stantially simplified the formulation of the delay bound computation (inequality (1)), and to Eberhard Mueller-Menrad, who discovered several bugs in an earlier version of Tables 1 and 2. He is also thankful to Ellen Boyle for her invaluable help with the preparation of the manuscript.

References

[BaMa91] A. Banerjea and B. Mah. "The Real-Time Channel Administration Protocol". *Proc. Second International Workshop on Network and Operating System Support for Digital Audio and Video*, Springer-Verlag, Heidelberg, Germany, 160–170, November 1991.

[FeBZ92] D. Ferrari, A. Banerjea, and H. Zhang. "Network Support for Multimedia - A Discussion of the Tenet Approach". Technical Report TR-92-072, International Computer Science Institute, Berkeley, October 1992.

[Ferr92] D. Ferrari. "Real-Time Communication in an Internetwork". *Journal of High Speed Networks*, 1(1):79–103, 1992.

[Ferr93] D. Ferrari. "Distributed Delay Jitter Control in Packet-Switching Internetworks". *Journal of Internetworking: Research and Experience*, 4(1):1–20, 1993.

[FeVe90] D. Ferrari and D. Verma. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks". *IEEE Journal Selected Areas of Communication*, 8(4):368–379, April 1990.

[FKKM92] A. G. Fraser, C. R. Kalmanek, A. E. Kaplan, W. T. Marshall, and R. C. Restrick. "Xunet 2: A Nationwide Testbed in High-Speed Networking". *Proc. INFOCOM'92*, Florence, Italy, 1992.

[JSZC92] S. Jamin, S. Shenker, L. Zhang, and D. D. Clark. "An Admission Control Algorithm for Predictive Real-Time Service". *Proc. Third International Workshop on Network and Operating System Support for Digital Audio and Video*, Springer-Verlag, Heidelberg, Germany, 349–356, 1992.

[LaPa91] A. A. Lazar and G. Pacifici. "Control of Resources in Broadband Networks with Quality of Service Guarantees". *IEEE Communications Magazine*, 29(10):66–73, October 1991.

[ZhKe91] H. Zhang and S. Keshav. "Comparison of Rate-Based Service Disciplines". *Proc. of ACM SIGCOMM '91*, Zurich, Switzerland, 113–122, September 1991.

of course, makes the tasks of the destination heavier (and reduces those of the "intermediate destinations" to zero), as well as calling for much longer establishment messages, but simplifies the channel establishment code very substantially. It should also be easier to design and implement this code so that it is truly portable to a wide variety of internetworking environments.

All of these considerations should obviously (and will) be confirmed by experience with applying the new admission control method in several internetworking testbeds.

5 Conclusion

A more general and intellectually satisfying approach than the one previously proposed by the Tenet Group to admission control for real-time communication services in packet-switching internetworks has been presented. The new approach is based on a real-time channel model consisting of a cascade of single queueing servers, each one of which may be of one of two types (J or NJ) and contains given amounts of three types of resources; the connections between servers are characterized only by their (constant) delay. The admission tests and computations are designed and have to be executed for each server along the path of the channel to be established. The only aspect of a server that influences (some of) the tests and computations is the scheduling discipline implemented by the server. Servers with different scheduling disciplines can coexist along a channel's route, but the appropriate tests and computations will have to be run in each server during the establishment procedure. J-type and NJ-type servers can also coexist, of course. To show an example of tests and computations, we have presented those for the case of the D-EDD and J-EDD disciplines.

The advantages and the drawbacks of the new method have been briefly discussed. The method's generality is its most important characteristic: it is at the root of such advantages as the portability of the tests and computations as well as of the code, and the simpler one-level approach to admission control in internetworks; it is also responsible for such relative disadvantages as the higher establishment overhead and the longer establishment messages.

It seems clear, on the basis of these considerations, that the new method makes a positive and useful contribution to the technology of real-time communication. Only careful testbed implementation and experimentation, however, can confirm the validity of this conclusion.

Acknowledgements

The author is grateful to the members of the Tenet Group for being intelligent and creative sounding boards in several discussions of these topics. He is particularly indebted to Amit Gupta, who provided many suggestions and sub-

cases. The new method replaces it with a single server to be used as a building block for much more complicated and realistic node models. Since the new tests and computations pertain to this building block and not to the node model, the node model can be changed without having to modify the tests or the computations to be performed for admission control. Some of the tests and computations depend on the scheduling discipline implemented by the server; the correct ones will obviously have to be used during the channel establishment procedure in each server, the only condition (fortunately an easy one to satisfy) being that different tests and computations in different servers will interoperate, so that the destination will receive all the information it needs to make correct admission and allocation decisions.

It should be noted that most of the tests and computations devised for Schemes 0 and 1 are valid for each single server, as they were developed for nodes that were represented by single servers. This observation has allowed us to benefit greatly from our previous work; for instance, we did not have to prove inequalities (1), since the proof had already been given in [FeVe90]. The only difference, but a non-trivial one, is that tests and computations are now to be applied to each server, not to each node. Since it is to be expected that there will be one processor in each node running tests and computations whenever a channel establishment request is received, that processor will have to run as many sets of tests and computations as there are single servers in the model chosen for that node. One could argue that this will increase the overhead and the time of channel establishment; the counterargument, however, is that the new method is reliable, while the old one was rather shaky, either because it reduced the entire, possibly quite complicated, node to a single server, or because it relied on the assumption that the node had a severe bottleneck that could be correctly identified and would never change.

Another important consequence of the new approach is found in the implementation of admission control within an internetwork. With the old method, heterogeneous internetworks were treated in a hierarchical way: each subnetwork had its own characterization, and was dealt with, to a large extent, independently of the rest of the internetwork. The channel's destination allocated resources (i.e., assigned local performance bounds) only to highest-level subnetworks and links; within each subnetwork, such assignment was done by the "local destination" (the endpoint of the channel's path within the subnetwork), and so on recursively [Ferr92, BaMa91]. All of this can be done without violating the single roundtrip requirement of the Tenet channel establishment procedure. However, while the hierarchical procedure is elegant and efficient, its implementation, debugging, and testing are quite complicated. The new method does not require resorting to such a procedure, as long as each internetwork component along a channel's path can be modeled as a cascade of single servers as in Figure 2. If all heterogeneity can be absorbed by the modeling generality of the new method, a condition that is likely to be satisfied in practical cases, we can deal with one-level paths even when the internetwork is complex. This,

Table 2: Destination Tests for Admission Control

Name	Computation	Test	Forward
D-Test	$D_{prop,i} \leftarrow \sum_{n=1}^{N} D_{prop,n}$; $D_{min,i} \leftarrow \sum_{n=1}^{N} D_{min,n}$	$D_{min,i} \leq D_{max,i} \leq D_{prop,i}$	–
W-Comp	if $W_{min,i} < 1$, $w_{n,i} \leftarrow {W_{min,i}}^{1/N}$ $(n = 1,..N)$ if $W_{min,i} = 1$, $w_{n,i} \leftarrow 1$ $(n = 1,..N)$	–	$w_{n,i}(n = 1,..N)$
Z-Test	if $Z_{min,i} < 1$, $Z_{max,i} \leftarrow \prod_{n=1}^{N} zmax_{n,i}$, $z_{n,i} = zmax_{n,i}\left(\frac{Z_{min,i}}{\prod_{n=1}^{N} zmax_{n,i}}\right)^{1/N}$ $(n = 1,..N)$	$Z_{min,i} \leq Z_{max,i}$	$z_{n,i}(n = 1,..N)$

d-Comp	if $\sum_{n=1}^{N} dmax_{n,i} \leq D_{max,i}$, then
	$D'_{max,i} \leftarrow \sum_{n=1}^{N} dmax_{n,i}$,
	$d_{n,i} \leftarrow dmax_{n,i}(n = 1,..N)$
	if $\sum_{n=1}^{N} dmax_{n,i} > D_{max,i}$, then
	$D'_{max,i} \leftarrow D_{max,i}$
	$d_{n,i} \leftarrow dmin_{n,i} + \frac{1}{N}(D_{max,i} - D_{min,i})$
	$(n = 1,..N)$
		–	$d_{n,i}(n = 1,..N)$
			$D'_{max,i}$

J-Test	if $J_{max,i}$ is defined, then m = position of the last J-type server and $J_{N,i} \leftarrow \sum_{k=m}^{N} (d_{k,i} - t_{k,i})$; $D_{prop,n,i} \leftarrow \sum_{k=1}^{n} d_{prop,k}$ $(n = 1,..N)$ $D^*_{n,i} \leftarrow D_{prop,n,i} + \sum_{k=1}^{n-1} d_{k,i}$ $(n = 1,..N)$	$J_{N,i} \leq J_{max,i}$	$D^*_{n,i}(n = 1,..N)$
b-Comp	$\Delta_{n,i} \leftarrow \sum_{k=m}^{n} (d_{k,i} - t_{k,i})$ if $J_{max,i}$ is undefined, then $m = 1$ if $J_{max,i}$ is defined, then m = position of the previous J-type server $b_{n,i} \leftarrow s_{max,i}\left\lceil \frac{d_{n,i} + \Delta_{n,i}}{x_{min,i}} w_{n,i} \right\rceil$ $(n = 1,..N)$	–	$b_{n,i}(n = 1,..N)$

Table 1: Server Tests for Admission Control

Name	Condition	Computation	Test	Record	Forward
Deterministic Test	$Z_{min,i} = 1$	$U_{det,n} \leftarrow U_{det,n} + \frac{t_{n,i}}{x_{min,i}}$ where $U_{det,n}$ is initialized to 0	$U_{det,n} < 1$	$U_{det,n}$	–
Statistical Test	$U_n \geq 1$ where $U_n = \sum_{k=1}^{c_n+1} \frac{t_{n,k}}{x_{min,k}}$	$p_k = \frac{x_{min,k}}{x_{ave,k}}$ for $k = 1, 2, ..c_n + 1$ H: set of overflow combinations of channels $P_{do,n} \leftarrow \sum_H \Big[\prod_{k \in H} p_k \prod_{l \notin H} (1 - p_l) \Big]$ $zmax_{n,i} \leftarrow 1 - P_{do,n}$ if $Z_{min,i} < 1$; $z_{n,i} = zmax_{n,i} \leftarrow 1$ if $Z_{min,i} = 1$	$z_{n,k} \leq 1 - P_{do,n}$ (for $k = 1, 2, ..c_n$, and $z_{n,k} < 1$)	$P_{do,n}$	$zmax_{n,i}$
Delay Bound Computation	–	Let $d_{n,1} \leq d_{n,2} \leq .. \leq d_{n,c_n}$ for $k = 1, 2, ..c_n$; if $d_{n,h} - \sum_{l=1}^{h} t_{n,l} - T_n - t_{n,c_n+1} \geq 0$ for $h = k, k+1, ..c_n$, then $dmin_{n,i} \leftarrow \sum_{l=1}^{k} t_{n,l} + T_n + t_{n,c_n+1}$	–	$dmin_{n,i}$	$dmin_{n,i}$
Buffer Space Test	–	$bmax_{n,i} \leftarrow B$, where $B \leq B_n - \sum_{k=1}^{c_n} b_{n,k}$; $b_{n,i} \leftarrow bmax_{n,i}$ $dmax_{n,i} \leftarrow \frac{bmax_{n,i} x_{min,i}}{s_{max,i}} - \sum_{k=m}^{n-1} (dmax_{k,i} - t_{k,i})$ (i) if $J_{max,i}$ is undefined, then $m = 1$; (ii) if $J_{max,i}$ is defined, then m = position of the previous J-type server	$dmax_{n,i} \geq dmin_{n,i}$	$b_{n,i}$ $dmax_{n,i}$	$dmax_{n,i}$

(a) the addition of a new channel with a deterministic delay bound (i.e., $Z_{min,i} = 1$) would cause the worst-case utilization of the server by deterministic channels to rise above 1 (**deterministic test**);

(b) the addition of the new channel would cause the worst-case utilization of the server by all channels to reach a point where the probability of violating the local delay bound of an established channel would be greater than its local bound for that probability (**statistical test**);

(c) the amount of buffer space that is available for the new channel in the server would allow its packets to spend there a maximum amount of time larger than the minimum local delay bound the server can offer the new channel (**buffer space test**).

The minimum local delay bound $dmin_{n,i}$ offered by the server n to new channel i is the minimum value of $d_{n,i}$ that makes the packets on channel i schedulable. Computing $dmin_{n,i}$ is the goal of the **delay bound computation** to be performed by the server before the buffer space test. As shown in Table 1, which summarizes the admission tests for EDD-scheduled servers, this computation entails sorting the established channels in order of increasing local delay bound, and determining the first position in this list where the new channel can be inserted without causing any of the local delay bounds of the channels below it to violate the condition

$$d_{n,h} \geq \sum_{l=1}^{h} t_{n,l} + T_n + t_{n,c_n+1} \ (h = k, k+1, ..c_n; k = 1, 2...c_n), \qquad (1)$$

where T_n is the maximum service time server n may give to any packet (including the non-real-time packets), c_n is the number of channels established in server n, and $c_n + 1$ is the index locally assigned to the new channel. The value of $dmin_{n,i}$ is the value of the right-hand side of (1) for $h = k_o$, where k_o is the smallest value of k such that inequalities (1) are all satisfied. The proof of this result can be found in [FeVe90].

The admission tests and computations to be performed by the destination are collected in Table 2. Note that there are several different ways of organizing the tests and computations; the choices made in building Tables 1 and 2 have been dictated by the desire to simplify understanding and to avoid local computations during the reverse trip; they are by no means optimum in any sense of the term.

4 Discussion

The admission control method described in the previous section is much more general than that on which the Tenet Schemes 0 and 1 have been based so far. The assumption made by the latter, that each node could be satisfactorily represented as a single queueing server, is clearly inadequate in most practical

Figure 2: The model of a channel's path.

- *delay resource*,

where, depending on the component modeled by the server, computing power may represent the processor power or the link bandwidth, and where the *delay resource* refers to the feasibility of scheduling all packets by the server's scheduler within the respective local delay bounds.

A server on the modeled channel's route is in general traversed by a number of other, already established real-time channels. Each channel's end-to-end characterization is the 8-tuple of traffic and performance parameters listed in Section 2 above. In each server n, an established channel k has the following characterization:

- $d_{n,k}$: local delay bound (used to compute the EDD deadline),
- $z_{n,k}$: local minimum probability of delay bound violation,
- $w_{n,k}$: local minimum probability of buffer overflow,
- $t_{n,k}$: maximum packet service time.

While the first three parameters of this 4-tuple are computed during the establishment procedure, $t_{n,k}$ is assumed to be known in each server before the server is visited by the establishment message during its forward trip. In a J-type server, a channel whose delay jitter bound J_{max} is specified is characterized also by the canonical delay $D^*_{n,i}$, which is the delay with respect to the departure time from the source each packet on that channel will be forced to have when it is given to the scheduler in that server.

Another quantity that is known beforehand (and that we assume is transmitted to the destination through the establishment message for each visited server) is the propagation delay $d_{prop,n}$ of the link preceding server n along the new channel's route. Note that this quantity is the only one that characterizes each link in the model in Figure 2.

During the forward trip of the establishment message, we can see whether in each server

of the Multi-class EDD discipline in its two versions: the so-called Delay-EDD (or D-EDD) [FeVe90, ZhKe91] and Jitter EDD (or J-EDD) [Ferr93, ZhKe91].[3] Rate control, either at the periphery of the network or in all of its nodes, is needed to protect well-behaving channels from the misbehavior of faulty or malicious sources, and from the occasional bunching of packets on a channel due to traffic fluctuations. While the latter effect is automatically cured by the J-EDD scheduling discipline, which restores the original traffic pattern in each node implementing this policy, the former effect (and both effects in a D-EDD node) can be eliminated by suitably postponing the offending packets' deadlines in the node's scheduler [FeVe90].

3 The new admission control method

Figure 2 illustrates the model that we adopt for a simplex unicast real-time channel. The additional arrows symbolically allude to the fact that each server is shared among many channels. The crucial difference, with respect to the one used in the original Schemes 0 and 1, is that here each node visited by the channel may be represented by more than one single server. Thus, the new model is a much more general one.

For example, the node model in Figure 1 will contribute two servers for each node to the channel model in Figure 2. Each server is of one of two types:

(i) *J-type*, if it is scheduled by a discipline that can control jitter (in our example, the J-EDD discipline); and

(ii) *NJ-type* otherwise.

The admission tests we discuss here assume that the NJ-type servers are scheduled by the D-EDD discipline, but both D-EDD and J-EDD can easily be replaced by other disciplines, as long as the tests are modified appropriately. There is no need for all servers of either type to be homogeneous: servers may be scheduled by a variety of policies even within the same network and the same channel route, but of course we have to use the correct set of tests for each one of them.

Each server is endowed with given amounts of the following resources:

- buffer space,
- computing power,

[3] D-EDD chooses for transmission the packet with the earliest deadline on a deterministic channel if transmitting the one with the earliest deadline on a statistical channel were to violate the former's deadline; otherwise, it chooses the latter, unless the first non-real-time packet can be transmitted without violating the statistical packet's deadline, in which case the non-real-time packet is shipped. J-EDD is based on the same discipline but holds each packet that has arrived too early before giving it to the scheduler.

- $J_{max,i}$: the end-to-end delay jitter bound,
- $W_{min,i}$: the minimum probability that a packet is lost due to buffer overflow.

The Tenet approach is connection-oriented and reservation-based: before a real-time channel can be used by its requestor, it must be established (i.e., resources for the channel must be set aside along its route), so that the desired performance guarantees are supported. Note that these resources are not reserved for the channel's exclusive use; the channel only has higher priority for using them, but, in keeping with the principles of packet switching, they can be used by non-real-time traffic whenever no packets from the channel are present.

Channel establishment is a distributed process. A message issued by the source (though it could be issued by the destination instead) visits each node (switch, router, gateway) on the route of the channel. This message causes several admission tests and computations to be performed at each node. If the new channel passes all the tests in a node, the message is forwarded, with some state information about the current node, to the next node on the route. The final tests are performed by the destination; if they are successful, a channel-established message is sent by the destination to the source along the reverse route; when each node is revisited, the message corrects the tentative reservations made in that node by the forward message, and informs the node about the performance bounds assigned to it. Some of these bounds will be used during data transfer operations, others in tests for the admission of future channels, and some in both types of circumstances.

If any of the tests in the nodes or in the destination fails, the channel cannot be established, and a channel-rejected message is immediately sent back to the source. This message removes in each node it visits the tentative reservations made for the new channel.

A new channel's route may be computed by the source (or destination) if this host has the necessary topological information; knowledge by this host of the current real-time load (i.e., a measure of how much of each resource is currently earmarked for use by a real-time channel), and of such additional information as propagation delays and error rates of the links involved, is also quite useful to increase the probability that the chosen route will be able to support the new channel. An alternative is the construction of the route in a hop-by-hop fashion, with individual nodes usually not knowing much beyond the real-time loads of their immediate neighbors.

To complete this general description of the Tenet approach, we only have to mention two special aspects of data transfers: scheduling and distributed rate-control. Most scheduling policies can be used for real-time communication under fairly liberal conditions [Ferr92]. Since some of the tests to be performed in a node during channel establishment are dependent on the scheduling discipline implemented in that node, we have chosen to restrict this discussion to the case

Figure 1: A node model showing the three interface processors serving the three output links.

included in the new node model. This solution was obviously unsatisfactory: it lacked generality, and required determining where, in each node, was the bottleneck processor, a designation that could clearly change with the load on the node.

A general, clean solution was obviously needed. Such a solution is described here. Section 2 provides the background of our investigation by summarizing the Tenet approach. The new admission control method is presented in Section 3, and its characteristics are discussed in Section 4. Section 5 contains concluding remarks.

2 The Tenet approach

The Tenet real-time communication Schemes 0 and 1 are based on a simplex unicast network-layer abstraction, the *real-time channel*, characterized by traffic and performance bounds specified by the client. The traffic bounds for channel i are the following:

- $x_{min,i}$: the minimum inter-packet interval,
- $x_{ave,i}$: the minimum of the average inter-packet intervals,
- I_i : the averaging interval used in the definition of $x_{ave,i}$,
- $s_{max,i}$: the maximum packet size.

The performance bounds for channel i are:

- $D_{max,i}$: the end-to-end delay bound,
- $Z_{min,i}$: the minimum probability that packet delay is not greater than $D_{max,i}$,

number of real-time connections[1] admitted to share simultaneously the resources of a packet-switching network, the absence of congestion, hence the quality of the received message streams, could not be guaranteed.

Many methods can be resorted to in order to determine whether a new real-time connection can be added to those already existing in the network, and in fact several have been proposed in the literature (see for example [LaPa91] and [JSZC92]). One of the earliest algorithms presented [FeVe90] is the first method based on the Tenet approach, a method which was later called Scheme 0 [FeBZ92]; this scheme, which had been developed for simple homogeneous networks, was subsequently extended to complex heterogeneous internetworks, and became Tenet's Scheme 1 [Ferr92]. The principle on which both schemes are based is that of verifying, at the time a request for a new connection is made, whether the resources available along the path of the new connection are sufficient, even in the worst possible case, to

(i) provide the new connection with the performance guarantees it needs, and

(ii) allow the guarantees offered to all the existing real-time connections to continue being satisfied.

These conditions are tested in each of the nodes along the path, and the request message is allowed to proceed to the next node only if all tests in a node are satisfied. Since the Tenet schemes have been designed to reach an admission or rejection decision in only one roundtrip, the final tests take place in the destination[2] of the connection. Both types of tests (node tests and destination tests) depend on the types of bounds the client wants for the new connection, and on the model assumed for the node. In building Scheme 0 and then Scheme 1, we implicitly assumed that a node could be represented by a single queueing server with a finite amount of buffer space in its queues and with a Multi-Class Earliest Due Date (EDD) scheduler [FeVe90]. However, when Scheme 1 had to be applied to a real internetworking testbed such as Xunet 2 [FKKM92], we found that our node model was inadequate, primarily because nodes did not just consist of one processor, but included other processors on their network interface boards. A more realistic model for a node consisting of a general-purpose computer system is shown in Figure 1 as an example.

This problem was temporarily circumvented by introducing in some of the tests a variable representing, rather than the packet service time of the central processor, the packet service time of the bottleneck processor among all those

[1] The term *real-time connection*, strictly speaking, denotes a connection characterized by upper bound on packet delays; however, it has for several years been applied to connections offering bounds on some other performance or reliability indices (e.g., throughput, loss rate) as well, or instead. It will be used here with the latter, extended meaning.

[2] This discussion assumes that the request message is issued by the source of the new connection. A trivial modification of the schemes allows them to be used also when the message starts from the destination and proceeds backwards toward the source.

A NEW ADMISSION CONTROL METHOD FOR REAL-TIME COMMUNICATION IN AN INTERNETWORK

Domenico Ferrari
The Tenet Group
Computer Science Division
EECS Department
University of California at Berkeley
and International Computer Science Institute

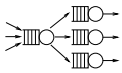
Abstract

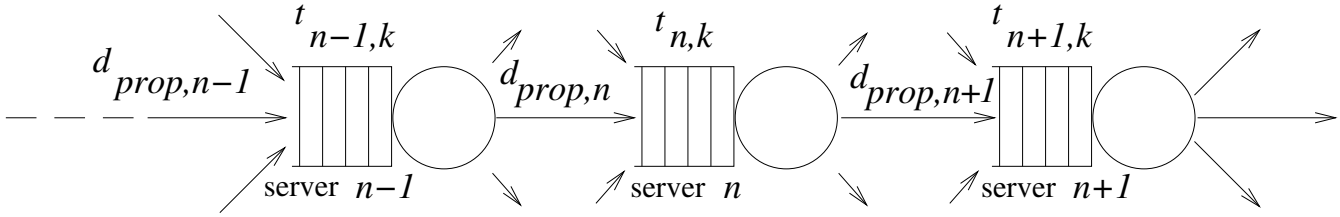
Admission control is a necessary component of a real-time communication service. The admission control method adopted so far in the Tenet approach to real-time communication was based on a very simple, not very realistic node model. The method described here can be applied to a wide variety of node models with a wide spectrum of accuracies, and is therefore much more general than the previous one. The admission control tests and computations for the case of the Earliest Due Date scheduling discipline are shown as an example of application of the new method. The main advantages and drawbacks of this method are discussed; the predominance of the advantages is expected to be confirmed by experience with testbed implementations.

1 Introduction

Admission control for connections is now universally recognized as a necessary component of a real-time communication service. Without a limitation of the

[0]This work was supported by the National Science Foundation and the Defense Advanced Research Projects Agency (DARPA) under Cooperative Agreement NCR-8919038 with the Corporation for National Research Initiatives, by AT&T Bell Laboratories, Hitachi, Ltd., Hitachi America, Ltd., Pacific Bell, the University of California under a MICRO grant, and the International Computer Science Institute. The views and conclusions contained in this document are those of the author, and should not be interpreted as representing official policies, either expressed or implied, of the U.S. Government or any of the sponsoring organizations.




$t_{n-1,k}$
$t_{n,k}$
$t_{n+1,k}$
$d_{prop,n-1}$
$d_{prop,n}$
$d_{prop,n+1}$
server $n-1$
server n
server $n+1$