A DECENTRALIZED 'ṣm' PROTOCOL FOR A LOCAL AREA NETWORK

by

Zvi Rosberg

Technical Report #310

February 1984
A Decentralized \( c \mu \) Protocol for a Local Area Network

Zvi Rosberg

Computer Science Department
Technion - IIT, Haifa, Israel

February 1984

Abstract

We consider a local area network. A lower bound to the long-run average waiting cost of the optimal decentralized access protocol is given. A new decentralized protocol is suggested and its long-run average waiting cost is bounded from above. For a typical local network these bounds are shown to be quite tight.

Keywords: Local area networks, Distributed protocols, Access protocols, Optimal scheduling, Single server queue.
1. Introduction

A local area network is composed of stations as processors, intelligent terminals and disks, interconnected through a common channel which supports digital communication over a limited geographical area. The stations are usually dispersed over an area which is less than a kilometer in diameter. The interconnection channel is wide-band, usually in the range of 1-50 megabits per second and is essentially error-free.

Most studies on access protocols have tried to achieve high throughput and small average delay. Several protocols were suggested and analyzed (e.g., CSMA/CD protocol in [Kito], BRAM protocol in [CFL], Ethernet in [MeBo], HYPERchannel in [TCJ] and a protocol in [HaSh]). A different approach was taken in [BuWa] where a 'quasi-optimal' implementation was suggested.

In this paper we restrict ourselves to the single bus topology (see [Bu]), however the technique which we use here, can easily be implemented to other topologies as well. As in [BuWa] we use information about the heterogeneity of the stations to design an access protocol.

Stations usually differ in their messages length, urgencies and time arrivals distributions. In previous studies, these differences were usually ignored for tractability of the analysis. Clearly, by considering them, performance can only be improved. Furthermore, we incorporate into the model (as in [BuWa]) the behavior of messages which require a reply as an acknowledgement.

This paper is motivated by the idea of [BuWa]. In contradistinction, we suggest a different access protocol which is compatible with the protocol proposed by the National Bureau of Standards [Bu]; by adding some small delay before each transmission we distributively implement the optimal centralized control. Moreover we analyze the model and provide a lower bound to the cost of the optimal decentralized protocol as well as an upper bound to the cost of our protocol. These bounds are shown to be quite tight when the end to end propagation delay is much smaller than the average
transmission time of a message.

In Section 2 we formulate a model for a local area network. In Section 3 we give a lower bound to the cost of the optimal decentralized protocol. In Section 4 we propose a new decentralized protocol and bound from above its cost in Section 5. A short discussion concludes the paper.

2. The Model

The network consists of \( n \) stations (see Fig. 1) connected to a coaxial cable through \( T \) tap port couplers. The \( T \) taps are already connected to the cable, however the designer is still free to locate the stations. (When the stations are also connected, the analysis is the same with some slight changes in the protocol).

The messages arrive at the stations as independent Poisson processes with rates \( a_{qi}, 1 \leq i \leq n, \sum_{i=1}^{n} q_i = 1 \). Every station is capable of storing in its buffer an infinite number of messages. The transmission time of a message sent by station \( i \) (excluding propagation delay) is a generally distributed random variable \( b_i \) with finite first and second moments \( \beta_{i,1}, \beta_{i,2} \) respectively.

Let \( p_{ij} \) be the probability that a message from station \( i \) is destined to station \( j \) and requires a reply from it. \( \sum_{j=1}^{n} p_{ij} \leq 1 \).

Let \( P = (p_{ij}) \), \( aq = (aq_1, aq_2, \ldots, aq_n) \) and \( \Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \) be the solution of \( \Lambda(I - P) = aq \), where \( I \) is the identity matrix.

If no message can generate an infinite sequence of replies then \( I - P \) is invertible (see [Kl]).

Denote this single server queuing system by \( \Theta = \{aq_i, b_i, p_{ij} | 1 \leq i, j \leq n \} \). It was shown in [Kl] that, if the channel does not pause when there are messages in the
buffers, the system is stable (ergodic) if and only if

$$
\sum_{i=1}^{n} \lambda_i \beta_{t,i} < 1.
$$

We assume that messages are transmitted without interruption.

To reflect the differences in urgencies among the stations we charge each message from station $i$, a cost of $C_i$ per unit of waiting time in the queue. The goal of this paper is to design a decentralized access protocol which will reduce the waiting cost of the messages.

Suppose for the moment, that every station has undelayed information about the buffer occupancies at all the stations and the access to the channel can be controlled without any time overhead (as propagation delay and switch on/off operation).

Under this assumption we have a time-sharing single server system and the long-run average waiting cost of using a protocol $\pi$ is (see [Kl])

$$
V(\pi) = \sum_{i=1}^{n} C_i E_{\pi}(l_i),
$$

where $E_{\pi}(l_i)$ is the expected buffer occupancy at station $i$ under stationary conditions (at steady state). Clearly, this expectation depends on the protocol $\pi$.

Let

$$
V = \inf_{\pi} V(\pi),
$$

where the infimum is taken over all protocols $\pi$ which use the buffer occupancies information and operate without time overhead. Clearly, $V$ bounds below the average waiting cost of the optimal decentralized protocol. In the next sections we explicitly express $V$ and define a decentralized protocol whose cost is close to $V$.

Remark 2.1 When $C_i = \frac{1}{\lambda_i}$, it follows from Little's Lemma that $V(\pi)$ is the long run average delay per message.
3. A Lower Bound

To express $V$ we use the results which were obtained in [Kl]. Throughout this section we assume the unrealizable assumption that every station has undelayed information about the buffer occupancies and the access can be controlled without any delay (i.e. a centralized control without time overhead).

It is this relaxation which yields the lower bound $V$. It was shown in [Kl] (see also [BuWa]) that the optimal protocol is a static priority rule. That is, after a transmission is ended, permission to transmit is given to the highest priority non-empty station; and when a message arrives at an empty system, it is transmitted immediately.

The optimal priority order among the stations is computed recursively

$$ C_i - \sum_{j \in B_k} p_{ij}(k)C_j $$

$$ i_k = \max_{i \in B_k} \frac{C_i}{C_i - \sum_{j \in B_k} p_{ij}(k)C_j}, \quad 1 \leq k \leq n, \quad (3.1) $$

where

$$ p_{ij}(1) = p_{ij}, \quad \beta_{i,1}(1) = \beta_{i,1}; \quad \Omega_1 = \{1, 2, \ldots, n\} $$

$$ \Omega_{k+1} = \Omega_k - \{i_k\} $$

$$ p_{ij}(k+1) = p_{ij}(k) + p_{i,ik}(k) p_{i,j}(k), \quad i, j \in \Omega_{k+1} $$

$$ \beta_{i,1}(k+1) = \beta_{i,1}(k) + p_{i,ik}(k) \beta_{i,j}(k), \quad i \in \Omega_{k+1} $$

Now, station $i_k$ has priority over station $i_j$ if and only if $k < j$

From [Kl], $E(t_k)$ under the optimal rule is explicitly expressed in the following form:

Let

$$ \lambda = \sum_{i=1}^{n} \lambda_i $$

$$ \rho = \sum_{i=1}^{n} \lambda_i \beta_{i,1} $$

$$ b_{ij} = a q_i \beta_{j,1} + p_{ij} $$

$$ e'_{ij} = \frac{1}{\lambda} \left\{ \sum_{k=1}^{n} \lambda_k (\alpha q_i q_j \beta_{k,1} + a q_i \beta_{k,1} p_{kj} + a q_j \beta_{k,1} p_{ki}) + 2 \lambda_i \lambda_j \delta_{ij} - \gamma \lambda_i b_{ij} - \lambda_j b_{ji} \right\} $$

Technion - Computer Science Department - Technical Report CS0310 - 1984
\[ e_{ij} = \begin{cases} 
\delta_{ij} & \text{for } i \neq j, \\
\frac{\delta_{ij}}{2} & \text{for } i = j.
\end{cases} \]

\[ \delta_{ij} = \begin{cases} 
1 & \text{for } i = j, \\
0 & \text{for } i \neq j.
\end{cases} \]

\[ a_{ij} = \delta_{ij} - \delta_{ij} \]

Define the following matrices and vectors:

For \( i < j \), let \( A_{ij} \) be a \( j \times i \) matrix which has non-zero elements only in its \( i \)-th row.

\[
A_{ij} = \begin{bmatrix}
0 & 0 & \ldots & 0 \\
a_{j1} & a_{j2} & \ldots & a_{ji} \\
0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0
\end{bmatrix}
\]

\[
A_{jj} = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1j} \\
a_{21} & a_{22} & \ldots & a_{2j} \\
\vdots & \vdots & \ddots & \vdots \\
a_{j1} & a_{j2} & \ldots & a_{jj}
\end{bmatrix}
\]

Also, let

\[
A = \begin{bmatrix}
A_{11} & 0 & 0 & 0 \\
A_{21} & A_{22} & 0 & 0 \\
A_{31} & A_{32} & \ldots & 0 \\
A_{j1} & A_{j2} & \ldots & A_{jj} \\
A_{n1} & A_{n2} & \ldots & A_{nn}
\end{bmatrix}
\]

\[
X_j = \begin{bmatrix}
x_{1j} \\
x_{2j} \\
\vdots \\
x_{jj}
\end{bmatrix}, \quad e_j = \begin{bmatrix}
e_{1j} \\
e_{2j} \\
\vdots \\
e_{jj}
\end{bmatrix}
\]

(3.2)
Let $X$ be the solution of the equation

$$AX = e.$$  \hspace{1cm} (3.3)

Now, identify the $i$-th station with the $i$-th priority. The long-run expected buffer occupancy at station $i$ using the optimal policy is

$$E(t_i) = \lambda \sum_{j=1}^{i} x_{ji} \beta_{j,i} + \frac{aq_i}{2} \sum_{j=1}^{i} \lambda_j \beta_{j,i} + \beta_{i,i} [1-(1-p)aq_i - \lambda_i].$$  \hspace{1cm} (3.4)

From (2.1), (2.2) and (3.4) we have

**Theorem 3.1** The long-run average waiting cost of a message using the optimal decentralized protocol is bounded below by

$$LB = \sum_{i=1}^{n} C_i E(t_i).$$

where $E(t_i)$ is given by (3.4)

When $p = 0$, the optimal priority order as well as the optimal $E(t_i), 1 \leq i \leq n$, are much simpler. This assumption can be made to approximate a model which has a negligible transmission time of acknowledgements comparing to the rest of the messages.

Under this restriction the optimal priority order becomes

$$i_k = \max_{i \leq k} \frac{C_i}{\beta_{i,i}}, \quad 1 \leq k \leq n.$$  \hspace{1cm} (3.5)

where
\[ \Omega_1 = \{1, 2, \ldots, n\}, \]
\[ \Omega_{k+1} = \Omega_k - \{i_k\} \]

Again, we identify the label of the station with its priority and obtain (see also [CMN, p.164])

\[ E(t_i) = \frac{aq_i \sum_{j=1}^{n} aq_j \beta_{j,2}}{2 \left[ 1 - \sum_{j=1}^{i-1} \rho_j \right] \left[ 1 - \sum_{j=1}^{i} \rho_j \right]}, \quad (3.5) \]

where

\[ \rho_j = aq_j \beta_{j,1}. \]

This policy is also known as the 'c \( \mu \)' priority rule. (c stands for the accumulated rate of waiting cost - \( C_i \); and \( \mu \) stands for the standard notation of the message completion rate - \( \frac{1}{\beta_{i,1}} \).)

In the next section we use the following properties to design a decentralized protocol whose average waiting cost is close to the lower bound LB.

1. When there is no time overhead and the buffer occupancies are known by all the stations, the optimal protocol is a static priority rule.
2. The end to end propagation delay in a local area network is relatively small with comparison to the average transmission time of a message.
3. The cost \( V(\pi) \) is a continuous function in \( \beta_{i,1} \), \( 1 \leq i \leq n \).

4. A Decentralized 'c \( \mu \)' Protocol

Consider a network with \( T \) taps located at nodes 1, 2, ..., \( n \). (The nodes are labeled from left to right as in Fig. 1.)

Let us connect the \( i \)-th priority station (according to (3.1)) to tap \( i \), thus the highest priority station is the most left one and the lowest priority is the most right one.

Let \( d_{ij} \) be the propagation delay between node \( i \) and node \( j \).
When the stations are already connected to the taps, the protocol will be slightly different. These changes are left to the reader.

We assume that every message receives a time stamp upon arrival. These time stamps are kept at the stations and are not transmitted.

The decentralized 'cμ' protocol which we propose is similar to the HYPERchannel protocol and to the protocol which was proposed by the National Bureau of Standards, [Bu]. The main difference stems from the fact that we use the time stamps to force transmission according to the 'cμ' priority order given in (3.1)

The buffer at each of the stations is handled as a FIFO queue. We use two types of messages, a token and a regular message. Adopting terminology from queuing theory, the protocol has two modes of operation.

(1) Busy period mode.

(2) Idle period mode.

During the busy period, regular messages are transmitted using a similar protocol to the HYPERchannel. During the idle period only the token is transmitted. A regular message contains its origin identity. The token contains the identity of the last transmitter and the time stamp of some message, if one has already arrived, or "high-values" otherwise.

For the sake of simplicity we assume that the transmission time of the token from one station to another is their propagation delay. (Clearly, the transmission time depends on the channel capacity and the implementation of the time stamp.)

The idle period starts when station n (the lowest priority) receives its turn to transmit and its buffer is empty; then, it originates the token. As soon as a regular message arrives at some station and the event is identified by the rest of the stations, the idle period ends and another busy period starts.

Let \( t_i \), \( 1 \leq i \leq n \), be the time required for a signal to propagate from station \( i \) to station 1 and back.
Clearly,

\[ t_1 = 0 \]

\[ t_{i+1} = t_i + 2d_{i,i-1}. \]

For simplicity we ignore the delay of the electronic devices and the algorithm execution time (they can easily be added to the \( t_i \)'s). The \( t_i \)'s are the only elements in the protocol which require a modification when the stations are already connected to the T taps.

The current time is available in the variable `TIME`. The token contains three fields: `token-origin`, `message-id`, and `message-time`.

Now, the algorithm at station \( i \), when the system has already started, is the following.

```plaintext
procedure initialization;
begin
  \( t_0 := 0 \);
  count := 0;
  direction := left;
  idle := false;
end;
```

```plaintext
procedure main;
begin
  while (true) do
  begin
    wait for an end of a regular or a token message;
    \( \tau := TIME \);
    if (a regular message) then
    begin
      if (idle = true) then initialization;
      busy - period;
      end;
    if (a token message) then idle - period;
  end;
```
procedure busy-period;
label finish;
begin
  j = {message origin};
  \( t_0 := r - d_y \);
  \( \text{wait for } t_i \text{ units; } \)
  if (a regular message has started transmission during the last \( t_i \) units) 
    then goto finish;
  if (there is a message in the buffer whose time stamp is smaller than \( t_0 \)) 
    then begin
        send the first message;
        goto finish;
    end;
  if (i=n) then 
    begin
      direction := right;
      token-origin := n;
      if (there is a message in the buffer) then 
        begin
          message-id := n,
          message-time := {time stamp of the first message};
          count := 2;
        end;
        if (the buffer is empty) then 
          message-time := {high-value};
          send the token,
    end;
finish: end.
procedure idle-period;

label finish, terminate;

begin idle := true;

if (the token is not from a next neighbor), then goto finish,

if ((token-origin = i+1 and direction = left) or
 (token-origin = i-1 and direction = right)) then

begin token-origin := i,

if (buffer is empty) then

begin send the token;

goto terminate,

end;

\[ t := \{\text{time stamp of the first message in queue}\} \]

\textbf{if} \((t < \text{message-time})\) \textbf{then}

\textbf{begin} message-time = t,

message-id := i,

count := 1;

\textbf{if} \((i=1 \text{ or } i=n)\) \textbf{then} count := 2;

send the token;

goto terminate;

\textbf{end};

\textbf{if} \((t > \text{message-time})\) \textbf{then}

\textbf{begin} send the token;

goto terminate;

\textbf{end};

\textbf{if} \((\text{message-origin} = i)\) \textbf{then}

\textbf{begin} if \((\text{count} = 1)\) \textbf{then}

\textbf{begin} count := 2;

send the token,
goto terminate;

end;

if (count=2) then send the first regular message in the queue
end;

end;

terminate: if (direction = left) then direction := right else direction := left,

if (i = n) then direction := right;
if (i = 1) then direction := left;

finish: end.

The algorithm starts when an end of a message is detected by the station. As long as regular messages are transmitted, the busy-period procedure is executed. During this period, messages are transmitted according to the priority order given in (3.1). This order is maintained by using the time stamps.

When station \( i \) detects an end of a regular message, it waits for \( t_i \) units before transmission. This waiting time is required for station \( i \) to receive an arbitrated transmission from the immediately higher priority port (station \( i-1 \)) if it has something to transmit. If nothing was transmitted during this waiting time (that is, stations \( 1,2,...,i-1 \) have nothing to transmit) station \( i \) will transmit a message provided its buffer contains a message whose time stamp is less than \( t_0 \) - the absolute finish time of the last transmission.

If nothing was transmitted during the waiting period of station \( n \), \( t_n \), and station \( n \) (the lowest priority) has nothing to transmit either, the current busy period ends and station \( n \) originates the token.

An idle period starts when a station receives the token for the first time after a busy period. During the idle period, every station transmits the token upon its turn.
The token travels through the stations as an elevator through the stores. It starts at station \( n \) moves toward station \( 1 \), then back toward station \( n \) and so forth. It continues traveling until a message enters the system and is identified by all the stations. Till the first arrival the token contains 'high-values'; afterwards it contains the time stamp and the identity of some message (not necessarily the first one).

When the token reaches a station, a comparison is made between the time stamps of the token and the first message in the queue (if the queue is not empty); if the station has a message with an earlier time stamp it updates the token.

If the token already contains the details of the first message in the queue and the details have been observed by the rest of the stations (condition: message-id=1 and count=2) then the station transmits the first regular message in its queue. This regular message starts a new busy period.

5. An Upper Bound

Let \( D \) be the time overhead per message of the decentralized 'c\mu' protocol. If the message starts a busy period, \( D \) is the time between its arrival and its beginning of transmission. Otherwise, \( D \) is the time between the end of the last transmission and the beginning of the current transmission. The end and the beginning of a transmission are measured by absolute time.

Let \( \Theta = \{aq, b, pq \mid 1 \leq i, j \leq n \} \) and \( \Theta' = \{aq, b_i + D, pq \mid 1 \leq i, j \leq n \} \) be two systems which differ only by the message transmission times.

Let \( \pi^* \) be the optimal priority order for system \( \Theta \), as given in (3.1). Regard the time overhead \( D \) of our decentralized protocol in system \( \Theta \) as transmission time. Furthermore, attach it to the real transmission time of the message immediately follows this time overhead. Thus, we obtain the queuing system \( \Theta' \) under the same priority protocol \( \pi^* \). Therefore,
Proposition 5.1 the local area network under the decentralized 'cμ' priority protocol is stochastically equivalent to the queuing system θ under the priority service policy π.

From (2.1), the local area network under the decentralized 'cμ' protocol is ergodic if and only if

\[ \sum_{i=1}^n \lambda_i (\beta_{i,1} + E(D)) < 1. \]

Let \( \bar{d} \) be a constant which bounds \( D \) from above. Now consider the queuing system

\[ \bar{\Theta} = \{ a_{ij}, b_i + \bar{d} \eta_{ij} \mid 1 \leq i, j \leq n \} \]

under the 'cμ' priority policy π.

Let \( E_{\pi^*}(l'_i) \) be the long-run expected buffer occupancy at station \( i \) in system \( \Theta' (\bar{\Theta}) \) using policy π as given in Section 3.

By stochastic dominance

\[ E_{\pi^*}(l'_i) \leq E_{\text{eq}}(l'_i) \quad (5.1) \]

\( E_{\text{eq}}(l'_i) \) is obtained from (3.3) and (3.4) by replacing \( \beta_{i,1}, \beta_{i,2} \) with

\[ \bar{\beta}_{i,1} = \beta_{i,1} + \bar{d}, \quad \bar{\beta}_{i,2} = \beta_{i,2} + \bar{d}^2 + 2\bar{d}\beta_{i,1}. \quad (5.2) \]

It is easy to verify that for a message which starts a busy period, \( D \) is bounded by \( 4d_{1,n} \) and by \( 3d_{1,n} \) for the rest of the messages. Thus

\[ \bar{d} = 4d_{1,n} \quad (5.3) \]

5.1 The General Case

Let \( V_\Theta(\pi^*) \) be the expected waiting cost using the decentralized 'cμ' protocol.

From (5.2) and (5.3) we have a computable expression for \( E_{\pi^*}(l'_i) \) and from (5.1) and Proposition 5.1 we obtain the following theorem:
Theorem 5.1

$$V_{\theta}(n^*) \leq \sum_{i=1}^{n} C_i E_{n^*}(\tilde{l}_i).$$

By standard error analysis technique from numerical analysis theory (see e.g. [RaRa, p. 430]) it can be shown from equations (3.3), (3.4) that

$$\frac{E_{n^*}(l_i) - E_{n^*}(l'_i)}{E_{n^*}(l'_i)} \quad 1 \leq i \leq n$$

are small if $\frac{d_{1,n}}{\beta_{i,1}}$ are small.

In the general case we do not specify the close form for the expressions in (5.4). However, it can be shown by straightforward calculation, that they are bounded by the condition number of the matrix $A$ (see (3.3)). $||A|| \cdot ||A^{-1}||$, times a second order polynomial function of the variables $\frac{d_{1,n}}{\beta_{i,1}}$, $1 \leq i \leq n$. ($||A||$ is any norm of the matrix $A$)

When $p_{ij} = 0$ we do give a close form for the upper bound.

5.2 The case $p_{ij} = 0$

A model with $p_{ij} = 0$ is a good approximation to a model where the transmission time of acknowledgements are negligible.

Let $\varepsilon = \max_{i=1}^{n} \frac{4d_{1,n}}{\beta_{i,1}}$

From (5.2) and (5.3)

$$\bar{\beta}_{i,1} \leq (1 + \varepsilon)\beta_{i,1}, \quad (5.5)$$

$$\bar{p}_{i} \leq (1 + \varepsilon)p_{i}. \quad (5.6)$$

Moreover,

$$\bar{\beta}_{i,2} = \beta_{i,2} \left[ 1 + \frac{(4d_{1,n})^2 + 8d_{1,n} \beta_{i,1}}{\beta_{i,2}} \right].$$
\[
\leq \beta_{i,t} \left[ 1 + \frac{(4d_{1,n})^2 + \delta d_{1,n} \beta_{i,t}}{\beta_{i,t}^2} \right] \\
\leq \beta_{i,t} (1 + 2\varepsilon + \varepsilon^2).
\] (5.7)

Further, let \( \rho_{(i)} = \sum_{j=1}^{i} \rho_j \).

From (3.6), (5.5)-(5.7)

\[
\frac{E_{\pi'}(L)}{E_{\pi'}(P_1)} \leq (1 + 2\varepsilon + \varepsilon^2) \frac{(1-\rho_{(i-1)})(1-\rho_{(i)})}{(1-\rho_{(i-1)} - \varepsilon \rho_{(i-1)})(1-\rho_{(i)} - \varepsilon \rho_{(i)})}.
\] (5.8)

Let \( \alpha_i = \frac{(1-\rho_{(i)})(1-\rho_{(i-1)})}{\rho_{(i-1)}(1-\rho_{(i)}) + \rho_{(i)}(1-\rho_{(i-1)})} \)

and \( \alpha = \min_{1 \leq i \leq n} \alpha_i \).

By standard algebraic manipulation

\[
\frac{(1-\rho_{(i-1)})(1-\rho_{(i)})}{(1-\rho_{(i-1)} - \varepsilon \rho_{(i-1)})(1-\rho_{(i)} - \varepsilon \rho_{(i)})} \leq 1 + \frac{\varepsilon}{\alpha_i} \leq 1 + \frac{\varepsilon}{\alpha}.
\] (5.9)

From (5.8) and (5.9)

\[
\frac{E_{\pi'}(L)}{E_{\pi'}(P_1)} \leq 1 + \varepsilon \left[ 2 + \varepsilon + \frac{1+2\varepsilon+\varepsilon^2}{\alpha-\varepsilon} \right].
\] (5.10)

The following theorem shows that the cost of the decentralized 'cµ' priority protocol is relatively close to the lower bound of the optimal decentralized protocol.

**Theorem 5.2** If \( \rho_{ij} = 0 \) then

\[
\frac{V_{\theta}(\pi')}{LB} \leq 1 + \varepsilon \left[ 2 + \varepsilon + \frac{1+2\varepsilon+\varepsilon^2}{\alpha-\varepsilon} \right]
\]

**Proof:** From Theorem 3.1 and Proposition 5.1
To demonstrate the ratio between the average waiting cost of the decentralized $c\mu$ policy and lower bound, consider the Ethernet over a 1 kilometer distance between the extreme nodes at the busiest minute. From the specifications and the statistical data which are given in [DIG, Ch. 10] it follows that $d_{1,n} = 3.3\mu s$, the average transmission time of a message is about 800 $\mu s$ and the channel utilization is 17 percent. Thus, $\varepsilon \leq 0.16$ and $\alpha = 2.44$.

Now, from Theorem 5.2 it follows that the average waiting cost of the decentralized $c\mu$ protocol is within at least 104% of the lower bound.

6. Discussion

For practical reasons one can simplify the decentralized $c\mu$ protocol by avoiding the time stamps and letting station $j$ (after the $t_i$ unit waiting period) to transmit a message if it has one, regardless of its time stamp.

This clearly destroys the $c\mu$ priority order. However, one would expect both protocols to behave similarly. It is not easy to analyze the protocol without the time stamps and the designer of the local area network should use a simulation study for this purpose.

A token passing protocol with the conjunction of time stamps can also be used to maintain the $c\mu$ priority order. The performance will not be much different from the performance of the protocol suggested in Section 4.
REFERENCES


Figure 1 - Local Area Bus Network