Wireless Networks 9, 535–544, 2003

A Distributed Multicast Routing Scheme for Multi-Layered Satellite IP Networks

IAN F. AKYILDIZ
Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

EYLEM EKICI *,**
Department of Electrical Engineering, the Ohio State University, Columbus, OH 43210, USA

GAOFENG YUE
Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract. In this paper, a distributed multicast routing scheme is introduced for multi-layered satellite IP networks, which include GEO, MEO, and LEO layers. This scheme aims to minimize the total cost of multicast trees in the satellite network. Multicast trees are constructed and maintained in the dynamic satellite network topology in a distributed manner. Simulation results are provided to evaluate the performance of the new scheme in terms of end-to-end delay and multicast tree cost.

Keywords: satellite networks, multicast routing, shortest path tree, core based tree

1. Introduction

Many applications such as software distribution, electronic commerce, and teleconferencing rely on multicast services. The multicast routing problem in terrestrial wireline networks has already been studied extensively in the past [9]. However, none of the existing multicast routing protocols are well-suited for satellite networks. Some multicast protocols including Internet Group Management Protocol (IGMP) [3], Reverse-Path Multicast (RPM) [4], Distance Vector Multicast Routing Protocol (DVMRP) [10], and the Multicast Extensions to OSPF (MOSPF) [8] employ some type of periodic message exchanges to form or maintain multicast trees. In the Core Based Tree (CBT) approach [2], multicast packets are first sent to a *core* node, from which they are relayed to multicast group members. The choice of the core can affect the tree performance greatly and it is not easy to select a suitable core in a dynamic architecture such as a satellite network. Moreover, none of the other existing protocols are applicable for satellite networks where satellites are moving with respect to the Earth as well as to each other. The only existing multicast routing algorithm [7] developed for the satellite networks is designed primarily for LEO satellite constellations.

GEO, MEO, and LEO layers have their own advantages. A combination of different layers of satellites can provide a more efficient network with better performance than these layers individually. The so-called the *Multi-Layered Satellite Routing algorithm* (MLSR) [1] for unicasting is designed for a satellite network that consists of satellites in three layers. In MLSR, the logical location concept is used to isolate the mobility of the LEO satellites from the satellites in upper layers. In order to reduce the computational complexity in satellites and the communication load in the network, the satellite network is organized hierarchically, where satellites are grouped and their management is given to a satellite in the upper layer. The hierarchical organization is used for routing table calculations. The data packets are forwarded independent of this hierarchy.

Adapting the method used by MLSR [1] algorithm to handle the mobility, we propose a new distributed multicast routing scheme for multi-layered satellite configuration. Our new multicast routing scheme aims to minimize the cost of multicast trees rooted at the source. The cost of a multicast tree is the sum of the cost of all links in the tree, and it reflects the performance of the multicast routing scheme used to build the tree.

The rest of the paper is organized as follows. In section 2, we introduce the satellite network architecture. The new distributed multicast routing scheme is presented in section 3. Section 4 evaluates the performance of the new multicast routing scheme. Finally, section 5 concludes this paper.

2. Satellite network architecture

The hierarchical satellite network consists of three layers of satellites, namely, LEO, MEO, and GEO satellite layers. LEO and MEO satellites are moving with respect to the Earth. The mobility of the LEO satellites is captured by the logical location concept [6]. Logical locations are fixed grid points in the space which are embodied by the nearest satellites. At any given time, a satellite is associated with only one logi-

* Corresponding author.
** Eylem Ekici was with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, when this work was performed.

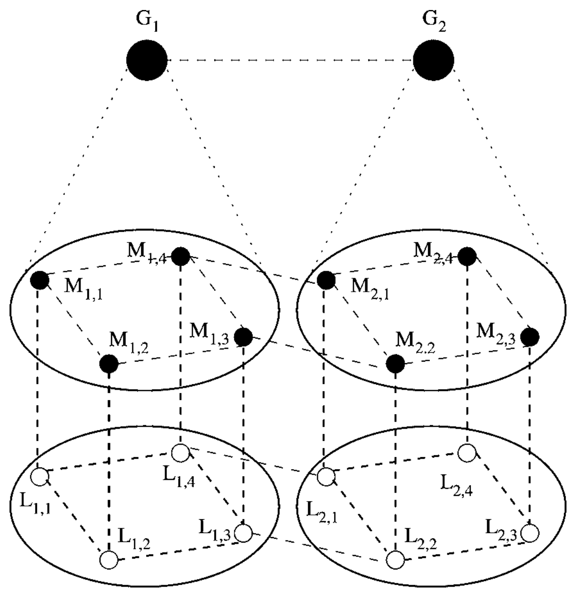


Figure 1. The architecture of the hierarchical satellite network.

cal location which it is closest to, and is represented by the ID of that logical location. Satellites in the same layer are connected to each other via *Inter-Satellite Links* (ISLs), while the communication between different layers is accomplished over *Inter-Orbital Links* (IOLs). An ISL from a satellite A to another satellite B is denoted by $ISL_{A\to B}$. Similarly, an IOL from A to B is represented by $IOL_{A\to B}$. The sources and destinations of information are assumed to be the gateways on the Earth. Satellites communicate with the terrestrial gateways over *User Data Links* (UDLs). A terrestrial gateway can be directly connected to multiple satellites in different layers.

Each link in the network is associated with *delay* and *cost* metrics. The *delay* of a link includes processing, propagation, and queuing delays. The *cost* of a link is related to the available bandwidth and the type of the link in the satellite network. The type of links include ISL, IOL, UDL. These metrics are measured for the downstream direction by the upstream node. In our scheme, the delay information is not needed in the calculation of the multicast tree. Our objective is to minimize the cost of the tree in our new multicast routing scheme.

The LEO satellites in the coverage area of a MEO satellite form a LEO group. All LEO satellites in a LEO group are managed by the MEO satellite that covers them. The period in which the LEO group memberships do not change is called a *snapshot* period. LEO groups are represented as virtual nodes in GEO satellites. GEO satellites do not know the details of the LEO satellite layer topology. The MEO satellites in the coverage of a GEO satellite form a MEO group. All MEO satellites in a MEO group are managed by the GEO satellite that covers them.

A partial picture of the hierarchical satellite network is depicted in figure 1. G_i, $i = 1, 2$, are the GEO satellites. $M_{i,j}$ are the MEO satellites and $L_{i,j}$ correspond to LEO groups, $i = 1, 2$; $j = 1, 2, 3, 4$. The LEO satellites within the LEO groups are not shown. The nodes in the satellite network are connected by the dashed lines.

3. Description of the routing scheme

3.1. Definitions

Definition 1 (Cost of a link). The cost $C(l)$ of a link l is the product of the weight of the link and the utilization of the link:

$$C(l) = \frac{W(Cap - A)}{Cap}, \tag{1}$$

where A is the available bandwidth, Cap is the capacity of the link, and W is the weight of the link l according to its type (ISL, IOL, or UDL).

Definition 2 (Path). A path from one node S to another node D is denoted by $P_{S\to D}$. $P_{S\to D}$ includes all nodes along the path and the directed links from S to D along the path.

Definition 3 (Cost of a path). The cost $C(P)$ of the path P is the sum of the cost of the links on the path:

$$C(P) = \sum_{l\in P} C(l), \tag{2}$$

where l is the link on the path P, and $C(l)$ is the cost of the link l.

Definition 4 (Least cost path). The least cost path $P^*_{S\to D}$ from node S to the node D is defined as the path from S to D with the minimum cost among all the possible paths $P_{S\to D}$.

Definition 5 (Entry LEO satellite of a LEO group). Let $L_{i,j}$ be a LEO group in the satellite network. The Entry LEO satellite $ENLEO_{A\to L_{i,j}}$ of $L_{i,j}$ from any node A is defined as a LEO satellite within $L_{i,j}$ which has a link from A with the least cost among all the incoming links from A to $L_{i,j}$. $ENLEO_{A\to L_{i,j}}$ is formally defined as follows:

$$ENLEO_{A\to L_{i,j}} = \begin{cases} \{L_{i,j,q} \mid \min_k C(IOL_{G_i\to L_{i,j,k}})\}, \\ \quad \text{if } A = G_i; \\ \{L_{i,j,q} \mid \min_k C(IOL_{M_{i,j}\to L_{i,j,k}})\}, \\ \quad \text{if } A = M_{i,j}; \\ \{L_{i,j,q} \mid \min_{k,r} C(ISL_{L_{p,q,r}\to L_{i,j,k}})\}, \\ \quad \text{if } A = L_{p,q},\ (p,q) \neq (i,j). \end{cases} \tag{3}$$

Definition 6 (Exit LEO satellite of a LEO group). Let $L_{i,j}$ be a LEO group in the satellite network. The Exit LEO satellite $EXLEO_{L_{i,j}\to A}$ of $L_{i,j}$ to any node A is defined as a LEO satellite within $L_{i,j}$ which has a link to A with the least cost among all the outgoing links from $L_{i,j}$ to A. $EXLEO_{L_{i,j}\to A}$ is formally defined as follows:

$$EXLEO_{L_{i,j}\to A} = \begin{cases} \{L_{i,j,q} \mid \min_k C(IOL_{L_{i,j,k}\to G_i})\}, \\ \quad \text{if } A = G_i; \\ \{L_{i,j,q} \mid \min_k C(IOL_{L_{i,j,k}\to M_{i,j}})\}, \\ \quad \text{if } A = M_{i,j}; \\ \{L_{i,j,q} \mid \min_{k,r} C(ISL_{L_{i,j,k}\to L_{p,q,r}})\}, \\ \quad \text{if } A = L_{p,q},\ (p,q) \neq (i,j). \end{cases} \tag{4}$$

Definition 7 (Source-based multicast tree). A source-based multicast tree T is a tree which is rooted at the source S and spans all destinations. It is composed of nodes and links in the tree T. Every intermediate node in the tree receives multicast datagrams from its upstream node and forwards them to its downstream nodes.

3.2. Overview of Multi-Layered Satellite Routing algorithm

Multi-Layered Satellite Routing algorithm considers hierarchical satellite network architecture including GEO, MEO, and LEO layers. The *logical location concept* is employed to isolate the mobility of LEO satellites from the satellites in the upper layers. LEO and MEO satellites are grouped and their management is accomplished by the corresponding MEO and GEO satellites covering them. Summary links are introduced to represent the links connecting the LEO groups and other nodes. In order to calculate routing tables, satellites measure the delay of adjacent links and encapsulate the delay measurement in a data unit called *delay measurement report* (DMR). Satellites exchange delay measurement reports to create a picture of the topology of the network.

DMRs are sent from lower layers to upper layers. MEO satellites create DMRs for the LEO groups in their coverage, and report their own DMRs and the DMRs of their LEO groups to the GEO satellites they are connected to. GEO satellites exchange the delay measurement reports to create the total topology of the network, including LEO groups rather than individual LEO satellites. Each GEO satellite calculates the routing tables for all MEO satellites and LEO groups in its coverage. Upon receiving the routing table of its LEO group from the GEO satellite, each MEO satellite generates individual routing tables for the LEO satellites in its LEO group. The details of the routing table calculation can be found in MLSR [1], consisting of a series of computation and communication events.

In our scheme, the collection and exchange of the link costs can be achieved by the method employed in MLSR [1]. The following modifications are needed for the cost exchange procedures used in our scheme:

- We use cost measurement report rather than delay measurement report.
- The gateways report the costs of *User Data Links* to the satellites they are connected to via UDLs.
- A summary link is chosen as the link with the least cost that connects the members of a LEO group with another node in the network.
- Only the first nine steps of the procedure described in MLSR [1] are adopted in the multicasting scheme. The rest of them are not needed since the multicast trees are created on demand.

In our scheme, we assume that underlying unicast routing calculates the shortest delay path. The tree calculation is accomplished in a distributed manner and consists of two stages. First, the GEO satellite of the source gateway creates an initial tree in the *Initial_Stage*. The initial tree includes LEO groups rather than individual LEO satellites since it is calculated by GEO satellites. The information about the initial tree is sent to MEO and GEO satellites in the initial tree, and to the MEO satellites whose LEO groups are in the initial tree. Then, the tree calculation enters the *Enhancement_Stage*, where these MEO and GEO satellites expand the subtrees in their corresponding coverage areas.

3.3. The initial stage

The tree calculation is initiated by the source gateway. The source S creates an *Init* message, which contains the source and the group members of the multicast group. If the source S has a UDL to a GEO satellite, it sends the *Init* message to the GEO satellite. Otherwise, it sends the *Init* message along the shortest delay path to its GEO satellite. Receiving the *Init* message, the GEO satellite follows the steps below to compute an initial tree rooted at the source according to the topology information at the GEO satellite, spanning all destinations:

(a) The initial tree (T_i) only has the source node, i.e., $T_i = \{S\}$.

(b) The GEO satellite uses Dijkstra's algorithm [5] to determine the least cost paths from the source to the destinations. Assuming N destinations $D_1, \ldots, D_N$, the GEO satellite calculates $P^*_{S\to D_i}$ (definition 4), for $i = 1, 2, \ldots, N$.

(c) The minimum cost path among all the paths obtained above is added to the initial tree. Select $P = \{P^*_{S\to D_j} \mid \min_{i\in\{1,2,\ldots,N\}} C(P^*_{S\to D_i})\}$, and extend the tree as $T_i = T_i \cup P$.

(d) A destination is selected from the destinations not included in the tree, such that the added cost is minimum when the least cost path from a node in the tree to this destination is added to the tree. In other words, the destination which is closest to the tree is connected to the tree. The destination D to be added and the node t in the tree from which the tree will be expanded to D are selected as follows:

$$(t, D) = \{(t, D_j) \mid \min C(P^*_{t\to D_i})\},$$

$D_i \notin T_i$, $t \in T_i$. T_i is updated as $T_i = T_i \cup P^*_{t\to D}$.

(e) Step (d) is repeated until all destinations are included in the multicast tree.

This process is depicted in figure 2. The links that are not part of the multicast tree are omitted for clarity. The source S sends an *Init* message to the GEO G_1. The *Init* message is composed of the source (S) of the multicast group and the group members D_i, $i = 1, \ldots, 7$. The procedure described above is used by the GEO G_1 to build the initial tree. The path from S to D_5 is added to the tree first. Then, D_2, D_3, D_1, D_6, D_4, and D_7 are sequentially appended to the tree. The links between satellites are represented by dashed lines, and UDLs by solid lines.

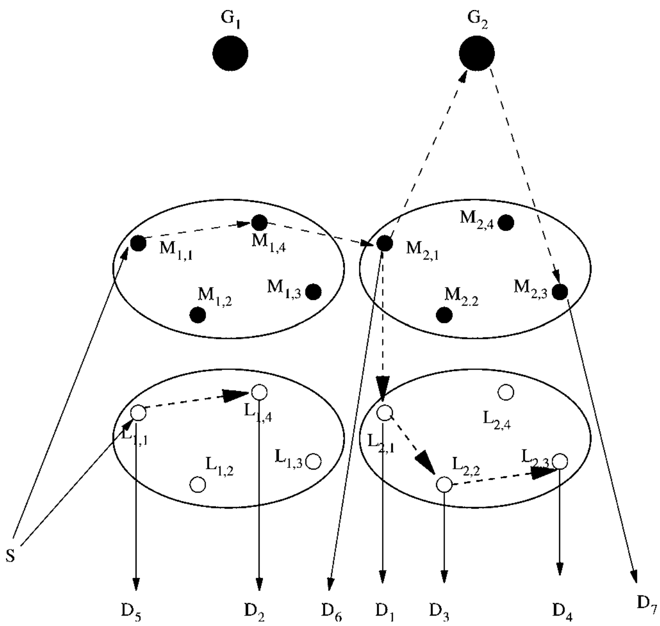


Figure 2. The setup of the initial tree.

Table 1
The *Connectivity* message received by $M_{2,1}$ for itself.

ownerFlag	1
upstream	$M_{1,4}$
downstreams	$L_{2,1}$, G_2
destinations	D_6

Table 2
The *Connectivity* message received by $M_{2,1}$ for $L_{2,1}$.

ownerFlag	0
upstream	$M_{2,1}$
downstreams	$L_{2,2}$
destinations	D_1

3.4. The enhancement stage

The GEO satellite uses the *Connectivity* message to download sequentially the necessary information about the initial tree to MEO and GEO satellites in the tree, MEO satellites of the LEO groups in the tree, and the source via the direct links or the shortest delay paths. The *Connectivity* message includes an *ownerFlag* field, an *upstream* node field, a *downstream* nodes field, and a field consisting of *destinations* connected to the node receiving the *Connectivity* message. The *ownerFlag* field tells whether the *Connectivity* message is for the node receiving it (*ownerFlag* $= 1$) or for the LEO group of the node receiving this message (*ownerFlag* $= 0$). If both a MEO satellite and its LEO group are in the initial tree, the MEO satellite will receive two *Connectivity* messages, one for the MEO satellite, and the other for its LEO group. Tables 1 and 2 show the *Connectivity* messages for $M_{2,1}$ and $L_{2,1}$ of figure 2.

After receiving the *Connectivity* message from the GEO satellite, the source gateway sends a *Setup* message to its downstream nodes, which triggers the tree setup and calculation of missing tree segments in the LEO groups. When a satellite receives a *Setup* message for which it has not received a *Connectivity* message, it buffers the *Setup* message for a specific period of time.[1] If the *Connectivity* message is received on time, *Setup* message is processed, otherwise, the *Setup* message is discarded. If a LEO group is a downstream node, the *Setup* message is delivered to the managing MEO satellite. The *Setup* message has only one field, *ownerFlag*. The *ownerFlag* has the same meaning as the *ownerFlag* in the *Connectivity* message. The satellite receiving the *Setup* message adds corresponding forwarding entries to its routing table to reach the downstream nodes and the destinations connected to it. Then it sends *Setup* messages to the downstream nodes, as the source gateway does.

If the LEO group of a MEO satellite is in the initial tree, then the MEO satellite handles the *Setup* message differently according to its position relative to its LEO group in the initial tree. The MEO satellites use the procedure used by the GEO satellites to calculate the subtrees for their LEO groups or for both themselves and their LEO groups at the same time. The difference is the selection of the source and the destinations for different cases. After the subtree calculation is completed, the MEO satellites inform the LEO satellites to add corresponding forwarding entries to their routing tables and send *Setup* messages to their own downstream nodes and/or the downstream nodes of their LEO groups.

When a satellite K, which is a GEO satellite or a MEO satellite whose LEO group is not adjacent to itself in the initial tree, receives a *Setup* message for itself, it processes the *Setup* message as follows:

1. K adds the corresponding forwarding entries to its routing table to reach the downstream nodes and the destinations directly connected to K. If a downstream node is a LEO group (L), the corresponding forwarding entry should be from K to the $ENLEO_{K \to L}$.
2. K sends *Setup* messages to its downstream nodes.

As an example, in figure 2, $M_{1,4}$ adds one forwarding entry to reach $M_{2,1}$ and sends a *Setup* message to $M_{2,1}$.

A MEO satellite follows the steps below to calculate the subtree for its LEO group or for its LEO group and itself at the same time:

1. If a MEO satellite ($M_{i,j}$) receives a *Setup* message for its LEO group ($L_{i,j}$) and $M_{i,j}$ and $L_{i,j}$ are not adjacent in the initial tree:
   (a) The Entry LEO satellite from the upstream of $L_{i,j}$ (or the source gateway if the source is the upstream node of $L_{i,j}$) is taken as the "source".
   (b) The destinations connected to $L_{i,j}$ and the Exit LEO satellite to the downstream nodes of $L_{i,j}$ are treated as the "destinations".

[1] The maximum buffering time for *Setup* messages should be several times longer than the usual packet transmission time to account for unexpected delays in packet transmission.

(c) Only the links within $L_{i,j}$, and the links going from $L_{i,j}$ to the destinations connected to the LEO group are involved in the subtree calculation.

(d) $M_{i,j}$ follows the same steps as the GEO satellite of the source to calculate the subtree for $L_{i,j}$.

(e) $M_{i,j}$ informs the LEO satellites within $L_{i,j}$ to add corresponding forwarding entries.

(f) $M_{i,j}$ sends *Setup* messages to the downstream nodes of $L_{i,j}$.

2. If a MEO satellite ($M_{i,j}$) receives a *Setup* message for itself and $M_{i,j}$ is the upstream node of its LEO group $L_{i,j}$ in the initial tree:

(a) $M_{i,j}$ is taken as the "source".

(b) The destinations connected to $M_{i,j}$ and $L_{i,j}$, and the Exit LEO satellite to the downstream nodes of $L_{i,j}$ are treated as the "destinations".

(c) Only the links within $L_{i,j}$, the links from $M_{i,j}$ to $L_{i,j}$, and the links connecting the destinations are involved in the subtree calculation.

(d) $M_{i,j}$ follows the same steps as the GEO satellite of the source to calculate the subtree for $L_{i,j}$ and itself.

(e) $M_{i,j}$ informs the LEO satellites within $L_{i,j}$ to add corresponding forwarding entries.

(f) $M_{i,j}$ adds corresponding forwarding entries to its routing table to reach its downstream nodes.

(g) $M_{i,j}$ adds corresponding forwarding entries to its routing table to reach the LEO satellites which are connected to $M_{i,j}$ in the calculated subtree.

(h) $M_{i,j}$ sends *Setup* messages to the downstream nodes of $L_{i,j}$ and $M_{i,j}$.

3. If a MEO satellite ($M_{i,j}$) receives a *Setup* message for its LEO group ($L_{i,j}$) and $M_{i,j}$ is the downstream node of $L_{i,j}$ in the initial tree:

(a) The Entry LEO satellite from the upstream of $L_{i,j}$ (or the source gateway if the source is the upstream node of $L_{i,j}$) is taken as the "source" in the procedure.

(b) The destinations connected to $M_{i,j}$ and $L_{i,j}$, and the Exit LEO satellite to the downstream nodes of $L_{i,j}$, and $M_{i,j}$ are treated as the "destinations".

(c) Only the links within $L_{i,j}$, the links from $L_{i,j}$ to $M_{i,j}$, and the links connecting the destinations are involved in the subtree calculation.

(d) $M_{i,j}$ calculates the subtree for $L_{i,j}$ and itself.

(e) $M_{i,j}$ informs the LEO satellites within $L_{i,j}$ to add corresponding forwarding entries.

(f) $M_{i,j}$ adds corresponding forwarding entries to its routing table to reach its downstream nodes in the initial tree.

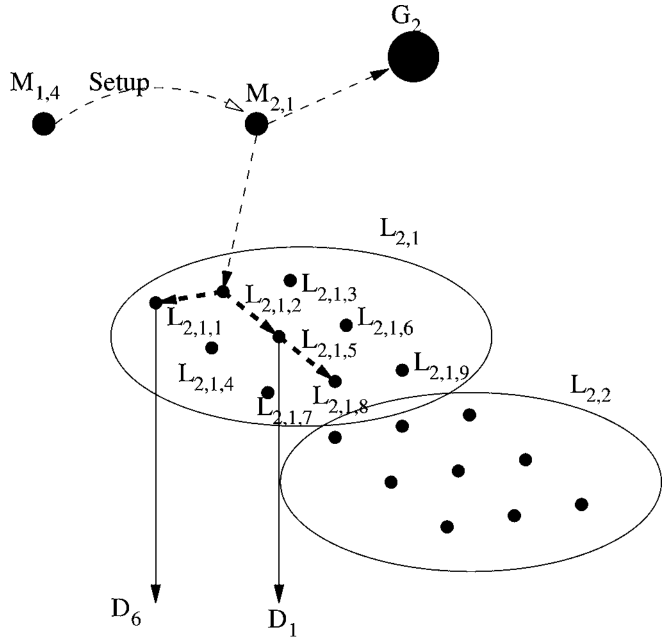


Figure 3. Tree expansion by a MEO satellite.

(g) $M_{i,j}$ sends *Setup* messages to the downstream nodes of $L_{i,j}$ and $M_{i,j}$.

In figure 3, upon receiving the *Setup* message from $M_{1,4}$, the MEO satellite $M_{2,1}$ starts to calculate the subtree for itself and its LEO group. The MEO satellite $M_{2,1}$ takes itself as the source. D_6, D_1 and $L_{2,1,8}$ are treated as the destinations. The paths in the subtree within the LEO group $L_{2,1}$ are marked with bold dashed lines.

The resulting complete tree is shown in figure 4. When each node in the initial tree calculates its subtree, it sends a *Setup_Ack* message to the GEO of the source gateway. After receiving all *Setup_Ack* messages from the nodes in the initial tree, the GEO satellite sends a *Setup_Complete* message to the source to start multicast session.

3.5. Dynamic group membership

3.5.1. Gateway joining

When a terrestrial gateway wants to join a multicast group, it sends a *Join_Request* message to the source of the multicast group. The source forwards the *Join_Request* message to its GEO satellite in charge of constructing the initial tree. Based on the existing tree, the GEO satellite calculates the least costs path from the nodes in the tree to this joining gateway and chooses the minimum cost path. The satellite in the tree having the minimum cost path to the joining gateway is called the *branching satellite*. The GEO satellite adds the minimum cost path to the initial tree and downloads the new partial tree information to the nodes in the new minimum cost path. The GEO satellite sends a *Join* message to the branching satellite to expand the new partial tree. The branching satellite and other satellites in the minimum cost path will follow the same steps as the satellites receiving the *Setup* message in the Enhancement stage. This is illustrated in figure 5. D_8 is the

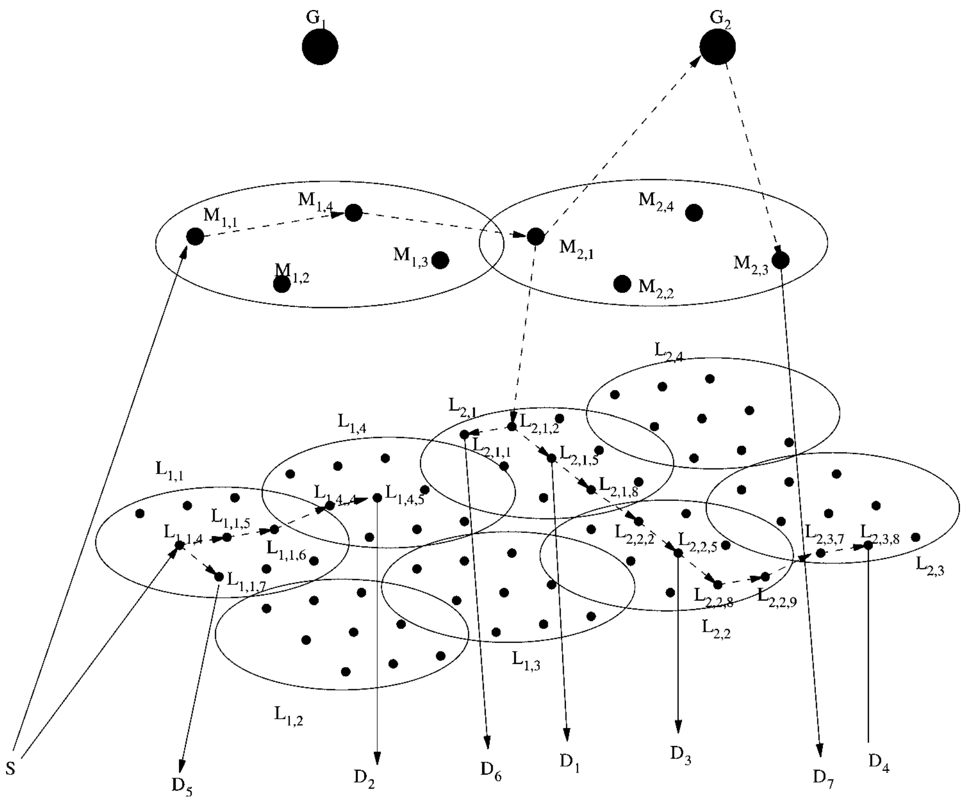


Figure 4. The complete multicast tree.

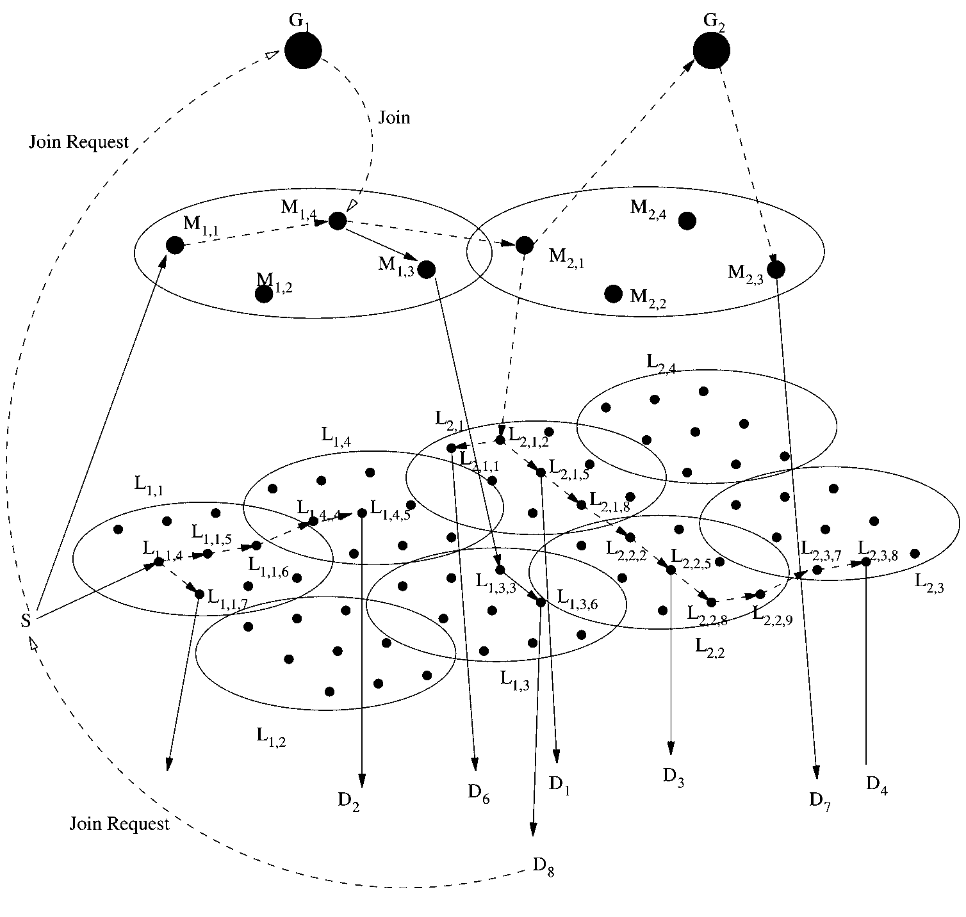


Figure 5. Gateway joining.

gateway willing to join the multicast group. G_1 uses the same procedure as in constructing the initial tree to find that $M_{1,4}$ has the minimum cost path to D_8. $M_{1,4}$ and $M_{1,3}$ process the *Join* message to jointly find a path from $M_{1,4}$ to D_8.

3.5.2. Gateway leaving

If a gateway on the Earth is leaving a multicast group, it will send a *Prune* message to its upstream satellite:

1. If the satellite receiving the *Prune* message has only one entry for this session, it deletes that entry and continues to forward this *Prune* message to its upstream satellite.
2. If the satellite receiving the *Prune* message has more than one entry for this session, it removes the entry for the branch through which it receives the *Prune* message.

3.5.3. Tree update

In order to reduce the cost of the multicast tree and maintain the property of the multicast tree, the *Tree_Update* operation is activated by the source gateway, occurring under two conditions:

- When the number of destinations added to or removed from the multicast tree exceeds a threshold value.
- At the beginning of each snapshot period, when the group memberships change.

The Update operation constructs a new multicast tree with the updated destinations.

4. Simulation results

For performance evaluation of the new multicast routing scheme, we performed three sets of experiments:

- We compared the tree cost difference and the end-to-end delay difference between our multicast scheme and Shortest Path Tree [5].
- We compared the tree cost difference and the end-to-end delay difference between our multicast scheme and Core Based Tree protocol [2].
- We performed further experiments to illustrate the effect of dynamic group membership.

4.1. System description

In this simulation, satellite positions and orbits are taken from the GEO constellation *Inmarsat-3*, the MEO constellation *MEONET*, and the LEO constellation *Iridium*. The interconnection structure of the satellites in the hierarchical network is as follows: GEO, MEO, and LEO satellites are connected to their two adjacent neighbors in the same plane via Intra-plane ISLs. LEO satellites in a plane have an inter-plane ISL to each of the adjacent co-rotating planes. We assume that ISLs crossing the seam are not considered for multicast tree generation and there are no inter-plane ISLs in the area above the latitude 70° and below the latitude −70°. In the MEO constellation, each MEO satellite can establish an Inter-plane ISL with each of the two adjacent planes.

Table 3
Regional data traffic flow shares (in %).

Source	Destination					
	N.A.	Eur.	Asia	S.A.	Afr.	Ocea.
N.A.	90	3	2	2	2	1
Europe	20	70	4	3	2	1
Asia	23	5	65	1	2	4
S.A.	30	7	2	58	2	1
Africa	15	7	4	2	71	1
Oceania	15	2	7	1	1	74

Table 4
Source distribution (in %).

Region	Ratio
North America	50
Europe	18
Asia	17
South America	5
Africa	5
Oceania	5

A satellite in a lower layer has an IOL with a satellite which provides the longest coverage service time in each of upper layers. IOLs in reverse directions are also established to provide duplex communications.

A terrestrial gateway communicates with a LEO and a MEO satellite via UDLs, which can provide largest coverage time. A terrestrial gateway is connected with a GEO satellite with the largest elevation angle via a UDL.

The terrestrial gateways can be on any piece of land on the Earth. Usually, developed areas have more gateways than the developing areas. Here, we consider two types of source and destination distribution: uniform distribution, and non-uniform distribution. For uniform distribution, the longitude of the source and the destination is a uniform distribution variable over [−180° 180°], and the latitude of the source and the destination is a uniform distribution variable over [−90° 90°].

For non-uniform distribution, different areas have different terrestrial gateway densities. Unlike voice traffic, data traffic is asymmetric. We have adopted the voice traffic distribution from existing literature [11,12] by tailoring it to data traffic distribution. The table 3 of [11] gives the voice traffic flow between six continental regions of the Earth. In [12], the regional traffic flow shares are calculated according to the result in [11]. With the fact that more percentage of data traffic are generated from and destined to North America, the regional traffic flow shares in [12] are adjusted accordingly to form the data traffic flow shares shown in table 3. In a similar way, from [11, table 3], we can get the source distribution of a region which is the traffic from that region divided by the total traffic. The source distribution is shown in table 4, which reflects the distinction of the data traffic from voice traffic.

For the simulation experiments, we selected the number of sources proportionally from different regions complying with table 4. For instance, 50% of sources are selected from North America, and 18% from Europe. Given a source in a simulation, the number of destination gateways chosen from a region is determined by the corresponding item in table 3.

The capacity of all UDLs, ISLs, and IOLs are chosen as 200 Mbps, and each outgoing link has a buffer space of 5 MB. Each link's background traffic has a uniform distribution. We analyze two cases. The first case is a lightly loaded network, where the utilization of each link is between 10% and 50%. The second case is a heavily loaded network, where the utilization of each link is between 50% and 95%.

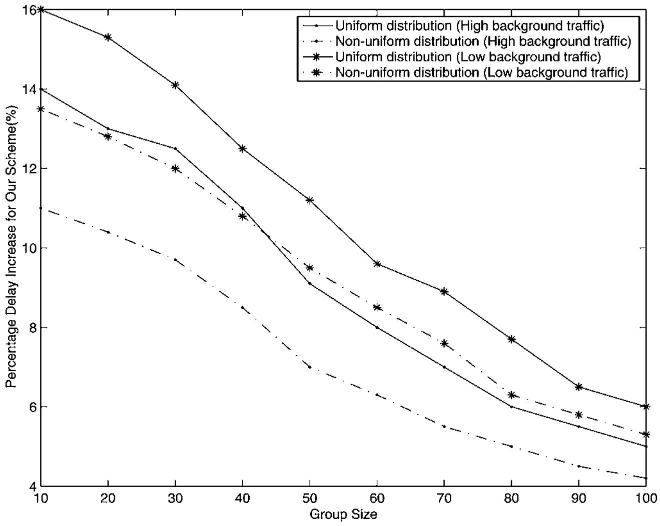
4.2. The comparison with shortest path tree algorithm

In the first set of experiments, we compare the performance of the trees created by our scheme with SPTs. An SPT is composed of the shortest delay paths from the source to the destinations. Our scheme tries to reduce the bandwidth consumption and to increase link sharing. Figure 6 demonstrates how much cost the new scheme can reduce, and how much end-to-end delay increase it can incur. It can be seen that the performance of the SPT protocol and our scheme vary with group size, member distribution. For a group size, 1000 multicast groups are produced for networks with heavy and light background traffics, respectively. These simulations are executed independently and the comparison results are averaged over the corresponding simulations.

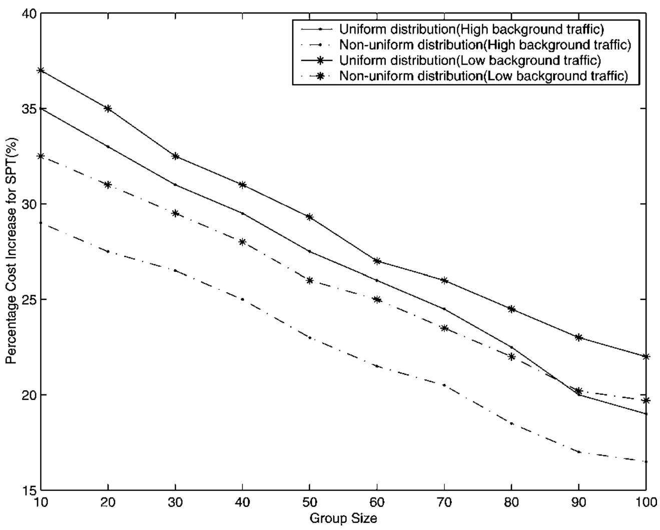
The solid lines in figure 6 represent the comparison result for uniform member distribution. For uniform member distribution, in highly loaded networks, our scheme introduces a delay increase ranging from 5% to 14%. With the group size increasing, the delay difference is decreasing. The SPTs have an average cost increase from 19% to 35% over the trees generated by our scheme. In lightly loaded networks, our scheme introduces a delay increase ranging from 6% to 16%. The SPTs have an average cost increase from 22% to 37% over the trees generated by our scheme.

We also conducted simulations for non-uniform member distribution. The results are shown by the dashed lines in figure 6. In networks with high background traffic, the average delay of the paths on the trees created by our scheme is 4.2–11% larger than the shortest paths. The cost of SPTs exceeds the cost of the trees built by our scheme by 16.5–29%. In networks with low background traffic, the average delay of the paths on the trees created by our scheme is 5.3–13.5% larger than the shortest paths. The cost of SPTs exceeds the cost of the trees built by our scheme by 19.7–32.5%.

The delay and cost difference have similar curves with uniform member distribution. However, these differences are slightly smaller compared to those of uniform member distribution. A noticeable observation is that the delay increase by our scheme is much smaller than the cost increase by SPT protocol. This indicates that we can sacrifice a small delay loss to achieve a higher bandwidth gain.



(a)



(b)

Figure 6. Comparison with SPT. (a) End-to-End delay comparison. (b) Cost comparison.

4.3. The comparison with core based tree protocol

In this set of experiments, we compare the performance of the trees created by our scheme with CBT [2]. Under the CBT protocol, a core node is selected and a shortest path tree based the core is constructed. All data packets are first sent to the core, which forwards the packets through the shortest path tree.

In our simulation, the core of the multicast group is selected as follows: the location of each terrestrial gateway is represented as a location vector in a Cartesian coordinate system. The vector of the location center of terrestrial gateways in one multicast group is assumed to be the sum of the location vectors of these terrestrial gateways. The vector of the location center is converted into spherical coordinates. The LEO satellite which is closest to this position is selected as the core. The motivation of using LEO satellites rather than MEO or GEO satellites as cores is that the selection of LEO

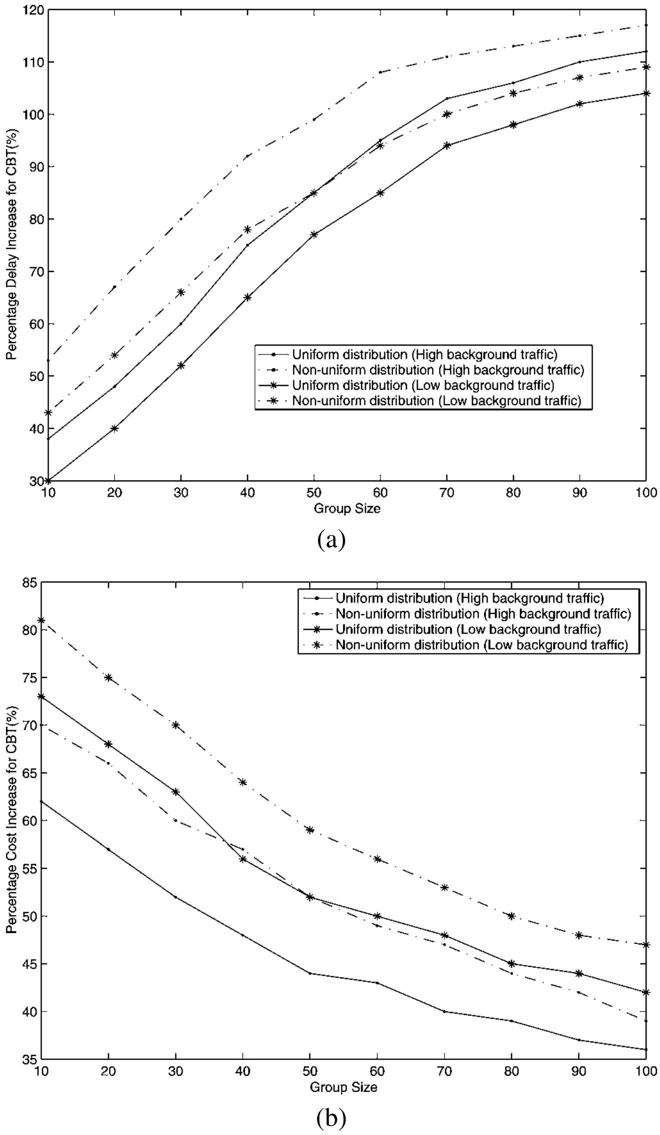


Figure 7. Comparison with CBT. (a) End-to-End delay comparison. (b) Cost comparison.

satellites as cores can incur smaller end-to-end delays.

The CBT delay increase and cost increase over the tree created by our scheme are shown in figures 7(a) and (b), respectively. The solid lines in figure 7 represent the comparisons for uniform member distribution, for which in highly loaded networks, the delay of the trees created by CBT protocol is 38–112% larger than the delay of the trees generated by our scheme. Also, the multicast based core trees have cost 36–62% greater than the trees constructed by our scheme. In lightly loaded networks, the delay of the trees created by CBT protocol is 30–104% larger than the delay of the trees generated by our scheme and the multicast based core trees have cost 42–73% greater than the trees constructed by our scheme.

The comparison results for non-uniform member distribution are shown by the dashed lines in figure 7. In networks with high background traffic, the delay of the trees created by CBT protocol is 53–117% larger than the delay of the trees generated by our scheme. Also, the multicast based core trees have cost 39–70% greater than the trees constructed by our scheme. In networks with low background traffic, the delay of the trees created by CBT protocol is 43–109% larger than the delay of the trees generated by our scheme and the multicast based core trees have cost 47–81% greater than the trees constructed by our scheme. The delay and cost increase curves have the similar shape for uniform and non-uniform member distribution. However, the increase in cost and delay for non-uniform distribution is higher than for uniform distribution.

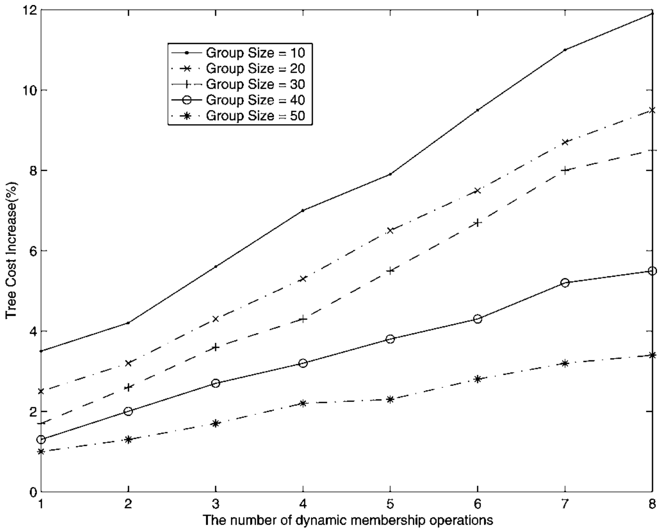


Figure 8. Effect of dynamic group membership.

4.4. The effect of dynamic group membership

The terrestrial gateways can freely join or leave a multicast group. The joining and leaving of multicast members may make the multicast tree lose its characteristic. When this happens, the tree update procedure should be activated to recalculate the tree. In this experiment, we show how well our scheme can accommodate dynamic membership.

A large amount of different size multicast groups with non-uniform member distribution are produced and corresponding trees are generated by our scheme. Joining and leaving gateways are randomly selected. When the dynamic operations are conducted for certain times, the tree is updated and the cost increase of the old tree over the new tree is computed. Here, we consider the group sizes of 10, 20, 30, 40 and 50 and perform the tree update after 1, ..., 8 dynamic operations, respectively. For each case, the tree cost increase percentage is averaged.

Figure 8 exhibits that the tree cost increase goes up with the number of dynamic operations before the tree update procedure operates. Also, the trees with smaller group member size are more subject to the dynamic operations than larger size groups.

5. Conclusions

In this paper, we proposed a distributed multicast routing scheme for multi-layered satellite IP networks. Our proposed scheme utilizes the approach used in MLSR [1] to capture the dynamics of the satellite network, where the mobility of the LEO satellites are captured using the logical location concept and the mobility of the MEO satellites are captured with snapshots. The objective of our distributed multicast routing scheme is to create and maintain multicast trees for which the cost is minimized. The simulation results demonstrate that our scheme generates multicast trees with lower tree costs at the expense of a small delay increase when compared with shortest path trees. With respect to core based trees, our scheme has better performance in terms of both delay and cost. The simulations have also shown that our scheme can support dynamic multicast group membership efficiently.

Acknowledgements

This work is supported in part by the National Science Foundation under grant number ANI-0087762, and NASA Glenn Research Center under grant number NAG3-2580.

References

[1] I.F. Akyildiz, E. Ekici and M.D. Bender, MLSR: a novel routing algorithm for multi-layered satellite IP networks, IEEE/ACM Transactions on Networking 10(3) (2002) 411–424.

[2] A. Ballardie, B. Cain and Z. Zhang, Core Based Trees (CBT), in: *Proc. of ACM SIGCOMM'93* (September 1993) pp. 85–95.

[3] S.E. Deering, Host extensions for IP multicasting, Internet RFC 1112 (August 1989).

[4] S.E. Deering and D.R. Cheriton, Multicast routing in datagram Internetworks and extended LANs, ACM Transactions on Computer Systems 8 (1983) 85–110.

[5] E.W. Dijkstra, A note on two problems in connection with graphs, Numerische Mathematik 1 (1959) 269–271.

[6] E. Ekici, I.F. Akyildiz and M.D. Bender, A distributed routing algorithm for datagram traffic in LEO satellite networks, IEEE/ACM Transactions on Networking 9(2) (2001) 137–147.

[7] E. Ekici, I.F. Akyildiz and M.D. Bender, A multicast routing algorithm for LEO satellite IP networks, IEEE/ACM Transactions on Networking 10(2) (2002) 183–192.

[8] J. Moy, Multicast routing extensions to OSPF, Communications of ACM 37 (1994) 61–66.

[9] L.H. Sahasrabuddhe and B. Mukherjee, Multicast routing algorithms and protocols: a tutorial, IEEE Network 14(1) (2000) 90–102.

[10] D. Waitzman, C. Partridge and S. Deering, Distance vector multicast routing protocol, Internet RFC 1075 (November 1988).

[11] M. Werner, Analysis of system connectivity and traffic capacity requirements for LEO/MEO S-PCNs, in: *Mobile and Personal Communications, Proceedings of the 2nd Joint COST 227/231 Workshop*, Florence, Italy, April 1995, ed. E. Del Re (Elsevier, Amsterdam, 1995) pp. 183–204.

[12] M. Werner and G. Maral, Traffic flows and dynamic routing in LEO intersatellite link networks, in: *Proceedings of the 5th International Mobile Satellite Conference (IMSC'97)* (1997) pp. 283–288.

Ian F. Akyildiz received the B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively. Currently, he is 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. He has held visiting professorships at the Universidad Tecnica Federico Santa Maria, Chile, Universite Pierre et Marie Curie (Paris VI), Ecole Nationale Superieure Telecommunications in Paris, France, Universidad Politecnico de Cataluna in Barcelona, Spain, and Universidad Illes Baleares, Palma de Mallorca, Spain. His current research interests are in satellite networks, sensor networks, wireless networks, and the next-generation Internet. Dr. Akyildiz is an Editor-in-Chief of Computer Networks Journal (Elsevier) and of the newly launched AdHoc Networks Journal (Elsevier). He is an editor for ACM–Kluwer journal, Wireless Networks, and a past editor for ACM–Springer Journal for Multimedia Systems (1995–2002), IEEE/ACM Transactions on Networking (1996–2001), Kluwer journal, Cluster Computing (1997–2001) and IEEE Transactions on Computers (1992–1996). He guest-edited more than ten special issues for various journals in the last decade. He was the technical program chair of the 9th IEEE Computer Communications Workshop in 1994, for ACM/IEEE MobiCom'96 (Mobile Computing and Networking) Conference, IEEE InfoCom'98 (Computer Networking Conference), as well as IEEE ICC'2003 (International Conference on Communications). He is the General Chair for the premier conference in wireless networking, ACM MobiCom'2002. He is an ACM Fellow (1996) and IEEE Fellow (1995). He received the Don Federico Santa Maria Medal for his services to the Universidad of Federico Santa Maria in Chile in 1986. He served as a National Lecturer for ACM from 1989 until 1998 and received the ACM Outstanding Distinguished Lecturer Award for 1994. He received the 1997 IEEE Leonard G. Abraham Prize award (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 (JSAC) in January 1996. He received the 2002 IEEE Harry M. Goode Memorial Award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking".
E-mail: ian@ece.gatech.edu

Eylem Ekici received the B.S. and M.S. degrees in computer engineering from Bogaziçi University, Istanbul, Turkey, in 1997 and 1998, respectively. He received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2002. He is currently an Assistant Professor with the Department of Electrical Engineering at The Ohio State University, Columbus. His research interests include satellite networks, wireless networks, sensor networks, routing protocols, and Internet.
E-mail: ekici.2@osu.edu

Gaofeng Yue received the M.S. degree in electrical engineering from Shanghai Jiao Tong University, P.R. China in 1995. He worked in Shenzhen Zhongxing Telecom Company from 1995 to 2000 as a R&D software engineer. He is pursuing the Ph.D. degree in electrical engineering at Georgia Institute of Technology.
E-mail: yuegf@ece.gatech.edu