A Multicast Routing Algorithm for LEO Satellite IP Networks

Eylem Ekici, *Student Member, IEEE*, Ian F. Akyildiz, *Fellow, IEEE*, and Michael D. Bender, *Senior Member, IEEE*

***Abstract*—Satellite networks provide global coverage and support a wide range of services. Since *Low Earth Orbit (LEO)* satellites provide short round-trip delays, they are becoming increasingly important for real-time applications such as voice and video traffic. Many applications require a mechanism to deliver information to multiple recipients. In this paper, a multicast routing algorithm for datagram traffic is introduced for LEO satellite IP networks. The new scheme creates multicast trees by using the Datagram Routing Algorithm. The bandwidth utilization and delay characteristics are assessed through simulations.**

***Index Terms*—Low Earth Orbit (LEO), multicast routing, satellite networks.**

I. Introduction

SATELLITE networks are becoming important for worldwide communication [1]. They not only provide global coverage, but they are also capable of consistently sustaining high bandwidth levels. Moreover, they support flexible network configurations. Currently, two thirds of the world still does not have a wired network infrastructure. Locally built networks or individual hosts can be connected to the rest of the world via satellites by simply installing satellite interfaces. Satellite networks can also be used as a backup for the existing networks. In case of congestion or link failures, traffic can be routed through satellites.

The connection-oriented routing has been the focus of the routing research for LEO satellite networks. The existing connection-oriented routing protocols assume ATM-like switches in the satellites. The heuristic routing algorithm proposed in [2], [3] aims to reduce the number of path handovers due to the mobility of satellites. The algorithm presented in [4] uses the snapshots of the constellation to optimize the paths. In [5], a two-layered satellite network architecture consisting of LEO and MEO satellite networks and a routing algorithm are proposed. A QoS-based satellite network is described in [6], which includes a routing scheme that resembles minimum hop routing in Manhattan Street Networks [7]. The probabilistic routing protocol (PRP) introduced in [8] aims to maintain the initial paths as long as possible in order to minimize the signaling overhead.

However, as the Internet is becoming very popular and the efforts regarding the *Next Generation Internet* (NGI) are on the way, there is an initiative in the commercial and also in the military world to use IP routing technology also in satellite networks. In the literature, there are only a few attempts to address the connectionless routing problem in satellite networks. The so-called "Darting" algorithm delays the exchange of topology update information until it is necessary to send data packets [9]. However, it is shown in [10] that the Darting algorithm does not reduce the protocol overhead. The Datagram Routing Algorithm [11] aims to route the packets on minimum propagation delay paths. The routing protocol presented in [12] uses a hybrid approach that uses geographic-based routing and shortest path routing with limited scope.

The IP-based LEO satellite networks can provide lower delays to multicast applications such as tele-education and IP-based teleconferencing at global scale. The multicast routing problem in terrestrial datagram networks has already been studied extensively in the past [13]. However, none of the existing multicast routing protocols are well-suited for LEO satellite networks. Reverse-path multicast (RPM) [14], distance vector multicast routing protocol (DVMRP) [15], and the multicast routing extensions for OSPF (MOSPF) [16] cannot be used because they employ some form of periodic message exchange to form or maintain the multicast trees, which is not favorable due to the limited processing power and power supplies of the satellites. The core-based tree (CBT) [17] concentrates the traffic at the core of the tree by requiring all multicast packets to be sent to the core. Finally, the protocol independent multicast (PIM) [18] switches between a CBT (sparse mode) and the shortest path tree (dense mode). The PIM protocol requires the monitoring of individual flow rates to trigger the switching from CBT to the shortest path tree, which is an additional burden for the satellites. To our knowledge, there is no multicast routing protocol so far specifically designed for satellite networks.

By using the *Datagram Routing Algorithm* [11], our new multicast protocol creates multicast trees in the LEO satellite constellation rooted at the source of each multicast group. The multicast tree is created such that the multicast packet replication is minimized in each satellite. Unless the multicast group membership changes, no tree maintenance is required. Using the *logical location* concept [11], our new multicast protocol preserves the initial tree structure despite the satellite mobility.

This paper is organized as follows. In Section II, we describe the satellite constellation. The new multicast protocol is pre-

Manuscript received November 3, 2000; revised May 1, 2001 and September 23, 2001; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Orda. This work was supported in part by the Department of Defense under Grant MDA904-98-C-A839, the National Science Foundation under Grant ANI-0087762, and NASA Glenn Research Center under Grant NAG3-2580.

E. Ekici and I. F. Akyildiz are with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: eylem@ece.gatech.edu; ian@ece.gatech.edu).

M. D. Bender is with the Department of Defense, Fort Meade, MD 20755 USA (e-mail: mdbende@afterlife.ncsc.mil).

Publisher Item Identifier S 1063-6692(02)03109-6.

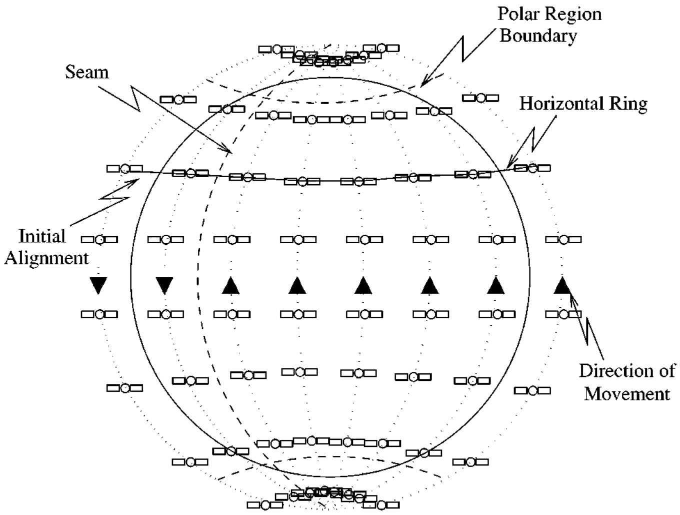


Fig. 1. Orbital planes around the Earth.

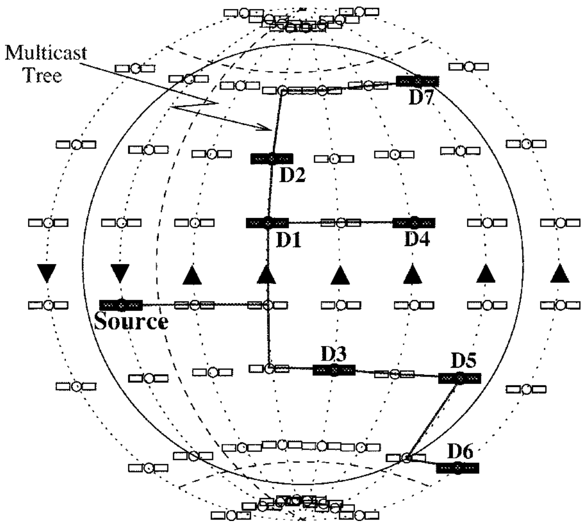


Fig. 2. Example multicast group and multicast tree.

sented in detail in Section III. Section IV is dedicated to performance evaluation of the new multicast protocol. Finally, Section V concludes the paper.

II. Satellite Constellation Model

The satellite network is composed of N separate orbits (*planes*), each with M satellites at low distances from the Earth as shown in Fig. 1. The planes are separated from each other with the same angular distance of $360^\circ/(2 \times N)$. They cross each other only over the North and South poles. The satellites in a plane are separated from each other with an angular distance of $360^\circ/M$. Since the planes are circular, the radii of the satellites in the same plane are the same at all times and so are the distances from each other. This satellite constellation can be classified as Walker Star type [19].

The *geographical location* of a satellite S is given by $[\text{lon}_S, \text{lat}_S]$ indicating the *longitude* and *latitude* of the location of S, respectively. We assume that the satellite constellation is divided into *logical locations* [11], which are equally spaced holes in the spherical grid of the LEO satellite constellation and are filled by the nearest satellites. A similar static location concept is proposed in [6]. Hence, the identity S of the satellite is not permanently coupled with its logical location, which is taken over by the successor satellite in the same plane. The logical location of a satellite S is given by $\langle p, s\rangle$ where p for $p = 0, \ldots, N-1$, is the plane number and s, for $s = 0, \ldots, M-1$, is the satellite number. The routing is performed considering these logical locations as hops. By this way, we do not need to be concerned with the satellite movements. Any routing tables used by the routing or multicasting protocol are associated with the logical locations rather than the individual satellites. Therefore, each time a satellite moves and fills another logical location, its routing tables must be updated. A satellite leaving a logical location transfers its routing tables to its successor and receives the new routing tables from its predecessor. In this work, we assume that the logical locations are embodied by the satellites assigned to them.

Each satellite has four neighboring satellites: two in the same plane and two in the left and right planes. The links between satellites in the same plane are called *intra-plane ISLs*. The links between satellites in different planes are called *inter-plane ISLs*. On intra- and inter-plane ISLs, the communication is bidirectional.

The intra-plane ISLs are maintained at all times, i.e., each satellite is always connected to the rest of the network through its *up* and *down* neighbors. The propagation delay on the intra-plane links is always fixed. All satellites are moving in the same circular direction within the same plane. As a consequence, any satellite that is observed from the Earth moving from South to North will be observed to start moving from North to South when it crosses the North pole. Hence, the 0th and Nth planes rotate in opposite directions. The borders of counter-rotating satellites are called *seams* as shown in Fig. 1.

The inter-plane ISLs are operated only outside the polar regions. When the satellites move toward the polar regions, the inter-plane ISLs become shorter. When two satellites in adjacent planes cross the poles, they switch their positions. In order to allow this switching, the inter-plane ISLs are shut down in polar regions and re-established outside of the polar regions.

III. The New Multicasting Protocol

Our new multicast routing protocol creates and maintains multicast trees that span the multicast group members for each multicast session. The multicast groups consist of logical locations (i.e., satellites closest to these locations) where the multicast packets need to be sent. In Fig. 2, a multicast tree with seven multicast group members spanning ($D1, \ldots, D7$) is shown. The term "group membership" involves the logical locations. Each member logical location (or satellite) may actually be associated with multiple recipients on the Earth. For a satellite to leave the multicast group, all actual recipients on Earth associated with that satellite must leave the multicast session. We assume that a satellite becomes a member even if there is only one recipient in its coverage area. In the following subsections, we present the underlying unicast protocol, the addressing mode, creation and modification of multicast trees, and the actions taken in case of congestions and satellite failures.

A. The Underlying Unicast Protocol

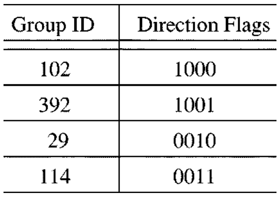
The structure of LEO satellite networks resembles the *Manhattan Street Networks*, which have been investigated extensively in the last decade [7]. However, there are very basic distinctions between these two networks. First of all, LEO satellite networks have a twisted circular connection structure. The ISLs across the seam connect satellites belonging to counter-rotating *planes,* which would correspond to point-symmetric nodes of the first and last *rows* with respect to the center of the connection grid in a circular Manhattan Street Network. Secondly, unlike in the Manhattan Street Networks, the links connecting nodes, i.e., satellites, have different weights in the LEO satellite networks, which are the propagation delays of the ISLs. Inter-plane ISLs are longest over the equator and become shorter as the polar regions are approached. These properties are considered in the design of the Datagram Routing Algorithm [11]. The Datagram Routing Algorithm aims to forward packets on minimum propagation delay paths between source–destination pairs and guarantees that the propagation delay experienced by a packet is smaller than or equal to the propagation delay on the *longest minimum hop path* between the same source–destination pair.

Our new multicast scheme is based on the Datagram Routing Algorithm [11]. As discussed in Section II, the logical locations of the satellites are regarded as holes in a spherical grid, filled by the nearest satellite. The routing is performed between the ideal logical locations, i.e., the holes of the grid. The mobility of satellites is captured by the logical location concept. For each packet, each satellite calculates the next hop independent of the previous hops. The routing algorithm ensures that:

- packets follow the minimum propagation delay route between the ideal locations of the source and destination;
- congested regions are avoided;
- resulting path is loop-free.

The Datagram Routing Algorithm [11] processes every incoming packet independently. Each satellite computes the next hop for each packet they receive. The next hop on the path is determined in three phases.

1) *Direction Estimation Phase:* In this phase, the directions of possible next hops are determined assuming that all ISLs have equal lengths. Under this assumption, the minimum hop paths are also minimum propagation delay paths. This phase calculates for each packet the *minimum hop metrics,* which consists of one or two tuples in the form of (direction, hop number). The minimum hop metrics show how many horizontal and vertical hops should be taken to reach the destination on a minimum hop path. This information is used as input in the next phase.
2) *Direction Enhancement Phase:* Since the lengths of ISLs are different in satellite networks as shown in Fig. 1, we have the direction enhancement phase, where we consider that the inter-plane ISLs have different lengths and refine our decision about the next hop accordingly. If the minimum hop metrics consists of only one tuple, i.e., if either no horizontal or no vertical hops are needed, it is possible that the enhancement phase finds another direction with higher priority that is a part of the minimum propagation delay path. If the minimum hop metrics consist of two tuples, then the enhancement phase chooses one of these two directions as the primary direction. As a result, the direction calculated in the second phase lies on the minimum propagation delay path.
3) *Congestion Avoidance Phase:* The decision made in the previous phase is revised in the congestion avoidance phase to avoid overly congested links and to route packets around failed satellites. In the Datagram Routing Algorithm, satellites do not exchange delay information. Therefore, when link congestions are discovered, the satellite that discovers the congestion forwards the packets on alternative links. When the number of packets in an outgoing buffer exceeds a threshold ξ, it indicates that there is a congestion on that particular link. If the link in the primary direction of a packet is congested, then the packet is forwarded on the secondary direction if available. Similarly, in case of satellite failures, the packets that go through the failing satellite are sent in their secondary direction. If the secondary direction is not available, then the packet is still placed in the primary direction in case of a congestion. In case of a satellite failure, if the packet does not have a secondary direction, then the packet is forwarded in a direction perpendicular to the primary one, from which it was not received.



Group ID	Direction Flags
102	1000
392	1001
29	0010
114	0011

Fig. 3. Example multicast routing table.

The details of the Datagram Routing Algorithm can be found in [11].

B. Addressing Mode

Our new multicast scheme creates source-based multicast trees for each multicast session. Each multicast session is assigned a unique *multicast session ID*. Once the tree is setup, the multicast packets are routed according to the multicast session ID in their headers. Multicast routing tables in satellites contain entries consisting of a multicast session ID and a *direction flag*. The direction flag shows in which direction a multicast packet must be forwarded. Fig. 3 shows an example multicast routing table in a satellite.

This routing table belongs to a satellite through which four multicast trees pass. Each digit in the direction flags corresponds to a direction. If a digit is 1, then the multicast packets are forwarded in that direction. No multicast packets are sent in directions having a 0 in the direction flags. For the multicast sessions 102 and 29, there is only one direction the packets must be sent to. For the other two multicast sessions, 392 and 114, there are two directions, hence this satellite is a *branching point* in the respective multicast trees.

C. Creation of Multicast Trees

Multicast trees are created using *tree-setup packets* $P = (\alpha, \beta)$, where α is the session ID and β is a partial list of destination satellites. First, the source satellite creates a tree-setup packet for its own use with a complete list of destination satellites. Like any other satellite that receives a tree-setup packet, the source satellite uses the Datagram Routing Algorithm [11] to determine the high- and low-priority directions for each destination as outlined in Section III-A. Following this, subgroups within the partial list β are created. The key point in subgroup creation is to keep the number of subgroups as small as possible. There is a one-to-one correspondence between the subgroups and the directions. The subgroup creation is accomplished as follows.

1) Using the Datagram Routing Algorithm [11], the high- and low-priority directions are determined for each destination in β. Note that the congested links and the links not available due to satellite failures are discarded at this stage for each destination.
2) A bin is created for each direction and destination satellites are assigned to these bins according to the directions calculated in Step 1. Each bin contains a candidate subgroup.
3) A minimum number (L) of bins are selected such that all destination satellites are contained in the selected bins. If there are multiple combinations with equal number of bins, the combination that contains the greatest number of priority directions is selected. Note that L can vary between 1 and 4.
4) If a destination is included in two selected bins, then it is removed from the bin of the low-priority direction. After this step, selected bins contain partial lists of destinations $\beta_i, 1 \leq i \leq L \leq 4$, such that $\bigcup_{i=1}^{L} \beta_i = \beta$ and $\beta_i \cap \beta_j = \{\}, i \neq j$.
5) A tree-setup packet P_i, $1 \leq i \leq L \leq 4$, is created for each selected bin such that $P_i = (\alpha, \beta_i)$.

Then the routing table is updated according to the directions of the formed subgroups. The tree-setup packets for each subgroup are sent in corresponding directions. The satellites receiving these packets perform the same operations as described in Steps 1–5.

The termination of the multicast session is accomplished by deletion of the multicast tree. A *tear-down message* is sent on the branches of the tree. Satellites receiving this message delete the entry for that multicast session from their routing tables and forward the tear-down message in the tree. If the satellites along the path are also destination satellites, then they send a message to the recipients on Earth, informing them about the termination of the multicast session.

Example: Let us consider a multicast session with the source satellite S and multicast group of destinations $\{1, 2, 3, 4, 5\}$. S starts to build up the multicast tree by creating a tree-setup packet $P = (\alpha, \{1, 2, 3, 4, 5\})$ and processing it. Fig. 4 shows subgroups created in the source satellite.

Group Member	Primary Dir.	Secondary D
1	→	↑
2	↑	-
3	↓	→
4	→	-
5	←	↑

(a)

↑	→	↓	←
1	①	③	⑤
②	3		
5	④		

(b)

Fig. 4. Example of subgroup creation in a satellite. (a) Members and possible directions. (b) Bin formation.

First, using the Datagram Routing Algorithm [11], high- and low-priority directions are calculated for each member in the multicast group (Step 1). The direction estimation phase of the Datagram Routing Algorithm calculates the directions on the minimum hop path, and the direction enhancement phase determines the high-priority direction that lies on the minimum propagation delay path. The resulting directions are shown in Fig. 4(a). Note that the group members 2 and 4 have only one direction to go. Each destination is then assigned to the bins of directions they can take (Step 2). Fig. 4(b) shows the bins with members. The circles indicate that the bin belongs to the high-priority direction of that multicast group member. Then a minimum number of bins must be selected such that all destinations are covered (Step 3). Choosing the first and second bins, we can cover all members with a minimum number of bins ($L = 2$). If a member appears in two selected bins, then it is assigned to the bin of its high-priority direction (Step 4). Therefore, the first subgroup (to be sent upward) consists of members $\{2, 5\}$ and the second subgroup (to be sent to right) consists of members $\{1, 3, 4\}$. As a result, two tree-setup packets, $P_1 = (\alpha, \{2, 5\})$ and $P_2 = (\alpha, \{1, 3, 4\})$, are created (Step 5), where P_1 is sent upwards and P_2 to the right. When the upper neighbor satellite receives P_1, it processes the received tree-setup packet following Steps 1–5 and creates new tree-setup packets. The right neighbor satellite follows the same Steps 1–5 for P_2.

D. Dynamic Group Membership

Dynamic group membership is supported by allowing members to join and leave the multicast session while it is in progress. The complete group membership is known only to the source of the multicast session. To support dynamic group membership, we define the following operations.

- **Join:** A satellite willing to join a multicast session sends a *Join_Request* message to the source of the corresponding multicast tree. The first satellite on the multicast tree that receives a *Join_Request* stops forwarding it to the source. It updates its routing table for that session, creates a tree-setup packet containing the session ID and the new destination satellite, and initiates the tree-setup operation as described in Section III-C. To inform the source of the session, it also creates a *Join_Notification* packet and sends it to the source. Fig. 5 shows the sequence of these events.
- **Leave:** A destination satellite willing to leave the multicast session sends a *Leave_Request* that traverses the tree

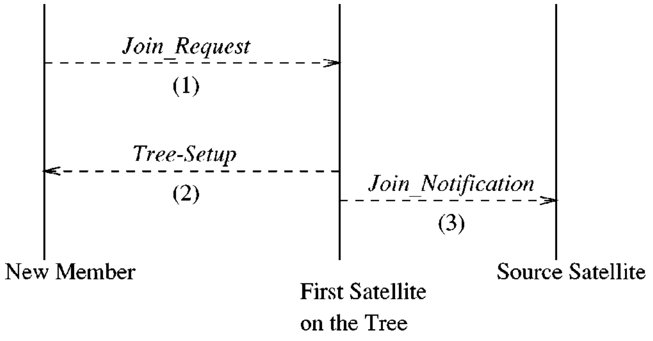


Fig. 5. Join operation.

backward. If the satellite receiving a *Leave_Request* is not a branching point, then it deletes the entry for that session. The first branching satellite receiving the *Leave_Request* updates its routing table so that the multicast packets are not sent to that branch, and sends a *Leave_Notification* to the source. If the leaving destination satellite is an intermediate satellite on the tree, it simply sends a *Leave_Notification* to the source and continues forwarding the multicast packets.

- **Update:** Members joining and leaving a multicast session causes the multicast tree to degenerate. To preserve optimality of the tree, it must be updated. When a notification (either *Leave* or *Join*) is received, the source increments the *notification counter*. When the notification counter exceeds a threshold value Ω, the source initiates the *update* operation. The multicast tree is updated using *update packets*. Update packets are created and processed like tree-setup packets as described in Section III-C up to Step 5, which is modified as follows.
  - —If a direction is already being used, then create an *update packet* and send it in this existing direction.
  - —If a new direction is used, then create a *tree-setup packet* with the list of destinations in the corresponding subgroup and send it in this new direction.
  - —If a direction is no longer used, then send a *tear-down message* in that direction.

E. Link Congestions and Satellite Failures

When the multicast tree is generated, the tree-setup process avoids the links that are already congested or not available due to satellite failures. However, link congestions and satellite failures may occur also after a multicast tree is created. If such a link is on a multicast tree, the tree should be updated such that none of the branches include that link. Since the complete group membership information is only maintained in the source of the multicast tree, the tree update must be initiated by the source satellite. When an intermediate satellite discovers that a particular link on a multicast tree cannot be used, it sends a *Tree_Update_Request* to the source of the multicast tree. Upon receiving the *Tree_Update_Request*, the source satellite resets the notification counter to zero and initializes the update operation described in Section III-D. The resulting multicast tree no longer includes the congested or unavailable links.

IV. Performance Evaluation

The experiments we performed to assess the performance of our multicast scheme fall into three groups.

- We compare the end-to-end propagation delay between the source and each multicast group member for unicast connections and on the multicast trees generated by our routing algorithm. We also demonstrate the bandwidth savings by using our multicast scheme instead of individual unicast connections. These experiments are performed for uniform and nonuniform multicast group member distributions.
- We analyze the effect of the dynamic multicast group membership on the tree length.
- We compare our multicast scheme with PIM [18], MOSPF [16], and CBT [17] schemes. The experiments cover uniform as well as nonuniform member distributions.

In all experiments, a satellite constellation with 288 satellites is used. These 288 satellites are distributed equally among 12 planes, which results in $M = 12$ and $N = 24$ as constellation parameters. We assumed the initial alignment of the satellites, where each satellite resides exactly on one of the logical locations of routing [11]. Since tree delay characteristics change as in the unicast routing protocol, the effect of mobility of satellites on delay is not investigated separately. In [11], it is shown that change in the propagation delay due to satellite mobility is 0.3% on the average. All experiments presented in this section reflect the average of 10 000 independent simulations and cover multicast groups of size 5 to 70, which is almost a quarter of the number of all available satellites.

A. Generation of Multicast Groups

The performance of our new multicast routing scheme is assessed by considering different multicast member distributions. For the uniform case, the sender as well as the group members are selected randomly in the satellite network. However, in a real-life situation, multicast group members may not be uniformly distributed. Multicast data may be destined to several locations that are geographically close to each other. It is apparent that North America, Western Europe, and Southeastern Asia would contain more destinations than any other part of the Earth. A small number of satellites is needed to serve these areas. Densely populated areas have limited but variable radii. Satellites serving these areas can be regarded as *islands* (subgroups) in the satellite constellation. The terms “island” and “subgroup” are used interchangeably.

In order to mimic the correlated member distributions, we express the distribution pattern of the destinations with three parameters $(\mathcal{G}, \mathcal{M}, \mathcal{R})$, where $\mathcal{G}$ is the total number of satellites in the multicast group, $\mathcal{M}$ is the maximum number of member satellites in an island, and $\mathcal{R}$ is the maximum radius of each subgroup. The generation of multicast group members has the following steps.

1) The sender satellite is determined randomly. This satellite may be any of the satellites in the constellation.

2) The satellite islands are formed. For each island:
   a) The number of members in that island is chosen randomly. This random number is between 1 and $\mathcal{M}$.
   b) The radius of the island is determined. Island radii are randomly chosen between 1 and $\mathcal{R}$.
   c) A subgroup core is chosen such that the rest of the subgroup is formed around it.
   d) Other satellites are chosen from that island until the number of satellites for that subgroup is reached.
3) Step 2 is repeated until $\mathcal{G}$ is reached.

This procedure generates multicast groups of a fixed size that have different number of subgroups (islands) with a fixed maximum radius. The maximum size (number of members) and maximum radius of subgroups determines the average density of islands. Determining the subgroup size and radius randomly, we generate subgroups of different size and a fixed average density.

In our experiments, we consider four cases of member distribution.

Uniform: All multicast group members are uniformly distributed.

Nonuniform 1: The group members are distributed nonuniformly as described above with parameters $\mathcal{M} = 5$ and $\mathcal{R} = 4$.

Nonuniform 2: The group members are distributed nonuniformly with parameters $\mathcal{M} = 5$ and $\mathcal{R} = 2$.

Nonuniform 3: The group members are distributed nonuniformly with parameters $\mathcal{M} = 10$ and $\mathcal{R} = 2$.

Note that the uniform distribution corresponds to the least-density case for a given multicast group size and the third nonuniform case corresponds to the highest member density.

B. Performance of the Multicast Routing Algorithm

Our new multicast routing algorithm generates multicast trees by minimizing the packet replication in each hop. Thus, the resulting paths do not always have minimum propagation delays. In the first experiment, we demonstrated the deviation of the propagation delays on the multicast tree created by our new multicast algorithm from the minimum propagation delay paths determined by the Datagram Routing Algorithm [11] from source to each destination. This deviation is reflected as percentage increase in propagation delays. The results of this experiment are shown in Fig. 6.

For all multicast group member distributions, the deviation from the minimum propagation delay path increases as the number of multicast group members increases up to 10 and 15. After this point, as the group size increases, the percentage increase in the end-to-end propagation delay starts decreasing. When the group size is small, each additional group member increases the average percentage deviation because the path to that additional member shares the hops in the tree. These hops may be secondary hops, and therefore, the end-to-end propagation delay is larger than the one of the minimum propagation delay path. When there are more destination satellites, the packets are assigned to their preferred directions with higher probabilities. Therefore, the paths on the multicast tree are very close to the minimum propagation delay paths.

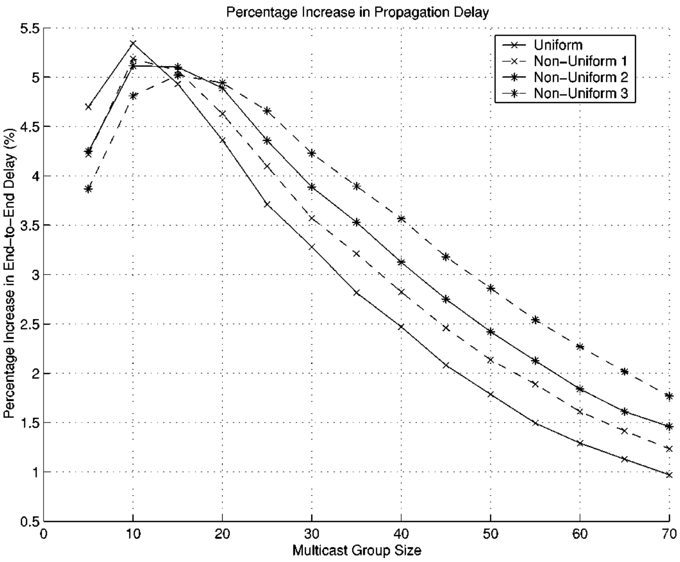


Fig. 6. Percentage increase in propagation delay.

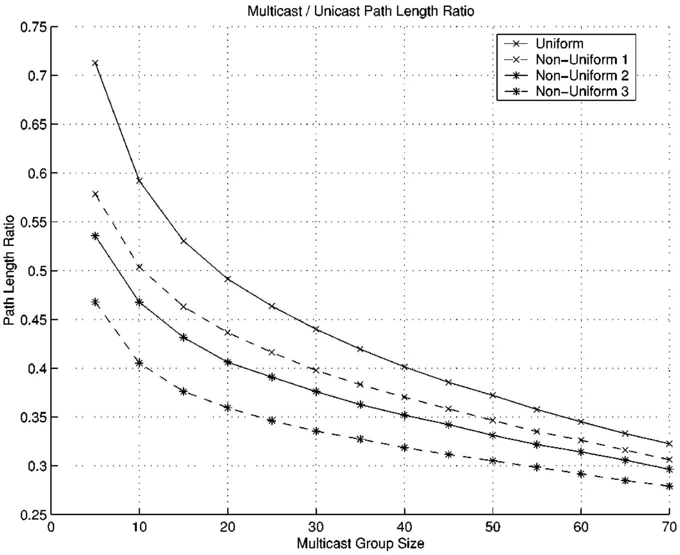


Fig. 7. Multicast/unicast path length ratio.

Comparing the different member distributions, we can see that the distributions with higher densities have smaller deviations for smaller multicast groups and tend to have larger deviations as the multicast groups grow. It is also important that the range of percentage deviation becomes larger for less dense distribution cases and is highest for the uniform distribution case. Under any scenario, the percentage deviation changes between 1 and 5.5%.

In the second experiment, we focus on the bandwidth saved by using our multicast scheme rather than sending independent packets to each destination. For this purpose, we calculate the length of the multicast trees and the sum of the lengths of the individual paths to all destinations. The results are shown in Fig. 7.

The multicast/unicast path length ratio should be interpreted as follows. If the ratio is 1, then the sum of the unicast path lengths to all destinations and the length of the multicast tree are the same, i.e., there is no link sharing at all. The more this ratio approaches to zero, the higher is the link sharing. As an example, if this ratio is 0.5, then this means that using unicast packets consumes $1/0.5 = 2$ times the bandwidth that the multicast packets would use.

Fig. 7 shows that link sharing increases as the group size grows for all group member distributions. When there are more receivers, there is a higher probability that an outgoing link is shared by more than a single destination. As group size grows, the slope of the curve approaches zero. This means that adding a new member increases link sharing more when there are few group members. Since link sharing depends on the location of the destinations, it grows as the destination density in the islands grow. However, the difference in link sharing becomes smaller as the multicast group size increases.

C. Effect of Dynamic Group Membership

The addition and removal of multicast group members causes deviations from the original structure of the multicast tree. As described in Section III-D, the changes to the group membership are reported to the source of the session. The source of the tree initiates the *tree update procedure* when the number of received notifications exceed a threshold value Ω. This threshold value affects the degree the tree degenerates when the group membership changes.

In order to show the effect of the threshold value Ω on the tree length, we perform a set of experiments, where a uniform member distribution in the satellite network is assumed. First, a multicast tree is generated using our new multicast algorithm. Then, based on the assumption that the numbers of members joining and leaving are equal on the average, $\lceil \Omega/2 \rceil$ random member additions and $\Omega - \lceil \Omega/2 \rceil$ random member removals are performed, which adds up to Ω modifications. The length of the tree after these membership changes is recorded. Then the tree is updated using the *tree update procedure* (Section III-D), and its length is also recorded. The length difference between the trees before and after the update is expressed as the percentage increase with respect to the length of the updated tree. Note that the tree before the update is usually longer than the tree after the update. The results are average values of 10 000 random simulations. The experiment is repeated for group sizes 5, 10, 20, 30, and 40. The results are depicted in Fig. 8 for Ω values between 1 and 10.

As shown in Fig. 8, as the threshold value Ω grows, the tree deviates more from its original structure and becomes longer. Larger Ω values correspond to delaying the tree update. The changes in the tree membership affect smaller multicast groups more. As the group size increases, the effect of delaying the tree update decreases. For example, the tree size increases by 4% on the average for the first new member when the group size is 5. For a group of 40 members, the increase in the tree length is only 2.5%, on the average, even after 10 modifications. According to these results, a variable update threshold scheme can be deployed, which updates the smaller multicast trees after fewer notifications are received, and delays the tree update for larger multicast groups.

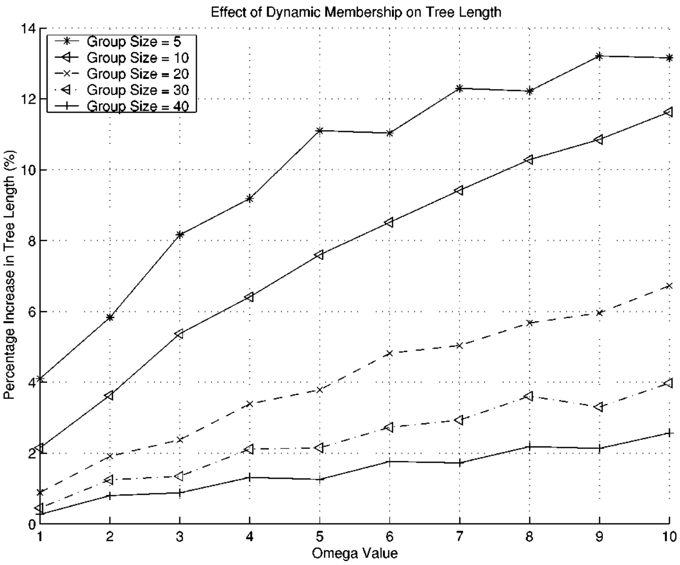


Fig. 8. Effect of dynamic group membership.

D. Comparisons With Other Multicast Schemes

As outlined in Section I, none of the existing multicast schemes are well suited for LEO satellite networks. We now compare our multicast scheme with PIM [18], MOSPF [16], and CBT [17] schemes.

1) PIM Scheme: We first compare our new multicast scheme with PIM [18] because PIM does not require centralized calculations and creates *shortest path trees* (SPTs) when the traffic flow increases. PIM performs well when the multicast group members are densely located in certain areas. Since multicast groups in satellite networks have this type of structure, we compare the length of the trees generated by our new multicast scheme with the length of the PIM trees.

Like many other multicast protocols, PIM [18] is based on constructing and maintaining a multicast tree. The construction of the multicast tree is independent of the underlying unicast routing protocol. The initial multicast trees generated with PIM are shared trees, i.e., different senders use the same tree to reach the members of a certain multicast tree. When the traffic from a specific sender exceeds a threshold, the sender and receivers switch to an SPT, leaving the shared tree used in the first step. PIM is designed to avoid the overhead of broadcast packets and to support low delays for heterogeneous applications.

In our simulations, we assume that the multicast members receive data in high rates so that eventually every destination switches to SPTs. In this case, the SPTs delay characteristics are the same as the unicast connections between the sender and individual receivers. In this set of simulations, we focus on the tree lengths, i.e., the bandwidth usage after PIM switches to SPTs. We compare bandwidth demand of PIM and our multicast scheme for different multicast group member distributions.

The trees of our multicast scheme are as long as the trees generated by PIM protocol in the worst case. PIM tries to establish shortest paths between source and destinations, and links are shared only if they belong to multiple shortest paths. Our scheme, on the other hand, tries to merge the links wherever possible at the expense of increasing the delay. In Fig. 9, we

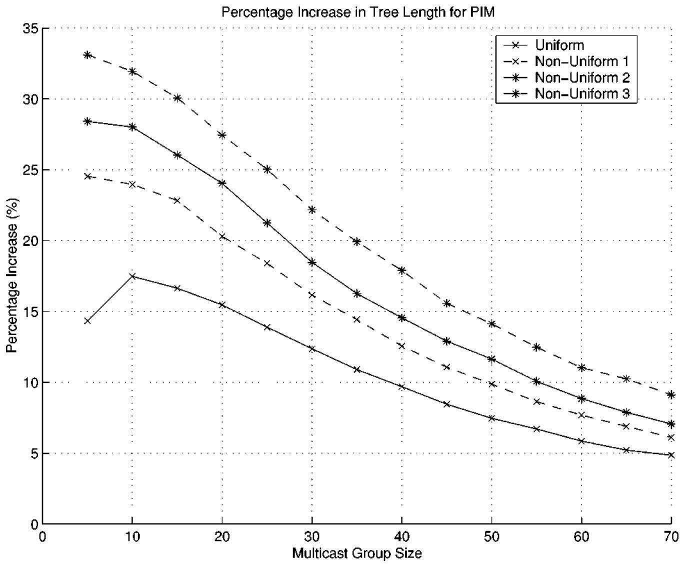


Fig. 9. Percentage increase in tree length for PIM.

demonstrate the savings in the bandwidth. The results presented here show the tree length differences in percentages.

For uniformly distributed group members, the PIM tree length exceeds the tree lengths of our scheme at rates between 17.5% and 9.5%. The multicast trees generated by the PIM protocol are 14% longer than the trees generated by our multicast algorithm when the multicast group has five members. This difference increases to 17.5% for multicast group size of ten. As the multicast group size grows, the difference between the lengths of two trees decreases, but still our multicast scheme produces shorter trees. When the multicast members are distributed in islands, we observe that the savings obtained using our multicast scheme also grows. The highest savings are obtained for the distributions with the highest subgroup densities. As in the uniform distribution case, the difference between the tree lengths decreases as the number of multicast group members increases.

2) MOSPF: The Multicast Routing Extensions for OSPF (MOSPF) [16] enables multicast delivery of packets in the Internet. The group membership information between the hosts and the network is handled by the IGMP procedures [20]. Inside an OSPF area, the protocol creates an SPT rooted at the source spanning all destinations. Individual OSPF areas are connected over the backbone area. The backbone area receives a summary of the group members in every OSPF area to which it is connected. This way, the complete membership information is not broadcast to all OSPF areas. All multicast packets are sent also to the routers in the backbone area, which forward these packets to other OSPF areas if necessary.

In order to use MOSPF in a LEO satellite network, the satellite network must be partitioned into OSPF areas. If the OSPF areas are formed by grouping the logical locations, the traffic between two areas passes over the backbone area, which results in suboptimal paths. For example, even though the source and the destination satellites in two different areas may be only two hops away from each other, the packets may have to be forwarded to the backbone area first. This may cause the packets to take paths multiple times longer than the optimum path. Another problem associated with the OSPF areas is the placement of the backbone area, which all other areas must be connected to. Therefore, it is not feasible to divide the satellite network into OSPF areas. If the entire satellite network is considered as a single OSPF area, then the following observations can be made.

1) Under MOSPF, the group membership information is flooded inside the area using LSA packets, which corresponds to flooding the membership information in the entire satellite network. Our multicast routing algorithm maintains the multicast group membership information at the source satellite and does not require any kind of broadcasting to maintain the group membership information.
2) To create the multicast tree, the MOSPF protocol requires every router on the tree to run independently a shortest path algorithm to determine the next hops to reach every destination. This requires that the satellite topology is maintained in all satellites. Our new multicast routing algorithm is based on the Datagram Routing Algorithm, which does not require the entire network topology and calculates the next hops with very low overhead.
3) If the MOSPF protocol is modified such that the group membership information is maintained without flooding and the next hop to every destination is obtained from the Datagram Routing Algorithm, then MOSPF would create SPTs like the PIM scheme does. Under this scenario, the delay and bandwidth consumption performance of the trees generated by MOSPF would be the same as in the PIM case as presented in Section IV-D-1.

3) Core-Based Tree Scheme: In this section, we compare our multicast routing scheme with CBT protocol [17]. In the CBT scheme, the multicast packets are sent to a designated node called the *core*, which relays these packets to multicast group members. The packets are routed from the source to the core as unicast packets. The connection between the core and group members is accomplished via an SPT. The choice of the core is an important issue affecting the performance of the CBT scheme. In the RFC version of CBT specification [21], a *bootstrap mechanism* is suggested, which selects the core based on the hashing of the router IDs. Although this strategy distributes the cores in the network, it may perform poorly in many cases since it is not possible to consider all possible multicast groups in the network when designing the hashing function.

In our experiments, we compute the core for each tree independently, although it may not be feasible in real implementations. The node closest to the center of the gravity of a multicast group is chosen as the core. The metric used for this calculation is the hop count. The procedure for core calculation is as follows.

1) Using the direction estimation phase of the Datagram Routing Algorithm [11], determine the minimum hop metrics from the source to all multicast group members.
2) Multiply the horizontal and vertical hop counts by -1 for the directions *left* and *down*.
3) Calculate the average horizontal and vertical hop counts and round them to the closest integer.

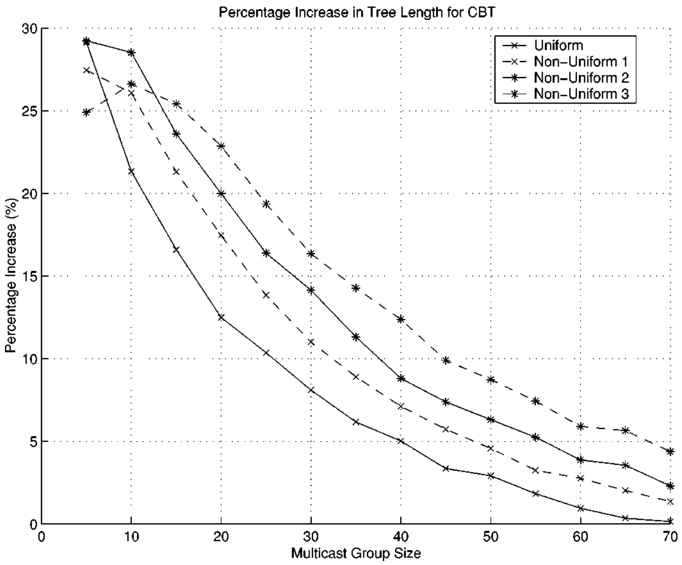


Fig. 10. Percentage increase in tree length for CBT.

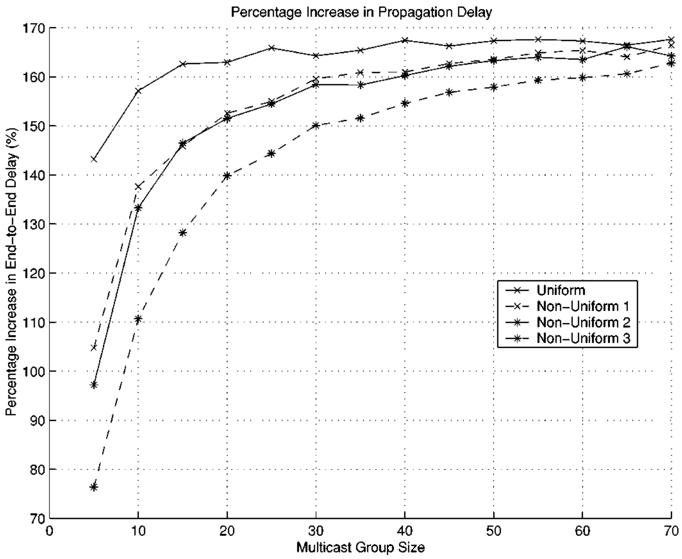


Fig. 11. Percentage increase in propagation delay for CBT.

4) Designate the satellite that can be reached from source by the calculated mean horizontal and vertical number of hops as the core of the tree.

In our simulations, we compare the lengths of the trees generated by our algorithm and the CBT scheme. We also compare the propagation delays of the packets routed on the minimum propagation delay paths and trees generated by the CBT algorithm. These tests are performed for uniform and nonuniform member distributions.

In Fig. 10, the increase in tree length is depicted, which is observed when the CBT protocol is used instead of our multicast scheme. From this figure we can conclude that our multicast scheme creates shorter multicast trees than the CBT protocol. For small group sizes, the difference is above 25%. This difference, however, becomes smaller with the increasing multicast group size. When the number of group members increases, the shortest path tree between the core and the group members includes more branches. Therefore, the unicast connection between the source and core takes up a smaller portion of the total tree length, and the difference decreases. Especially for the uniform distribution, the difference approaches zero, which means that the tree length is the same on the average. Furthermore, denser subgroups benefit from our multicast routing scheme more than the sparse subgroups.

We also present experimental results to show the delay performance of the CBT protocol. In Fig. 11, the percentage difference of the propagation delay on the CBTs and the minimum propagation delay paths between the source and each destination are depicted. For the uniform member distribution, the propagation delay on the core-based multicast tree is on the average 143% to 167.5% longer than the minimum propagation delay paths. The propagation delay difference stabilizes around 165% for multicast group sizes 20 and over. This means that the packets on the core-based multicast tree experience approximately 2.5 times the delay they would experience on the unicast paths. As the subgroups get denser, the difference curve is pushed downwards. For the case with most dense subgroups, the difference starts growing from 75% for a multicast group size of 5. It grows up to 162% for a group size of 70. The other two nonuniform cases show similar results, where they start around 100% and grow up to 165%. Note that the increase in propagation delay for our scheme is between 1% to 5.5% for the same range of group sizes, which is shown in Fig. 6.

V. Conclusion

In this work, we proposed a new multicast routing scheme for datagram traffic in LEO satellite IP networks. The new scheme is based on the Datagram Routing Algorithm. Multicast trees are generated such that the number of branches going out of a satellite is minimized at each step. The simulation results show that the multicast trees provide delays exceeding the minimum propagation delay by at most 5.5% on the average. Multicast trees are multiple times shorter than the sum of unicast paths. They also outperform trees generated by PIM, MOSPF, and CBT schemes. We also present simulation results regarding the dynamic group membership and show that the dynamic group membership scales well with the increasing multicast group size.

References

[1] I. F. Akyildiz and S. Jeong, “Satellite ATM networks: A survey,” *IEEE Commun. Mag.*, vol. 35, pp. 30–44, July 1997.

[2] M. Werner, C. Delucchi, H. Vogel, G. Maral, and J. De Ridder, “ATM-based routing in LEO/MEO satellite networks with intersatellite links,” *IEEE J. Select. Areas Commun.*, vol. 15, pp. 69–82, Jan. 1997.

[3] G. Berndl, M. Werner, and B. Edmaier, “Performance of optimized routing in LEO intersatellite link networks,” in *Proc. IEEE 47th Vehicular Technology Conf.*, vol. 1, May 1997, pp. 246–250.

[4] H. S. Chang, B. W. Kim, C. G. Lee, Y. Choi, S. L. Min, H. S. Yang, and C. S. Kim, “FSA-based link assignment and routing in low-Earth orbit satellite networks,” *IEEE Trans. Veh. Technol.*, vol. 47, pp. 1037–1048, Aug. 1998.

[5] J. Lee and S. Kang, “Satellite over satellite (SOS) network: A novel architecture for satellite network,” in *Proc. IEEE INFOCOM 2000*, vol. 1, Mar. 2000, pp. 315–321.

[6] R. Mauger and C. Rosenberg, “QoS guarantees for multimedia services on a TDMA-based satellite network,” *IEEE Commun. Mag.*, vol. 35, pp. 56–65, July 1997.

[7] N. Maxemchuk, “Routing in the Manhattan street network,” *IEEE Trans. Commun.*, vol. COM-35, pp. 503–512, May 1987.

[8] H. Uzunalioglu, I. F. Akyildiz, and M. D. Bender, “A routing algorithm for LEO satellite networks with dynamic connectivity,” *ACM–Baltzer J. Wireless Networks (WINET)*, vol. 6, no. 3, pp. 181–190, June 2000.

[9] K. Tsai and R. Ma, “Darting: A cost effective routing alternative for large space-based dynamic topology networks,” in *Proc. IEEE MILCOM’95*, 1995, pp. 682–687.

[10] R. A. Raines, R. F. Janoso, D. M. Gallagher, and D. L. Coulliette, “Simulation of two routing protocols operating in a low Earth orbit satellite network environment,” in *Proc. IEEE MILCOM’97*, vol. 1, Nov. 1997, pp. 429–433.

[11] E. Ekici, I. F. Akyildiz, and M. D. Bender, “A distributed routing algorithm for datagram traffic in LEO satellite networks,” *IEEE/ACM Trans. Networking*, vol. 9, pp. 137–147, Apr. 2001.

[12] T. R. Henderson and R. H. Katz, “On distributed, geographic-based packet routing for LEO satellite networks,” in *Proc. IEEE GLOBECOM 2000*, vol. 2, Dec. 2000, pp. 1119–1123.

[13] L. H. Sahasrabuddhe and B. Mukherjee, “Multicast routing algorithms and protocols: A tutorial,” *IEEE Network*, vol. 14, no. 1, pp. 90–104, Jan.–Feb. 2000.

[14] S. E. Deering and D. R. Cheriton, “Multicast routing in datagram internetworks and extended LANs,” *ACM Trans. Comput. Syst.*, vol. 8, pp. 85–110, May 1983.

[15] D. Waitzman, C. Partridge, and S. Deering. (1988, Nov.) Distance vector multicast routing protocol. [Online] RFC 1075. Available: http://www.ietf.org/rfc/rfc1075.txt

[16] J. Moy, “Multicast routing extensions to OSPF,” *Commun. ACM*, vol. 37, pp. 61–66, Aug. 1994.

[17] A. Ballardie, B. Cain, and Z. Zhang, “Core-based trees (CBT),” in *Proc. ACM SIGCOMM’93*, Sept. 1993, pp. 85–95.

[18] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, “The PIM architecture for wide-area multicast routing,” *IEEE/ACM Trans. Networking*, vol. 4, pp. 153–162, Apr. 1996.

[19] J. G. Walker, “Satellite constellations,” *J. British Interplanetary Soc.*, vol. 37, pp. 559–571, 1984.

[20] S. E. Deering, “Host extensions for IP multicasting,”, RFC 1112, Aug. 1989.

[21] A. Ballardie. (1997, Sept.) Core-based trees (CBT version 2) multicast routing. [Online] RFC 2189. Available: http://www.ietf.org/rfc/rfc2189.txt

Eylem Ekici (S’99) received the B.S. and M.S. degrees in computer engineering from Bogaziçi University in 1997 and 1998, respectively. He is currently working toward the Ph.D. degree in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

His research interests include satellite networks, wireless networks, routing protocols, and Internet.

Ian F. Akyildiz (M’86–SM’89–F’96) received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nürnberg, Germany, in 1978, 1981, and 1984, respectively.

He is currently the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, and Director of the Broadband and Wireless Networking Laboratory. His current research interests are in wireless networks, satellite networks, and the Next-Generation Internet.

Dr. Akyildiz is a Fellow of the Association for Computing Machinery (1996). He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria, Chile, in 1986. He served as a National Lecturer for ACM from 1989 to 1998, and received the ACM Outstanding Distinguished Lecturer Award for 1994. He received the 1997 IEEE Leonard G. Abraham Prize from the IEEE Communications Society for his paper entitled “Multimedia Group Synchronization Protocols for Integrated Services Architectures,” published in the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS in January 1996. He received the 2002 IEEE Harry M. Goode Memorial award from the IEEE Computer Society with the citation “for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking.” He is the Editor-in-Chief of *Computer Networks* (Elsevier Science) and an editor for ACM/Kluwer *Journal of Wireless Networks* and ACM/Springer *Journal for Multimedia Systems*. He is a past editor for IEEE/ACM TRANSACTIONS ON NETWORKING (1996–2001), Kluwer *Journal of Cluster Computing* (1997–2001), and IEEE TRANSACTIONS ON COMPUTERS (1992–1996). He was the technical program chair of the 9th IEEE Computer Communications workshop in 1994, the ACM/IEEE Mobile Computing and Networking (MOBICOM’96) conference, and the IEEE INFOCOM’98 Computer Networking conference, as well as for the IEEE ICC 2003. He is the General Chair of ACM MOBICOM’2002.

Michael D. Bender (M’99–SM’00) received the B.S.E.E. degree from The Johns Hopkins University, Baltimore, MD, and M.S. degrees in electrical engineering and computer engineering from Loyola College, Baltimore.

He is currently the Chief of Emerging Communications Technologies for the National Security Agency, where he serves as the Technical Program Director for Wireless Communications Research. He also teaches graduate courses in networking at Loyola. He has made major contributions that resulted in the advancements in networking and signal processing that have improved understanding of emerging communications systems. His current research interests include services infrastructures for telecommunications networks, and broadband and multimedia for wireless and next-generation communications.

Mr. Bender is a member of the Science and Engineering Society and a Senior Member of NSA’s Technical Track.