Wireless Networks 7, 413–424, 2001

A Virtual Topology Based Routing Protocol for Multihop Dynamic Wireless Networks

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

JAMES I. PELECH
Lucent Technologies, 67 Whippany Rd PO Box 903 Whippany, NJ 07981-0903, USA

BÜLENT YENER
Bell Laboratories, Lucent Technologies, 700 Mountain Ave., Murray Hill, NJ 07974, USA

Abstract. In this paper, a new hierarchical multihop routing algorithm and its performance evaluation is presented for fully dynamic wireless networks. The routing algorithm operates on a *virtual topology* obtained by partitioning the routing information for mobile terminals and mobile base stations into a hierarchical, distributed database. Based on the virtual topology, each mobile base station stores a fraction of the routing information to balance the complexity of the location-update and the path-finding operations. Mobility of the network entities changes the load distribution and causes processing and memory bottlenecks in some parts of the network. However, since the network routing elements are also mobile, their movement can be used to distribute the load. Thus, new load balancing schemes are intoduced to distribute the routing overhead uniformly among the mobile base stations. The performance of the hierarchical multihop routing algorithm is investigated through simulations. It is shown that the routing protocol can cope with high mobility and deliver packets to the destinations successfully.

Keywords: dynamic wireless networks, location update operation, mobile base stations, multihop routing algorithm, path finding operation, virtual location area, virtual cell, virtual topology

1. Introduction

Dynamic wireless networks are more reliable and efficient than the conventional packet radio networks because (1) they save bandwidth for communication between remote terminals by providing routes through the wired part of the network, and (2) multihop wireless links are used between mobile terminals close to each other. In general, the dynamic wireless network distinguishes itself from the packet radio networks by its unique architecture which can be used for, but not limited to, exploring and provisioning dangerous zones, military, and journalism applications [7,10]. Since the dynamic wireless network is a relatively new architecture, there are many open research issues to be addressed. One of these issues is the routing problem.

Dynamic wireless networks are made up of heterogeneous nodes. These nodes are classified as mobile terminals (MT), mobile base stations (MBS) and fixed switching centers (SC). The MTs can only communicate with the MBS they are currently assigned to. The MBSs are connected to each other and SCs such that they form trees rooted at the SCs. In this model, these trees are connected to each other via a high speed, wireline network at their roots, i.e., their respective SCs. The topology information is incrementally collected at the MBSs and SCs.

The connection structure imposes a hierarchical ordering of nodes that also reflects their complexity. The MTs are small, low complexity devices that do not consume much power. In contrary, the MBSs are more complex nodes with high power consumption potentials capable of managing the connections of many MTs in its designated working area. The mobility of the nodes are increasing with decreasing complexity and power consumption characteristics. Hence, the MTs are portable, handheld devices; whereas the MBSs should be carried in motor vehicles. SCs do not move at all. The detailed architecture is presented in section 2.

The architectural structure of dynamic wireless networks has many advantages. First of all, since the network elements are connected forming trees, the number of MTs and corresponding total coverage area can be increased at a cost proportional to the logarithm of the number of elements in the tree. Hence, a single tree can cover vast areas at very low communication and management costs. Second, since the trees are connected to each other via a high capacity backbone network, many trees can be connected to each other even if they are sparsely distributed. The use of the backbone network maintains the delay between far away trees at acceptable levels. Hence, it would be possible to cover and connect countries, or even continents, using a single network while providing mobility and scalable expandability. Furthermore, having low complexity terminals decreases the cost of MTs and increases their number within the network.

There exist some similarities between the network architecture presented here and the mobile ad hoc network architectures defined by IETF. The mobile ad hoc networks consist of hosts complex enough to take part in the routing of packets. The routing problem in mobile ad hoc networks is being studied in the IETF MANET working group [6]. The routing protocols for mobile ad hoc networks are outlined in [3,4], and compared for performance of various parameters. The mobile ad hoc networks and our proposed network model both assume mobile hosts for communication. However, there are major architectural differences between these two models:

- *Specialized mobile entities.* The elements of our proposed network architecture have specialized capabilities. The MTs can receive and send information, but, unlike the nodes of the mobile ad hoc networks, they cannot route packets. In our architecture, peer-to-peer communication between a pair of MTs goes through a MBS which is specifically designed for routing purposes.
- *Fixed network.* The proposed model here requires a fixed backbone network with SCs as access points, which is not present in mobile ad hoc networks. The fixed network reduces the probability of connection interruption by routing the packets over a reliable, fixed network over long distances. Fixed networks are not part of the ad hoc network architectures. The coverage area of wireless ad hoc networks and the number of participating nodes is much smaller when compared to the similar measures of the dynamic wireless network.
- *Hierarchical network structure.* Our proposed network architecture has a hierarchical node organization. The hierarchical structure increases the scalability of the communications system. Ad hoc networks do not assume any kind of node organization. In applications such as military operations and emergency situations there is a command and control chain which can be captured by the hierarchical architecture proposed here.
- *Load balancing with mobility.* Finally, none of the previous work considers *load balancing* which is expected to be an important issue in tactical networks in which base stations may have different processing and storage capabilities. To avoid bottlenecks in the network, the processing and memory overhead must be distributed among the base stations uniformly. In this work we show how the mobility of base station can be used to achieve load balancing.

We believe the existing multihop routing algorithms that are designed to be used in purely wireless, relatively small ad hoc networks with uniform node complexity have significant drawbacks when applied in dynamic wireless networks which may not be overcome easily. For example, in a dynamic multihop wireless network, only mobile base stations (MBSs) can communicate with their peers, i.e., with other MBSs. On the other hand, mobile terminals (MTs) cannot communicate with other MTs. Thus, the existing multihop routing protocols [2,8,9] may be applicable on the MBS level. However, they cannot handle the mobility of the MT because a MT can move from a MBS to any other MBS.

In contrast with the mentioned approaches, we adapt the model suggested in [1] and consider a distributed database that organizes the network nodes into trees. Partitioning of the network nodes and the global state information induces a virtual network on which the amount of state information is minimized at the expense of optimal routing. This work extends the previous work in [1] in several ways. First, we eliminate flooding from the routing algorithm. As a result, the amount of control messages is reduced. Second, we enhance the protocols to prevent packet loss during the transient behavior of the protocols. Third, we devise protocols for load balancing by moving base stations to heavily loaded areas. Finally, we present an extensive simulation study to demonstrate the performance of the protocols under various network conditions.

The remaining portion of this paper is organized as follows. In section 2, we describe the system model of the dynamic wireless network and explain the virtual topology construction. Then, we explain the proposed hierarchical multihop routing protocol in section 4. In section 5, we present protocols for location update. In section 6, we discuss the load balancing issues. In section 7, we investigate the performance of the proposed algorithm through simulations. Finally, we conclude the paper and point out possible future research in section 8.

2. The physical network architecture

We assume that the dynamic wireless network consists of (i) a wired backbone network with an arbitrary topology, (ii) a collection of fixed *switching centers* (SCs) that are connected to the backbone nodes, (iii) a collection of *mobile base stations* (MBSs) linked to the SCs (although we use the name "BASE stations", our so-called "base stations" are mobile, even highly mobile), and (iv) *mobile terminals* (MTs) which can communicate with mobile base stations as shown in figure 1. The physical network architecture is hierarchical and consists of four levels.

Each switching center (SC) is connected to at least one backbone node through physical links. Switching centers are considered to be the access points to the wired high-speed backbone for long distance communications. We assume that each backbone node can serve up to S SCs.

Each SC can support at most B mobile base stations using wireless channel connections. Furthermore, two MBSs can communicate with each other without going through any SCs. Note that we assume an underlying mechanism which assures each MBS can directly communicate with one and only one SC at any time instance.

Each MBS can support at most T mobile terminals (MTs). Since the base stations are mobile, they do not have a well-defined cell structure found in cellular networks. We call the domain that an MBS serves a *virtual cell* (VC). The communication between an MBS and MTs in the same virtual cell are made via broadcast medium. However, no communication among MTs is allowed even if they are in the

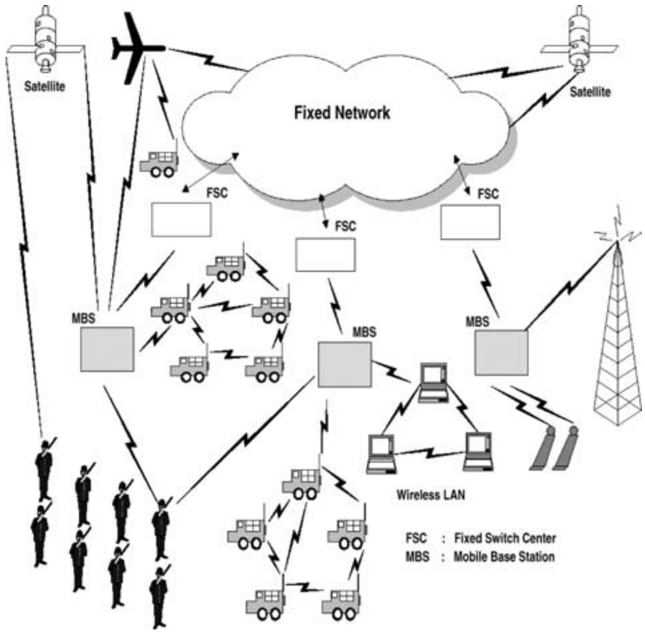

Figure 1. Network model.

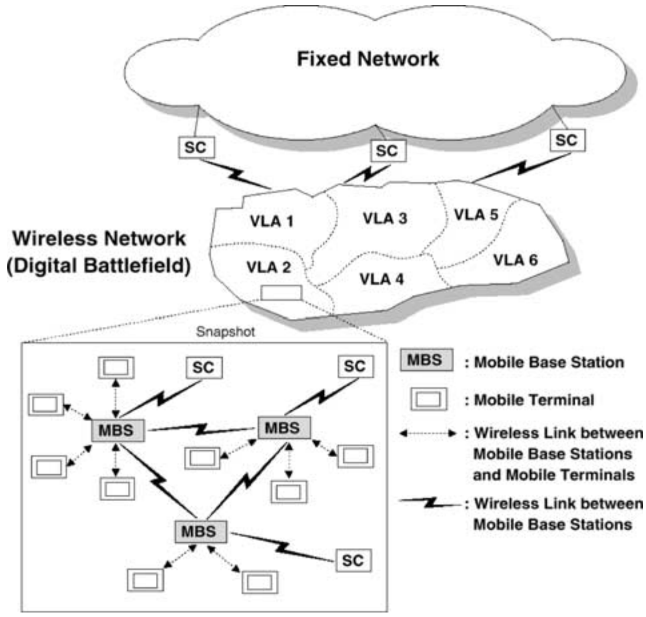

Figure 2. Trees, directories etc.

same virtual cell. We assume that *each MT is supported by exactly one MBS at any time instance* as suggested in [1]. Finally, the union of virtual cells induces a *virtual location area* (VLA) which is served by a single SC.

We assume parameters S, B, and T are given. Thus, the total number of MTs in the system is $N = S \cdot B \cdot T$. In the next section, we explain the basic concepts of the new hierarchical multihop unicast routing protocol.

3. The virtual network architecture

The routing algorithm presented in this paper operates on a *virtual topology* which is a distributed database structure. A pair of MBSs are considered to be adjacent in the virtual topology if they appear as a next-hop in each others routing tables. Thus, there is a direct physical link between adjacent nodes but not every physical link defines adjacency.

3.1. Construction of virtual topology

The virtual topology is constructed in two steps. First, the MBSs served by a single SC are partitioned into disjoint subsets. Second, the routing tables of these MBSs that reside within the same VLA are organized in a hierarchical database. We introduce a simple algorithm to construct the virtual topology. The algorithm partitions the MBSs within a VLA into disjoint *Breadth First Search* (BFS) trees [5]. Each tree is rooted at one mobile base station and has three constraints: (i) the diameter is at most D, (ii) the size is at most Δ, and (iii) maximum degree of each node in the BFS tree is at most d. The objective behind these constraints is as follows: the diameter bound will be used for routing length, the size of each tree determines the bounds of the state information. The degree bound is motivated by the processing and memory limitations of a MBS. If a tree violates one of these constraints it is called a *saturated* tree. In each VLA, the following algorithm is invoked only once during the network setup. Therefore, the overhead incurred by this partition algorithm is negligible.

1. Initially, all MBSs are unmarked.
2. While there is an unmarked MBS do:
   (a) choose a root MBS b from the unmarked MBSs to build a new BFS tree;
   (b) expand the BFS tree by adding an unmarked MBS adjacent to that BFS tree;
   (c) REPEAT step 2 UNTIL tree is saturated;
   (d) mark all the MBSs in this tree and go to step 2.

We note that the resource requirements of a MBS are dependent on their location in the tree (i.e., the maximum requirements occurring at the root nodes). This introduces *asymmetry*. As a result only certain MBSs may become a root. However, this asymmetry can be prevented by choosing the parameters Δ, D, B and T so that *any* MBS can become a root.

3.2. Distribution of routing information

The routing information is distributed to the MBSs in a hierarchical way as shown in figure 2. The state information at each node reflects the hierarchy and has the following components:

- the ID of its parent node,
- the IDs of the nodes in the subtree below this node (both MBSs and MTs), and

- the *next node* for each node in subtree (i.e., next hop in path to node).

The objective of storing the routing information in a distributed database is achieving to limit the location update overhead. Since the topology information is partial, its update is local. Each switching center (SC) maintains an entry in each tree and the terminals in its VLA, and can communicate with each of the root MBSs. The directory size is at most BT, where B is the maximum number of MBSs in a VLA, and T is the maximum number of terminals served by a MBS.

Each MBS in a BFS tree is a unique *home* to $O(T)$ mobile terminals (MTs) residing in its virtual cell. The MBS has to maintain a directory for these MTs. We call such an MBS as the *home MBS* and the corresponding BFS tree as the *home tree* of these MTs. Each MBS is capable of communicating only with their immediate children and their parent.

Each MBS in a BFS tree knows the topology of the *subtree* rooted under it. Thus, the root MBS of a BFS tree knows all the MTs covered by the MBSs on this tree. The total directory size in an MBS is $O(T\Delta' + \Delta')$ where T is the maximum number of MTs allowed in a virtual cell and Δ' is the size of the subtree under this MBS. Thus, each root MBS has a directory in the size of $O(T\Delta + \Delta)$.

4. The path finding procedure

In dynamic wireless networks, routing has two components: (i) locating the destination MT (*location update procedure*), and (ii) finding a feasible path from the source to the destination (*path finding procedure*). In this section, we present a novel path finding procedure which distributes the complexity and overhead among the MBSs and SCs. The routing algorithm is based on finding a *common predecessor* for the source and destination nodes in the virtual hierarchy. Let s denote the source and d denote the destination nodes. We consider the following three cases.

4.1. Case 1: Intra-tree routing – s and d are in the same tree

In this case, there are two possibilities: source and destination both reside (i) in the same virtual cell, or (ii) in different virtual cells. In the first case, they have the same parent MBS node in the tree. Thus, routing is trivial: s forwards the packet to its parent which in turn delivers it to t. If s and t are in different virtual cells, then the packet follows a path $P = \{p_s^1, p_s^2, \ldots, p_s^k \mid k \leqslant D\}$ where p_s^i is the ith predecessor of s toward the root of the tree and p_s^k is also a predecessor of destination d. Note that routing provides the shortest path over the virtual topology, not on the physical one. For example, if there are two MBSs that are within communication distance of each other but reside in different subtrees, the message must travel up to the first common node of the two subtrees and then down the other subtree to the destination. Thus, the cost is $O(D)$.

4.2. Case 2: Inter-tree routing – s and d are in different trees

Since the information kept in each tree is local to the members of that tree, inter-tree routing operations are performed via the SC that coordinates the roots of the trees in the same VLA. In the following, we specify the path finding procedure for this case.

1. The parent MBS of the source forwards the packet from the source MT to the root MBS of its tree.
2. The root MBS of the source tree does not have the information about the destination since its knowledge is also limited to this tree. Thus, it forwards the packet to the SC. Since the information kept in an SC covers only the MTs within the LA, we must consider the following two cases:

   Case 2.1. *Source and destination are in the different trees of the same VLA.* In this case, the SC has the routing information and can determine which tree host the destination d. Thus, the packet is forwarded to the root MBS of the destination tree. For this case, the protocol will terminate with steps 1 through 2.1.

   Case 2.2. *Source and destination are in different VLAs.* The fixed part of the dynamic wireless network plays an important role in this type of communication. Since the source and destination may be far away from each other, it is unreliable and inefficient to solely use multihop routes as shown in figure 3. Therefore, we utilize the fixed part of the network to achieve better reliability and share the load in the wireless part of the network.

The packet from the sources is routed over the wired-backbone. In addition to steps 1 and 2, the protocol has the following steps performed by a SC upon receiving the packet:

3. Discard the packet if the destination d is *not* in its VLA (i.e., if it does not have valid routing information for d).
4. Forward the packet to the root MBS of the destination tree.
5. Only one root MBS will forward the packet to the destination MBS which serves as the destination of the packet. This is true since each root MBS knows all MTs residing in this tree.

Cost of path finding. In this case, we need two multihop routes (i.e., source → source root MBS and destination root MBS → destination) and a unicast between a SC and the root MBS in its VLA. Therefore, the routing length will be bounded by at most twice the tree depth plus extra hop from and to the SC which is $O(D)$.

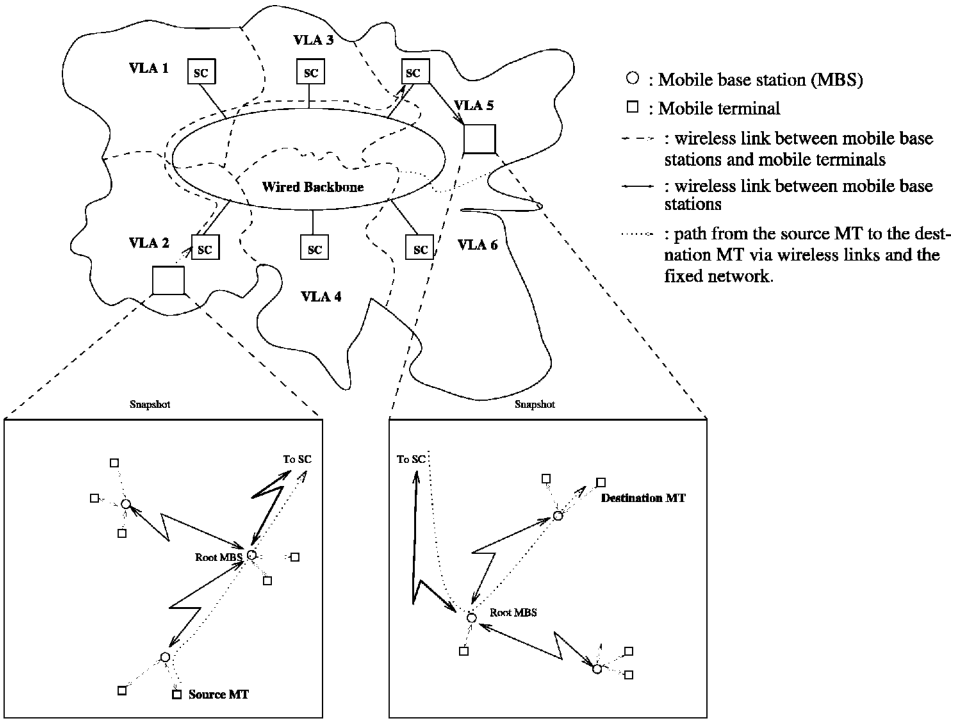

Figure 3. Source and destination in different VLAs.

5. The location update procedure

The location update procedure aims to maintain an up-to-date virtual topology for the path finding operation. Let t be a MT and let b be its home-MBS in the tree T_i which is rooted at node r_i. Let b' be the new MBS of the MT t at the time of update. For simplicity we assume that a VLA covers a large area. Thus, we analyze MBS or MT movement within a VLA. There are several applications in which a VLA is defined based on application specific partition of the entities. For example, in a digital battlefield operation [7,10] a VLA may be defined based on the command–control chain. In this case, the MBSs and the MTs correspond to armored vehicles and foot soldiers, and they usually stay within their VLA.

We analyze the location update procedure in four cases.

Case 1. $t, b, b' \in T_i$, i.e., the move is local to the home tree of the terminal.

Case 2. $t, b \in T_i$ and $b' \in T_j$, where $j \neq i$, i.e., the move causes a home tree change.

Case 3. Let $t, b' \in T_i$ and $b \in T_j$, where $j \neq i$. Thus, in this case, the MT t moves to a new VC on the same home-tree. However, the previous home MBS of t moves to a different tree.

Case 4. $t, b, b' \in T_j$. Here both terminal and the home base move to a new tree.

5.1. Control message format

The location update procedure uses a control message which is exchanged among network entities during the procedure. An UPDATE message is a quadruple: ⟨type, terminal_ID, home-MBS_ID, tree_ID⟩. The type field is 3 bits. Thus, there are at most eight types of UPDATE messages. Since there are at most T mobile terminals in each MBS and B MBSs in a VLA, the total number of bits, ζ, in an update message is

$$\zeta = 3 + \log T + \log B + \log\left(\frac{B}{D}\right). \qquad (1)$$

The field "tree_ID" contains the ID of the root MBS in that tree. We will now explain how the location update procedure handles the previously mentioned four cases.

5.2. Case 1: Tranquil home MBS and moving MT are in the same home-tree

1. Upon moving to a new MBS b' (i.e., detecting a stronger MBS), the MT t sends an UPDATE$(000, t, b, r_i)$ message to b'. The type field of that update message indicates that it originated from a MT.
2. MT sends an UPDATE$(101, t, b', r_i)$ message to its old MBS to inform the old MBS that it is in the process of moving to a new base station b'.
3. Upon receiving that update message, the new MBS b' informs the home MBS b of the MT t by sending TAKEOVER$(001, t, b', r_i)$. This message indicates this base is the new parent of the terminal. The TAKEOVER message will propagate through the old base and update all the base stations on this path.
4. Upon receiving UPDATE$(101, t, b', r_i)$ message, the old MBS marks the mobile as moving by setting a

flag in its database and waits for the control message TAKEOVER$(001, t, b', r_i)$ to come from the new MBS b'. During the time period that step is executed and the TAKEOVER message sent from the new base arrives in b, messages designated to the MT will be stored at the old MBS b. Thus, the time that the MT is out of reach is the time required by the mobile to "re-tune" to the new MBS, typically a small fraction of a second.

5. In this case, since both b and b' are on the same tree, the MBS b remains to be the home MBS for the MT t.

Cost. The UPDATE messages are propagated only to nodes in the delivery path. Thus, the cost of this location update procedure is the diameter of the tree because the new MBS b' needs, at most, that many wireless links to reach the MBS b.

5.3. Case 2: Tranquil home MBS and moving MT are in different trees

1. The MT t, upon moving toward a new MBS b', sends an UPDATE$(000, t, b, r_i)$ message to b'.
2. Upon receiving that update message, b' compares the tree_ID field with its tree_ID, e.g., r_j, and decides that the MT is a visitor which comes from a different home-tree. The MBS b' makes an entry in its directory for the MT t and sends to b TAKEOVER$(001, t, b', r_j)$.
3. MT sends an UPDATE$(101, t, b', r_j)$ message to its old MBS to inform the old MBS that it is in the process of moving.
4. Upon receiving the UPDATE$(101, t, b', r_j)$ message, b marks the mobile and waits for TAKEOVER$(001, t, b', r_j)$ to come from the new MBS b'.
5. ACK-PHASE: Upon receiving the takeover message, old base b sends back HANDOVER$(011, t, b, r_i)$ message to the new base b' and deletes the entry for t in its directory. Note, in this case the home base of the terminal is changed.
6. Upon receiving the HANDOVER$(011, t, b, r_i)$ message, all the MBS in the path from the old base b to the new base b' update their directories accordingly.

Cost. In addition to the update message sent by MT t, each takeover request costs 4 messages, $b' \to r_j \to SC \to r_i \to b$, and each handover ACK costs another 4 messages in a reverse order. Thus, the reliable protocol is expensive. However, it may be argued that the ACK phase can be dropped (as we did in the simulations).

5.4. Case 3: Moving home-MBS and MT in different home-trees

There are three questions related to the mobility of a MBS: (1) how does a MBS decide which tree it belongs to after it moves, (2) what happens to the MTs that it was serving, and (3) if the MBS moves to a different tree than its home-tree, how does it get inserted to the new tree?

In the following we address these issues.

1. The moving MBS b broadcasts an UPDATE(100, NULL, b, r_i) message whenever it suspects that its connection to the associated BFS tree is no longer valid.
2. MT sends an UPDATE$(101, t,$ NULL, NULL) message to its old MBS to inform the old MBS that it is in the process of moving but does not yet know its virtual location (i.e., tree and parent ID).
3. Each adjacent MBS b' on tree rooted at r_j (note that it is possible that $i = j$) that hears this message, broadcasts the UPDATE$(110, b', b, r_j)$ message. Note that the MBS b' copies its ID to the NULL (empty) field in the update message received from b.
4. Upon receiving the UPDATE$(110, b', b, r_j)$ message sent by b', the moving MBS b uniquely identifies that this message is directed to itself (this is important if multiple MBSs change their home trees).
5. Note that the moving MBS b may receive some UPDATE$(110, b', b, r_i)$ messages and some UPDATE$(110, b', b, r_j)$ messages. The question is *how it decides which tree it belongs to*. The moving MBS b makes the decision by applying the *majority function* to these messages. If the numbers of two update messages are the same, then the moving MBS b randomly selects its new home tree.
6. After deciding which tree it belongs to, the moving MBS b chooses one MBS as its parent in the tree rooted at r_j. There are several ways of choosing the parent. The simplest one is to choose a MBS with the minimum ID (among the MBSs which sent the UPDATE$(110, b', b, r_j)$ message). In another approach, the MB can request from the root of the new tree that the MBS to listen in order to maintain the bound on the virtual tree diameter[1]. The moving MBS b further informs its new parent MBS b' of this decision by sending UPDATE$(111, b', b, r_i)$.
7. Upon receiving UPDATE$(111, b', b, r_i)$ message, the MBS b' forwards this message to the root r_j if $i \neq j$. Root r_j informs all MBSs in the tree T_j.
8. Root r_j also informs root r_i via the switching center.
9. Root r_i informs all MBSs on tree T_i about the topology change. (We omit the maintenance of the tree, since it is a well-studied problem.)
10. Regarding the MTs that the moving MBS b used to serve (i.e., the orphans), we assume that these MTs are distributed among the neighboring cells. We consider

[1] In the simulations we took the later approach and attached the mobile MBS closer to the root to obtain minimum diameter.

two approaches for handling the orphans. In the first approach the an orphan is assigned to a new parent. For example, a tranquil MT previously served by b may initiate a location update procedure as explained in case 1. However, the root MBS r_i intercepts the message sent by the new MBS b' to the old MBS b (case 1, step 2) and assigns b' to the new home MBS of that tranquil MT. In the second approach, the orphan determines itself based on the signal strength. Note that a (logical) move of the MBS does not necessarily correspond to a significant physical move. Thus, the MTs may remain connected to the (logically) moving MBS when it moves between locations within the network.

Cost. In this case, the moving MBS b initiates the location update procedure by sending a broadcast message. As we described, it receives $O(\Delta)$ replies from its adjacent MBSs which hears its update message. The moving MBS b then sends a message to its new parent MBS b' which in turn, sends another message to its root, r_j. The root MBS r_j is then required to inform the old root MBS r_i about the departure of the moving MBS b. Finally, all MBSs on both the old and new home trees of the moving MBS b are informed and updated by $O(\Delta)$ messages. Thus, the total number of messages in this case is still $O(\Delta)$.

5.5. Case 4: Moving home MBS and MT in the same new home tree

The location update procedure for this case is very similar to that for case 3. However, since the home MBS and MT move to the same new home tree, the location update procedure will be transparent to MT t. In other words, in case 3, the MTs previously served by the moving MBS may need to initiate a location update procedure as explained in case 1 while it is not necessary in this case.

Note that in all four cases discussed above, the location update procedure never required network-wise broadcast. All the location updates are achieved by modifying the corresponding entries in the distributed databases of the adjacent network entities. Due to this desirable characteristic, the distributed databases converge very quickly in response to network topology changes. In other words, the new hierarchical multihop routing protocol can accommodate this very high mobility.

6. Load balancing using mobile base stations

As the MTs move from one virtual cell to another, the properties of the virtual topology (i.e., the diameter and the size of the BFS trees) may be violated. As a result, the number of MTs that a MBS serves may increase to cause asymmetry in the load distribution. Unbalanced load will cause some base stations to bottleneck and overall system performance will degrade. In the extreme case if the number of MTs in a virtual cell exceeds the constant T, then there will be service blocking.

There are two types of resources allocated to each base station: a fixed amount of bandwidth, and buffers. A packet drop occurs if a node runs out of its resources. Thus, the number of terminals a base station serves impacts the performance of the system.

Let us define an upper and a lower bound on the number m of MTs that a MBS can serve (denoted by *UB* and *LB*, respectively). In order to keep packet dropping below a desired level following inequality must be ensured: $T \geqslant UB \geqslant m \geqslant LB$. A base station is *overloaded* if $T \geqslant m \geqslant UB$, and it is *underloaded* if $m \leqslant LB$. For example, in the simulations (see figures 13 and 14 in section 7), we set $LB = 10$ and $UB = 15$ to ensure a specific bound on the packet dropping.

In this section, we suggest a load balancing algorithm which capitalizes on the *mobility* of the base stations. Our algorithm is based on linking the motion of MBSs to the spatial distribution of the MTs. In other words, we suggest to move additional base stations into the crowded virtual cells so that $T \geqslant UB \geqslant m \geqslant LB$ is maintained across the network. The main constraint is to ensure that no MTs of an underloaded base station will be dropped due the mobility for load balancing. This is necessary to prevent oscillation in the protocol.

The load balancing algorithm used to control the motions of the MBSs is independently layered on top of the routing and network updating protocols. These algorithms are only necessary in systems where the network functions autonomously. We consider two protocols for load balancing of the MBSs: a distributed, and a centralized one.

6.1. The distributed protocol

Let δ denote the distance that an underloaded MBS can move without dropping the MTs that it is serving. Note that this property depends on the location of furthest MT that it is serving. Thus, δ can be determined as a function of physical limitations, e.g., signal power.

In the distributed protocol an overloaded MBS (denoted by b) broadcasts a control message (e.g., SOS signal which contains the ID of this base station in addition to its tree_ID) which can be answered by *any* underloaded MBS that receives it. Upon receiving such a signal an underloaded MBS (denoted by b') estimates if b is within distance δ. We consider two cases: (i) the stronger signal received due to the movement of b', initiates an update procedure, and (ii) a forced update is needed to take over some of the terminals of b. The former case can be handled using the protocols in section 5. The latter one requires exchange of additional control messages between b and b' to identify the terminals to be handed over. Once the terminals identified protocols in section 5 can be used for takeover.

If an underloaded MBS receives multiple SOS signals, then it randomly chooses one target MBS that it can assist within distance δ. As a function of the maximum search area, it is possible to significantly increase the performance of the network as will be shown by the simulations.

6.2. The centralized algorithm

Here the SC must take an active role in directing the motion of the MBSs. The SC maintains a list of all MBSs, their load and (physical) location by using a global positioning system GPS. Periodically, the SC determines overloaded and underloaded MBSs. It runs an algorithm to determine the assignment of underloaded base stations to the overloaded ones as follows.

Let O and U denote the set of overloaded and underloaded MBSs, respectively. We construct a directed bipartite graph $G = (O, U, E)$ with edge set E such that there is a link $e = \langle i, j \rangle$ from $i \in O$ to $j \in U$ if

(1) distance between i and j is at most δ and

(2) the weight $w_{i,j} = m_i - m_j > 0$, where m_i is the number of MT served by MBS i.

The centralized algorithm is iterative and at each iteration it solves maximum weight matching problem on G in time $O(n^3 \log n)$ using the standard techniques. At the end of each iteration k, a new bipartite graph G_{k+1} is obtained for the next iteration as follows:

(i) the edges that are taken by the matching M_k are removed from the graph G_k,

(ii) weights of the remaining edges incident to node $i \in O$ are readjusted by subtracting the weight of the edge that matches i to some node $j \in U$. The algorithm stops when there is no connected component left in the graph.

Let $A_i = \{j \mid \exists M_k$ found by the algorithm s.t. edge $\langle i, j \rangle \in M_k\}$ where $i \in O$ and $j \in U$. Thus, set A_i contains the IDs of the MBs that are assigned to move closer to overloaded node i.

7. Performance evaluation

7.1. Simulation model

This section introduces a simulation study to verify the proposed routing protocols and measure their performance. The performance measures of interest are (i) the percentage of the offered load that was dropped, and (ii) the delay. The network traffic is generated. Each MT creates a message with probability P. The length of each message is determined randomly and uniformly between minimum and maximum allowed length. The messages are then broken down into packets for transmission. The destination for each message is a randomly and uniformly selected among the MT in the network. In the simulations, we maintain a distributed database, pass messages between the nodes, and allow the nodes to move independently. Each node maintains a portion of the database based on the content of the received control message.

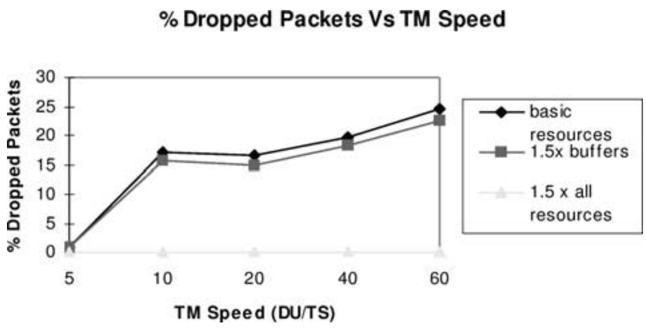

Figure 4. Dropped packet rate versus MT mobility.

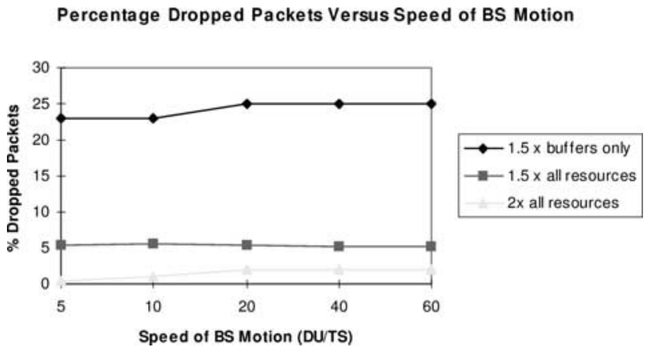

Figure 5. Dropped packet rate versus mobility of the MBSs (limited asymmetry in the network).

7.2. Packet loss as a function of mobility and resource allocation

One of the aims of the simulation study is to capture the relation between mobility and resource increase. Intuitively, as the mobility increases, the amount of resources to support mobility will also increase. Thus, we define *min-set* as the amount of resources (bandwidth and buffers) that each *stationary BS* needs for routing. We use the *min-set* as a basis for comparison for the impacts of changing parameters in the simulation.

We consider a network with 4 VLAs, 28 MBSs and 112 MTs. In figures 4 and 5, we show the percent of packet losses as a function of (i) the mobility of the MTs, and (ii) resource allocation to MBSs. The resources are normalized to those required for the *min-set* case where the MTs and MBSs were all stationary.

MT Mobility. In figure 4, we consider the MT mobility of 60 *Distance Unit/Time Step* (DU/TS) which corresponds to approximately 9,000–10,000 MT moves between MBSs during the 5000 step simulations (i.e., each of the MTs moved 80 times during the simulation).

The main point in figure 4 is that packet dropping can be avoided by providing 1.5 times of the *min-set* resources. In other words, the resource requirements due to mobility is 1.5 times the stationary case.

MBS Mobility. When the mobility of the MBSs is added into the simulation, the conclusions are essentially unchanged. For this set of simulation runs, the MBSs were set

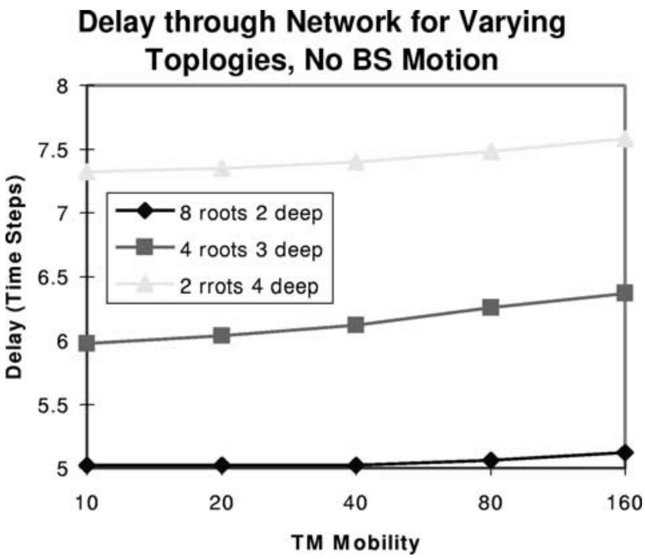

Figure 6. Delay through network as a function of network topology when the MBSs are fixed. MBSs at a particular level in the network were given a fixed amount of bandwidth.

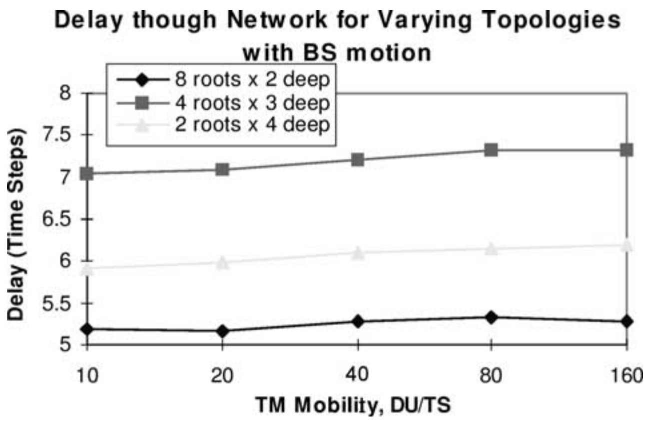

Figure 7. Delay through the network when the MBSs are allowed to move. The motion of the MBSs was set to have a probability of occurring during a time step of 0.5. When the MBSs do move, they move 10 distance units.

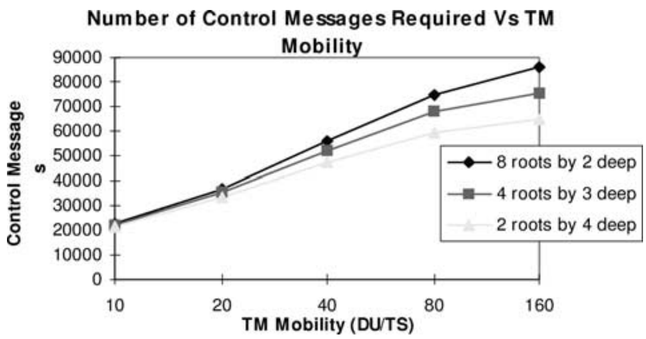

Figure 8. Number of control messages required as a function of the MT mobility. MBS mobility was fixed at a probability of 0.5 of moving 10 DU each time step.

to move so as to minimize the average distance to the MTs that it serves (rather than a pure random motion). Again, the number of MTs that each MBS was allowed to service was limited in order to limit the asymmetry in the network. Figure 5 shows the dropped packet rate as a function of the mobility of the MBSs. In these runs, the mobility of the MTs was fixed at 20 DU/TS. As can be seen from figure 5, the rate of dropped packets was due entirely to the amount of resources that were allocated.

At the 60 DU/TS rate, a MT could make it completely through a MBS's serving area in only 10–12 time steps. This indicates that the rates of motion of the MTs and MBSs in the simulation runs were much higher than might be seen in an actual system. The reason for the high rates in the simulations is to prove the robustness of the protocols. Simulation results shown in figures 7 and 8 illustrate that the underlying protocols are properly delivering messages and updating the network.

7.3. The performance as a function of virtual topology and traffic patterns virtual topology

To study the network topology effects on performance, we utilize a network with 2 VLAs, 64 MBSs and 640 MTs. Initially, each MBS had 10 MTs communicating with it. Within a VLA, the network was setup with three different topologies:

- 2 root MBSs (i.e., 2 trees) each with trees 4 levels deep,
- 4 root MBSs each with trees 3 levels deep,
- 8 root MBSs each with trees 2 levels deep.

Traffic pattern. Two traffic distributions were used in the simulation: (1) homogeneous uniform destination, and (2) heterogeneous uniform destination distribution. In the homogeneous case, the MT randomly chooses a destination for a message from among the entire population of MTs. In the heterogeneous case, a MT generates a message and sends it with probability P1 to another MT which is communicating with its own MBS, with probability P2 that is in its own tree, and with probability P3 with a random MT.

Figure 6 shows the number of hops required for the performance of the three network topologies when the MBSs were restricted from moving. The data rate allocated to each of the MBSs was based on their layer in the network. More resources were allocated to MBSs that are higher in the network. Since the 8 root case had more MBSs higher in the network than the 2 root case, the 8 root case had a larger total data rate available to it in comparison to the 2 root case.

Figure 7 shows the delay through the network as a function of the MT mobility when the MBS moves with probability 0.5 and at a rate of 10 distance units per time step (DU/TS). Note that the performance of the network actually improved when the MBSs were allowed to move for the cases of 4 and 2 roots, respectively. For these cases, as the MBSs move, they tend to reconnect to the network at higher levels, thereby reducing the mean distance a message has to travel to get to the target MBS.

The number of control messages that were required to keep the network routing information up to date for three network topologies is shown in figure 8. We see that the

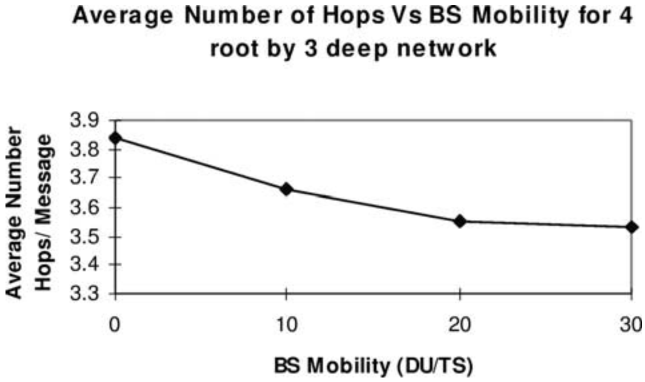

Figure 9. Average number of hops for message delivery as a function of MBS mobility for a network with 4 trees each with diameter 4.

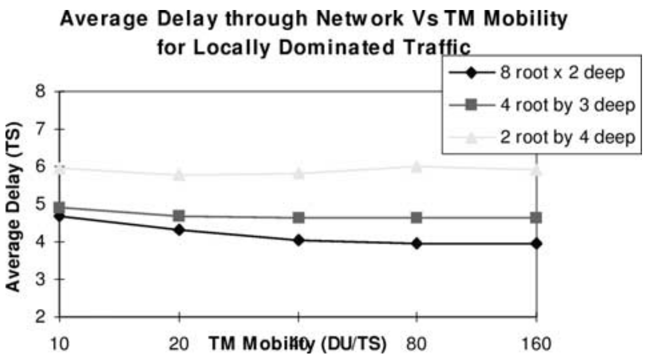

Figure 10. Average delay through the network for the case where the messaging is dominated by local traffic. For this plot, it was assumed that 60% of the traffic would remain within the tree of the sending MT.

2 root topology requires the least number of control messages to remain up to date. In fact, the number of control messages increases with decreasing depth of the trees. However, although the number of control messages decreases, the number of total hops required for the control messages increases. As an example, when the mobility of the MTs was 160 DU/TS, the total number of hops required for the 2 root case (64,800 control messages) was 514,400. However, for the 8 root case (85,800 control messages), it was only 477,000. So, although the number of control messages was less, the total cost (total number of hops) was more.

Figure 9 shows the effect on the average number of hops required to deliver a message as a function of the mobility of the MBSs for the 4 root by 3 deep topology. As shown in figure 2, the MBS motion increases as the average number of hops required to deliver a message decreases. This is due to the maintenance of the virtual topology. As the MBS motion increases, tree depths decrease since the new MBSs are attached closer to the root. The decrease in the number of hops levels off as figure 9 indicates (around 30 DU/TS).

As noted earlier, a second traffic model was also analyzed. For the local traffic model case, it was assumed that of the traffic that the MT generated, 60% would remain in its own tree with the other 40% randomly distributed amongst all of the MTs (including those in its own tree). Figure 10 shows the average delay through the network as a function of

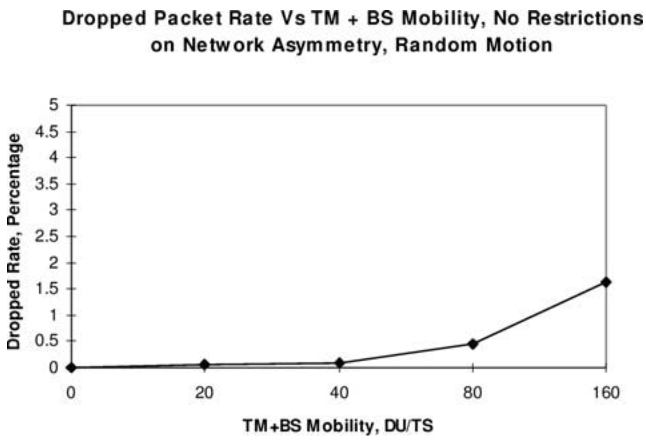

Figure 11. Dropped packet rate vs rate of motion of the MTs with no limit on network asymmetry, random motion of the MTs and MBSs.

the mobility of the MTs (MBS mobility was fixed at $P = 0.5$ of 10 DU/TS motion). The delays through the network for these cases were lower than the homogeneous destination case.

7.4. The performance as a function of load balancing

Here we compare the performance of our load balancing protocols on a network with 2 VLAs, 32 MBSs and 320 MTs. In our simulations, we set $LB = 10$ and $UB = 15$. Figure 11 shows the performance of the network if no load balancing protocol is employed. In figure 11, the dropped packet rate has increased as a result of the dropping of the MBS loading restriction. When the motion of the MTs is random (as was the case for this simulation), the probability indicates that the MTs will remain relatively spread out in the service area. Because of this, the uneven loading on the MBSs is unlikely to become extremely unbalanced. If the conditions of the simulation are changed so that the MTs now have a direction to their motion, the results are dramatically different. To simulate a case where the motion of the MTs are coordinated (as might be the case for military action, for example), 50% of the MTs moved in a directed fashion. That is, they were directed at different times in the simulation to move to specific areas of the service area. As might be expected under such a case, the dropped packet rate sharply increases when there is no attempt to balance the load on the MBSs. To handle these problems, the algorithms presented in section 5 were used to direct the motion of the MBS to help load balance the network. Figure 12 shows the results when the distributed algorithm was used versus no load balancing. The performance of the network with the addition of the distributed algorithm, outperforms the case where no attempt is made to load balance the network.

Upon examining the output of the simulation, it becomes apparent that the dropped packets occur in bursts (clusters). Because the motion of the MBSs is a reaction to the motion of the MTs, the area of the network in which the MTs start to cluster becomes overloaded until more MBSs can arrive.

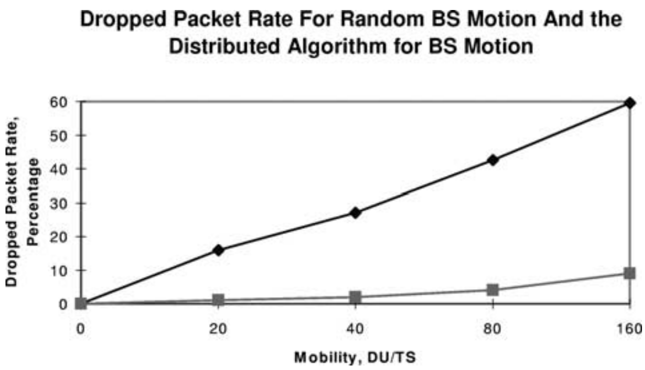

Figure 12. Comparison of the performance of the network for directed MT motion with no restrictions on network asymmetry with random MBS motion and a distributed algorithm to direct the MBS motion. The steeper curve is the case without load balancing.

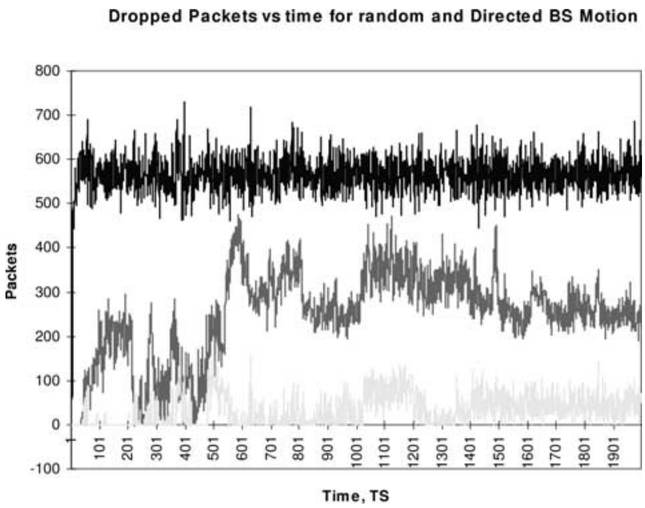

Figure 13. Dropped packets as a function of time for random MBS motion and directed MBS motion. Mobility rate was 80 DU/TS for MBSs and MTs. The top curve is the total packets offered to the network, the middle curve is the lost packet rate with no balancing, the bottom curve is the lost packet rate with the distributed balancing algorithm.

Once the extra MBSs have arrived, the load becomes more balanced and the dropped packet rate is reduced. However, the packet rate does not go back to zero. Figure 13 shows the dropped packets as a function of time for the cases where the MBSs were allowed to move randomly and when they moved to balance the load. The mobility rate for the plot was 80 DU/TS for the MTs and the MBSs.

When a centralized algorithm was used for load balancing, the performance of the network improved further. We used a simpler implementation of the protocol presented in section 6.2 for the centralized load balancing. In this greedy approach a SC chooses one pair of overloaded and underloaded base stations from O, and U sets one at a time for best-fit. Figure 14 shows the lost packets versus time for the centralized approach.

The centralized algorithm outperformed the distributed one, as can be seen from the figures. The distributed algorithm reduced the lost packet rate from approximately 60% (with no load balancing) to less than 9%. The centralized algorithm reduced the lost packet rate down to approximately 6%.

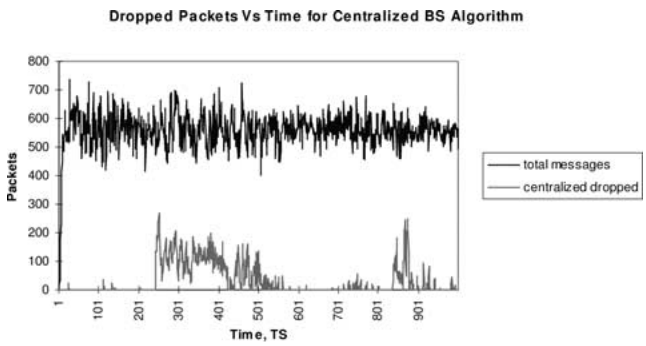

Figure 14. Lost packets versus time for the centralized load balancing algorithm. Conditions were the same as for figure 16 with the distributed algorithm.

8. Conclusions

A new multihop routing algorithm for dynamic wireless networks is introduced in this paper. The routing algorithm operates on a *virtual topology* which is realized by a distributed database for the routing information.

The virtual topology concept partitions the *mobile base stations* (MBSs) of a *virtual location area* (VLA) into multiple virtual trees. A message originated from a source *mobile terminal* (MT) is transmitted to its destination via these virtual trees and possibly the fixed network. The *virtual topology* controls and limits the routing information used by the network entities. The updates required in the routing tables, due to MBS and MT mobility, are not propagated over the entire network and are localized within VLAs. Furthermore, the hierarchical structure of the proposed algorithm effectively shares the load on the multihop links with the fixed network.

References

[1] I.F. Akyildiz, W. Yen and B. Yener, A new hierarchical routing protocol for dynamic wireless networks, in: *Proc. IEEE INFOCOM'97* (April 1997).

[2] M.S. Corson and A. Ephremides, A distributed routing algorithm for mobile wireless networks, Wireless Networks 1 (1995) 61–81.

[3] S.R. Das, R. Castaneda, J. Yan and R. Sengupta, Comparative performance evaluation of routing protocols for mobile ad hoc networks, in: *Proc. of IEEE ICCCN '99* (1999) pp. 153–161.

[4] S.R. Das, C.E. Perkins and E.M. Royer, Performance comparison of two on-demand routing protocols for ad hoc networks, in: *Proc. of IEEE INFOCOM 2000*, Vol. 1 (2000) pp. 3–12.

[5] R.K. Ghosh and G.P. Bhattacharjee, Parallel breadth-first search algorithms for trees and graphs, International Journal of Computer Mathematics 15(3–4) (1984) 255–268.

[6] J. Macker and S. Corson, Mobile ad hoc networks (manet) (1997), `http://wwww.ietf.org/html.charters/manet-charter.html`.

[7] R.D. Mouldin and P.A. Young, Wireless battlefield tactical networking supporting c20tm and c41 for army warrior, in: *Proc. of Milcom'94* (1994).

[8] S. Murthy and J.J. Garcia-Luna-Aceves, A routing protocol for packet radio networks, in: *Proc. of ACM Mobicom'95* (1995).
[9] W. Stallings, *Data Communications*, 5th ed. (McMillan, 1996).
[10] T.J. Wheeler, J.D. Richardson, M. Rosenzweig and D. Cummings, Designing a tactical ATM network integrating performance engineering and design, in: *Proc. of Milcom '94* (1994).

I.F. Akyildiz. Photograph and biography not available at time of publication.
E-mail: ian@ece.gatech.edu

J.I. Pelech. Photograph and biography not available at time of publication.
E-mail: jip@lucent.com

Bülent Yener is a Member of Technical Staff at the Bell Laboratories in Murray Hill, New Jersey. Dr. Yener received B.S. and M.S. degrees in Industrial Engineering from the Technical University of Istanbul, Turkey, and M.S. and Ph.D. degrees in computer science, both from Columbia University, in 1987 and 1994, respectively. His current research interests include reliable and realtime multicasting, quality of service in the IP networks, wireless mobile networks, and the Internet security. He has served on the Technical Program Committee of leading IEEE conferences and workshops. Dr. Yener is a member of the IEEE and DIMACS and serves as an editor for the Computer Networks Journal and the IEEE Network Magazine.
E-mail: yener@research.bell-labs.com