TOPOLOGY RELATED ARBITRATION OF A SHARED COMMUNICATION CHANNEL

by

I. Chlamtac and W.R. Franta

Technical Report 282

June 1983

* Computer Science Dept., Technion-IIT, Haifa, Israel.

** Computer Science Dept., University of Minnesota, Minneapolis Minnesota, USA.
ABSTRACT

We describe a model of a general topology distributed system with a shared communication resource. The access to the resource is sequential and users are required to cooperate in order to achieve sequential-interference free access in minimum average time. A positive cost which is a function of users' location in the system is involved in exchanging information needed for user cooperation.

A set of policies based on user negotiation via reservation is presented and analyzed. It is shown that an optimal policy can be found under which groups of users negotiate in parallel to choose representatives who negotiate serially for resource access. The optimal group size is found to be a function of the system topology, the cost of communication and the rate of resource requests generation in the system. The relation between this model and existing communication systems is presented and their behavior under the optimal policy is analyzed.

Key words: optimal communications resource sharing, allocation policy, distributed system, network topology, delay analysis.
1. INTRODUCTION

In many communication systems a single communication channel is shared between the users. In satellite, radio and other distributed communication systems the channel is allocated to a single user at a time. To obtain such allocation the actions of the distributed users have to be coordinated by means of a channel access communication protocol. A large number of protocols have been proposed and implemented so far [3,4,5].

Most of these protocols use timing, reservation or channel sensing mechanisms to achieve orderly channel access. The protocols have been analyzed and it was shown how channel sharing can be affected to obtain, maximum channel throughput [12,13].

These research efforts have concentrated on the design and optimization of protocols in which the system is fully connected by the communication channel. In other words, a channel access, or transmission made by any user will be received by all remaining users.

This assumption is central to protocol design since it also affects the mechanisms that can be used by the users to coordinate their actions and it reduces the importance of the system topology in the protocol design process. In this paper we investigate a different approach to the protocol, channel allocation problem. We assume an arbitrary system topology where users must explicitly negotiate for channel access rights by use of reservation. A pair of users wishing to communicate must forward the (reservation) message in a way which accounts for the topology so that, naturally, the communication cost may differ from pair to pair depending on the users' location in the network. It is shown that for a given topology an optimal negotiation process can be found which minimizes
the delay cost involved in the channel allocation process in an arbitrary network.

In terms of current systems such protocol can be considered for satellite, radio and local communication networks. With satellite communication we look at the case where a network of ground stations with arbitrary topology is to be allocated the satellite link for communication with other satellite covered regions. To obtain an orderly access under existing reservation protocols the satellite link is used as the reservations carrier [14]. With the proposed approach the users can alternatively conduct direct negotiations using terrestrial facilities, so that the cost of reservation due to satellite associated propagation delays can be significantly reduced. The case of radio networks with general topology introduces the so called hidden node problem [2,3,5]. By the use of explicit, topology optimized, reservation algorithm the channel allocation can be obtained in these systems so that interference from mutually hidden nodes (i.e., nodes not in line of sight) is avoided, while the reservation cost is minimized.

Lastly, we can consider local area networks with general topologies as candidates which can potentially benefit from a topology dependent access algorithm. For instance, in loop networks utilizing token passing protocols nodes are queried in their order on the loop, until a ready user is found. The negotiation delay depends on the loop size and may become significant especially for low traffic loads [6]. In braided loops, double loops or look ahead loops, additional communication links are introduced to improve reliability. The total
communication paths give rise to a more complicated topology [15, 6].

The proposed protocol can utilize these additional communication links to achieve a more efficient channel reservation by relating the token negotiation process to the topology.
2. THE MODEL

We view the distributed system as a connected graph $G_k$ consisting of $K$ nodes where each arc corresponds to a communication link. A fixed cost $a_{ij}$ is associated with each communication, message passing, between two connected nodes $i$ and $j$. The cost of communication between two nonadjacent nodes is the sum of costs incurred in several link traversals, or hops, needed to pass a message. For a given routing algorithm $R$, let $d_{ij}$ represent the number of hops between nodes $i, j$, with the maximum path cost bounded by

$$K' = \max d_{ij} \cdot a, \quad a = \max a_{ij} \text{ for all } i,j.$$ 

We assume that nodes generate resource requests at random so that only some of the users will be active at any time. We want to effect a sequential channel allocation in which active nodes negotiate access by means of reservation packets and a timing mechanism. The nodes are ordered and distinguishable (i.e., have unique names) in every negotiation process. The highest order active node is allocated the channel and will engage it until all its accumulated requests are honored.

Under a general reservation policy starting at $T_r$ node $i$ can be in states:

S1. Waiting $t_{\text{reservation}} \leq w_1(i)$ to transmit reservation following $T_r$.

S2. Waiting $t_{\text{data}} \leq w_2(i)$ to transmit data following $t_{\text{reservation}}$.

Condition (c): The waiting times $w_1, w_2$ (in units of $a$) must together guarantee that in every negotiation process a lower node will defer to reservations made by higher order nodes.
Combining states S1, S2 with condition (c) a set of allocation policies is obtained as follows:

For active node i, starting at \( T_r \),

\( (* \ T_r \ - \ the \ beginning \ of \ the \ next \ negotiation *) \)

Procedure Elect \((w_1(i),w_2(i),T_r)\);

begin
while not higher priority reservation received do
begin
wait \( w_1(i) \);
send reservation;
wait \( w_2(i) \);
send data (* engage resource *)
end;

If a higher priority reservation is received following \( T_r \) and prior to resource acquisition, the procedure is repeated from next \( T_r \).

Comments: 1. Reservation information is relayed to all nodes.

2. Each relay of reservation between adjacent nodes is bounded by \( a \), taken as the basic unit of time, s.t. the maximum delay incurred in relaying reservation information is bounded by \( K' a \).

In the set of allocation policies given by Elect \((w_1,w_2)\) we define the optimal policy as that which minimizes the average node access delay to the resource for a given topology \( G_k(K,K',R) \), communication cost \( a \), and the total request arrival rate \( \lambda \).

To restrict the search for potential optimal policies we first establish bounds on \( w_1 \). With \( K' \) the bound of communication cost and from procedure Elect we have: \( 0 \leq w_2 \leq K' \), and similarly from condition (c) \( w_1(i) + w_2(i) = K', \forall i \).
We thus obtain the following boundary policies:

(Sequential) \( w_2(i) = 0, \quad w_1(i) = i \cdot K' \), and

(Parallel) \( w_2(i) = K', w_1(i) = 0, \forall i \).

With the sequential policy we obtain for the reservation interval 
\( KI \): the duration of negotiation beginning at \( T \) and ending at channel acquisition:

\[
0 \leq KI \leq K \cdot K'a \quad \text{with} \quad RI(\lambda \to 0) = K \cdot K'a, \quad RI(\lambda \to \infty) = 0. \tag{1}
\]

For the parallel policy we have

\[
RI = K'a, \quad \text{for all} \quad \lambda \to s. \tag{2}
\]

In other words, the "reservation cost" curves for the two boundary policies cross at \( \lambda = \lambda^* \), so that for \( \lambda < \lambda^* \) the parallel policy and for \( \lambda > \lambda^* \) the sequential policy incur lower cost.

Let us now choose \( w_2 = \xi, 0 < \xi < K' \). For a given \( \xi \) and routing algorithm \( R \) a reservation sent by node \( i \) will be received by all nodes no more than \( \xi \) hops away from node \( i \).

Consequently, we can reduce the total number of sequential negotiations by collecting nodes into subgraphs of diameter \( \xi \) so that:

1) nodes in a given subgraph \( M \) send reservation following \( w_1(M) \) and 2) within \( w_2(M) = \xi \) all nodes in \( M \) can establish highest order active node in the group. In other words, groups of nodes conduct "sequential" negotiations, while nodes in the same group negotiate in "parallel". To reduce the total cost \( R_I \) equations (1) and (2) above suggest that for a given request arrival rate an optimally balanced sequential/parallel negotiation process exists,

Let us assume \((M_0, M_1, \ldots, M_m)\) to be an ordered partition of \( G_k \) into \( m \), the minimal number, of groups with maximum diameter \( \xi \).
From condition (c) node $\epsilon M_i$ may engage the channel only if no active node $\epsilon \cup M_j$ exists. Consequently we must have:

$$w_1(M_i) = (w_1(M_{i-1}) + K' - w_2(M_{i-1})) \cdot a$$

and setting $w_1(M_o) = 0$ we have:

$$w_1(M_i) = (K' - a)_i \cdot a$$

$$w_2(M_i) = K', \ 0 \leq k \leq K' \ \forall i$$  \hspace{1cm} (3)

Equation (3) represents discrete time policies obtained by making reservation/acquisition decisions in time multiples of $a$, the basic cost of communication between connected nodes. Setting $w_2$ to the two extreme values 0 and $K'$ we again obtain the (parallel) and (sequential) policies defined earlier.

With $w_1, w_2$ cost given by (3) we see that negotiations in the successive groups $M_o, M_1, ...$ partially overlap so that as suggested earlier all policies in (3) represent various combinations of the sequential and parallel policy. The total cost of negotiation given by the reservation interval $R_i$ in (5) will be:

$$R_i = \sum_{j=1}^{m} w_1(M_j)x_j + w_1(M)$$

where $m$ - the total number of groups in the network and

$$x_j = \begin{cases} 
1, & j \leq i \\
0, & j > i 
\end{cases}$$

with $i$ - the index of the first group with at least one active node for a given order in each reservation interval.

Figure 1 shows the reservation intervals resulting from the allocation policies discussed above.
3. CHOOSING THE OPTIMAL ALLOCATION POLICY

We assume \( G_k, a, K', \lambda \) and \( m \) as defined before and that each request is of a fixed duration \( T \). We shall normalize \( a \) by \( T \) and without lack of generality we choose \( T = 1 \). We further assume that the system is synchronized in basic time slots of length \( a \).

We define \( \lambda \) to be the total arrival rate of requests at nodes from homogeneous Poisson processes, \( D \) the expected resource access, or data transmission delay and \( S \) the channel resource utilization normalized by \( T \).

As before we set the reservation interval to begin at \( T_{r} \), the end of preceding resource acquisition, or in case no active nodes are present at this time the negotiation will begin at the first \( m.a \) multiple following the time of first request arrival at one or more nodes counting from \( T_{r} \). With these assumptions we develop the delay formulas and the following optimization algorithm.

3.1 Expected Channel Access Delay

Consider the sequence of channel idle and busy periods shown in Fig. 2. When \( m = 1 \), the only allowable values is \( \lambda_1 = K' \).

When a node becomes active it must wait until the beginning of the next \( a \)-slot to schedule its transmission request. The busy period consists of a series of \( 1 + \lambda_1 a \) blocks such that within each block the first \( a \)-slot is used for reservations transmission, the next \( \lambda_1 - 1 \) \( a \)-slots for relay of reservation throughout the network, followed by the data transmission. With these observations the expected delay,
The delay $D_1$, when $m = 1$ is given by:

$$D_{m=1} = (1 + k_1 a) + \frac{1}{2} \frac{\rho'}{1 - \rho'} + \frac{a}{2} (1 - \rho')$$  \hspace{1cm} (4)$$

with $\rho' = S(1 + k_1 a)$ and $S$, channel throughput - the number of data packets transmitted per unit time. The result is had by observing that the channel can be viewed as a server in an M/D/1 queue, with the transmission time for each packet transmitted given by $(1 + k_1 a)$ time units. Expected delay is then given as the sum of packet transmission time (the first term of (4)) plus the expected wait an active user node suffers in an M/D/1 queue (the second term of (4)) plus the delay suffered by the active node (packet arrival) which ends the idle period, times the probability of its occurrence (the third term of (4)). From (4) substitution of $\rho'$ gives

$$D_1 = 1 + k' a + \frac{1}{2} \frac{S(1 + k'a)}{(1 - S(1 + k'a)))} + \frac{a}{2} (1 - S(1 + k'a))$$  \hspace{1cm} (5)$$

To realize an expression for $D_m$, $1 < m \leq K$, the $m$ groups can be considered as being "polled" within the reservation interval. For purposes of this view, the mechanism by which the nodes in a group select the node with highest priority is unimportant.

We have as before that each transmitted packet requires $1 + k a$ normalized units of channel time, so that under steady state the fraction $\rho$, of channel busy time is given by

$$\rho = (1 + k a) \cdot S$$  \hspace{1cm} (6)$$

For a stable channel we must have $0 \leq \rho < 1$. 

Technion - Computer Science Department - Technical Report CS0282 - 1983
To obtain \( D_m^k \) we begin with the formula

\[
D = 1 + \frac{-\rho}{2(1-\rho)} + \frac{a}{2} \left[ 1 - \frac{\rho'}{m} \right] \left[ 1 + \frac{m}{1-\rho'} \right],
\]

(7)

with \( \rho' \) the channel utilization. Equation (7) is obtained by mapping the Konheim-Meister formula, [6], for loop networks with \( r \), representing the polling delay required for transferring control from one node to another, in a fixed round robin sequence. To adapt (7) to our model we can establish an expression for \( \rho' \), and an expression for \( r \), by noticing \( \rho' = \rho \) as given by (6), and

\[
r = K' - k
\]

(8)

which is established by noting that a delay of \( K' - k \) a-units is required to pass control to the next node or group of nodes in the logical or virtual polling sequence, in the distributed system.

Using (6) and (8) in (7), and simplifying yields (9).

\[
D_m^k = 1 + \frac{a}{2} + \frac{S+S_{ka}}{2(1-S-S_{ka})} + \frac{a}{2} \left( 1 - \frac{S+S_{ka}}{m} \right) \left( 1 + \frac{mK'-ml}{1-S-S_{ka}} \right)
\]

(9)

where integer \( k \) is constrained by \( m \leq k \leq K' \), as determined by the diameter of the subgraphs for a given \( G_k \) and \( m \). Notice that setting \( m = 1 \) in (9) yields (5) so that (9) is valid for \( m \), \( 1 \leq m \leq K \).

This analysis is strictly correct if and only if the aggregate packet arrival rates for the groups are equal, otherwise (9) must be considered an approximation, a point discussed further in Section 5.

To establish the normalized channel capacity \( C \) for a given \( k \), we note that the channel is saturated when one node (the highest priority node) captures it for an indefinite but long period.
Since each transmission requires \( 1 + \frac{\lambda a}{a} \) normalized units of channel time of which only 1 unit is used to actually transmit the packet, we obtain

\[
C = \frac{1}{1 + \frac{\lambda a}{a}}
\]

(10)

3.2 Finding the Optimal Policy

For given \( \alpha, K, K' \) we wish to choose parameters \( \ell \) and \( m \) which minimize the total delay for a given throughput \( S \).

Notice that the reservation delay \( R_I \),

\[
R_I(\alpha, m, \ell) = (K' - \ell)a + 1 + \frac{\lambda a}{a}
\]

(11)

with \( i \) an integer, \( 1 \leq i \leq m \), is determined by the time needed to locate the first group with an active node.

When \( S \to 0 \), the probability of a group having a ready node becomes small so that \( R_I \) tends to its worst case expected length given by

\[
(K' - \ell)a \cdot m + \frac{\lambda a}{a}
\]

with \((K' - \ell)a \cdot m\) dominant term in (11) so that \( R_I \) is minimized by setting \( \ell = K' \) yielding \( m = 1 \).

For large \( S \) values, \( S \to \infty \), few \((K' - \ell)a \) slots will pass before one is used so that \( \ell \cdot a \) becomes the dominant term in (11) and \( R_I \) is minimized by setting \( \ell = 0 \), yielding \( m = K \).

These observations suggest that for a given \( S \) there exists a pair \((m, \ell)\) which optimally balances the two terms of (11).
Definition: For a given $S$, we call optimal that $(m, \lambda)$ pair which produces the minimum expected delay $D$, over the range of allowable $(m, \lambda)$ values.

From the preceding discussion we have:

1. For $S \leq 0$ the optimal values are given by $m = 1$, and $\lambda = K'$,
2. For $S \geq C$ the optimal values are $m = K$, $\lambda = 0$.

It remains to determine the $S$-ranges over which settings (1) and (2) are optimal, and to determine the optimal $(m, \lambda)$ for intermediate $S$ values. To obtain an efficient optimization procedures we next significantly reduce the number of curves to be considered for optimization.

Let (GP) denote a grouping algorithm which produces an optimal partitioning, i.e., for a given $\lambda$ it produces the minimum number of groups, $m$, as given in Appendix A.

For $\lambda$, $0 \leq \lambda \leq K'$ we may, of course, obtain the same $m$ value for several consecutive $\lambda$ values. Let us define $\hat{\lambda}_m$ to be the smallest of the $\lambda$ values for which the same value $m$ is obtained. Similarly, for $m$, $1 \leq m \leq K$, there exist $m$ values which are not produced by optimal partitioning.

Let us define $\underline{m}$ to be the vector of $m$ values produced by (GP). Together we define the vectors $\lambda$, $m$ by:

$$
\lambda = (\hat{\lambda}_m, \hat{\lambda}_m, \ldots, \hat{\lambda}_{m_{\text{max}}})
$$

$$
m = (m_1, m_2, \ldots, m_{\text{max}}).
$$
Theorem: The only \( m \) values which must be considered to minimize delay are those contained in \( \mathbb{m} \). For each \( m \in \mathbb{m} \), the only \( \ell \) values which must be considered are \( \ell = \hat{\ell}_m \) and \( \ell = K' \).

Corollary: For a given \( m \), the minimum \( D \) in (9) over the range \( 0 \leq S \leq C(\ell) \) is given by setting

\[
\ell = \begin{cases} 
K' & 0 \leq S \leq S'(m) \\
\hat{\ell}_m & S > S'(m)
\end{cases}
\]  

(12)

where

\[
S'(m) = \frac{1}{2} (S + T) - \frac{W}{3V} - \frac{1}{2} \sqrt{-3(S - T)},
\]  

(13)

\[
U = am^2(K'-2) - aK' - 2m(K'-2) + m^2(K'-2) - 2amK'(K'-2)
\]

\[
V = a(K'-2) - 2aK' + a^2(2K'(K'-2)^2 - a^3K'(K'-1)
\]

\[
W = 3aK' - a(K'-2) + a^2(2K'(K'-2)^2 + m(K'-2) + amK'(K'-2) + a^2m(K'-1)(K'-2)
\]

\[
Z = 2m(K'-2) - m^2(K'-2)
\]

and

\[
Q = \frac{3UV - W^2}{9V^2}
\]

\[
R = (9WUV - 27ZV^2 - 2W^3)/(54V^3)
\]

\[
S = (R + (Q^3 + R^2)^{1/2})^{1/3}
\]

\[
T = (R - (Q^3 + R^2)^{1/2})^{1/3}
\]

For proof of corollary, see Appendix B.
Proof of Theorem. Put differently, Theorem 1 guarantees 1), that \((m, \xi)\) pairs which are topologically possible but not produced by grouping algorithm \((GP)\) are not required i.e. would never appear in a "composite minimal delay curve" and further, 2) among \((m, \xi)\) pairs for which \(m\) is repeated, only the pair with smallest \(\xi\) need be considered. We consider each part in turn.

To show that \((m, \xi)\), pairs not in \((m, \xi)\) are not included in the composite minimal delay curve, imagine any other grouping algorithm which given \(m\) produces groups such that the largest group diameter is \(\tilde{\xi}\). The alternate algorithm thus produces the pair \((\xi, \tilde{m})\). From the output of the grouping algorithm we find the entry \((m, \tilde{\xi})\) associated with \(\tilde{\xi}\). (An entry for \(\tilde{\xi}\) must be present as the grouping algorithm produces groups for all \(\xi, 1 \leq \xi \leq \xi'\).) By the nature of the grouping algorithm \((GP)\) we must have \(\tilde{m} \geq m, \xi\). If \(\tilde{m} = m, \xi\) then the pair is in \((m, \xi)\) i.e. the alternate algorithm and grouping algorithm \((GP)\) have produced the same pair. If \(\tilde{m} > m, \xi\) then use of the pair \((\tilde{m}, \tilde{\xi})\) can be avoided in the composite delay curve since \(D_{m, \xi}^{\tilde{\xi}}\) is an increasing function of \(m\) and better performance is possible with \((m, \xi)\) as \(D_{m}^{\xi} > D_{m, \xi}^{\tilde{\xi}}\).
To show that entries with repeated m values need not be considered for inclusion in the composite minimal delay curve we consider the output of the grouping algorithm (GP).

\[ \ell: \hat{\lambda}_m \hat{\lambda}_m + 1 \ldots \hat{\lambda}_m + j \hat{\lambda}_m + j + 1 \]

\[ m: \underbrace{m \ m \ m}_{j+1 \ repetition} \] some \( \hat{m} < m \)

From the corollary for \( S < S'(m) \) we would use \( \ell = K' - 1 \) instead of \( \ell + \hat{\lambda}_m + 1, \ldots, \hat{\lambda}_m + j \), so that observed performance would be independent of \( \ell \). For \( S > S'(m) \), however, \( D_m^\ell \) would be larger than \( D_m^{\hat{\lambda}_m} \) and, therefore, would not be used. Hence, best performance is obtained for a given repeated \( m \), with \( \hat{\lambda}_m \) and the curves \( (m, \ell) \), \( \ell = \hat{\lambda}_m + 1, \ldots, \hat{\lambda}_m + j \) need not be considered.

Q.E.D.
3.3 The Optimization Algorithm

The algorithm we present in this section is constructive and generates the composite minimal delay curve by beginning with the curve \((1,K')\) and forming the composite as the lower envelope of the curves eligible, from \(m, \ell\). The algorithm is efficient in that:

1) the \(S\) point at which the currently optimal curve can branch to a different \((m,\ell)\) pair is known without continuously tracing all curves, and

2) a minimal number of basic comparisons is required.

The procedure is organized, so that:

1. a sub-interval, \(\tilde{S}_i = \{S_t, S_r\}\) of the interval \(0 \leq S \leq 1\) is identified; then

2. for the interval \(S_i\) given by 1, the curves eligible for inclusion in the composite curve, from \(m, \ell\), are identified; after which

3. the minimal curve for the interval is determined from those eligible, and

4. once the optimal curve for a sub-interval \(\tilde{S}_i\) is found, a new sub-interval, \(\tilde{S}_{i+1}\) is identified and the procedure is repeated until the point \(S = C\) is reached.

The successive intervals \(\tilde{S}_i\) migrate to the right and the next sub-interval \(\tilde{S}_{i+1}\) to be used is determined by the last sub-interval, \(\tilde{S}_i\), and the minimal \(m, \ell\) curve determined (by 3, above) for the interval \(\tilde{S}_i\).

As further explanation we state that the composite minimal delay curve is composed of \((m,\ell)\) curve segments with the values of \(m\) and \(\ell\).
changing only at the branching points \( S'(m) \) as given by (13) or \( S''(m_1,m_2) \) as given next by (14), the intersection of the curves \( D_{m_1} = D_{m_2} \) with \( \ell_1 = K' \) or \( \ell_2 = K' \) or \( \ell_2 = K' \), and \( m_1, m_2 \in \mathcal{M} \). The relationship \( D_{m_1} = D_{m_2} \) gives a cubic equation in \( S \), with \( S''(m_1,\ell_1,\ell_2,\ell_2) \) the positive root given by:

\[
S'' = (m_1,\ell_1,\ell_2,\ell_2) = -\frac{1}{2}(Y + T) + \frac{W}{3V} - \frac{1}{2}\sqrt{-3}(Y - T),
\]

where

\[
Y = (R + (Q^2 + R^2)^{1/2})^{1/3},
\]
\[
T = (R - (Q^2 + R^2)^{1/2})^{1/3},
\]
\[
R = \frac{9WUV - 27V^2}{54V^3}, \quad Q = \frac{3UV - W^2}{9V^2},
\]

and

\[
V = e^{2aq + ax + a^2y + 2a^2\ell_1 \ell_2}e^{a^3\ell_1 \ell_2}q
\]
\[
W = -2e^{-ax - 3aq - a^2\ell_1 \ell_2}e^{m_1m_2i + a^2m_1m_2\ell_1 \ell_2} + am_1m_2ij - a^2y
\]
\[
U = e^{aq - 2m_1m_2i - am_1m_2ij} - m_1m_2Ke - am_1m_2K'i
\]
\[
+ m_1m_2i + am_1m_2\ell_1 \ell_2 e - am_1m_2K'j
\]
\[
Z = 2m_1m_2i + m_1m_2Ke - m_1m_2x
\]

where

\[
e = m_1 - m_2
\]
\[
q = m_1 \ell_2 - m_2 \ell_1
\]
\[
x = m_1 \ell_1 - m_2 \ell_2
\]
\[
y = m_1 \ell_2 - m_2 \ell_1
\]
\[
i = \ell_1 - \ell_2
\]
\[
j = \ell_1 + \ell_2
\]
Additionally, we have that the endpoints of the intervals \( S_i \) considered above are also given by the points \( S', S'' \) such that the rightmost endpoint of \( S_i \) (i.e., \( S_r \)) is an \( S' \) point and the left boundary of \( S_i \) (i.e., \( S_l \)) is an \( S' \) or \( S'' \) point.

We proceed by 1) establishing the criteria which, for a given \( S_i \), determines the set of eligible curves, and 2) establishing the rules governing the migration of the interval endpoints \( (S_l, S_r) \) in going from \( S_i \) to \( S_{i+1} \).

Rules:

R1. For \( S \in S_i = (S_l, S_r) \), the only curve with \( \xi = K' \) to be included in the set of eligible curves is that curve \((m, K')\) with

\[
m = \min \{ m^* | S'(m^*) \neq S' \}
\]

This follows from the observation that \( D_{r_1} > D_{r_2} \) for all \( \xi \), and any \( m_1, m_2 \) with \( m_1 < m_2 \).

R2. For \( S \in S_i \), and \( \xi < K' \) (i.e., \( \xi = \xi_m \) for some \( m \)) the only curves to be included in the set of eligible curves are those \((m, \xi)\) curves for which \( m_0 < m < m^* \), with \( m_0 \) corresponding to the curve optimal at \( S \), and \( m^* \) the \( m \) corresponding to the \( S'(m) \) which is \( S_r \).

Proof: For all \( m < m_0 \) we have \( \xi_m > \xi_{m_0} \) by the structure of \( m, \xi \). Since at \( S \), the optimal curve is \((m_0, \xi_{m_0})\) we also have \( D_{m_0} < D_{m} \).

Further \( C(\xi_m) < C(\xi_{m_0}) \) by (10). Therefore there exists an \( S \) value, say \( S^* > S \), such that \( D_{m_0} < D_{m} \) at \( S^* \). Further for all \( S > 0 \), the curves \( D_{m} \) are convex (see Appendix B), for
all \( \ell \in \mathcal{L}, \ m \in \mathcal{M} \), so that any pair of curves can intersect at most twice. Additionally, a necessary condition for \((m, \hat{\ell}_m)\) curve to become optimal is given by the existence of an \(S\)-value, say \(S < \hat{S}^* < S^*\), such that \(D_{m}^{\ell_m} < D_{m_o}^{\ell_o}\) at \(\hat{S}^*\). The existence of \(\hat{S}^*\) contradicts the fact that there exists an \(0 < \hat{S} < S\) for which \(D_{m_o}^{\ell_o} = D_{m}^{\ell}\), by virtue of the fact that \(D_{m}^{\ell} \leq D_{m_o}^{\ell_o}\) at \(S = 0\). \(\text{Q.E.D.}\)

The set of eligible curves is initialized to consist of the curves \((1, K')\), \((m_2, K')\) for the interval \((\hat{S}_x, \hat{S}_x) = (S'(1) = 0, S'(m_2))\). The endpoints migrate according to the following rules. If \(\hat{S}_i = (S_x', \hat{S}_x')\) and \(S_{i+1} = (\hat{S}_x, \hat{S}_x)\) then

\[ R3. \quad \hat{S}_i = \min_{m, \hat{\ell}_m} \{ S'(m, \ell, m, \hat{\ell}_m), S_x'(S''(m, \ell, m, \hat{\ell}_m) > S_x) \} \]

with \((m, \hat{\ell}_m)\) curves contained in the set of eligible curves as given by \(R1\) and \(R2\) above, and

\[ R4. \quad \hat{S}_x = \begin{cases} \min_{m, \hat{\ell}_m} \{ S'(m, \ell, m, \hat{\ell}_m), S_x'(S''(m, \ell, m, \hat{\ell}_m) > S_x) \} & \text{if } \min_{m, \hat{\ell}_m} \{ S'(m, \ell, m, \hat{\ell}_m), S_x'(S''(m, \ell, m, \hat{\ell}_m) > S_x) \} > \hat{S}_x \\ S_x'(m) \exists m = \min_{m \in \mathcal{M}} \{ S'(m) > S_x' \} & \text{else } \hat{S}_x = S_x' \end{cases} \]

The following algorithm which produces the composite delay curve is based on rules \(R1-R4\).
Optimization Algorithm

Given:

\( a, K' \)

\( m = \{m_1, m_2, \ldots, m_{\text{max}} = K\} \),

\( \hat{x} = \{\hat{x}_{m_1}, \hat{x}_{m_2}, \ldots, \hat{x}_{m_{\text{max}}}\} \) such that

\( \#m = \text{the number of elements in } m, \hat{x} \)

we compute:

\[ S'(m_i), S''(m_i, \hat{x}_{m_1}, \hat{x}_{m_2}, \hat{x}_{m_{\text{max}}}) \]

\[ C(\hat{x}_{m_1}) \text{ with } \hat{x}_{m_i} = \hat{x}_{m_{i+j}} \text{ or } K' \]

as necessary to construct and maintain

\( m\text{-set} = \text{the set of curves eligible for consideration in the continued construction of the composite minimal delay curve, such that} \)

\( (m_o, \hat{x}_o) \) reflects the optimal \((m, \hat{x})\) curve for an \(S\)-value, \(S\), being considered, to finally produce,

\( mp \) that \(m\) value from \(m\) reflecting the optimal \(m\) value that precedes \(m_o\), so that we can compute

\( 0 = \{<m, x, S_x, S_x>\} \) - The output of the algorithm given as a set of tuples which specify the sequence of \((m, \hat{x})\) curves of which the minimal delay curve is composed, together with the respective \(S\)-range, \((S_x, S_x)\), of their optimality.

I. [initialization]

*/ compute \(S'(m_i) \forall m_i \in m */

\[ \text{if } \hat{x}_{m_1} \geq K' \text{ then } S'(m_i) = 0. \]

\[ \text{else } S'(m_i) \text{ computed from (13) } \]

\[ C = 1/(1+a) \]

*/ establish first curve in composite curve */
i+1

mp ← mᵢ

mₒ + mᵢ; ℓₒ + ℓₘᵢ /* ℓₘᵢ = K' */

Sₓ + Sᵣ ← 0

0 + 0, <mₒ,ℓₒ,Sₓ,Sᵣ> /* comma implies concatenation */

i ← i+1

Sᵣ ← S'(mᵢ)

m-set ← (mᵢ,K')

2. [compute next branching point]

∀(mⱼ,ℓⱼ) ∈ m-set

[j,Sⱼ] ← ⊥ min {S''(mⱼ,ℓⱼ,mₒ,ℓₒ) | S''(mⱼ,ℓⱼ,mₒ,ℓₒ) > Sₓ}

/* Sⱼ' denotes that S'' value corresponding to mⱼ */

3. [update (mₒ,ℓₒ), m-set and (Sₓ,Sᵣ)]

if Sⱼ' > Sᵣ then /* handles the case Sᵣ < Sⱼ */

[if ℓₘₚ = 1 then [0 + 0, <mₒ,ℓₒ,Sₓ,Sᵣ> ; terminate]

   else

   [if mₒ = mᵢ then ℓₒ ← ℓₘᵢ

   else [for (mᵢ,ℓₘᵢ) m-set ℓₘᵢ ← ℓₘᵢ]

   mp ← mᵢ;

   i + i+1;

   m-set ← m-set ∪ (mᵢ,K'); /* ∪ = union */

0 + 0, <mₒ,ℓₒ,Sₓ,Sᵣ>;  

Sₓ + Sᵣ;

if ℓₘₚ = 1, then Sᵣ ← C

else Sᵣ ← S'(mᵢ);

goto 2]
else /* handles the case $S_k < S'' < S_r$ */

\[ 0 \rightarrow 0, <m_o, l_o, S_k, S''>; \]
\[ S_k + S''; \]
\[ m_o + m_j; \]
\[ l_o + l_j; \]
\[ if \ m_o = m_i \ then \ goto \ 2 \]

else

\[ [m-set \leftarrow m-set \Theta \{(m_n, l_m) | n < j}\] /* $\Theta \mapsto$ deletion */

\[ \text{goto} \ 2] \]
4. EXAMPLES

To demonstrate applications of the optimization procedure we present optimized delay curves for a variety of network topologies.

For the network given in Figure 3 the delay is given in Figure 4, for $m = K$, and $m = 1$, representing pure sequential and parallel policies as well as the optimal mixed sequential/parallel policy. The optimal grouping is given in Figure 3. Figure 4 further demonstrates the earlier claim that as the arrival rate increases sequential policies yield lower expected delays.

Figure 5(a) shows the minimal composite curve segments which show the optimal policy, obtained from the optimization algorithm given by the $(k, m)$ columns as a function of channel throughput $S$. Figure 5(b) shows how sequential parallel control in the optimized policy is affected by change in the basic cost of communication, $a$.

We find that small $a$ values allow more extensive use of small $m$ values (as the optimal $m$) than is true for large $a$ values. This tendency is generally true for all topologies, and can be understood by examination of equation (9).

Figure 6 compares the expected delay in the network of Figure 3 under the optimized allocation policy with two other extreme policies applicable under similar modeling assumptions. These are simple time division multiplexing TDMA, whereby every node is assigned a dedicated time slot of length $T$ [7], and random allocation, which in a slotted system yields the "slotted Aloha" protocol [17].
Figures 7 and 8 show the average response time as a function of the resource utilization, or channel throughput, for other network topologies. As intuitively expected, the relative performance improves from the linear to the lattice network. This is due to the fact that reducing $K'$ and/or increasing the average degree of network nodes enables us to collect more nodes per group for a given group diameter $l$ and given number of nodes $K$. In other words, for the same small number of groups $m$ will decrease the average reservation interval. Notice that the delay curves are most removed from the "optimal" curve given by $M/D/1$ queueing system representing zero negotiation cost for medium utilization values.

For intermediate $S$-values, the reservation interval tends to be longer than for large $S$ values, owing to the need for larger $m$, but with the probability of a group being idle (i.e. containing non active nodes) remaining significantly non-zero.

Figure 9 gives the specification of the composite delay curve as produced by the optimization algorithm. Using the table, the channel access can be optimally controlled by selecting the correct $(m, \xi)$ value for given channel throughput.
5. DISCUSSION

In the preceding sections we have presented, in principle, an approach for controlling sequential access to a shared communication channel. As is often the case tractable analysis and clarity of presentation require simplifying assumptions. The following remarks deal with some of these issues.

5.1 Further Simplification of the Optimization Procedure

Inspection of obtained composite delay curves, see figures 5 and 8, shows that 1) no \( m \in \mathcal{M} \) is skipped in the composite curve, and 2) no curve \((m, \xi) \in \mathcal{M}, \xi = K'\) is included. If we could guarantee that this order always holds, the optimization procedure would become a simple construction of \((m, \xi)\) curves with \( m \in \mathcal{M} \) and \( \xi \in \xi \) obtained from (GP) grouping algorithm and \( S', S'' \) as given earlier.

To guarantee this situation two conditions must hold, namely:

a) \( \forall m_i \in \mathcal{M} \) we must have that \( \exists \) an \( S \) value, say \( S_i^o \), such that

\[
S''(m_i \ell_{m_i} m_{i+1}, \ell_{m_{i+1}}) \text{ is given by } D(S_i^o, m_i \ell_{m_i} m_{i+1}) = D(S_i^o, m_{i+1} \ell_{m_{i+1}}),
\]

with \( \ell_{m_i} = \ell_{m_i} \), or \( K' \) so that \( D(S_i^o, m_i \ell_{m_i} m_i) < D(S_i^o, m_k \ell_{m_k}) \), \( \forall K > i \), so that the sequence of points \( S_i^o \) form a strictly increasing sequence \( i = i, \ldots, \#m-1 \); and

b) \( \forall m_i \in \mathcal{M} \)

\[
S''(m_i \ell_{m_i} m_{i+1}, \ell_{m_{i+1}}) > S'(m_{i+1})
\]
Requirement a) is sufficient to guarantee that all \((m,k_m)\) curves for \(m \in M\) are included in the composite curve and b) guarantees that only \(k_m = \hat{k}_m\) values are used for \(k\). To guarantee a, and b a set of cubic inequalities involving \(S', S''\) values must hold. The set contains \(O((\# m)^2)\) inequalities and it is difficult to show that this set of inequalities always holds. Further requirement b) is easily violated as demonstrated by considering the linear network with \(K=50\). In this case \(S'(50)=0.1688\) and \(S''(25,1,50,48)=0.1650\).

b). Demonstrating violations of a) is more difficult. In view of these facts it is difficult to prove that a simplified procedure is adequate.

5.2 Groups with Imbalanced Traffic

Equation (9) requires that for a given \(m\), the groups place equal demands upon the channel. This requirement can be violated if the groups are imbalanced in terms of their demands upon the channel.

The grouping algorithm does not attempt to balance either the number of nodes in the groups, nor the channel demand rate (as given by requests per unit times) of the groups, although revised grouping algorithms are possible.
For the groups produced by the algorithm (GP) we can see by inspection if groups are balanced or nearly balanced (in terms of total number of nodes). When groups are imbalanced our resulting composite delay curve provides an upper bound on the actual composite delay curve for the following reasons. Say a group associated with $m_j \in m$ is imbalanced, while neighboring groups are balanced or nearly balanced. Then if the actual delay curve associated with the $(m_j, \lambda_{m_j}^{\lambda_m})$ grouping is less than $D_{m_j} \lambda_{m_j}^{\lambda_m} = \hat{\lambda}_{m_j}$ or $K'$, then our composite delay curve, which uses $D_{m_j}^{\lambda_{m_j}}$, yields an upper bound on actual performance. If the actual delay curve associated with $(m_j, \lambda_{m_j}^{\lambda_m})$ grouping is worse than the curve $D_{m_j}$, then not using $m_j$ groupings, i.e. using $m_{j-1}$ and $m_{j+1}$ groupings and skipping $m_j$ grouping's improves performance. Therefore, if the delays associated with $m_j$ were known and larger than $m_j$ grouping delays, $m_j$ groupings would be skipped in construction of the composite curve. Therefore the composite curve would again be accurate if $m_{j-1}, m_{j+1}$ are at least nearly balanced. Since the assumptions leading to (9) are strictly valid for the $(m, \lambda)$ curves $(1, K'), (K, 0)$, a composite curve which contains at least these two curves is always possible. Thus producing a composite curve which uses only balanced (or recently balanced) groups yields a composite curve which represents an upper bound on actual performance as stated.
5.3 Improving Performance by Relating (Node) Naming to Topology

In Section 2 we made the assumption that nodes are ordered, but no relation between ordering, or naming, and the topology given by $G_k$ was assumed. In a totally distributed system where every node has only its own local view of the network, or in dynamically changing topologies, e.g. in mobile radio networks, topology-name relation is not practical. If on the other hand, a global view of the network exists the order, or naming of nodes, may be topology related. Since the network is connected, a path through all nodes or groups always exists so that, for example, topologically consecutive naming can be chosen. Consequently, given $\ell$ the maximum group diameter (3) can be rewritten as

$$\begin{cases} w_1(M_i) = \min ((K'-\ell)i, \ell - i) \\ w_2(M_i) = \ell \end{cases} \quad (15)$$

In (15) the maximum intergroup cost is $2\cdot\ell$ given that every two groups traversed in the given order are adjacent on the network graph. In other words for throughputs higher than some $\bar{s}$ the average delay can be further reduced. The analytic techniques presented to solve policies defined in (3) apply also to policies given by (15) and clearly the earlier results form an upper bound on the performance given by (15).
6. SUMMARY

We have presented a theoretical approach (allocation policy and optimization procedure) to accessing a shared resource in a distributed system. Random request arrivals and positive cost on communication between the system users were assumed. It was shown that given this cost, network topology and total request arrival rate an allocation policy which minimizes the average resource access time in the system can be found. This policy was found to be an optimal combination of sequential and parallel negotiation policies.

Using the given approach and algorithms a reservation based protocol can be constructed through which orderly access of shared communication channel can be obtained. The performance of such protocol has been analyzed for various system configurations and compared to alternative fixed and random access based protocols.
APPENDIX A: THE OPTIMAL NODE GROUPING PROCEDURE (GP)

Problem: Given $G_k$, $\ell$ find the minimal number $m$ of subgraphs $g_i$ (the groups) so that:

1. $\max (d_{ij} | i, j \in g_n) \leq \ell$ for $n = 1, 2, \ldots, m$
2. for $i \neq j$, $g_i \cap g_j = \emptyset$
3. $\bigcup_{n=1}^{m} g_n = G_k$.

Procedure (GP): Step 1. Produces all allowable groups for given $\ell$ so that only groups constructed for $\ell = i-1$ are required to construct groups for $\ell = i$.

For a given $\ell$, the groups are by construction organized into blocks, $B^\ell_i$, $B^\ell_i$ constructed from groups $B^\ell_{i+1}$, $B^\ell_{i+1}$, that is, from two consecutive blocks associated with the previous $\ell$ value. To initialize the process the nodes are organized into blocks, $B^0_1$, that is, blocks for which the group diameter is zero. The construction is with blocks $B^0_i$ composed of groups of nodes, $g_k$, such that diameter $(g_k) = \ell$.

1. [initialization]
   
   /* create blocks of groups for which group diameter, \ell, is zero */
   la. choose any node-$i \in g_k$ to begin, node-$i$ becomes $B_1^0$. Set $k = 1$.
   1b. While $\exists$ nodes-$i \in g_k$ not used do
   
   $K \leftarrow K+1$
   
   [construct $B^0_K$ consisting of all unused nodes-$j$ such that
   
   $d_{ij} = 1$ for any node-$i \in B^0_{K-1}$;
   
   tag nodes-$j$ used;
   
   if for node-$j$ included, $d_{ij} \leq 2 \forall$ nodes-$i \in B^0_{K-1}$ mark it$^2$
   
   /* on tableau marked groups are denoted by $< > */]
2. [construct blocks $B_i^2$, $1 \leq i \leq K'$, $i \geq 1$, from blocks $B_i^{k-1}, B_i^{k+1}$]

Let $\phi$ denote the null set. Let

- $b_i$ denote the block given by all unmarked groups from $B_i \cup \phi$.
- $b'_i$ denote the block composed of all marked groups from $B_i \cup \phi$.

If $B_i \phi$ does not exist then $b_i, b'_i$ are given by the null set, $\phi$.

Let $a_i^*$ denote the groups composed of all possible unions of groups $b'_i \cup \phi$.

Then

$$B_i^{k+1} = \{\alpha \cup \beta \cup \gamma \cup \delta | \alpha \in b_i, \beta \in a_i^*, \gamma \in b_{i+1}, \delta \in a_{i+1}^* \}$$

and diameter $(\alpha \cup \beta \cup \gamma \cup \delta) \leq k - \phi$

mark groups added

Figure 1 illustrates the group construction procedure obtained from step 1.

Step 2. The input to this step are the groups in the sequence of $B_i$ blocks. This step uses a set covering algorithm [9] with minor alterations.

Comment: Procedure (GP) above has high complexity, for regular graphs.

For general topology graphs with a large number of nodes an efficient heuristic has been constructed.

For regular graphs, e.g., linear or lattice topology, the grouping is immediate [17].

Examples: For a lattice network, figure 8 shows the $(l, m)$ vector for $K=100, K'=9$.

Figure 10 (b) shows actual group construction for a six node network.

---

2 A group is marked when it is added to a block $B_i \phi$ if all nodes in the group are within $k+1$ hops of all nodes in another already in the block.
we have $K' = \sqrt{K} - 1$

and $m = \left( \frac{\sqrt{K}}{i+1} \right)^2 + \delta_{kl} \left[ \frac{2\sqrt{K} - 1}{i+1} \right]$

with $\delta_{kl} = \begin{cases} 1 & \text{if } \sqrt{K} \mod(i+1) = 0, \\ 0 & \text{otherwise} \end{cases}$

and $1 \leq l \leq K'$

with $K = 100$ and $K' = 9$, and we obtain

<table>
<thead>
<tr>
<th>$l$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>25</td>
<td>16</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

so that

$m = 1, 4, 5, 9, 16, 25$

and

$l = (9, 6, 4, 3, 2, 1)$.

Figure A.2 - Lattice network Grouping
The Complexity of the problem: The complexity of any procedure for (GP) is established as follows. Assume we are given a graph, $G_k$, such that the maximal group diameter permissible is $\lambda$. For this network we will call a pair of nodes $(i, j)$ $\lambda$-connected if there exists a path between $(i, j)$ which contains at most $\lambda$ hops (or $(\lambda - 1)$ intermediate nodes). Similarly, a group is $\lambda$-connected if the diameter of the group is $\lambda$, that is, if each pair of nodes in the group is $\lambda$-connected. Further a group is called $\lambda$-maximal if no node can be added to the group without violating the $\lambda$-connectivity constraint. We define an $\lambda$-maximal group to be a clique of order $\lambda$, so that $\lambda = 1$ yields a clique of order 1 as defined in standard graph theory. Consequently, for $\lambda = 1$, only one of possibly $K'$ values for which the grouping is required becomes a clique covering problem, which is NP-complete, [9]. The same conclusion can be reached by considering the set dominating problem created by $\lambda = 2$ [9, 13].
Heuristics for General Graphs:

In the general case heuristics may be required when \( K, K' \) are not small. Simple efficient heuristics can be constructed based on "greedy" approaches, although they can not be expected to yield optimal groupings.

We consider a grouping heuristics (GH) which is linear and totally distributed both in terms of required information and lack of synchronization requirements. Under (GH) every node has a local knowledge of all nodes within distance \( k \) (no. of hops) of itself. Groups are built around the highest order node by applications made from lower order nodes. A node whose application has been accepted is added to the group. Nodes which were rejected apply to remaining nodes in order until they are accepted or until they themselves become the highest order node among nodes unassigned. The GH procedure is given next.

Assumptions and Definitions:

1. Every node in \( G_k \) can be of one or both types

   \[ \text{host} \quad - \quad \text{the highest order node in set } d_i \]

   \[ \text{satellite} \quad - \quad \text{all but the highest order node in set } d_i \]

   \[ d_i \quad - \quad \text{set of nodes within } k \quad \text{(hops) of } i \text{ in } G_k \]

   *Comment:* \( d_i \) changes dynamically throughout the algorithm s.t.

   \[ d_i \quad \text{(initial)} = d_{ij} \quad \text{i.e., set of all nodes within } k \quad \text{hops of } i \]

   \[ d_i \quad \text{(final)} = g_i \quad \text{group of nodes assigned to host node } i. \]

Consequently node type may change from satellite to host but not from host to satellite.
2. Events - describe arrivals or departures of control message at node $i$

Event types:

- $A_i(j)$ - application from satellite node $i$ to join group of host $j$
- $C_i(j)$ - confirmation from host node $i$ to accept satellite node $j$ in the group
- $R_i(j)$ - rejection of application from node $j$ by host or satellite $i$
- $O_i(j)$ - offer from host $i$ to node $j$ to join group of $i$

3. Every node $i$ has a FIFO buffer for incoming events of type $A_i(i); C_i(i), C_j(k), R_j(i), O_j(i)$.

4. Nodes execute on incoming events; if no expected event type is found in buffer node execution is suspended until event arrival.

5. Node synchronization is assumed. Grouping procedure can be initialized by all satellite nodes in any order leading to the same final graph partition.
Procedure Grouping Heuristics (GH) for node $i$:

Begin while $\exists j \in d_i$ s.t. $j < i$

{satellite node} do begin send $A_i(j_{\min})$ to all $j$, $j < i$ {j_{min}=min d_i} all $j < i$

wait for event $(j_{\min})$ {event(j_{min})=event in buffer} incoming from node $j_{\min}$

case of event $(j_{\min})$

$R_{j_{\min}}(i): d_i = d_i - j$

end;

$0_{j_{\min}}(i): i \in g_{j_{\min}}$ {in final partition node $i$ is allocated to group $g_{j_{\min}}$}

send $C_i(j_{\min})$ to all $j \in d_i, j < i$

repeat wait for event $(x)$

{host node} $g_i := i$;

while $d_i - g_i \neq \emptyset$

do begin wait for events $(A_j(x), \forall j \in d_i - g_i)$

$g_i = g_i + j/A_j(x=i)$ s.t. $d_{g_i,j} \leq$ in ascending order of nodes $j$

construct group $g_i$ and denote

$
\{g_i\} = \{j_{\text{accepted}}\}$

send $0_i(j_{\text{accepted}})$

end;
Wait for events \( A_j, C_j \) from \( \forall j \in d_1 \) so that:

1) \( g_i \neq i \) then \( C_j \) accepted \( (i + A_j(x=i) \) or \( C_j(x \neq i) / j \in d_1-j \) accepted

\{denote \( J_{\text{new}} = \{j \in A_j(x=i) \) and diameter \( (g_i, j) \leq x/j \in d_1-j \) accepted

then begin \( g_i = g_i + J_{\text{new}} \); send \( O_i(j) \); end;

\( \text{send } R_i(j) / j \in J_{\text{new}}^c \) \( d_1 = d_1-j / j \in (C_j(x \neq i) \) or \( J_{\text{new}}^c \) \( / j \in d_1-j \) accepted

end; \{of while of host node\}

\( g_i = d_1 \); node \( i \) is allocated to be the host of group \{\( d_1 \)\}

END.

Notations: \( + J \): group addition

\( - J \): group subtraction

\( J^c \): group I complement.

An example we consider the graph given in Figure 6. Since \((GH)\) uses the node ordering in the process of group construction different partitions obtain when node ordering is changed. For the two orderings (left, right) as shown in each node heuristic \((GH)\) provides optimal solutions for all \( \leq 1 \leq K^* \) for \( \leq 2 \), shown in Figure A.1 i.e three groups in each case. Similar to optimal procedure \((GP)\), heuristics \((GH)\) does not however attempt to balance the sizes of constructed groups, so that for "left" ordering totally balanced groups are obtained while for "right" ordering one group dominates.
Figure A.1 - Groupings created by heuristics (GH) for two possible node orderings.
APPENDIX B. FINDING OPTIMAL $\xi^*$.  

Let $\xi^*$ denote the optimal values of $\xi$ which minimize $D$. Then $\xi^*$ values are characterized by equations (12) and (13) in Sect. 3.2.

Proof: Unconstrained solution of $dD/d\xi = 0$ does not guarantee

$$0 \leq \xi^* \leq K', \quad \xi^* \text{ integer}, \quad (B.1)$$

$$0 \leq S < C_m, \quad (B.2)$$

To obtain $\xi^*$ which satisfies (B.1) and (B.2), we view $\xi$ as a continuous rather than integral valued variable and form $dD/d\xi$ which gives

$$\frac{dD}{d\xi} = \frac{aA}{2\pi[1-S(1+\xi a)]^2}$$

with

$$A = 2m^2 - 2S(S+2a+S^2 + S^2 m + S^2 a + S^2 m a + S m k' + S^2 a^2).$$

From (10) $[2m(1-S(1+\xi a))^2] > 0$ so that the critical points, $\xi_1$ and $\xi_2$, of $dD/d\xi = 0$, obtain when $A = 0$:

$$\xi_{1,2} = \frac{-Sm + m - Sa + S^2 a \pm B^{1/2}}{Sa(m - Sa)} \quad (B.3)$$

with

$$B = B(S) = (Sm + Sa - S^2 a)^2 - (m - Sa)(2m^2 - 2Sm - Sa + S^2 m + S^2 a + S m k' + S^2 a).$$
For $\xi_1, \xi_2$ to be real,

$$S_a(m - S_a) \neq 0$$

and

$$B \geq 0$$

must hold, (B.4) is always true since by assumption we have $a \leq 1$, $m > 2$ and $S \leq 1$.

To determine the $S$ region which guarantees (B.5) we set $B(S) = 0$, obtaining a quadratic function in $S$ which yields the roots

$$S_1 = \frac{m}{a}$$

and

$$S_2 = \frac{1}{\frac{m}{a} + \alpha K'}$$

$S_1$ does not satisfy (B.2) since $S_1 = \frac{m}{a} > 1 > 0$, while from (B.6) $S_2$ does. Examination of the coefficient of $S^2$ in $B(S)$ shows that $B(S) > 0$ for $0 < S < S_2$ and $S > S_1$. Since we require (B.2), we consider root $\xi_1, \xi_2$ behaviour in $S \in [0, S_2]$.

For the endpoint $S = 0$, we have from (B.3) $\xi_1 \to \infty$ as $S \to 0$ and $\xi_2 \to -\infty$ as $S \to 0$, since $K > S_a$ and $\sqrt{K} > 1$. For $S = S_2$ we have, since $B(S_2) = 0$, that $\xi_1 = \xi_2$ with

$$\xi_1 = \xi_2 = a(S_2^2 - S_2) + m(1 - S_2)$$

From knowledge of the roots $\xi_1, \xi_2$ at the endpoints together with examination of $dD/d\xi$, we find that $dD/d\xi$ changes sign only twice: from plus to minus at $\xi_2$ ($\xi_2$ given by using the plus sign in (B.3)) so that the maximum of $D$ is given using $\xi_2$. Hence the minimum is obtained using $\xi_1$ (obtained using the minus sign in (B.3)). From consideration of the values of $\xi$ for $S = 0$, $S = S_2$, and the fact
that \( x_1, x_2 \) are smooth continuous functions in \( S \), the unconstrained values which minimize and maximize \( D \) can be qualitatively shown, see Figure 11, as a function of \( S \). From (B.3) we further have

\[
K' + \frac{1}{a(m-1)} \leq x_1 \leq +\infty,
\]

(3.7)
a fact we will use later.

![Figure 11 - Characterization of \( x_1, x_2 \) and \( D(x) \).](image)

We can now use the unconstrained results for \( x_1, x_2 \) to find the integral \( x^*(S) \) values constrained by (B.1)-(B.2) which minimize \( D \). To do so we notice that \( x_1 \) falls outside the range of constraint (B.1) while \( x_2 \) passes through the range \([1,K']\) when \( S \) obeys (B.3). Since \( x_1(S) > K' \) always holds, and since \( D(x) \) is continuous with only two critical points \((x_1, x_2)\) we can obtain \( x^* \) by examining the behavior of \( D \) in each of the three \( S \) regions shown in Figure 11.
\[ S \in [0, \alpha]: \] For \( S \) in \( A = [0, \alpha] \) we have \( \ell_2 < 1 \) and \( \ell_1 > K' \). Therefore in \( [0, \alpha] \) \( D(\ell) \) has no critical points and the smallest constrained \( D(\ell) \) value is obtained by setting \( \ell^* = K' \).

\[ S \in [\beta, S_2]: \] For \( S \) in \( C = [\beta, S_2] \) we have \( K' < \ell_2 < \ell_1 \), again with \( \ell_1, \ell_2 \) the only critical points of \( D(\ell) \) in the region. Consequently, the smallest constrained \( D(\ell) \) value is obtained by setting \( \ell^* = 1 \).

\[ S \in (\alpha, \beta): \] For \( S \) in \( B = (\alpha, \beta) \) we have \( 1 < \ell_2 < K' \), again with \( \ell_1 > K' \) and \( \ell_2 \) the only critical values of \( D(\ell) \) in the region. Thus, the constrained minimum must be obtained either by setting \( \ell^* = \hat{\ell}_m \), or \( \ell^* = K' \) depending on which of the pairs \( D(\ell = \hat{\ell}_m) \) or \( D(\ell = K') \) is smaller. To determine the smaller pair we have \( \ell^* = K' \) for \( S = \alpha \) (from the left) and \( \ell^* = \hat{\ell}_m \) for \( S = \beta \) (from the right). As a consequence, for \( S \) in this region we must have \( \ell > 1 \) for \( S \) near \( \alpha \) and \( \ell < \hat{\ell}_m \) for \( S \) near \( \beta \). Since \( \nu \) is continuous in \( S \) there must exist an \( S \) point, \( S', \alpha < S' < \beta \), such that \( D(\ell = \hat{\ell}_m) = D(\ell = K') \), that is a point at which \( \ell^* \) will change from \( K' \) to \( \hat{\ell}_m \). Setting \( D(\ell = \hat{\ell}_m) - D(\ell = K') \) gives a cubic in \( S \), which when solved gives (13). Thus, we have that the constrained minimum of \( D \) is obtained using \( \ell^* = K' \) for \( S \in [0, S'] \) and \( \ell^* = \hat{\ell}_m \) for \( S \in [S', S_2] \), as required for \( S \in [0, S_2] \).

For \( S \in (0, C \hat{\ell}_m) \times S_2 \), notice that \( dD/d\ell \) has no real roots in the region \( (S_2, C) \) and \( D \) becomes an ascending function at \( S_2 \). Therefore \( D \) is an ascending function in the region \( (S_2, C) \). Thus, \( \ell^* = \hat{\ell}_m \) yields the minimum \( D \) value in the region \( (S_2, C) \) since \( D(\ell = \hat{\ell}_m) \) was minimal at \( S_2 \).
REFERENCES


REFERENCES (Cont'd)


16. Goyal, A., Lipovský, G.J. and Malek, M., "Reliability in Ring Networks", COMCON 1982, Fall, Washington, D.C.,

\[ w_2 \text{ (all nodes) } = K'a \]

\[ \text{data transmission (node 2)} \]

\[ w_1 \text{ (all nodes) } = 0 \]

\[ \text{RI } = K'a \]

(a) Parallel Policy

\[ w_2(0) = 0 \quad w_2(1) = 0 \quad w_2(2) = 0 \]

\[ \text{data transmission (node 2)} \]

\[ w_1(1) = K'a \]

\[ w_1(2) = 2K'a \]

\[ \text{RI } = 2K'a \]

(b) Sequential Policy

\[ w_2(0) = \lambda a \quad w_2(1) = \lambda a \quad w_2(2) = \lambda a \]

\[ \text{data transmission (node 2)} \]

\[ w_1(1) = (K' - \xi)a \quad w_1(2) = 2K(K' - \xi)a \]

\[ \text{RI } = 2(K' - \xi)a + a = (2K' - \xi)a, \ 0 \leq \xi \leq K' \]

(c) Parallel/Sequential Policy

See text for notations

**Figure 1** - Length of reservation interval (RI) for Parallel, Sequential and Sequential/Parallel allocation policies given highest order active node is node number 2.
Figure 2 - Channel idle, busy periods with $m = 1$, $l = l_1 = K'$

(a) Sample Network Connectivity graph. Assuming shortest path routing is used $K' = 4$.

<table>
<thead>
<tr>
<th>$m$</th>
<th>$l$</th>
<th>Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>$(1, 2, 3, 4, 5) (6, 7, 8, 9, 10)$</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>$(1, 2, 3) (4, 5, 6) (7, 8, 9, 10)$</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>$(1, 2) (3, 6) (4, 5) (7, 8, 9, 10)$</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>$(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)$</td>
</tr>
</tbody>
</table>

(b) Optimal grouping for $0 < l < K'$

Figure 3 - A Sample Network and its Grouping Using (GP) Algorithm
Figure 4 - Delay/throughput behaviour for network of Figure 3, under sequential, parallel and optimized sequential/parallel policies.
<table>
<thead>
<tr>
<th>m</th>
<th>k</th>
<th>$[S_k, S_r]$</th>
<th>$D_m^*(S_r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0, 0</td>
<td>1.9</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0, 0.29457</td>
<td>2.41</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.29457, 0.49061</td>
<td>4.05</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.49061, 0.56095</td>
<td>5.37</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.56095, c=.833333</td>
<td>D+∞</td>
</tr>
</tbody>
</table>

(a) Curve segments of composite curve for the network of figure 3 with $a=.2$

(b) Optimal $m$ values as a function of $S$ for network of figure 3.

**Figure 5** - Optimal policies as a function of arrival rates (channel throughput, $S$) and basic communication cost, $a$. 

Technion - Computer Science Department - Technical Report CS0282 - 1983
Figure 6 - Comparison of expected delay for network in Fig. 3 with:
1) optimized sequential/parallel allocation policy
2) random access (slotted ALOHA), and
3) time division (TDMA).
Sequential/parallel optimized policy

Figure 7 - Throughput/delay performance comparison for linear network under various allocation policies.
Figure 8 - Throughput/delay performance for lattice network under various allocation policies.
<table>
<thead>
<tr>
<th>(m)</th>
<th>(l)</th>
<th>([S^e_r, S^r_r])</th>
<th>(D^r_m(S^r_r))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49</td>
<td>[0.0.]</td>
<td>5.95</td>
</tr>
<tr>
<td>1</td>
<td>49</td>
<td>[0.0.12105]</td>
<td>7.16</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>[.12105.241849]</td>
<td>14.03</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>[.241849.303633]</td>
<td>21.79</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>[.303433.338173]</td>
<td>27.22</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>[.338173.438695]</td>
<td>54.5</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>[.438695.442515]</td>
<td>56.23</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>[.442515.446875]</td>
<td>57.86</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>[.446875.451748]</td>
<td>59.39</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>[.451748.45712]</td>
<td>60.83</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>[.45712.59427]</td>
<td>126.69</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>[.594269.632378]</td>
<td>161.09</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>[.632378.719115]</td>
<td>281.07</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>[.719115.(c=909091)]</td>
<td>(+\infty)</td>
</tr>
</tbody>
</table>

*Figure 5* - Curve segments of composite curve for linear network, \(K=50\), \(K'=49\), \(a=.1\)
Figure 10(a) - Group construction tableau

Figure 10(b) - Tableau for loop network, K=6