ESTIMATING TOPOLOGICAL CHANGE IN FULLY CONNECTED MOBILE NETWORKS — A LEAST UPPER BOUND ON THE WORST CASE

by

Y. Gold and S. Moran

Technical Report #321

June 1984

* Dept. of Electrical Engineering and CS, University of Connecticut

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

This work was supported in part by the National Science Foundation, grant No. ECS-8307186.
ESTIMATING TOPOLOGICAL CHANGE IN FULLY CONNECTED MOBILE NETWORKS -
A LEAST UPPER BOUND ON THE WORST CASE

by

Yaron I. Gold
Department of Electrical Engineering and Computer Science
the University of Connecticut

and

Shlomo Moran
Department of Computer Science,
Technion, Israel Institute of Technology

ABSTRACT

A least upper bound is derived on the amount of adjustment of virtual token
passing (VTP) time needed to assure collision-free access control in VTP networks
(i.e., networks that use "time-out" or scheduling-function based access-protocols)
and in which nodes change their spatial configuration due to motion. The bound is a
function of node maximal speeds, the time that passed since the previous adjustment
and the bound on network end-to-end propagation delay. The new bound allows VTP
times that are shorter than those found in previous publications, especially when
intervals between adjustments grow larger. For most VTP networks with large mixed
populations of mobile and stationary users, the average VTP time (which is a major
factor in performance of the access protocol) allowed by the new bound is much short-
er than that of any configuration independent protocol.

KEY WORDS - mobile communication, dynamic topology, virtual token passing, schedule-
ing overhead, access control.

This work was supported in part by the National Science
Foundation, grant No. ECS-8307186.
INTRODUCTION

"virtual-token-passing" (VTG) access protocols provide collision-free channel access, by means of distributed scheduling functions that determine the amount of time a node must wait (either from the end of its own previous transmission, or from the end of the last transmission on the channel) until it may next access the channel. TDMA [1] can be considered a VTP protocol. The time it takes to "pass the token" between two adjacent nodes in the transmission sequence (virtual token-passing time, or VTP time) for TDMA is a full message transmission time. BRAM [2], too, is a VTP access protocol. It is based on the observation that a node whose predecessor has not used its access-rights is sure to realize this if a certain amount of time equal to the network maximal (end-to-end) propagation delay has passed from the moment the predecessor was granted access-rights. BRAM assumes knowledge of this maximal delay and uses it as its VTP time. It is otherwise independent of node configuration.

SOSAM's [4] scheduling function provides "tighter" scheduling (i.e., shorter VTP times and, hence, less idle time for the channel) than other scheduling functions (TDMA, BRAM, or GBRAM [3]). Unlike BRAM that uniformly uses the "worst-case" propagation delay, SOSAM's token passing time is tuned to individual internode propagation delays. In a stationary network it is always equal to the minimum necessary for each node to "know" whether its predecessor in the round-robin sequence has or has not used its access rights. This minimum is equal to the signal propagation delay between the node and its predecessor, and is generally much smaller than the network end-to-end propagation delay. The price paid for this tighter scheduling is the need to maintain more accurate and detailed topological information (see [4],[5]). It is shown in [4] that protocol performance (in terms of channel utilization and message delays) is directly related to the average VTP time.

The performance advantage of SOSAM over protocols that use less topological information justifies extra overhead involved in automatic acquisition of the required information. This capability is especially desirable for networks with mobile nodes, due to the dynamic nature of these networks' topologies. Efficient distributed methods for measuring the required topological information for SOSAM are described in [5] and in [6].

For stationary networks, topological information can be obtained and stored at installation time, and used thereafter. However, when a network contains mobile...
nodes its topology changes with time, and topological information employed by a node must be updated during network operation.

To maintain updated information in a dynamic system, two update processes are required: measuring and extrapolating. Though the method described in [5] for measuring the required topological information for SOSAM is very efficient, it still uses system capacity, and hence cannot be applied continuously. Thus, this information must be sampled in a discrete manner, and during the intervals between samples the system must rely on extrapolation of necessary information. In our case of maintaining topological information for the purpose of conflict-free scheduling, worst-case predictions must be made (based on node maximal speeds) leading to "safety margins" (gaps) in scheduling and hence to longer VTP times and to some degradation in performance [5]. In order to minimize the sizes of these gaps and still maintain collision-free operation in networks with dynamic topologies, a tight upper bound on the worst case topological change is desirable. Determining this bound is a complex problem in geometry. It is shown in [5] that in order to assure collision-free operation the time to pass the virtual token from a node, say node-A to the next node in the round-robin sequence, say node-B, must be increased by not less than a certain quantity, INC(t). This quantity is a function of the time, t, that passed since the last update, as well as of node-A's and node-B's maximal speeds and of the maximal speed of the node that last transmitted, say node-C. INC(t) is an upper bound on the quantity INC(t), which is defined (using the notation of Figure 1, with all distances given in terms of signal propagation delays) as:

\[
INC(t) = |b(t) - b(0)| + |c(t) - c(0)| - |a(t) - a(0)|.
\]

Figure 1. - The geometry of a displacement and related notation.
In [5] we considered the case when the bound on the geographical size of the network is not taken into account, or when the amount of node travel between measurements is very small relative to network size. The initial configuration, denoted \((A(0), B(0), C(0))\), and the directions of motion of node-A, node-B and node-C that yield the worst-case displacement, \(D = ((A(0), B(0), C(0)), (A(t), B(t), C(t)))\) were derived for this case, and the corresponding value of \(INC(t)\) is given by:

\[
(2a) \quad INC(t) < 2(v_A^2 + v_B^2 + v_C^2)^{1/2}t
\]

where \(v_A\), \(v_B\) and \(v_C\) are maximal node speeds, normalized by signal propagation speed. In the special case where \(v_A = v_B = v_C = v\) we have:

\[
(2b) \quad INC(t) < 2(1+2^{1/2})vt = (4.8...)vt.
\]

As shown in [5], SOSAM with its scheduling function extrapolated using (2a) or (2b) will provide collision-free access control, even when the time interval, \(t\), between adjacent measurements must occasionally be increased (e.g., due to a temporary node failure, or the need to complete a high-priority process before a new measurement can begin). However, its performance advantage over configuration independent protocols may decrease. When \(t\) is large SOSAM's resulting VIP time may become equal to that of BRAM. In order to avoid further degradation in performance the network may then switch to BRAM or another configuration independent protocol, but for this purpose each node must support both the other protocol and SOSAM, and nodes must coordinate the switch from one protocol to the other.

In many practical cases the geographical size of the network is bounded at any given time by a known quantity (which can either be fixed, determined for example by national borders, or a more tightly estimated quantity that is based on currently measured topological information and node maximal speeds), and this bound may be used to obtain tighter scheduling. In this paper we derive the least upper bound on \(INC(t)\) as a function of the parameters \(r_i\), which are the proportions of network end-to-end distance that nodes may travel during the time since the last measurement. These parameters incorporate the following quantities:
(1) the given upper bound on the network's end-to-end distance or propagation delay.

(2) the time interval, \( t \), since the last measurement and

(3) node maximal speeds.

As demonstrated in Section 4, this new bound allows SOSAM to provide collision-free access with much less extrapolation overhead than allowed by the old bound and thus also to maintain its performance advantage (in terms of message delays and channel utilization) even when \( t \) becomes very large. For most networks with large mixed populations of mobile and stationary nodes performance obtained by the new bound will be much better than that of any configuration-independent VTP protocol even at large values of \( t \).

THE LEAST UPPER BOUND ON INC\( (t) \)

In this section we present procedure \( \text{COMPUTE inc}(r_A, r_B, r_C) \) that computes \( \text{inc}^*(t) \), which is the least upper-bound on \( \text{INC}(t) \) normalized by the network end-to-end propagation delay, \( R \):

\[
\text{inc}^*(t) = \frac{\text{INC}(t)}{R}.
\]

The parameters \( r_A, r_B \) and \( r_C \) are the maximal distances (in terms of signal propagation delay, and also normalized by \( R \)) that node-\( A \), node-\( B \) and node-\( C \) may travel during the time period \( t \) and incorporate the necessary information on network maximal geographical size, node maximal speeds and the time since the last measurement:

\[
\begin{align*}
  r_A &= \frac{v_A t}{R} \\
  r_B &= \frac{v_B t}{R} \\
  r_C &= \frac{v_C t}{R}
\end{align*}
\]

The procedure is given next, and the proof of its correctness follows in Theorem 1 below. \( r_B \) and \( r_C \) play symmetric roles in this procedure. It is assumed that not both are zero, and, in fact, that \( r_C \neq 0 \). (if both are zero, then, trivially,
inc*(t) := 2r_A

procedure COMPUTE_inc(r_A, r_B, r_C)
1. if r_A + r_B ≥ 1 or r_A + r_C ≥ 1, then inc*(t) = 2,
2. else (i.e. r_A + r_B < 1 and r_A + r_C < 1) compute inc*(t) by the following algorithm:
   2a. q := r_B/r_C;
   2b. M := r_B/(1 - r_A);
   2c. compute the positive root y of the equation
       \((1 - q^2y^2)(1 - y^2)^{1/2} - qy^2 + My = 0;\)
   2d. x := My;
   2e. z := qy;
   2f. x1 := (1 - x^2)^{1/2};
   2g. y1 := (1 - y^2)^{1/2};
   2h. z1 := (1 - z^2)^{1/2};
   2i. inc*(t) := 2(r_Ax_1 + r_By_1 + r_Cz_1 + 1 - x_1).

Computation time of this algorithm is mainly determined by step 2c. However, in the case where r_B = r_C (i.e. q = 1), this step can be reduced (see [7]) to:
2c'. y := (M + (M^2 + 8)^{1/2})/4.

Theorem 1: procedure COMPUTE_inc computes the (normalized) least upper bound on INC(t).

The proof of Theorem 1 is based on four lemmas. Lemma 1 establishes the geometric properties of the worst displacement, D*, as a function of r_A, r_B and r_C (see Figure 2). The proof of this lemma is given in [7] and since it is rather long we do not repeat it here. The other three lemmas use these properties to derive the corresponding inc*(t), denoted inc*(D*), which is the desired inc*(t).
Lemma 1:

a. If \( r_A + r_B \geq 1 \) then there is a worst displacement, \( D^* \) in which:
   - \( c(0) = 0 \) (and hence \( a(0) = b(0) \))
   - \( b(t) = c(t) = 1 \)
   - \( a(t) = 0 \).

b. If \( r_A + r_C \geq 1 \) then there is a worst displacement, \( D^* \) in which:
   - \( b(0) = 0 \) (hence \( a(0) = c(0) \))
   - \( b(t) = c(t) = 1 \)
   - \( a(t) = 0 \).

c. If \( r_A + r_C < 1 \) and \( r_A + r_C < 1 \) then there is a worst displacement
   \( D^* = ((A(0), B(0), C(0)); (A(t), B(t), C(t))) \) with the following properties:
   - (c1) \( D^* \) is planar;
   - (c2) \( A(t) \) lies on the bisector of the angle opposite to \( a \);
   - (c3) \( a(t) = 0 \), and \( b(t) = c(t) = 1 \);
   - (c4) both \( B(t) \) and \( C(t) \) coincide with the intersection of the bisectors of
     \( a \) and of the angles external to \( \beta \) and \( \gamma \);
   - (c5) \( A(0)A(t) = x_A \), \( B(0)B(t) = x_B \), \( C(0)C(t) = x_C \).

Moreover, the worst displacement is unique (up to translation and rotation). The worst displacement is given in Figure 2.

Lemma 2: If \( r_A + r_B \geq 1 \) or \( r_A + r_C \geq 1 \) then \( inc(D^*) = 2 \).

\((*)\) \( A(0)A(t) \) is the length of the line segment between \( A(0) \) and
\( A(t) \), and similarly for \( B(0)B(t) \) and \( C(0)C(t) \).
Proof: Let \( D^* \) be a displacement with the properties described in Lemma 1, part a. Then, from definition (1) we have:

\[
\text{inc}(D^*) = (a(0) - a(t)) + (b(t) - b(0)) + (c(t) - c(0)) = \]

\[
= (a(0) - 0) + (1 - a(0)) + (1 - 0) = 2
\]

which, by Lemma 1, part a implies the lemma for \( r_A + r_B > 1 \). The proof for the case \( r_A + r_C > 1 \) follows similarly from Lemma 1, part b.

Lemma 3: If both \( r_A + r_B < 1 \) and \( r_A + r_C < 1 \) then \( \text{inc}^*(t) \) is given (in terms illustrated in Figure 2) by:

\[
\text{inc}^*(t) = \text{inc}(D^*) = 2(r_A \cos(a/2) + r_B \cos\beta_1 + r_C \cos\gamma_1 + 1 - \cos(a/2)).
\]

Proof: Again, by definition (1) we have:

\[
\text{inc}(D^*) = (a(0) - a(t)) + (b(t) - b(0)) + (c(t) - c(0)).
\]

The following trigonometric equalities are easily verified by inspection (see Figure 2):

\[
a(0) = r_C \cos\gamma_1 + r_B \cos\beta_1, \quad a(t) = 0
\]

\[
b(0) = (1 - r_A) \cos(a/2) - r_C \cos\gamma_1, \quad b(t) = 1
\]

\[
c(0) = (1 - r_A) \cos(a/2) - r_B \cos\beta_1, \quad c(t) = 1.
\]

The lemma follows by simple substitution.

Lemma 3 provides an expression for \( \text{inc}^*(t) \) in terms of \( a/2, \beta_1 \) and \( \gamma_1 \). To compute \( \text{inc}^*(t) \), given \( r_A, r_B \) and \( r_C \), one must express \( a/2, \beta_1 \) and \( \gamma_1 \) in terms
of these parameters. By the Sine formula and the fact that 
\( \sin(180^\circ - x) = \sin(x) \), the following equalities are derived:

(i) \( r_B/\sin(a/2) = (1 - r_A)/\sin\beta_1 \)
(ii) \( r_C/\sin(a/2) = (1 - r_A)/\sin\gamma_1 \)

We also get by inspection that \( a = 2(\beta_1 + \gamma_1) - 180^\circ \), or:

(iii) \( a/2 = \beta_1 + \gamma_1 - 90^\circ \).

The above equations can be solved by first finding \( \beta_1 \), and then substituting

(iv) \( \sin(a/2) = r_B\sin\beta_1/(1 - r_A) \) and

(v) \( \sin\gamma_1 = (r_B/r_C)\sin\beta_1 \).

Our last Lemma provides a method to find \( \sin\beta_1 \):

**Lemma 4:** Let \( q = r_B/r_C \) and \( M = r_B/(1 - r_A) \), then \( \sin\beta_1 \) is the positive root \( y \) of the equation

\[
(1 - q^2y^2)^{1/2}(1 - y^2)^{1/2} - qy^2 + My = 0
\]

**Proof:** From (iii) above we have that \( \sin(a/2) = -\cos(\gamma_1 + \beta_1) \). By substituting in (i) we obtain:

\[-r_B/\cos(\gamma_1 + \beta_1) = (1 - r_A)/\sin\beta_1 \]

which can be rewritten as:

(vi) \( (\sin\gamma_1\sin\beta_1 - \cos\gamma_1\cos\beta_1)/\sin\beta_1 = r_B/(1 - r_A) = M \).

By the Sine formula we have \( \sin\gamma_1/\sin\beta_1 = r_B/r_C = q \). Hence, (vi) can be written as:
\[ q \sin^2 \beta_1 = (1 - q^2 \sin^2 \beta_1)^{1/2} (1 - \sin^2 \beta_1)^{1/2} / \sin \beta_1 = M \]

which implies the Lemma.

To see that the correctness of COMPUTE* is implied by the above lemmas, observe that the correctness of step 1 is implied by Lemma 2, steps 2a. through 2c. provide a computation of \( \sin \beta_1 \) as given in Lemma 4, steps 2d. to 2e. compute \( \sin(a/2) \) and \( \sin_1 \) respectively, by equations (iv) and (v), steps 2f.-2h. compute the cosines of the same three angles, and step 2i. computes \( \text{inc}^* (t) \) by Lemma 3.

Figure 3 shows a plot of \( \text{inc}^* (t) \). The old bound and the VTP time for a configuration independent protocol are also plotted, for comparison. We chose the case when \( v_A=v_B=v_C=v \) (and hence \( r_A=r_B=r_C=r \)) as representative. The time axis is normalized by \( R/v \) which is the time it takes a node to traverse the network from end to end when travelling at full speed. The curve demonstrates how the growth rate of the new \( \text{inc}^* (t) \) decreases gradually with time until \( \text{inc}^* (t) \) becomes constant. This curve is the basis for computing the extrapolation increments for node-B, given that node-C was last to transmit, and node-A is node-B's immediate predecessor in the round-robin VTP sequence. Implementation and use of these values are discussed in Section 3 below.

Figure 3 - \( \text{inc}^* (t) \) as a function of \( vt/R (=r) \)

3. IMPLEMENTATION

As described in [4], SOSAM's scheduling-function values (called D-values) for each node-i (i.e. the amounts of time node-i must wait from the instance the channel became idle to the instance node-i may safely access the channel without collision), are stored in a table in node-i's memory, with an entry \( D(i,j) \) for each node index, j. Entry \( D(i,j) \) is selected by node-i when node-j was the last
to transmit. The D-values are functions of inter-node propagation delays and of the desired priority discipline. D(i,j) is roughly equal to the sum of virtual token passing times from node-j to node-i, via the nodes that are between them in the priority sequence. If all nodes are stationary these values are fixed. In a network with mobile nodes the D-values change and must therefore be updated from time to time. At any given instance, t units of time after the most recent topology-measurement, the VTP time (denoted R_t(i)) between node-i's predecessor (node-i') and node-i, consists of two components: R_o(i), which is the result of the topology-measurement and INCT(i,j), which is the extrapolation increment used at time t. (Note that while R_o(i) is independent of j, INCT(i,j) and hence R_t(i) have different values for different j). INC(i,j) is ideally equal to (but, to avoid collisions, must not be smaller than) INC*(t) with A, B and C corresponding to i', i and j, respectively (i' is the node immediately preceding i in the round-robin token-passing sequence). If the intervals between measurements are short, regular and known, then a constant, pre-calculated value of each INC(i,j) can be permanently stored and added to each newly measured value of D_o(i,j). This sum can then be used to compute the scheduling function value, D_t(i,j), that will be used by node-i throughout the whole period until the next measurement. If intervals between measurements are not very short, an extrapolation adjustment that is a function of time is preferred (see [5]). A time-varying adjustment is a must if the intervals between measurements are not equal. A constant adjustment step, AINC(i,j), can be pre-calculated for a fixed time interval, At, and the scheduling function value D_t(i,j) can be incremented by that step every At:

\[ R_t(i) = R_o(i) + n \Delta INC(i,j) \]

where n is the smallest integer for which \( nAt > t \). This implementation method is suitable for the "old" bound on INC derived in [5], which grows linearly with t.

When the interval between measurements grows (e.g., due to a temporary node failure, or to the need to complete a high-priority process before a new measurement may begin), fixed increment steps will cause unnecessary deterioration of SOSAM's performance, as explained in Section 1. The new bound, derived in Section 2, can be used to compute INCT(i,j) that will provide better
performance. To take advantage of the new bound which does not grow linearly with time (see Figure 3), storing one pre-calculated increment step is no longer sufficient. Instead, the INC \( t \) curve can be approximated by a piecewise linear curve, and several values, corresponding to the different curve segments, must be stored. \( \Delta INC_k(i,j) \) denotes the increment step for the \( k \)-th segment. The duration of each segment is an integer multiple, \( N \), of \( \Delta t \). The value of \( R_t(i) \) used at time \( t \) after the most recent measurement may be computed as:

\[
R_t(i) = R_0(i) + INC_t(i,j)
\]

\[
= R_0(i) + N \sum_{m=1}^{k-1} \Delta INC_m(i,j) + n \Delta INC_k(i,j)
\]

where \( INC_t(i,j) \geq INC^*(N(k-1)+n)\Delta t \) and \( n \) is the smallest integer for which \( (N(k-1)+n)\Delta t \geq t \) (this defines both \( n \) and \( k \) uniquely, for given \( N \) and \( \Delta t \)). The protocol shifts to a new \( \Delta INC_k(i,j) \) value every \( N\Delta t \) units of time. The smaller \( \Delta t \) the closer \( INC_t(i,j) \) is to \( INC^*(t) \). The number of values that must be stored corresponds to the selected number of curve segments and it depends on the value of \( N \): the larger \( N \) the fewer (but longer) segments on the one hand, but the farther the extrapolation increment from its least upper bound on the other hand. This trade-off is illustrated in Figure 4.

---

**Figure 4** - Various piecewise-linear discrete approximations of the \( INC^*(t) \) curve.

---

Since the derivation of the new \( INC^*(t) \) takes into account the bound, \( R \), on network geographical size, the issue of adapting the scheduling function to new values of \( R \) is a natural one to be raised. Is it necessary to re-compute all \( \Delta INC_k(i,j) \) values whenever the changing network topology dictates a new value of \( R \)? The derivation in Section 2 indicates that \( INC^*(t) \) is really a function of the \( r_i \) values, which incorporate the dependency on \( v_i \) and \( R \): \( r_i = v_i t / R \).
Therefore, the effect of a change in $R$ on $INC(t)$ amounts to a different scaling of the time axis:

$$INC^*(t_2) = INC^*((t_1/v_1)R_2) = INC^*((t_1/v_1)(R_2/R_1)R_1) = INC^*((R_2/R_1)t_2).$$

The scaling factor is $R_2/R_1$.

By adjusting the rate at which the protocol increments $R_t(i)$ we can obtain the appropriate scaling. This can be done by adjusting $At$: if the protocol has been using $At=At_1$ when $R$ was equal to $R_1$, then when $R$ changes to $R_2$ the protocol must use $At=At_2=At_1(R_2/R_1)$. Since a change in $R$ can be accommodated by a change in $At$, a fixed (and relatively small) set of $\Delta INC_k(i,j)$ values can be used for each $(i,j)$ pair, for a given set of node maximal speeds and for a given segmentation of the $INC^*(t)$ curve.

This observation is especially important when the network re-estimates $R$ every time it measures the topological information ($R$ may be estimated from this information along with information on node maximal speeds). Network token passing times can thus be kept near optimal by simply adjusting $At$.

4. RESULTS AND EVALUATION

The purpose of using a tighter bound on $INC(t)$ is to reduce the additional scheduling overhead that is due to the nodes' changing topology. In [5] we showed that $inc(t)$, the average value of $INC(t)(i,j)$ is the main factor determining the extrapolation overhead at any given time, $t$. In order to evaluate the advantage of the new bound over the "old" one derived in [5] we computed $inc(t)$ by both methods for several representative distributions of maximal speeds in a 10-node example network. We then plotted the results on the same set of coordinates (see Figure 5). For additional insight we also plotted the minimal scheduling overhead obtainable by a configuration independent round-
robin VTP protocol (e.g. BRAM) that uses only the bound on network end-to-end distance.

The curves in Figure 5 provide the extrapolation overhead of the example network at various situations. The overhead is represented by the average increment in VTP time (inc), normalized by the network end-to-end propagation delay (\( a \)). Values of inc are plotted for various values of \( r_{\text{max}} \) (which is the proportion of network end-to-end distance that the fastest node can travel during the corresponding extrapolation period). To simplify computation it was assumed that all mobile nodes have the same maximal speed, \( v_{\text{max}} \). The percentage, \( p \), of mobile nodes in the network was varied as a parameter. The distribution of mobile nodes along the transmission sequence turned out to have minimal effect on the value of inc. Since \( r_{\text{max}} \approx v_{\text{max}} t/R \), each point on a curve represents the extrapolation overhead for a whole set of combinations of values for \( v_{\text{max}}, t \) and \( R \).

---

**Figure 5 - inc vs. \( r_{\text{max}} \) with \( p \) as a parameter**

---

From the curves we see that at low values of \( r_{\text{max}} \) (< 0.1) the old method and the new method provide very close values of inc, both much smaller than the scheduling overhead imposed by a configuration independent protocol. This has already been indicated in [5]. At high values of \( r_{\text{max}} \) the advantage of the new bound becomes substantial. We also see that if the old bound is used then the scheduling overhead will increase beyond that of a configuration independent protocol. The crossover point (for \( p=100\% \)) is at a value of \( r_{\text{max}} \) less than 0.41. The additional overhead due to motion (which is often the major component in the total scheduling overhead) that is implied by the new bound is always less than the total overhead implied by BRAM, and much less for small and medium proportions of mobile nodes (a situation that a configuration independent protocol can not take advantage of).
5. SUMMARY AND CONCLUSION

In this paper we introduced a new bound on the topological rate of change in a network with mobile nodes. The new bound takes into account the bound on network geographical size. If employed by virtual-token-passing access protocols (specifically - SOSAM), for the purpose of extrapolating required topological information, the new bound allows the protocol to provide performance that is in most cases better than that of configuration independent access protocols. Unlike a previously derived bound, performance degradation (due to extrapolation overhead) with increased time between topological measurements, tapers off and converges to a constant. This property also relieves the network from the burden of supporting and coordinating the use of more than one access protocol.

A possible implementation of the scheduling information derived from the new bound is also discussed: A node's extrapolation increments that are derived from the new bound are stored in a set of tables residing in that node's memory. Depending on the amount of time that passed since the last topological measurement, the appropriate table is used for some time. When the network updates the bound on its geographical size (this bound is derivable from the topology measurement results and from node maximal speeds), the nodes need not re-compute all table entries. Instead, the new information is simply translated into the appropriate rate at which the network uses table entries to increment its scheduling function values.
Figure 1 - The geometry of a displacement and related notation.
Figure 2 - The Worst Displacement $D^*$
Figure 3 - inc*(t) as a function of vt/R (=r)

![Graph showing the relationship between inc*(t) and vt/R](image)

- **Old** bound
- **New** bound

**Configuration independent protocol**

**Time since the last measurement**

**New bounds**

**Old bound**

---

Technion - Computer Science Department - Technical Report CS0321 - 1984
Figure 4 - Various piecewise-linear discrete approximations of the INC(t) curve.
Figure 5 - inc versus $r_{\max}$ with $p$ as a parameter.

- "old" bound
- new bound

$p=100\%$ of nodes are mobile

configuration independent protocol

$p=40\%$

$p=20\%$

$p=10\%$

Average increase in VTP time, inc
(in units of end-to-end propagation delay, s)

Time since last measurement
(in units of end-to-end traversal time by mobile node)
REFERENCES


