which the set of stations is arranged in the virtual ring. Stations are dynamically added and deleted from the CCL as indicated in the following paragraphs.

When a station transmits a packet, it passes the permission to transmit (the token) to the next station in the virtual ring. The first byte of every packet header is the token, i.e. the address of the next station to transmit. If a station has no data ready for transmission when it receives the token, it must transmit an empty packet, i.e. a packet which only contains the token. Stations which fail three times in a row to transmit any kind of packet when they receive the token, are excluded from every other station Cycle Control List. Therefore stations pass the token in a cyclic fashion.

At the end of every cycle, there is an interval, called contention interval (CI), which can be used by stations which are out of the logical cycle and want to be inserted in it. If no station transmits during the contention interval, then the CI ends after a timeout and a new cycle begins. If only one station transmits an insertion request during the CI, all other stations will listen to it (remember that the bus is a broadcast medium) and will add the new station at the end of the cycle. If two or more stations transmit an insertion request during the CI, there will be a collision that will be felt by all other stations (collisions are detected by an incorrect CRC). When this occurs all physically connected stations are inserted in every station's Cycle Control List. This procedure will add to the cycle all the stations which wanted to join the cycle (and collided during the CI), as well as some others which did not want to get inserted in the cycle. The latter ones will be automatically deleted in the next three cycles, as indicated previously, since they will ommit themselves when they receive the token.

Notice that this rather simple and apparently inefficient collision resolution mechanism will only be executed if two or more stations want to join the cycle during the same contention interval which is a low probability event. It should be remembered that the contention interval is not used for normal data transmission but only for re-insertion of out of order stations in the cycle.

The model presented in the following sections considers that the number of stations in the cycle remains constant and therefore the contention interval has a fixed duration equal to the timeout already mentioned.

Figure 1 presents three possible scenarios. The first one (figure 1.a) shows a cycle in which only one station is transmitting data packets and the remaining ones are only passing the token. The scenario illustrated in figure 1.b displays a situation where all stations in the cycle are sending data packets. Finally, figure 1.c shows an intermediary situation in which some stations transmit data packets and some transmit empty packets.

3. Performance Evaluation of an Isolated Network

This section presents the analytic models used to evaluate the performance of the local network described in the previous section. The first model allows one to obtain upper and lower bounds, for the average access delay, defined as the time interval between the arrival of a packet at the source station and the arrival of the last bit of the packet at the destination station. The second model allows one to obtain an approximation for the average access delay.

The list of input parameters and resulting performance measures is given below.

Input Parameters

C: bus speed (in bps).
P: number of stations.
λ: average packet arrival rate at each station (packets/s).
v: empty packet size (in bits)
r: duration of the contention interval (in sec.)

Random Variables of Interest

$\tilde{t}_{B_i}$: time during which station i uses the bus per cycle.
$\tilde{c}$: cycle duration
$\tilde{w}$: packet waiting time at each station (i.e. time interval since a packet arrival at a station and the start of its transmission).
$\tilde{m}$: packet length (not including the token), in bits.
$\tilde{d}$: packet delay (i.e. time interval since a packet arrival at a station and the arrival of its last bit at the destination station).
$\tilde{x}$: *data packet transmission time* ($\tilde{x} = \tilde{m}/C$) (it does not include the token)

Let $X^*(s)$, $B_i^*(s)$, $C^*(s)$ be the Laplace Transforms (L.T.s) of the p.d.f.s of the random variables $\tilde{x}$, $\tilde{t}_{B_i}$ and $\tilde{c}$, respectively. The following notation will be used.

$$\bar{b}_i \triangleq E[\tilde{t}_{B_i}],\ \overline{b_i^2} \triangleq E[\tilde{t}_{B_i}^2],$$

$$\bar{c} \triangleq E[\tilde{c}],\ \overline{c^2} \triangleq E[\tilde{c}^2],$$

$$W \triangleq E[\tilde{w}],\ D \triangleq E[\tilde{d}],$$

$$\bar{x} \triangleq E[\tilde{x}],\ \overline{x^2} \triangleq E[\tilde{x}^2].$$

Let S be the bus throughput, defined as the fraction of the bus capacity used to transmit data packets.

In the section that follows it will be assumed that the packet arrival process at each station is a Poisson one.

3.1 - Lower and Upper Bounds for the Average Packet Delay

The lower bounds for the packet delay occurs when only one station has packets to transmit and the remaining ones are just transmitting empty packets. The upper bound occurs when all stations always have a data packet to transmit.

The packet waiting time at each station can be obtained using an M/G/1 model where the server is the bus and the service time is the cycle time. Notice that in the situation defined for the lower and upper bound cases, the cycle time can be considered statistically independent from the packet interarrival time.

Let α_i = Pr [station i has a data packet ready for transmission when it receives the token]. Then

$$B_i^*(s) = \alpha_i X^*(s)\, e^{-\frac{sv}{C}} + (1-\alpha_i)\, e^{-\frac{sv}{C}} \tag{1}$$

and

$$\bar{b}_i = \alpha_i \bar{x} + \frac{v}{C} \tag{2}$$

$$\overline{b_i^2} = \alpha_i \left(\overline{x^2} + \frac{2v}{C}\bar{x}\right) + \frac{v^2}{C^2} \tag{3}$$

The cycle time is the sum of the times that all stations use the bus, either trasmitting a data packet or an empty packet, plus the contention interval duration. In general, $\tilde{t}_B$ for a given station is not independent of the same variable for other stations. This is easily understood by observing that during a long cycle more packets will tend to arrive at a given station than during a short cycle, therefore affecting the value of the probability, α, that a station has a data packet ready for transmission.

However, in the case of the upper and lower bounds, the time that a station occupies the bus can be considered independent for all stations. Therefore, $C^*(s)$ can be easily obtained as the product below.

$$C^*(s) = e^{-sr} \prod_{i=1}^{P} B_i^*(s) \tag{4}$$

Lower Bound for the Mean Packet Delay

Let k be the only station that always has data packets for transmission when it receives the token.

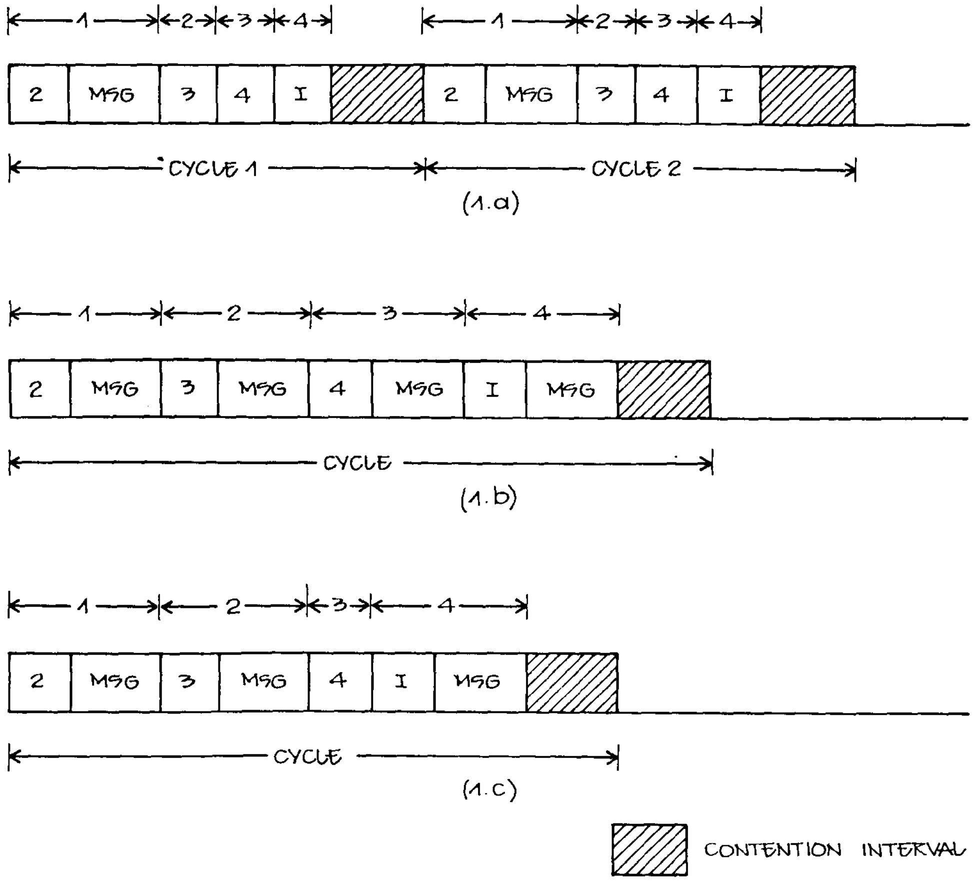

FIGURE 1 - THREE SCENARIOS OF THE LOCAL NETWORK PROTOCOL

The remaining ones only transmit empty packets Therefore,

$$\alpha_i = \begin{cases} 1 & \text{for } i = k \\ 0 & \text{for } i = k \end{cases} \qquad (5)$$

and

$$B_i^*(s) = \begin{cases} X^*(s)\, e^{-\frac{sv}{C}} & \text{for } i = k \\ e^{-\frac{sv}{C}} & \text{for } i \neq k \end{cases} \qquad (6)$$

Hence,

$$\bar{b}_i = \begin{cases} \bar{x} + \frac{v}{C} & \text{for } i = k \\ \frac{v}{C} & \text{for } i \neq k \end{cases} \qquad (7)$$

and

$$\overline{b_i^2} = \begin{cases} \overline{x^2} + \frac{2v}{C}\,\bar{x} + \frac{v^2}{C^2} & \text{for } i = k \\ \frac{v^2}{C^2} & \text{for } i \neq k \end{cases} \qquad (8)$$

Using (4) and (6) we have that the L.T., $C^*_{low}(s)$, of the cycle time in the lower bound case is given by,

$$C^*_{low}(s) = X^*(s)\, e^{-s\left(\frac{Pv}{C}+r\right)} \qquad (9)$$

and its first and second moments, $\bar{c}_{low}$ and $\overline{c^2_{low}}$, are given by,

$$\bar{c}_{low} = \bar{x} + \frac{Pv}{C} + r \qquad (10)$$

$$\overline{c^2_{low}} = \overline{x^2} + 2\left(\frac{Pv}{C} + r\right)\bar{x} + \left(\frac{Pv}{C} + r\right)^2 \qquad (11)$$

Finally, using the M/G/1 formulas [KLEI 75] we obtain the lower bound, D_{low}, for the average packet delay as,

$$D_{low} = \frac{\lambda\, \overline{c^2_{low}}}{2(1-\lambda\, \bar{c}_{low})} + \bar{x} + \frac{v}{C} \qquad (12)$$

Upper Bound for the Mean Packet Delay

In this case, since all stations always have a data packet ready for transmission when they receive the token,

$$B_i^*(s) = X^*(s)\, e^{-\frac{sv}{C}} \quad \text{for } i = 1, \ldots, P \qquad (13)$$

Therefore, the L.T. and the first and second moments of the cycle time in the upper bound case are given by,

$$C^*_{upp}(s) = [X^*(s)]^P e^{-s\left(\frac{Pv}{C} + r\right)} \qquad (14)$$

$$\overline{c}_{upp} = P\left(x + \frac{v}{C}\right) + r \qquad (15)$$

$$\overline{c^2_{upp}} = P[(P-1)\ (\overline{x})^2 + \overline{x^2} + \overline{x}\ (\frac{Pv}{C} + r)] \ +$$

$$\left(\frac{Pv}{C} + r\right)[P(\overline{x} + \frac{v}{C}) + r] \qquad (16)$$

Finally, using the M/G/1 formulas the upper bound, D_{upp}, for the average packet delay is given by

$$D_{upp} = \frac{\lambda\, \overline{c^2_{upp}}}{2(1 - \lambda\, \overline{c}_{upp})} + \overline{x} + \frac{v}{C} \qquad (17)$$

3.2 - An Approximation for the Average Packet Delay

The simulation studies that we carried out showed that assuming independence between the various queues at the network stations as was done in the literature [BUX 81] underestimates the second moment of the cycle duration. Bux and Truong [BUX 83] present a nice approximate analysis for the case of exhaustive service.

Therefore, a model to estimate the cycle duration should take into account the dependency of one cycle length on the previous cycles. The concept of conditional cycles, introduced by Kuehn [KUEHN 79], which only considers the influence of the immediately preceding cycle on the following one, was used to obtain an approximation for the average packet delay. Simulation was used to evaluate this approximation and the results are discussed in section 3.3.

The model to be discussed next considers the bus as a cyclic server which serves in a round-robin fashion all the stations. At most one message per queue is transmitted every time that the cyclic server (the bus) serves the queue, i.e. service is non exhaustive. After transmitting a message from a queue, control must be passed to the next station in the cycle.

Let $c_o = \frac{Pv}{c} + r$ be the portion of a cycle wasted due to control overhead (token passing and contention interval).

The average cycle duration, $\overline{c}$, can be easily obtained as follows.

$$\overline{c} = c_o + \sum_{j=1}^{P} \alpha_j\, \overline{x} \qquad (18)$$

The probability α_j can also be interpreted as the average number of packets from station j served by the bus in one cycle. But, in equilibrium, the average number served in a cycle must be equal to the average number of arrivals in a cycle ($\lambda\, \overline{c}$). Thus

$$\alpha_j = \lambda\, \overline{c} \qquad (19)$$

Using (19) in (18) we get that

$$\overline{c} = \frac{c_o}{1 - \lambda P \overline{x}} \qquad (20)$$

The conditional cycle model [KUEHN 79] distinguishes between two types of cycles, which we will refer as type 1 and type 2 ones. During type 1 cycles a given station does not transmit any data packet and during type 2 cycles it does transmit.

Let us define some additional random variables.

$\tilde{c}_1$: duration of type 1 cycles.

$\tilde{c}_2$: duration of type 2 cycles.

and let $\overline{c}_1 \triangleq E[\tilde{c}_1]$, $\overline{c^2_1} \triangleq E[\tilde{c}^2_1]$, $\overline{c}_2 \triangleq E[\tilde{c}_2]$ and $\overline{c^2_2} \triangleq E[\tilde{c}^2_2]$

Let β_i be the probability that a station transmits a data packet during a type i (i=1,2) cycle.

Then, $\overline{c}_1$ and $\overline{c}_2$ can be written as a function of β_1 and β_2 as follows

$$\overline{c}_1 = c_o + (P-1)\, \beta_1\, \overline{x} \qquad (21)$$

and

$$\overline{c}_2 = c_o + (P-1)\, \beta_2\, \overline{x} + \overline{x} \qquad (22)$$

But, β_1 and β_2 can be expressed as a function of $\overline{c}_1$ and $\overline{c}_2$ by observing that in equilibrium, the average number of packets of a given station served during a cycle is equal to the probability that a particular station has at least one data packet ready for transmission. Thus,

$$\beta_1 = \lambda\, \overline{c}_1 \qquad (23)$$

and

$$\beta_2 = \lambda\, \overline{c}_2 \qquad (24)$$

Combining (21), (22), (23) and (24) we find that

$$\overline{c}_1 = \frac{c_o}{1 - \lambda \overline{x}\ (P-1)} \qquad (25)$$

and

$$\overline{c}_2 = \frac{c_o + \overline{x}}{1 - \lambda \overline{x}\ (P-1)} \qquad (26)$$

The second moments of type 1 and type 2 cycles can be obtained from the expressions (11a.) and (11b.) given in [KUEHN 79] for the conditional cycle time variance making the appropriate variable substitutions. In our case, after some algebraic manipulation we have that.

$$\overline{c^2_1} = (P-1)[\beta_1\, \overline{x^2} - (\beta_1\, \overline{x})^2] + (\overline{c}_1)^2 \qquad (27)$$

and

$$\overline{c^2_2} = (P-1)[\beta_2\, \overline{x^2} - (\beta_2\, \overline{x})^2] + \overline{x^2} - (\overline{x})^2 + (\overline{c}_2)^2 \qquad (28)$$

As indicated in [KUEHN 79], renewal theory arguments and Little's law can be used to obtain the average packet waiting time W: an arriving test packet of a given station meets either a type 1 or type 2 cycle in progress. Since we are considering Poison arrivals, the probabilities of meeting a type 1 or type 2 cycle are $(1-\alpha)\ (\overline{c}_1/\overline{c})$ and $\alpha\ (\overline{c}_2/\overline{c})$, respectively, where $\alpha = \lambda \overline{c}$ is simply the probability that a station has a data packet ready to transmit when it receives the token. Since

$\tilde{c}_1$ and $\tilde{c}_2$ are being considered iid random variables (an approximation), the average residual cycle times are given by $\overline{c_1^2}/2\,\overline{c}_1$ and $\overline{c_2^2}/\,2\,\overline{c}_2$ for type 1 and type 2 cycles respectively, according to renewal theory [KLEI 75]. Finally the average packet waiting time is the sum of the average residual cycle time plus the time to serve all data packets found in the queue. By Little's Law, the average queue length is λW and each packet in the queue takes $\overline{c}_2$ seconds, in the average, to be served. Then,

$$W = (1-\alpha).\,\frac{\overline{c}_1}{\overline{c}}\,.\,\frac{\overline{c_1^2}}{2\,\overline{c}_1} + \alpha.\,\frac{\overline{c}_2}{\overline{c}}\,.\,\frac{\overline{c_2^2}}{2\,\overline{c}_2} + (\lambda W)\,\overline{c}_2 \qquad (29)$$

Solving for W we have that,

$$W = \frac{\overline{c_1^2}}{2\,\overline{c}_1} + \frac{\lambda\,\overline{c_2^2}}{2(1-\lambda\overline{c}_2)} \qquad (30)$$

and the average packet delay D is given by

$$D = W + \overline{x} + \frac{v}{C} \qquad (31)$$

The throughput, S, is defined as the fraction of the total bus capacity, C, used to transmit data packets (excluding the token). Thus,

$$S = \lambda\, P\, \overline{x} \qquad (32)$$

3.3 - Numerical Results

Figure 2 presents average delay versus throughput curves for five values of the number of stations. Figure 3 shows the variation of the average throughput as a function of the number of stations for several values of the average packet arrival rate at each station. Finally, figure 4 shows delay versus number of stations curves for the average, lower and upper bounds. The parameters used to generate the curves in figures 2,3 and 4 are representative of the current version of REDPUC and are: C = 10 Mbps, v= 94 bits , r = 15 μs. Packet transmission time was assumed to be exponentially distributed with an average of 160 μs.

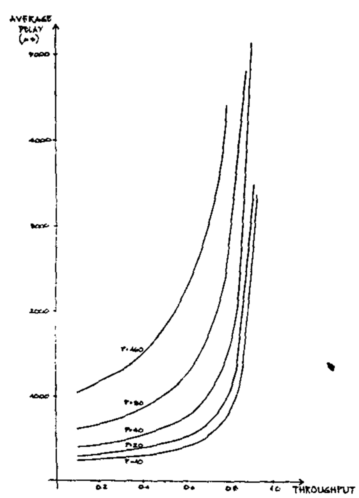

FIGURE 2 - AVERAGE DELAY VERSUS THROUGHPUT

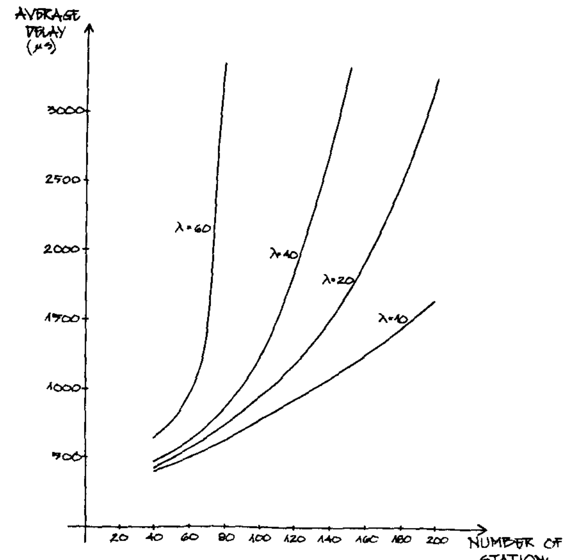

FIGURE 3 - AVERAGE DELAY VERSUS NUMBER OF STATIONS

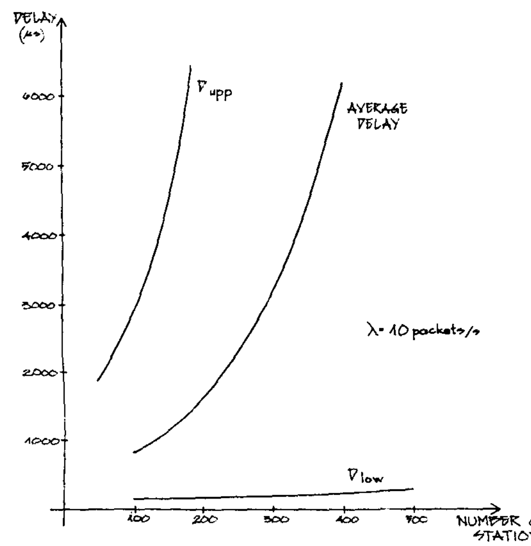

FIGURE 4 - DELAY VERSUS NUMBER OF STATIONS

Extensive simulations were carried out in order to validate the approximations used in the analytic model. The simulation program was written in Simscript II and it models the system with as few simplifying assumptions as possible.

Confidence intervals were obtained using the method based on the t-Student table ([LAW 82]). Each sample used a different seed for the pseudo-random number generator. Five samples were generated for each set of the parameter values. Each simulation run generated over 50.000 packets for all stations for each value of the seed. Under these considerations, 97.5 percent confidence intervals were obtained for the average packet delay as shown in Table 1, which presents a comparison between calculated and simulation results. As it can be seen, the approximation is very good even for high throughput and large number of stations.

		THROUGHPUT		
		0.256	0.512	0.768
P=10	CALCULATED	291	407	834
	SIMULATION	272	421	975
	Conf. Int.	±5.25	±8.78	±28.7
	Relative Error	6.9%	3.3%	14.5%
P=160	CALCULATED	1255	1881	4396
	SIMULATION	1150	1777	3936
	Conf. Int.	±5.25	±46.01	±461.5
	Relative Error	9.1%	5.9%	11.7%

Table 1 - Average Delay (in μs): Comparison between calculated and simulation results

4. Local Network Interconnection

From figure 3 one can observe that for a given load, per station the average packet delay increases in a non-linear fashion with the number of stations in the network. Therefore, if we partition the set of stations into several interconnected local networks, in such a way that most packets will not have to cross network boundaries to reach their destination, one may get a lower average end-to-end delay. In many local network applications, such as office automation, there are natural groups of user-stations which tend to exhibit higher volumes of intragroup message exchange if compared with intergroup communication (e.g. user-stations of the same department of a company). Another motivation for considering the partition of a single broadcast type local network into several interconnected networks lies in the fact that in the latter case several simultaneous transmissions may occur.

The set of local networks is interconnected by gateways, which are special purpose stations which implement the internetwork protocol.

In order to obtain expressions for the average end-to-end packet delay in the context of interconnected token bus networks, some definitions are in order. Let C_i, P_i, λ_i be, as before, the bus speed, number of stations and average arrival rate per station at network i, respectively.

Consider also the following input parameters.

P : number of stations in the set of interconnected networks.

R : number of interconnected networks

$p_{s,t}$: probability that a packet arriving at station s is bound to station t. ($p_{s,s} \triangleq 0$ for all s)

α_s : probability that an external packet arrives at station s.

G : number of gateways which interconnect the set of networks.

R_i : $= \{s_{i_1}, \ldots, s_{i_m}\}$: the set of stations of network i. (it does not include the gateways of networks i).

$\mathcal{J}_i$: $= \{G_\ell, \ldots, G_n\}$: the set of gateways in network i.

Let Π_{ij} be the path followed by a packet which originates at network i and has network j as its final destination. Let $\mathcal{R}(\Pi_{ij})$ be the set of networks which belong to the path Π_{ij}:

Let $\mathcal{R}(\Pi_{ij}) = \{R_i, \ldots, R_j\}$

$\mathcal{G}(\Pi_{ij})$ be the set of gateways in the path Π_{ij}:

$\mathcal{G}(\Pi_{ij}) = \{G_k, \ldots, G_\ell\}$

Let us define the performance measures of interest.

D_{ij} : average delay to send a packet from network i to network j.

D_{G_k} : average packet processing delay at gateway G_k.

D : average end-to-end packet delay

Clearly D_{ii} for any network is obtained using expression (31) with the appropriate values for network i, and an adjusted arrival rate per station described later.

Then the average delay, D_{ij}, to send a packet from network i to network j is given by

$$D_{ij} = \sum_{G_k \in \mathcal{G}(\Pi_{ij})} D_{G_k} + \sum_{\ell \in \mathcal{R}(\Pi_{ij})} D_{\ell\ell} \tag{33}$$

and the average end-to-end packet delay, D, is given by

$$D = \sum_{i=1}^{R} \left(\sum_{s \in R_i} \alpha_s \right) \sum_{j=1}^{R} D_{ij} \; t_{ij} \tag{34}$$

where t_{ij} is defined as the probability that a packet that arrives at network i goes to network j. This probability can be obtained as a function of $p_{s,t}$ and α_s in a straightforward manner:

$$t_{ij} = \frac{1}{\left(\sum_{s \in R_i} \alpha_s \right)} \sum_{s \in R_i} \alpha_s P_r[\text{packet arriving at } s \text{ goes to } R_j] \tag{35}$$

which implies that

$$t_{ij} = \frac{1}{\left(\sum_{s \in R_i} \alpha_s \right)} \sum_{s \in R_i} \alpha_s \left(\sum_{t \in R_j} p_{s,t} \right) \tag{36}$$

We are now left with the problem of finding the average packet processing delay at a gateway. Let,

I : number of instructions to process a packet at the gateway (in millions of instructions)

μ_G : gateway processing rate (in MIPS)

Then each packet takes I/μ_G seconds to be processed at the gateway. In order to estimate the average packet waiting time at the gateway, an M|D|1 model [KLEI 75] will be used for the gateway.

The average packet arrival rate, λ_{G_k}, at gateway G_k is the sum of the arrival rates at each network weighted by the probability that the packet crosses its originating network boundary and passes through gateway G_k in its way to the destination network. Therefore,

$$\lambda_{G_k} = \sum_{i=1}^{R} \lambda_i |R_i| \sum_{j=1}^{R} t_{ij} \, \epsilon_{ij} \, (G_k) \tag{37}$$

where

$$\epsilon_{ij} = \begin{cases} 1 & \text{if } G_k \in \mathcal{G}(\Pi_{ij}) \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

Finally, D_{G_k}, is given by

$$D_{G_k} = \frac{\lambda_{G_k} (I/\mu_G)^2}{2(1-\lambda_{G_k} I/\mu_G)} + \frac{I}{\mu_G} \tag{39}$$

A final adjustment in the value of λ_i has to be made for each network R_i in order to use the model presented in section 3. That model assumes that the average packet arrival rate is equal for all stations of the same local network. In the case of interconnected networks, the following approximation will be done: the load generated by all gateways of a given local network will be added to the load generated by user stations and divided by the number of stations (including the gateways) in the network in order to give the adjusted value of λ_i. Avoiding this approximation is still possible by obtaining the average waiting time at each station separately, using individual arrival rates as indicated in [KUEHN 79]. This would lead of course to a more complex model.

Hence, when applying the model described in section 3 to solve for network i, one should consider that the number of stations in this network, P_i, is equal to the number of user stations plus the number of its gateways. Namely,

$$P_i = |R_i| + |\mathscr{G}_i| \tag{40}$$

The average arrival rate at each of the P_i stations should be

$$\frac{\lambda_i |R_i| + \sum_{\substack{(j,\ell):R_i \in \mathscr{R}(\Pi_{j,\ell}) \\ j \neq i}} \lambda_j |R_j| t_{j\ell}}{P_i} \tag{41}$$

The expressions derived above allows us to obtain the average packet end-to-end delay for a set of interconnected networks independent of the topology and of the distribution of stations into networks.

Let us now consider some particular examples.

Let the external traffic be uniformly distributed through the set of stations and let the destination station be selected with equal probability among the set of stations. Let the number of stations in each network be the same and equal to N. So,

$$\alpha_s = \frac{1}{P} \qquad \text{for } s = 1, \ldots, P \tag{42}$$

$$P_{s,t} = \begin{cases} \frac{1}{P-1} & \text{for } s,t=1, \ldots, P \text{ and } s \neq t \\ 0 & \text{for } s = t \end{cases} \tag{43}$$

Then,

$$t_{ij} = \begin{cases} \frac{R\ N^2}{P(P-1)} & \text{for } i \neq j \\ \frac{RN(N-1)}{P(P-1)} & \text{for } i=j \end{cases} \tag{44}$$

The above expression follows directly from the general expression for t_{ij} given by formula(36) and from expressions(42) and (43). The reader should note that in this case N.R= P.

Now, consider three possible topologies for the interconnection of the networks: linear, ring and star.

Topology 1: Linear Partition

Consider the case shown in figure 5 where the set of stations is arranged into a set of linearly interconnected networks:

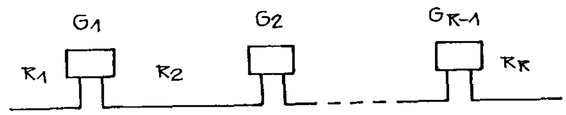

FIGURE 5 - LINEAR PARTITION

The delay, D_{ij}, can be obtained from expression (33) by considering

$$\mathscr{R}(\Pi_{ij}) = \{R_{\min(i,j)}, \ldots, R_{\max(i,j)}\} \tag{45}$$

and

$$\mathscr{G}(\Pi_{ij}) = \{G_{\min(i,j)}, \ldots, G_{\max(i,j)-1}\} \tag{46}$$

The values of ϵ_{ij} (G) to be used in expression (38) are given by,

$$\epsilon_{ij}(G_k) = \begin{cases} 1 & \text{if } \min(i,j) \leq k < \max(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{47}$$

Finally, the value of λ_{G_k} can be expressed in a straightforward manner if we observe that gateway G_k receives all traffic which originates at one of the k networks to its left and goes to one of the (R-k) networks to its right and vice-versa. Then, if $\lambda_i = \lambda$ for all networks,

$$\lambda_{G_k} = \frac{2\ \lambda N^3\ k\ (R-k)R}{P\ (P-1)} \tag{48}$$

Topology 2: Ring

The set of local networks is connected in a ring fashion as indicated in figure 6

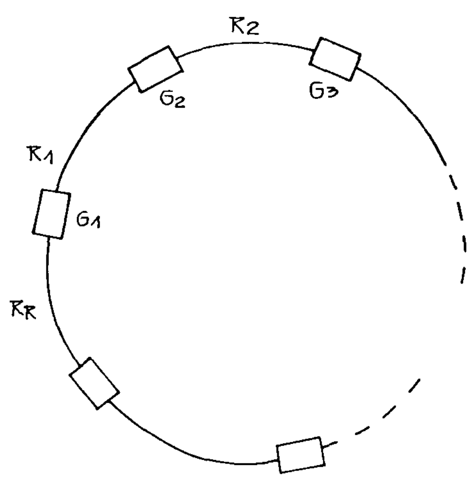

FIGURE 6 - RING TOPOLOGY

Routing in this case will be done following the path with the smallest number of gateways. If either direction from source to destination contains the same number of gateways, one of the two possible paths is chosen with equal probability.

So, for $\lambda_i = \lambda$ for every network and t_{ij} given by expression (44) the average packet arrival rate at each gateway is given by

$$\lambda_{G_k} = \frac{\lambda N^3 R}{P(P-1)} \left(\lfloor \frac{R}{2} \rfloor \left(\lfloor \frac{R}{2} \rfloor + R \bmod 2\right)\right) \tag{49}$$

The derivation of the above expression follows in a straightforward manner from expressions (37) and (44) and will not be included here due to space limitations. Two cases should be considered in obtaining this expression, namely: even number of networks and odd number of networks. Expression (49) integrates in a single formula the results for both cases.

The number of gateways traversed by a packet that goes from network i to network j is equal to $\min \{|j-i|, R-|j-i|\}$ and the number of networks crossed by the same packet is equal to $1 + \min \{|j-i|, R-|j-i|\}$

So,

$$D_{ij} = \min \{|j-i|, R-|j-i|\} D_{G_k} + (1+\min \{|j-i|, R-|j-i|\}) D_{ii} \tag{50}$$

D_{G_k} is obtained from expression (39) using the value of λ_{G_k} given by formula (49)

Topology 3: Star

As indicated in figure 7, in a star topology there is only one gateway through which all networks are interconnected.

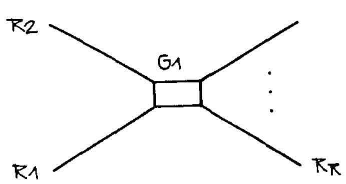

FIGURE 7 - STAR TOPOLOGY

$$D_{ij} = \begin{cases} D_{ii} + D_{G_1} + D_{jj} & \text{for } i \neq j \\ D_{ii} & \text{for } i = j \end{cases} \tag{51}$$

D_{G_1} is obtained by expression (39) with λ_{G_k} given by

$$\lambda_{G_k} = \frac{\lambda N^3 R^2 (R-1)}{R (P-1)} \tag{52}$$

4.1 - Numerical Results

In order to illustrate the results derived in the preceding section, several numerical examples of interconnected networks will be discussed here. The examples consider three types of topologies: linear partition, ring, and star. For each of the topologies, several values of the number of networks were considered. Also, for each case two types of inter-station communication probabilites were used. The first will be refered hereafter as the uniform case and implies that $p_{s,t}=1/(P-1)$. The second case, refered as the non-uniform case considers the following expression for $p_{s,t}$:

$$p_{s,t} = \frac{(P-|s-t|)^3}{\sum_{t \neq s} (P-|s-t|)^3} \tag{53}$$

The expression above implies that closer stations will have a higher probability of exchanging packets.

Consider the problem of interconnecting 320 stations in a local network. One possibility is to build a single local network into which all stations will be attached. An alternative approach would be to have several interconnected local networks. Which solution provides a lower average end-to-end delay? The models developed in the previous section allows us to answer to this and other questions. It will be considered for the rest of this section that C = 10 Mbps for the single network solution and C_i = 10 Mbps for every network i. The average packet arrival rate per station for all curves discussed below is 10 packets/s. It will also be assumed that each packet requires 200 instructions to be processed at each gateway. For the sake of comparison, the average end-to-end delay for the single network solution is 3570 μs.

The curves of figure 8 consider a ring topology for interconnecting the 320 stations into several interconnected networks. The gateway processing rate considered was 1 MIP. The number of networks varies from 2 to 32 networks. One of the curves considers uniform traffic and the other considers non-uniform traffic. As expected the non-uniform traffic case gives a lower average end-to-end delay. An interesting aspect of these curves is that the average delay has a minimum. This can be explained by observing that at the beginning, when the number of networks starts to grow, the number of stations per network decreases forcing the average intra network delay to decrease bringing the end-to-end delay down. On the other hand, the number of gateways also increases with an increase in the number of networks and after a certain point, the delay suffered by a packet in the gateways that it has to cross compensates for the decrease in the delay in each network traversed, forcing the average end-to-end dealy to increase again. Comparing with the single network solution, the interconnected network one provides a much lower average delay. The average delay in this case is roughly 1240 μs for the non-uniform case for six networks.

Figure 9 shows three delay versus number of networks curves, one for each topology for non-uniform traffic and a gateway with processing rate equal to 0.8 MIPS. The reader should be careful not to interpret these curves as a straightforward comparison between the topologies. Although one would be lead to think that the ring topology should always outperform the linear partition, one should remember that the linear partition has two networks (the end point ones) with one less gateways (one less station in the network). When the percentage of internal traffic is high, the intra

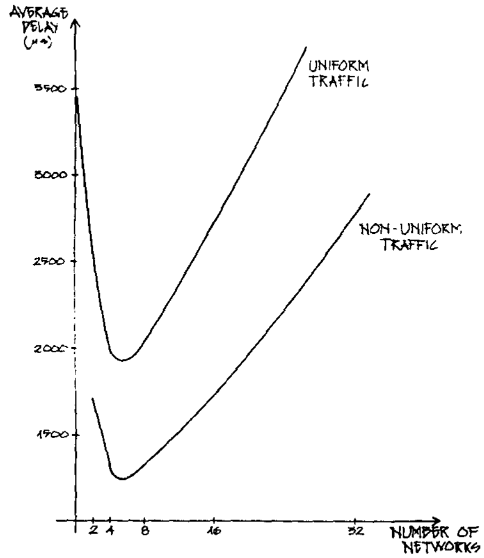

FIGURE 8 - AVERAGE DELAY VERSUS NUMBER OF NETWORKS

network delay tends to dominate the average end - to end delay. In some cases, depending on the average packet arrival rate and on the gateway processing rate, the linear partition case may start to exhibit better performance than the ring. As the number of networks increases, this behavior will tend to reverse. Another aspect not considered in this paper, but which should be taken into account when comparing different topologies, is the allocation of stations into networks. Consider the following problem: given the topology, the number of stations and the matrix $[p_{s,t}]$ find the allocation of stations into interconnected networks which minimizes the average end-to-end delay. This is an interesting optimization problem, which is a subject of ongoing research by the authors.

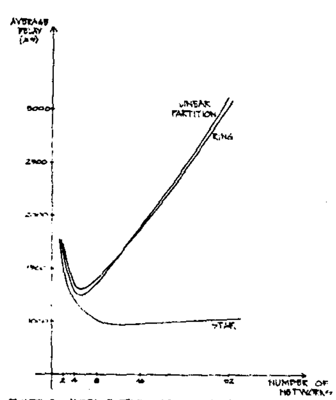

FIGURE 9 - AVERAGE DELAY VERSUS NUMBER OF NETWORKS

Finally, figure 10 shows three average delay versus gateway processing rate curves. This figure considers 8 networks with 40 stations each. The percentage of intra network traffic in this case is 28%. As expected, the gateway processing rate has a dramatic effect on the performance of the star topology, as can be seen.

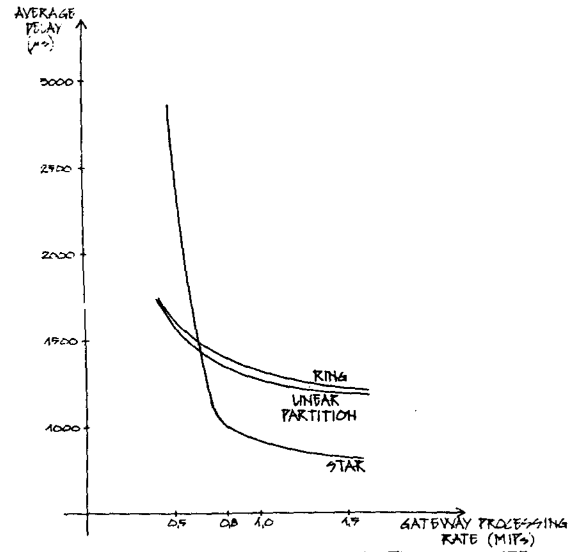

FIGURE 10 - AVERAGE DELAY VERSUS GATEWAY PROCESSING RATE

5 - Conclusions

A token bus local area network built at the Pontifícia Universidade Católica do Rio de Janeiro (PUC/RJ) was described. Lower and upper bounds on the average delay were derived. An approximation for the average delay was obtained using Kuehn's cyclic server model [Kuehn 79]. Simulation results are given to illustrate the accuracy of the model. The problem of interconnecting several local networks was treated and results for the average end-to-end delay are obtained. Numerical examples show that in certain cases a set of interconnected networks may exhibit a better performance that the single network solution for interconnecting a number of stations.

REFERENCES

[BUX 81] Bux, Werner, "Local Area Subnetworks: A Performance Comparison", IEEE Transactions on Communications COM-29 (1981), pp. 1465-1473.

[BUX 83] Bux, Werner and Hong L. Truong, "Mean-Delay Approximation for Cyclic-Service Queueing Systems", Performance Evaluation 3 (1983)pp.187-196.

[IEEE 82] IEEE, "Token-Passing Bus Access Method and Physical Layer Specifications", IEEE Project 802 Local Area Network Standards, Draft D, IEEE p 802.4/82/40, December 1982.

[KLEI 75] Kleinrock, Leonard "Queueing Systems: Theory", Vol. I, John Wiley, New York, 1975.

[KUEHN 79] Kuehn, P.J., "Multiqueue Systems with Nonexhaustive Cyclic Service", The Bell System Technical Journal, Vol. 58, nº3, March 1979.

[LAW 82] Law, A.M., Kelton, W.D., "Simulation Modeling and Analysis", McGraw-Hill, New York,1982.

[SOARES 83] Luis Fernando G.Soares, "Sintese de Protocolos para Redes Locais de Computadores"(On the Design of Protocols for Local Area Networks),Doctoral Dissertation, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, RJ, December 1983.

This research was supported by EMBRATEL, FINEP and CNPq.