Internet Multicast Routing Infrastructure

Proposal to the National Science Foundation Division of
Network, Communications and Research Infrastructure

David L. Mills, Ashfaq A. Khokhar, Guang R. Gao
Electrical Engineering Department
University of Delaware

Research Summary

This is a proposal for research in the area of distributed multicast routing algorithms. It is based on prior work done in support of DNCRI programs, including those in the areas of network time synchronization, network routing algorithms. and in support of DARPA collaborative programs with the DARTnet and CAIRN projects and in other research areas represented by the proposers. The proposed work involves the analysis, synthesis and evaluation of novel network infrastructure routing paradigms. The paradigms are designed specifically to support multicasting as an intrinsic capability of the infrastructure fabric, while building services such as unicast routing, load-deflection, hierarchical multicast groups, admission control and related services layered on this infrastructure.

Our approach is based on a hierarchy of clusters, each covering a manageable area, such as a campus or industrial park, together with a set of algorithms based on a hybrid combination of shortest-path-tree (SPT), minimum-spanning-tree (MST), and Steiner tree principles. The algorithms to be considered early in the project include heuristics recently developed for the automatic configuration of a large set of clients and servers and the subject of a nearly completed dissertation. They construct spanning trees with constrained degree and distance metric, and operate in an incremental add/drop fashion in response to changing network reachability and congestion state.

The significance of the proposed work is that it represents a fresh look at the Internet infrastructure with the specific goal of improving the scaling of the infrastructure to very large networks. With a new paradigm unhampered by legacy constraints, we expect to develop and test a suite of algorithms that will allow a significant increase in the size of the Internet, while reducing protocol overhead, routing database size and duplication of services.

1. Introduction

With the introduction of the information superhighway as one of the alternative way of communication, Computer Supported Collaborative Work (CSCW) environments such as group-decision support systems (GDSS), whiteboards, teleconferencing, interactive seminars, etc., have become the focus of many researchers worldwide [Grudin94]. At the heart of the superhighway infrastructure is the ability to support multicast communication paradigms. The present multicast infrastructure model developed by the IETF community has evolved to a layered architecture and a suite of routing, reservation and applications which depend on an underlying unicast routing paradigm. It is the thesis of this proposal that potentially the same performance hazards that affect other layered architectures, such as the OSI model, can develop if the layered architecture is followed to its natural conclusion. However, the thesis continues, these problems can largely be avoided by a modest degree of realignment in the basic unicast/multicast routing paradigm.

This is a proposal for research in the area of distributed multicast routing algorithms. It is based on prior work done in support of DNCRI programs, including those in the areas of network time synchronization, network routing algorithms and in support of DARPA collaborative programs with the DARTnet and CAIRN projects and in other research areas represented by the proposers.

With support from DARPA under contract DABT 63-95-C-0046, “Scalable, High Speed, Internet Time Synchronization,” one of us (Mills) has been working to develop routing, authentication and auto-configuration for widely distributed network protocols, such as the Network Time Protocol (NTP), which is now widely dispersed in the Internet. This work has led to a collaboration involving University of Delaware (UD), University College London (UCL) and Science Applications International Corporation (SAIC) in support of the DARTnet Networking Security and Mobility Research Collaboration RA96-15. This collaboration, which involves no specific additional funding for the collaboration partners, has been approved by DARPA.[1] The approval, together with the current DARPA contract, which carries with it funding for a 1.5-Mbps connection to the DARTnet research network, provides access to the CAIRN research network, which extends to the research infrastructure operating at 155 Mbps.

While no specific funding for research projects involving the DARTnet/CAIRN infrastructure has been requested from DARPA, we believe the present proposal goes beyond the specific research mission addressed by DARPA and that its scope is appropriate for funding from NSF/DNCRI. The Memoranda of Understandings (MOU) exchanged between the UD, UCL and SAIC collaborators are consistent with this view, in that the research programs of the three collaborators, together with the investigators listed in this proposal, are mutually supportive and synergistic.

The funds requested from NSF are to support a three-year program of analysis, design, implementation and testing. The primary product is a set of database convergence and routing algorithms suitable for a generic Internet of widely deployed unicast and multicast services. The algorithm analysis design phase of this project is to be conducted by UD, with assistance from UCL and SAIC. The protocol design phase is to be conducted primarily by UCL with assistance from UD and SAIC. Implementation of the protocol in the form of a Unix daemon is the primary responsi-

1. The Memorandum of Understanding between DARPA and UD is enclosed with this proposal.

bility of SAIC. The conduct of the testing program, including correctness verification and performance evaluation, will be coordinated by UCL with assistance from UD and SAIC.

2. Results from Previous Work

UD has been a significant contributor to Internet technology for the last decade and, in the case of one of the principal investigators (Mills), for the last two decades. Protocols developed include the Exterior Gateway Protocol (EGP) described in RFC-904 [Mills84] and the Network Time Protocol (NTP) [Mills91, Mills94]. In the case of NTP, the architecture, protocol and algorithms, produced with funding from NSF, DARPA, US Army and US Navy, have been implemented and deployed widely in the Internet in over 100,000 servers and clients. In addition, NSF funded the development of the Fuzzball router software, which was designed and implemented by Mills and used in the NSFnet Phase I backbone.

Prior work on the current NSF grant in the area of multicasting includes the developement of multicast support for the NTP [Mills94], which is now running in many places in the Internet, as well as being distributed with the standard operating systems to current Digital and HP customers. As described later in this proposal, our work on autonomous configuration leads directly to the work proposed.

3. Multicast Routing Algorithms

Multicast communication deals with the dissemination of information from single or multiple sources to multiple destinations. Since the beginning of the computer network era, research focus has been on point-to-point communication patterns. Now we are rediscovering the use of multicast communication as an efficient way of sharing information in collaboration projects. The current multicasting model being developed by the Internet Engineering Task Force (IETF) represents one approach. In what follows, we evaluate the current IETF model, highlight various shortcomings, and suggest an alternative approach to resolve them.

Due to the dynamic nature of multicast group membership , designing efficient multicast routing algorithms that scale well with the change in the size of group is an extremely challenging problem [Doar89]. The design philosophy of these routing algorithms should generally be driven by the following factors [Diot97]:

1. A routing algorithm should minimize the network load. The correctness of the algorithm should be independent of the route taken.
2. Different cost functions, such as bandwidth utilization, node connectivity, end-to-end delay, etc., should determine the optimality of the algorithm.
3. Resource requirements such as routing information stored inside routers should be minimized.
4. Performance of the algorithm should be scalable with the change in the group size, i.e., the problem size.

Several algorithms have been proposed in the literature which partially address some of the above issues. The simplest algorithm is to broadcast (flood) the messages to every node and let each node itself decide if the message is needed or not. Such an algorithm is bound to have low efficiency in terms of network utilization. Digressing from this basic approach several algorithms

have been proposed which build a routing tree based on shortest path distance between the source and the destination nodes. These algorithms are divided into three basic categories: source-based routing algorithms, Steiner tree-based routing algorithms, and core-based routing algorithms. The source-based algorithms [Dalal78, Waitzman88, Thyagajan95, Zhu95] compute a minimum spanning tree for each source based on the shortest path from each source to each destination node. Therefore these algorithms do not require additional resources other than the unicast routing tables.

The Steiner tree algorithms [Cimet87, Noronha94, Clare86, Takashi80, Winter87] design a tree that spans all the group members. The construction of Steiner trees for a large number of group members is extremely computation intensive and at the same time does not guarantee an optimal solution. The algorithm based on such a tree needs to be rerun each time group membership changes.

The core-based routing algorithms (CBT) [Ballardie93, Herzog95] are designed to reduce the size of the routing data base by using a shared tree. Multiple senders and receivers are involved in the computation. The main idea is to develop a tree for each group rather than for each flow or service. A center (also referred to as core) for a group is chosen and messages from multiple sources are first routed towards the center and then spread out to members. This approach suffers from traffic concentration towards the center. A detailed survey of these algorithms can be found in [Diot97].

4. The Current IETF Model

The present multicasting model developed by the IETF research community has evolved to a layered architecture and a suite of routing protocols which depend on a developing service model and resource reservation paradigm. It is the thesis of this proposal that potentially the same performance hazards that affect other layered architectures, such as the OSI model, can develop if the layered architecture is followed to its natural conclusion. However, the thesis continues, these problems can largely be avoided by a modest degree of realignment in the basic unicast/multicast routing paradigm.

The realignment consists of a hybrid unicast/multicast routing architecture, together with algorithmic enhancements to provide distributed, multi-path, multi-service routing. Specific architectural and algorithmic problems are identified below, along with recommendations for a development approach designed to fix them or at least reduce their adverse impact. Since this proposal targets primarily the architecture and algorithms, specific protocol engineering issues are not considered in any depth.

The existing IETF multicast architecture and protocol model has developed incrementally according he following rationale:

1. The unicast routing infrastructure, including the architecture, protocols and service models, is mature and, as a practical matter, resists change. Therefore, the multicasting service must be built on this foundation.

2. The multicast routing infrastructure consists of relatively small islands of service availability separated by explicitly configured encapsulating tunnels represented by the MBONE. The multicast service will not be ubiquitous, at least not in the near term. In order to protect local resources, somewhat ad-hoc scoping schemes and tunnel configurations are necessary. As a result, engineering the multicast routing fabric is fragile and scales awkwardly.

3. Specific multicast groups are relatively sparse; that is, receivers consist of a relatively small fraction of the Internet population and senders are usually a relatively small fraction of the group population. Therefore, it is not practical to invest routing state in routers other than those required for the actual group members. As a corollary, efficient deployment in very large groups typical of interactive simulation exercises is doubtful.

4. In the most common paradigm, group membership is determined by the receivers; that is, receivers join a group by specific protocol requests and the multicast routing for the group is developed from these requests. It follows that multicast routing state specific to each group must be set up and managed by real-time schemes and the responsiveness of these schemes win be a large factor in their perceived performance.

5. Multicast service is most critically important to real-time applications like speech and video. In many cases, real-time traffic cannot be handled efficiently with best-effort service and may require some specific network state, such as proposed in the Integrated Services (IS) working group. This state must be disseminated along the flow graphs in real time using protocols as yet unspecified.

6. In the IETF model, real-time multicast service requires resource reservation of some kind, such as RSVP. Such reservations are specific to a particular multicast delivery tree or aggregates of such trees. Therefore, changes to the underlying unicast routing will affect resource reservations, which may have to wait for soft-state recovery. The system response to a routing transient when large numbers of groups and members are present may seriously affect the performance of real-time applications.

7. Information about ongoing multicast applications is disseminated by a session protocol such as “sd”. The presence of various conferences and public events must be disseminated throughout the multicast service area, allowing promiscuous access to ubiquitous broadcast events. The span of the sd multicast tree must overlay the global multicast infrastructure. Therefore, at least one multicast service tree must span the universe of multicast service areas.

8. Present multicast routing schemes, in particular the MBONE, do not provide for the construction of multicast delivery trees dependent on type of service, available capacity, traffic aggregation or any metric other than hop count, group membership requests or manual configuration. Therefore, these schemes are vulnerable to changing traffic source characteristics and link utilization.

5. Approach

As a natural result of the extended discussion presented in following sections, the following approach is proposed:

1. A hybrid unicast/multicast routing paradigm should be developed that provides efficient routing computations, including shortest-path and shared-tree routes, as a function of resource availability and application requests. The paradigm should include multiple-path flow graphs and not necessarily require backtracking to construct reverse-path trees. In response to the continued rapid growth of the Internet, the paradigm must of necessity be hierarchical.

2. A distributed scheme should be developed that automatically constructs routing trees with specified constraints, such as aggregate bandwidth, maximum fan-in and fan-out degree and maximum path distance. This work follows on existing DARPA-funded work in autonomous configuration paradigms, as well as recent work reported in the ATM community and IETF drafts.

3. A unified model for multicast routing and service provision should be developed, so that distributed network management functions, such as hybrid unicast/multicast routing, resource reservation and traffic shaping, can be done using an intrinsic underlying hierarchical multicast capability (e.g., span-limited flooding).

In this approach a multipurpose database convergence protocol can be used to efficiently distribute such data to all nodes according to hierarchy. The result should be a multicast fabric as robust as the present unicast fabric, yet serve as a medium to exchange such things as routing trees for specific groups, service provisioning and resource reservation. The following sections discuss various specific issues critical to the performance and reliability of a scalable multicasting paradigm in the Internet. They are presented in no particular order and should be considered preliminary to a more thorough analysis and resolution.

6. Discussion

The IETF and its member working groups have proposed a number of schemes suitable for constructing multicast routing trees, including DVMRP, PIM. and CBT, which have been implemented and deployed in some places in the Internet. Each of these schemes constructs trees specific to each group and expands or contracts the tree spans in response to receiver-initiated requests. In addition, the working groups have proposed the IS model, which is intended to instill network flows with specified service parameters, such as guaranteed or probabilistic end-end delays. Finally, the working groups have proposed RSVP, a scheme to reserve resources and assign flows for various classes of applications with single or multiple sources and destinations. Clearly, these schemes represent an integrated, comprehensive model designed to serve a class of services of which real-time, multicast delivery is only one.

The various IETF schemes result in a de-facto layered model, with the existing unicast routing fabric as the lowest layer, the multicast routing schemes as the next upper layer, the IS scheme as the third layer and RSVP as the outermost layer. Somewhere in the middle are the media transport protocols, such as RTP. There are a few gaps in this model, such as a protocol to manage the distributed IS database and a generic admission control algorithm, but these gaps will probably be quickly filled.

There are some serious performance problems raised by the current approach. The various layers mentioned above make strong assumptions about the lower layers on which they depend. For instance, multicast routing assumes that unicast routing is long-term stable; that is, changes in the

unicast fabric occur at intervals normally long compared to the multicast layers ability to reconfigure. The same assumption exists at each layer interface. A hiccup in the unicast routing layer may uproot a tree in the multicast layer, which in turn could cause network flows using that tree to reconfigure, which would disturb resource reservation and probably the media transport applications that use it.

A much more more serious problem may be the limited scope of services assumed at each level about the next lower level. Where this is most critical is the set of assumptions made by the multicast routing layer about the unicast routing layer; in particular, the limitation to single path routes and a single routing metric. A much more flexible and effective service model would include more than just a single unicast route between two nodes in the multicast tree, and include possibly several deflection routes that could avoid traffic concentration on a small number of links while other links are underutilized.

One of the problems identified in the present working group activities is scaling the multicast infrastructure to much larger groups and many more groups than occur in the research community, such as the Defense Simulation Internet. This involves some form of aggregation, either in the fan-in of shared trees or some form of hierarchical routing. These schemes seek to minimize resources required by exploiting shared resources and shared state. This and related points are further explored in following sections.

6.1 On the Nature of Routing Algorithms

Algorithms that find minimum distance paths on a graph have been studied for a very long time. Those that construct shortest-path spanning trees (SPT) for a packet network with defined distances or costs assigned the links represent the most common routing algorithms in use today. Existing algorithms deliver the best (lowest cost) routing for conventional unicast service, but may not be the best choice for multicast service. Most network routing algorithms have as their sole purpose the construction of a SPT for each node, where the path selection is based on a metric, like hop count or link delay, possibly augmented by some traffic-sensitive statistic. The technology of routing algorithms which reliably construct and maintain SPTs is a mature art and proven in many implementations, including ARPAnet (old and new), OSPF, RIP, and the Hello algorithm used in the NSFnet Phase-I system. However, there are some characteristics of existing algorithms that represent significant shortcomings in multicast service. These include:

1. With few exceptions, all Internet routing algorithms construct a single SPT, where the only path found from one node to another is a minimum over all available paths. Therefore, all traffic must flow via that path, which can result in congestion, even if other uncongested paths are available.

2. The SPT for unicast service may not be the best tree for multicast service, especially if the link distances along each direction are unequal. Those multicast routing paradigms under discussion in the IETF assume that the multicast spanning tree can be constructed by reversing the forward path, which may not be the optimum solution.

There are two major classes of distributed routing algorithms in use today - link state, represented by the Dijkstra algorithm, and node state, represented by the Bellman-Ford algorithm. In principle, both classes of algorithms operate the same way. Starting from a distinguished node called the

root, the algorithm assigns a label to each node, where the label represents the minimum distance from the root to that node. Initially, the root is labeled zero and all others are labeled a large number interpreted as infinity. At each step, a node already labeled (the current node) is chosen and the distance to each of its neighbors constructed as the label assigned the current node plus the distance along the incident link to the neighbor. In principle, the various algorithms can be distinguished by the order the nodes are considered, FIFO for Bellman-Ford and sorted-by-distance for Dijkstra. The same algorithm can be used for both unicast and multicast SPTs, but there is an important difference. For unicast routing with unequal link distances in each direction, the distance incident toward the root is used to construct the label. For multicast routing, the distance away from the root is used.

There are two ways to develop the actual SPT. The classic Ford-Fulkerson algorithm, starts with the labeling algorithm above and then constructs the SPT by working backwards. Starting with the set of nodes at maximum distance from the root, the algorithm identifies for each node the (single) link between it and the node next closer to the root. To do this, one of these nodes (the current node) is selected and its label compared with the label of each of its neighbors. If the label of a neighbor node equals the label of the current node less the distance of the incident link to that node, then that link is marked as belonging to the SPT (if there is more than one, select one of them arbitrarily). This process is then repeated for the set of nodes next closer to the root and continued until all nodes other than the root have been processed.

In cases where sufficient topological information is available, as with link state algorithms, it is possible to build the SPT at the same time the nodes are labeled. This is how the Wiretap algorithm [Mills89] (and many others) operates. The result is a set of links that define the SPT for each root. The Wiretap algorithm expands on this technique to construct the k shortest paths and to avoid congested nodes. In this way both the unicast and multicast trees can be constructed. The unicast tree for a given destination represents the set of links used to reach that destination from every other node in the network; the multicast tree for a given source represents the set of links used from that source to reach every other node in the network.

Now, consider the reciprocal paths constructed by reversing the link distances used in calculating the unicast and multicast SPTs. Unless the graph is known to be undirected, so that the distance along each link is independent of direction, the path between the source and destination and the reciprocal path are not necessarily the same. However, reverse-path schemes such as PIM, CBT and RSVP overtly assume the two paths are identical. While for many network structures this may in fact be the case, there is no assurance that for arbitrary structures this will always be the case, especially if multiple paths are available or spanning trees other than SPTs are involved. From the above discussion, calculating the SPTs for either direction is straightforward; the only substantive issue is the size of the database involved.

Recently, multicasting has been modeled as a Steiner problem in networks (SPN) [Bauer95]. Several polynomial time heuristics have been suggested and are claimed to produce near optimal results. The fundamental problem in these solutions is that regardless of the size and sparsity of the multicast group, all the nodes in the network are included in constructing the tree. However, in applications where multicast members are present in the form of sparsely distributed clusters, the proposed solutions make the problem unnecessarily difficult. An interesting aspect that merits further investigation is to consider the network consisting of hierarchies based on the degree of membership of the nodes participating in the multicast. The degree of membership may be

defined as the number of neighbors involved in the multicast group. Based on various design criteria such as fixed number of Steiner nodes, link capacity, etc., multiple Steiner trees may be constructed among the nodes that have a certain range of membership degree. The degrees of nodes participating in the Steiner tree may be decided apriori, depending on the application and the type of the target network. Next, each node in the Steiner tree serves as a root for group members in its neighborhood and SPTs or MSTs may be constructed for neighboring members only. This solution has a promise of being scalable and faster compared with the solutions proposed in the literature. The scalability is from the observation that addition or subtraction of members in a group may affect the routing trees only up to a certain level of hierarchy. The proposed solution is also useful for routing in multiple multicast groups involving multiple data flows.

6.2 On the Advantages of Multiple-Path Routing

With few exceptions, all unicast and multicast routing algorithms currently used in the Internet are single-path algorithms, in that they construct a single SPT on a graph. Among the exceptions, IGRP and OSPF can do load sharing over multiple links, an algorithm proposed by BBN [Haverty82] can build multiple SPTs as a function of link utilization, and the Wiretap algorithm can build the k shortest paths and avoid congested links. It may be of some interest to examine the latter two algorithms in some detail.

The BBN algorithm, which is a modification of the Dijkstra algorithm, was so far as known never implemented. The SPF algorithm used in the ARPAnet and in OSPF is actually a modification of the Dijkstra algorithm in which incremental adjustments to the current SPT are computed without requiring recalculation of the entire tree. The proposed modifications to SPF are designed to provide a load-deflecting capability, where traffic on an overloaded link is deflected to another path. This is done by averaging the measured link utilization over a relatively long interval, like ten seconds. Periodically, these averages are flooded to all nodes using the routing update protocol. If, due to the current routing configuration and utilization, a link becomes overloaded, A new SPT is calculated from the current network graph with the overloaded link removed. The overload traffic is then routed via the recalculated SPT. As long as the overload traffic, or suitable fraction of the incident traffic, can be marked in such a way (perhaps using a probabilistic scheme), then the routers can distinguish which SPT to use - the original or the overload one.

In the Wiretap algorithm, the Dijkstra algorithm is modified to calculate the k shortest paths from a designated root node. The link state database includes factors such as measured link delay, link utilization and packet classes (connected virtual circuit, unconnected virtual circuit, etc.) Operating with this database, the algorithm uses a linear combination of distances and preassigned weights in order to construct source routes for individual packets as they arrive. While this algorithm requires a capability for source routing, there is no intrinsic problem in implementing it using local router databases.

The Wiretap algorithm, in common with the BBN algorithm, operates in a greedy way. The effective distance of each path depends on a composition of the distances calculated along each link, which can change as a function of previously allocated flows. A new flow does not displace existing flows, but can utilize additional paths not included in the baseline SPT. The two schemes thus distribute flows along deflection routes. The problem with both of these schemes is that the deflected route, while of minimum distance over the remaining links, may not be the best according to other factors. For instance, the Wiretap algorithm properly deflects around overloaded links

and nodes; however, like the BBN scheme, the usual result is only a local deflection, where most of the path is common with the original paths, and only the local links near the overload are protected. This has an unfortunate effect where overload tends to cascade, so that, as each link or node approaches saturation, the routing scheme makes a flurry of local adjustments and may itself contribute to the overload due its own control messages.

The situation becomes much more chaotic in the case of multicast routing, since not just single paths are involved, but the entire SPT. A load deflection algorithm could in some cases involve recalculation of the entire SPT and major routing disruptions, including transient loops. When shared trees are included in the scheme, the situation becomes even more critical.

A strategy that may come to bear on these issues is our work with autonomous configuration systems now being supported by ARPA. These systems involve the synthesis of SPTs with specified constraints, such as maximum fan-in and fan-out degree and/or path distance. It is expected that this work will be synergistic with the agenda suggested here and generate ideas useful for the synthesis of robust multicast routing schemes.

6.3 On an Underlying Multicast Fabric

In the various multicast routing protocols development projects, such as CBT and hierarchical multicast, there is a curious assumption that an underlying multicast capability exists in order to serve certain protocol requirements, such as group presence, border router exchanges and so forth. In particular, the basic multicasting functionality appears to be independently established by DVMRP, CBT and RSVP. However, some link-state routing protocols such as OSPF already implement a multicasting capability in the form of a link-state flooding protocol. While node-state routing protocols such as RIP do not need this capability, it is likely that recent enhancements to the basic Bellman-Ford protocol model may require information exchange beyond the current requirement, including only the immediate neighbors of a node. This does not necessarily mean each node must communicate with a set of nodes transparently via its neighbor nodes, just that state incidental to one node may be preserved and passed transparently from one node to another node via other intervening nodes.

A natural question to ask at this point is: why not integrate a multicast capability into the service model provided by the basic network routing-update paradigm? The intent in this approach is to provide an all-nodes connectivity for use by network management algorithms, including those used to construct routes for the actual data. The new capability would be specifically designed to have a robustness level equivalent to an out-of-band channel, so that network instabilities due to data misroute and congestion would have minimal effect on network management functions. This may take the form of a reserved (high) priority for packets of this class, or it could take the form of updates generated directly by the driver, as in the original Hello protocol.

In such a case, the basic network service primitives would include both unicast and multicast paradigms, much in the same spirit as proposed for some ATM switches. However, this hybrid unicast/multicast model is specifically targeted for use by network management functions, where the traffic volume is relatively small and where service models are well understood. As demonstrated from past experience, in particular the MBONE, the expense of true ubiquity required for the hybrid model would not be justified for most actual data traffic. The hybrid routing fabric would be used to disseminate link-state updates, group addresses (sd), border router advertisements,

shared-tree construction algorithms and the like. It would have a low-delay, low-volume utility and be strictly policed for these and similar functions.

With an intrinsic (low-rate) multicasting capability, a number of things become much more simple. Construction, grafting and pruning group-specific trees can be done using distributed algorithms, at least if the frequency of changes to the data structures is manageable. In any case, the intrinsic capability could be used to construct truncated trees for some services using existing paradigms based on DVMRP, for example, in order to shield irrelevant nodes from needless cycles.

Perhaps the most useful advantage of the underlying multicast capability is the ability to construct sophisticated routing structures, such as efficient shared trees, integrated service parameters, resource reservations and multiple-path routing as described previously.

6.4 On Constructing Shared Trees

Returning to the question of unicast versus multicast SPTs, consider the total distance (cost) of the two services. The cost of a unicast path is the sum of the link distances along the path, which is minimized if the path belongs to the SPT. The multicast cost is equal to the sum of all link costs, since the service requires transmission of every packet on every link of the tree. However, in the case of shared trees, the SPT may not represent the tree of lowest overall cost. A minimum-weight spanning tree (MST), where the weight is the sum of all included link costs, may be a better choice. There are a number of algorithms to construct the MST, including the Prim-Dijkstra algorithm. In principle, these algorithms are even simpler than the SPT algorithms and can be implemented in either a centralized or distributed form.

Now, consider the application of the above principles to the construction of shared trees. Schemes such as CBT and sparse-mode PIM operate by identifying one or more designated routers, called core routers, which serve to concentrate flows on a shared tree. First, consider the function which routes packets from a set of senders to a core router. Assuming senders do not usually emit packets at the same time, the minimum cost is achieved using a unicast SPT, with root at the core router and leaves at the senders. Then, consider the function which routes packets from the core router to the current set of destinations. Clearly, the minimum cost is achieved by a MST. If more than one core router is involved, the MST is simply expanded to include all the core routers. Note that is not necessary to designate a root of the MST, since only the total weight is necessary.

It follows from the preceding discussion that individual SPTs represent a considerable investment in router state and in many cases may not result in significant savings in delay or overhead and in some cases may be suboptimal relative to a MST used as a shared tree. Construction of a hybrid SPT-MST scheme might be the most desirable engineering choice; however, several issues remain to be resolved.

First, note that, while the shared tree is often only marginally less efficient than a set of per-node shortest-path trees, use of a shared tree for distribution within the same area as a sending node may be quite inefficient. In extreme, this may require long detours to reach local nodes. A solution for this may be a hybrid strategy involving a SPT-MST but augmented by a per-area local clustering scheme. This could be done using algorithms borrowed from our work in autonomous configuration, which builds SPTs constrained by maximum degree and distance. This of course is the same motivation as for the hierarchical multicast approach suggested recently. One objective

of the discussion here is to demonstrate a unification of both the shared tree and hierarchical approaches.

An appropriate algorithm to construct quasi-optimal shared trees is the cornerstone of an autonomous configuration paradigm. This paradigm is designed to solve the basic problem of automatically configuring a very large network of clients and servers for a specific service. Given a network of clients and severs interconnected by network links, a typical network design problem is to construct a MST specific to the service and instil the design in a set of configuration files disseminated to the set of network nodes involved. When such a design is constrained by such things as maximum fan-in or fan-out degree or maximum subnet diameter, the algorithms in most cases turn out to be NP-hard, that crafted heuristics become necessary.

The object of our autonomous configuration work is to devise distributed algorithms that construct quasi-optimal solutions to these problems in an efficient way. These algorithms are intended for services like network name resolution, time synchronization, resource management and similar applications. They depend in large part on a discovery, refinement and deployment strategy in which multicasting is an important factor. Accordingly, an intrinsic multicasting capability of the network routing fabric considerably enhances the utility and efficiency of the paradigm.

As an example of this approach, consider a network graph and set of (directed) links. Consider the set of SPTs, one rooted at each node, then select the one of minimum total weight. The (single) spanning tree provides a path from every source node to every destination node based only on the distance metric assigned the graph. While this problem can become somewhat ugly, since it scales as O(2^n), it may be an interesting challenge to devise distributed algorithms which approximate the solution in shorter lifetimes. Distributed hill-climbing algorithms come to mind as a starting point.

Continuing in this fashion, assume there are constraints, such as the maximum number of routes that traverse a link, use a node, etc. The problem now becomes much harder. Finally, consider the problem to select the best core router or rendezvous point in a network where the uplink distances to the router are different than the downlink distances. This model could be used to study the case where source nodes unicast packets to the router, which then fans them out via a (single) multicast tree. This is an example of what has been called the warehouse problem, which is a factor of our autonomous configuration research.

6.5 On Concern for Database Explosion

The integrated services (IS) group of IETF is developing a model suitable for providing real-time delivery services in the Internet. This is not the place to discuss this model, other than to observe it requires additional state at each participating router to manage resources and some sort of protocol to distribute the information necessary to manage these resources as the result of customer requests (admission control). For reliable management of these resources, it is necessary to assign them with respect to individual flows as recorded in the state space of each router along the path of that flow. This requires bookkeeping at fan-in and fan-out points of the distribution tree for each flow, as well as accounting for additions and deletions of resource as participants join and leave the flow.

At first glance, this model would seem to suffer acutely from a scalability constraint, since resources might have to be assigned to each link of each spanning tree and resource assignment individually managed for each link and node. The designers point out that individual flows, and presumably the resources assigned each flow, can be merged, thus decreasing the state space and protocol management overhead to acceptable levels. It would further reduce the overhead if the number of spanning trees could be reduced, possibly to one or a few for each region, as suggested above. Nevertheless, in order to accurately determine the flow parameters for each merged flow, it will be necessary somewhere, perhaps only at the edges of the service area, to account for each flow separately.

Current IETF engineering principles are to avoid centralized algorithms in favor of distributed ones in the interest of robustness and survivability. Favored algorithms are those that can recover state following destruction due to a crash or attack by exploiting redundancy implicit in their neighbors. This is a natural model for such things as routing algorithms, time synchronization and similar network infrastructure services, since the state associated with these services exists independently of the applications that use the network and any flows they might instantiate. The paradigm can be extended to multicast routing with dynamic groups on the supposition that group membership is implemented something like network membership - a new network (or group) never heard before is dynamically integrated in the routing fabric simply by announcing its presence.

Of considerable concern in the current IETF model, which includes databases for the unicast routing fabric, multicast routing fabric, service flows and resource reservation, is the volume of state space required, especially since some (maybe a considerable) fraction of this space must be replicated (or re-derivable) in possibly many places. Each of the service layers replicates in part a database maintained by a lower layer - DVMRP replicates much of RIP, RSVP rediscovers reverse paths, and so forth. The conclusion drawn from these observations is that the design of these layers should be approached as a functional unit with a common database and database convergence protocol.

While the database structure required for flow installation and management is not completely clear at this time, there is every suspicion that it will replicate at least some lower layer functionality. In addition, guaranteed-delay service requires an interaction with every router on the path; therefore, a resource reservation change may involve a number of adjustments at merge points in the multicast fabric. As the size of the network grows and the number of groups grows, significant scaling problems are created. It does not seem possible to deal with these problems, unless some degree of systematic aggregation is engineered into the fabric in the form of shared trees or, equivalently, hierarchical routing as described above.

7. Research Plan

Our research plan follows the model used by many network researchers, including ourselves in previous projects, of analysis, design, simulation, implementation and evaluation. The analysis phase will involve a systematic literature search to find and evaluate specific ideas that may be of use in later phases of the work. The design phase will consider candidate designs and select promising ones for further evaluation. The simulation phase provides a proof-of-concept and a vehicle for investigating various scenarios which may occur in practice. The implementation and evaluation phases provide insight in the actual performance of the design in a real-world environment

and a suitable vehicle for technology transfer. In particular we propose to investigate the following:

1. Conduct a critical study of the IETF model with particular emphasis on scalability, efficiency and processor/memory economy.

2. Develop efficient, scalable routing algorithms for a native multicast infrastructure with an automatic clustering capability as a function of resource availability and application requests.

3. Develop algorithms for multiple-source-multiple-destination communication patterns that exist in co-operative networking environments such as whiteboards, teleconferencing, etc., employing shared trees and localized information updates for individual clusters.

4. Develop algorithms to compute the routing trees in an automatic and distributed fashion with specified global constraints such as aggregate bandwidth, maximum fan-in and fan-out degree, and maximum path distance.

5. Simulate the proposed techniques in a realistic fashion as a proof of concept and make the simulation tools available for the experimental network community.

6. Implement a suitable prototype algorithm and protocol for testing in the DARTnet/CAIRN infrastructure. Design and conduct experiments designed to verify the basic concepts and scalability constraints.

We plan to use our ongoing work in other areas, such as progressive video reconstruction in a teleconferencing environment, as a testbed for the multicasting algorithms. We plan also to use existing traffic generators and measurement packages designed by DARTnet/CAIRN collaborators as well.

8. Research Facilities

A particular strength of the proposed UD effort is the local network configuration and interconnection with other research networks. The UD research network, including about two dozen workstations, routers and servers, is distinct from the campus network and is reserved for research only. It is connected to DARTnet/CAIRN by a router and T1 tail circuit that does not involve the public Internet infrastructure. Thus, experiments in potentially disruptive routing algorithms can proceed without fear that accidents will not disturb public Internet operations.

Of vital importance to the success of the proposed work is provisioning for DARTnet access by all participants in the collaboration. Since the proposed approach involves invasive surgery of the basic network routing functions, this work cannot be safely conducted in an operational network. According to the memoranda of understandings exchanged among the participants, in order to support this work, SAIC will provision their own high speed router and access line to a CAIRN PoP at no cost to the Government. With the use of other funds, UCL is to be connected to DARTnet as well. UD is already connected to DARTnet, which is itself connected to CAIRN, so no additional provisioning is required.

9. References and Bibliography

1. Ammar, H., G. Polyzos, and S. Tripathi. Guest editorial. *IEEE J. Selected Areas in Communications 15, 3* (1997).

2. Ballardie, T., P. Francis and J. Crowcroft. Core based trees (CBT): an architecture for scalable inter-domain multicast routing. *Proc. SIGCOMM 93 (September 1993, San Francisco)*.

3. Baqai, S., F. Khan, A. Ghafoor, A. Khokhar. Quality-based evaluation of multimedia synchronization protocols for distributed multimedia information systems. *IEEE J. Selected Areas in Communications 14, 7* (1996), 188-1403.

4. Bauer, F., and A. Varma. Degree-constrained multicasting in point-to-point networks. *Proc. IEEE INFOCOM 95* (April 1995, Boston).

5. Berry, L. Graph theoretic models for multicast communication. *Computer Networks and ISDN Systems 20* (1990), 95-99.

6. Birman, K., A. Schiper and P. Stephenson. Lightweight causal and atomic group multicast, *ACM Trans. Computer Systems 9, 3* (August 1991), 272-314.

7. Birman, K.P., R. Cooper and B. Gleeson. Design alternatives for process group membership and multicast. Research Report TR91-1257. Cornell University, 1991.

8. Cimet, I., and S. Kumar. A resilient algorithm for minimum weight spanning trees. *Proc. International Conference on Parallel Processing* (August 1987, St. Charles), 196-203.

9. Dabbous, W., and B. Kiss. A reliable multicast protocol for whiteboard application. INRIA Research Report 2100, Sophia Antipolis, November 1993.

10. Dalal, Y.K., and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Comm. ACM 21, 12* (1978).

11. Diot, C., W. Dabbous and J. Crowcroft. Multicast communication: a survey of protocols, functions, and mechanisms. *IEEE J. Selected Areas in Communications 15, 3* (1997).

12. Doar, M., and I. Leslie. How bad is naive multicast routing?. *Proc. IEEE INFOCOM 93* (San Francisco), 82-89.

13. Gong, L., and N. Shacham. Elements of trusted multicasting. *Proc. International Conference on Network Protocols (ICNP 94)*, (October 1994, Boston).

14. Grudin, J. Computer-supported cooperative work: history and focus. *IEEE Computer 27, 5* (1994).

15. Hambrusch, S., A. Khokhar, and F. Hameed. Communication Operations on Coarse-grained Mesh Architectures. *Parallel Computing 21* (1995), 731-751.

16. Hambrusch, S., A. Khokhar. C3: An architecture independent model for coarse-grained parallel machines. *J. Parallel and Distributed Computing. 32* (1996), 139-154.

17. Hambrusch, S., A. Khokhar, Y. Liu. S-to-P braodcasting on coarse-grained parallel machines. Proc. International Conference on Parallel Processing, 1996, 69-76.[Diot97].

18. Haverty, J., B. Hitson, J. Mayersohn, P. Sevchik, and G. Williams. ARPANET routing algorithm improvements. Report 4931, Bolt Beranek and Newman, Inc., March 1982.

19. Herzog, S., S. Shenker and D. Estrin. Sharing the "cost" of multicast trees: an axiomatic analysis. *Proc. ACM SIGCOMM 95* (September 1995).

20. Hwang, F.K., and D. S. Richards. Steiner tree problems. *Networks 22, 1* (January 1992) 55-89.

21. Jamieson, L., S. Hambrusch, E. Delp, and A. Khokhar. The role of models, software tools, and applications in high performance computing. In: U. Vishkin (ed). *Developing Computer Science Agenda for High Performance Computing*. ACM Press, 1994, 90-97.

22. Khokhar, A., C-L. Wang, M. Shaaban, and V. Prasanna. Heterogeneous computing: challenges and opportunities. *IEEE Computer Magazine, Special Issue on Heterogeneous Processing 26, 2* (1993), 18-27.

23. Law, K.L.E., and A. Leon-Garcia. Multicast and self-routing ATM radix trees and banyan networks. *Proc. IEEE INFOCOM 95* (April 1995, Boston).

24. Mills, D.L. Exterior gateway protocol formal specification. Network Working Group Report RFC-904, M/A-COM Linkabit, April 1984.

25. Mills, D.L. Internet time synchronization: the Network Time Protocol. I*EEE Trans. Communications COM-39, 10* (October 1991), 1482-1493. Also in: Yang, Z., and T.A. Marsland (Eds.). *Global States and Time in Distributed Systems*. IEEE Computer Society Press, Los Alamitos, CA, 1994, 91-102.

26. Mills, D.L., and A. Thyagarajan. Network Time Protocol Version 4 proposed changes. Electrical Engeneering Report 94-10-2, University of Delaware, October 1994, 24 pp.

27. Mills, D.L. Wiretap: an experimental multiple-path routing algorithm. *ACM Computer Communication Review 19, 1* (January 1989), 85-98.

28. Noronha, C.A., F.A. Tobagi. Optimum routing of multicast streams. *Proc. IEEE INFOCOM 94* (June 1994, Toronto).

29. Rajagopalan, B., Reliability and scaling issues in multicast communication. *Proc. ACM SIGCOMM 92*, 188-198.

30. Takahashi, H., and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Math. Japonica 6* (1980), 573-577.

31. Waitzman, D., C. Partridge, S. Deering, Distance vector multicast routing protocol. Thechnical Report RFC 1075, Internet Engineering Task Force, November 1988.

32. Waxman, B. Routing of multicast connections. *IEEE J. Selected Areas in Communications 6, 9* (December 1988), 1617-1622.

33. Winter, P. Steiner problem in Networks: a Survey. *Networks 17, 2* (1987), 129-167.

34. Zhu, Q., M. Parsa, and J. Garcia-Luna-Aceves. A source-based algorithm for delay-constrained minimum-cost multicasting. *Proc. IEEE INFOCOM 95* (April 1995, Boston).

Guang R. Gao, PhD
Professor of Electrical Engineering

Brief Biography

Guang R. Gao has received SM and Ph.D Degree in Electrical Engineering and Computer Science Massachusetts Institutes of Technology, in June 1982 and August 1986. In 1987, he became an Assistant Professor, School of Computer Science, McGill University, Canada. In June 1992, he was promoted to associate professor with tenure. In Sept. 1996, he joined the Department of Electrical Engineering of University of Delaware as an associate professor. His research interests include computer systems: architectures, compilers and networking for high-performance computing.

Awards and Honors

Senior Member of IEEE 1996, NSERC Senior Industrial Fellowship 1993-1994

Five Most Relevant Publications

Experience with Non-numeric Applications on Multithreaded Architectures. Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (To Appear), June, 1997. (with Angela Sodan, Olivier Maquelin, Jens-Uwe Schultz and Xin-Min Tian)

Thread Partition and Schedule Based on Cost Model. Proceedings of the ACM International Symposium on Parallel Algorithms and Architectures (SPAA97, to appear), June, 1997. (with X. N. Tang and J. Wang and K. Theobald)

A Study of the EARTH-MANNA Multithreaded System. International Journal of Parallel Programming, 24(4):319-347, August 1996. (with Herbert H. Hum, Olivier Maquelin, Kevin Theobald, Xinmin Tian, and Laurie J. Hendren

Polling Watchdog: Combining Polling and Interrupt for Efficient

Costs and Benefits of Multithreading with Off-The-Shell RISC Processors. Proceedings of the First International EURO-PAR Conference, pages 117-128, Stockholm, Sweden, August, 1995, Springer-Verlag. (with Olivier Maquelin and Herbert H. Hum)

Five Other Significant Publications

Latency Tolerance: A Metric for Performance Analysis of Multithreaded Architecture. Proceedings of the International Parallel Processing Symposium (IPPS-97, to appear), April, 1997. (with Shashank S. Nemawarkar)

Compiling C for the EARTH Multithreaded Architecture. Proceedings of the International Conference on Parallel Architecture and Compilation Techniques (PACT96), pp 12-23, Oct. 1996. (with Laurie J. Hendren, Xinan Tang, Yingchun Zhu, Xun Xue, Haiying Cai and Pierre Ouelle)

Data Locality Analysis for Distributed Shared-memory Multiprocessors. Proceedings of the Ninth Workshop on Languages and Compilers for Parallel Computing, Aug. 1986. (with Vivek Sarkar and Shaohua Han)

A Framework for Resource-Constrained Rate-Optimal Software Pipelining. IEEE Transaction on Parallel and Distributed Systems, pp 1133-1149, Vol 7, No. 11, Nov. 1986. (with R. Govindarajan and Erik R. Altman)

A New Framework for Exhaustive and Incremental Data Flow Analysis. Using {DJ} Graphs , ACM SIGPLAN '96 Conference on Programming Language Design and Implementation (PLDI), June 1996. (with Vugranam C. Sreedhar and Yong-fong Lee)

Advisor And Advisees

Gao's advisor was J. Dennis (MIT). Gao has graduated seven doctoral students: Robert Yates (LLNL), Herbert Hum (Intel), Ning Qi (Convex), Guy Tremblay (U. of Quebec at Montreal), Vugranam C Sreedhar (HP), Erik Altman (IBM), Shashank Nemawarkar (IBM).

NSF Research Grant Program for

Hybrid Technology Multithread Architectures, 1996-1997. This is a point-design study of peta-flops computers. (The NSF Grant No. is ASC-9612105, I am a co-investigator in collaborating with the PI's: Paul Messina and Tom Sterling at Caltech/JPL; another co-investigator is Prof. K. Likharev of SUNY Stony Brook.)

Sponsor: NASA (through JPL), Grant title: ''Hybrid Technology Architecture Evaluation and Hybrid Technology Multi-Threaded Architecture.'', This is a matching to the above NSF grant. I am a co-investigator with PIs and co-investigator listed above. Total amount: 35,000.

Sponsor: NSA (through JPL), Grant title: ''Hybrid Technology Multithreaded Architecture Project'', 1997-1999, I am a co-investigator, with a team of investigators from other institutions. (It has been selected for funding. Jointly from the DARPA program below, we at Delaware expect to receive: total amount 800,000 approx., with 400,000 per year.)

Sponsor: DARPA (through JPL), Title: "A Hybrid Technology Multithreaded Computer Architecture for Petaflops Computing". In respond to DARPA research program BAA-97-03, submitted Dec., 1996. I am a co-investigator, with a team of investigators from other institutions. (It has been selected for funding. Jointly from the NSA program above, we at Delaware expect to receive: total amount 800,000 approx., with 400,000 per year.)

Principal Investigator or Co-Investigator for a number of NSERC Research Grant Programs, awarded between 1988--1996. (detailed list omitted)

Recipient of industrial research grants from BNR, IBM T.J. Watson Research Center, IBM Canada and others.

Ashfaq A. Khokhar, PhD
Assistant Provessor of Electrical Engineering

Education

1993 Ph.D. in Computer Engineering, University of Southern California. Advisor: Viktor K. Prasanna
1988 M.S. in Computer Engineering, Syracuse University.
1985 B.Sc. in Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan.

Professional Experience

1995-present. Assistant Professor, Department of Electrical Engineering and Department of Computer and Information Sciences, University of Delaware.

1993-1995 Visiting Assistant Professor, School of Electrical and Computer Engineering, and Department of Computer Sciences, Purdue University.

1989-1993 Unix Consultant, University Computing Services, University of Southern California.

Research Interests

Network Computing, Parallel and Distributed Software Systems, Design and Analysis of Algorithms, Computation Geometry and Graph Algorithms, Dis- tributed Multimedia Systems, and Heterogeneous Computing.

Five Most Relevant Publications

“Quality-Based Evaluation of Multimedia Synchronization Protocols for Distributed Multimedia Information Systems,” with Baqai et. al., IEEE Journal on Selected Areas in Communications, Vol. 14, No. 7, pp. 188-1403, 1996.

“S-to-P Braodcasting on Coarse-grained Parallel Machines,” with Susanne Hambrusch and Yi Liu, Proceedings of International Conference on Parallel Processing, pp. 69-76, 1996. (Outstanding Paper Award)

“C3: An Architecture Independent Model for Coarse-grained Parallel Machines,” with Susanne Hambrusch, Journal of Parallel and Distributed Computing, Vol. 32, pp. 139-154, 1996.

“Communication Operations on Coarse-grained Mesh Architectures,” with Susanne Hambrusch and Farooq Hameed, Parallel Computing, Vol. 21, pp. 731-751, 1995.

“Issues in Storage and Retrieval of Multimedia Data,” with M. Farrukh Khan and Arif Ghafoor, ACM Multimedia Systems, Vol. 3, pp. 298-304, 1995.

Five Other Significant Publications

“The Role of Models, Software Tools, and Applications in High Performance Computing,” with Leah Jamieson, Susanne Hambrusch, and Edward Delp, Developing Computer Science Agenda for High Performance Computing, edited by Uzi Vishkin, ACM press, pp. 90-97, 1994.

“Application of Heterogeneous Computing in Multimedia Database Management Systems,” with Arif Ghafoor in Heterogeneous Computing, edited by Mary Eshaghian, Artech House, Inc. 1995, pp. 335-354.

“Scalable Data Parallel Implementations of Object Recognition using Geometric Hashing,” with Hyoung Joong Kim, Viktor Prasanna, and Cho-Li Wang, Journal of Parallel and Distributed Computing, Special Issue on Data Parallel Algorithms and Programming), Vol. 21, pp. 96-109, 1994.

“Contour Ranking on Coarse-grained Machines: A Case Study for Low-level Vision Computations,” with Farooq Hameed, Susanne Hambrusch, and Jamshed Patel, Concurrency: Practice and Experience, accepted to appear.

“Scalable Parallel Implementations of List Ranking on Fine-grained Machines,” with Jamshed Patel and Leah Jamieson, IEEE Transactions onParallel and Distributed Systems, accepted to appear.

Research Grants/proposals:

“Coarse-grained Data Structures for Spatial Data Sets,” Principal Investigaotr, submitted to Delaware Research Partnership Program, Potential industrial collaboraters: Siemens Research Corporate Center, amount $200,000, under review.

“High Perfromance Infrastructure at University of Delaware,” Affiliated Investigator, submitted to National Science Foundation, CISE Infrastructure Program, $856,000, under review.

Human Resources

PhD. Students Supervised: Lan Chen, “Coarse-grained Spatial Data Structures,” in progress.

Undergraduate Research Supervised: Michel Becht, University of Delaware, Fall 1996-present

Selected Professional Activities

Program Committees: IEEE Workshop on Heterogenous Computing (1994), SPIE Symposium on Information, Communication and Computer Technology (1995), 10th Conference on High Performance Computers (SUPER*CAN HPSC) (1996), Tutorials Chair, International Parallel Processing Symposium, 1995.

Reviewer for: National Science Foundation, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence Image and Vision Computing Journal Journal of Parallel and Distributed Computing Parallel Processing Letters

Member ACM and IEEE Computer Society

List Of Collaborators

Song Chen, Edward Delp, Mary Eshaghian, Arif Ghafoor, Susanne Hambrusch, Leah Jamieson, Jamshed Patel, Viktor Prasanna, Mohammad Shaaban.

David L. Mills, PhD
Professor of Electrical Engineering

Education

BSE Engineering Science, University of Michigan, 1960
BSE Engineering Mathematics, University of Michigan, 1961
MSE Electrical Engineering, University of Michigan, 1962
MS Communication Sciences, University of Michigan, 1964
PhD Computer and Communication Sciences, University of Michigan, 1971

Professional Responsibilities

Dr. Mills leads projects in high speed networks and internetworking research sponsored by the DARPA, NSF, US Navy and US Army. He is principal investigator for several projects in computer network time synchronization, gigabit-speed networks and network security. His research activities are concentrated in the areas of network architecture, protocol engineering and experimental studies involving the Internet system.

He has for many years been an active contributor to the field of computer network time synchronization. Protocols he developed, prototyped and deployed have evolved to the Network Time Protocol (NTP), which is now in widespread use by agencies of the US, including National Institute of Science and Technology (NIST) and US Naval Observatory (USNO), as well as in many other countries of the world. One of his current interests is extending the accuracy, stability and robustness of this technology to the submillisecond regime on desktop workstations. Another is cryptographically secure authentication for very large networks of distributed servers and clients.

He is a member of the Internet Research Steering Group (IRSG) and End-to-End (E2E) Research Group, which coordinate advanced research on the Internet system. He was chair of the Internet Architecture (INARC) and Gateway Algorithms and Data Structures (GADS) task forces, and member of the Internet Activities Board (IAB) and its predecessor Internet Control and Configuration Board (ICCB). He served on the NAS/NRC Committee on Survivable Telecommunications Networks, AT&T Network Architecture Advisory Panel, and NSF Network Technical Advisory Group (NTAG), as well as several other committees involved with national network infrastructure and the establishment of the National Research and Education Network (NREN) and the High Performance Computer and Communications Initiative (HPCCI).

Dr. Mills was the advisor and principal architect for the NSFnet Phase I backbone network, which interconnected six supercomputer sites during the period 1986-1988 and later evolved into the present Internet national backbone. Software and network routing protocols he developed for use in the DARPA Internet research program and commonly called the Fuzzball was used in the packet routers and gateways of that system and survives to the present.

Human Resources

Graduate students in the last five years: Qoing Li, Ajit Thyagarajan, Bradley Cain, Erik Perkins, Kenneth Monington, Brian Huffman, Donald Nelson

Undergraduate research and honor students in the last five years: Marie Conte, Steven Bijanski, Tyrone Thompson

New courses developed and taught: CPEG 419 and ELEG 651 Computer Communications and Networks, ELEG 867 Seminar on Cryptographic Techniques with Computer Network Applications, ELEG 867 Seminar on Computer Security

Selected publications

Mills, D.L. Authentication scheme for distributed, ubiquitous, real-time protocols. Proc. Advanced Telecommunications/Information Distribution Research Program (ATIRP) Conference (College Park MD, January 1997), 293-298.

Mills, D.L. The network computer as precision timekeeper. Proc. Precision Time and Time Interval (PTTI) Symposium (Reston VA, December 1996), to appear.

Mills, D.L. Proposed authentication enhancements for the Network Time Protocol version 4. Electrical Engineering Report 96-10-3, University of Delaware, October 1996, 36 pp.

Mills, D.L. Improved algorithms for synchronizing computer network clocks. IEEE/ACM Trans. Networks (June 1995), 245-254.

Mills, D.L. Precision synchronization of computer network clocks. ACM Computer Communication Review 24, 2 (April 1994). 28-43.

List Of Collaborators for the Last Five YearsU

J. Crowcroft and A. Bellardie (University College London, UK), J. Levine (NIST), R. Schmidt (USNO), C. Boncelet and J. Elias (University of Delaware)

Current and Prior Support for the Last Five Years

Co-Principal Investigator: “ARL Federal Research Laboratories Program,” University of Delaware, Electrical Engineering Department, for ARL Cooperative Agreement DAA L01-96-2-002 (January 1996 - September 2001, $3.5M) (with collaborators at other institutions).

Principal Investigator: “Scalable, High Speed, Internet Time Synchronization,” University of Delaware, Electrical Engineering Department, for DARPA Information Technology Office Contract DABT 63-95-C-0046 (June 1995 - June 1998, $370K).

Principal Investigator: “Advances in Computer Network Timekeeping,” University of Delaware, Electrical Engineering Department, for NSF Division of Network and Communications Research and Infrastructure Grant NCR-93-01002 (June 1993 - August 1997, $177K).

Principal Investigator: “SAFENET Time Management,” University of Delaware, Electrical Engineering Department, for Northeastern Center for Electrical Engineering Education Contract A3036-92 6192 43093 (July 1993 - March 1997, $113K).

Principal Investigator: “Performance and Policy Dimensions in Internet Routing,” University of Delaware, Electrical Engineering Department, for DARPA Information System Technology Office Contract NAG-2-638 (February 1990 - August 1994, $849K).