A New Hierarchical Routing Protocol for Dynamic Multihop Wireless Networks

Ian F. Akyildiz * Wei Yen * Bülent Yener, †

Abstract

The routing techniques used in conventional packet radio networks are not suitable for dynamic multihop wireless networks because of their unique architecture. In this paper, a new hierarchical multihop routing algorithm is introduced which balances the cost of location-update and path-finding operations by partitioning the terminals and mobile base stations to produce a virtual topology. *Based on the* virtual topology, *each network entity stores a fraction of the network topology information and maintains the routing efficiency. Finally, the performance of the hierarchical multihop routing algorithm is investigated through simulations.*

Key Words: 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 conventional packet radio networks because (1) they save bandwidth for communications between remote terminals by providing routes through the wired part of the network, and (2) multihop wireless links are used between 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 [4, 7].

Since dynamic wireless network is a relatively new architecture, there are a lot of open research issues to be addressed. One of these issues is the routing problem. In this paper, we introduce a new hierarchical multihop unicast routing protocol. First, we briefly review the related work regarding routing protocols in various network environments.

The routing algorithm used in ARPA and SINCGARS packet radio networks is the distributed version of the Bellman-Ford algorithm (DBF)[1][1]. In the DBF, each node in the network updates its routing table by exchanging routing information with its immediate neighbors periodically. Therefore, a network topology change caused by mobility requires very long time to be noticed by all nodes as the message exchanges "ripple" through the network. In other words, the DBF algorithm fails to accommodate the terminals with high mobility, because it does not converge quickly enough.

Another algorithm commonly used in packet radio networks is the flooding mechanism which creates a loop-free broadcast tree to carry packets from a source to all nodes in the network [6]. In general, the flooding mechanism can accommodate high mobility because network entities are not required to keep up with the network topology changes due to the broadcast nature. However, the broadcast approach used in this algorithm consumes excessive bandwidth which is especially intolerable in a wireless environment.

In [2], a routing algorithm containing three types of control messages is proposed for packet radio networks. This algorithm divides the communication period into two phases, construction and maintenance phases. A multipath route is established by using *query* and *reply* messages during the construction phase. Later, as the network topology changes, the route is updated by using *failure* and *reply* messages during the maintenance phase. Although this algorithm only has local effects when reacting to a topology change in the maintenance phase, it involves all nodes in the construction phase of each communication, even if the source and destination are immediate neighbors. Thus, the message complexity of this algorithm might be too high in some cases. Furthermore, as pointed out in [2], this algorithm cannot accommodate high mobility. The network topology should not change too frequently such that the multipath route obtained during the construction phase remains valid.

In [5], a *wireless routing protocol* (WRP) is introduced for packet radio networks. The WRP uses the concept of *predecessor* to reduce excessive update messages resulting from network topology changes. However, compared to the routing algorithm in [2] and the flooding mechanism [6], the WRP requires a large memory space to store various routing information. In particular, the storage complexity of the WRP increases with the number of nodes in the network. Thus, the WRP might not be suitable for large networks.

In summary, we believe the existing multihop routing algorithms for packet radio networks have significant drawbacks which may not be overcome easily. Besides, the performance of these multihop algorithms when applied in the dynamic wireless networks is unknown. 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 rout-

*Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, {ian,wei}@ee.gatech.edu, Tel:(404)894-5141 Fax:(404)894-0035

†Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102-1982, yener@cis.njit.edu, Tel.:(201) 596-2666

[1]SINCGARS stands for the primary Single Channel Ground and Airborne Radio System which has been widely adapted in the Army and Navy.

ing protocols [2, 5, 6] 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.

The remaining of this paper is organized as follows. In Section 2, we describe the network model of the dynamic wireless network. Then, we explain the proposed hierarchical multihop routing protocol in Section 3. In Section 4, we investigate the performance of the proposed algorithm through simulations. Finally, we conclude this paper and point out possible future research in Section 5.

2 System Description

We assume that the dynamic wireless network consists of (i) a wired backbone network, (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 highly mobile) and (iv) *mobile terminals* (MTs) that can communicate with mobile base stations as shown in Figure 1. More specifically, we have four levels of hierarchy

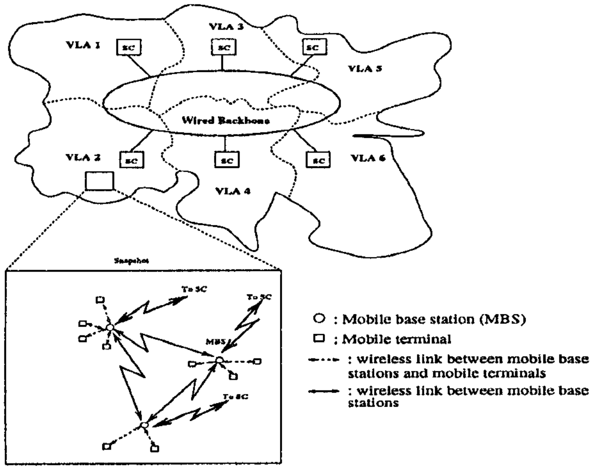

Figure 1: Network Model

in our architecture:

HIERARCHY 1. A wired backbone network represented as an undirected, connected, simple graph G=(V,E) with an arbitrary topology.

HIERARCHY 2. Each switching center is connected to at least one backbone node through physical links. We assume that each backbone node can serve up to S SCs.

HIERARCHY 3. Each SC can support at most B MBSs using wireless connections. Furthermore, two MBSs can communicate with each other without going through 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. Since the scope of this paper concentrates on the routing issues, this assumption is reasonable in the sense that it prevents us from considering physical layer problems.

Next, we define a virtual location area (VLA) as the collection of MBSs which are supported by a single SC. Those MBSs that reside within the same VLA can communicate with a *virtual topology* as we will explain in Section 3. The topology in this level is determined by choice and it is called virtual since there is no physical link between them but instead a routing table entry between connected MBSs.

HIERARCHY 4. Each MBS can support at most T mobile terminals (MTs). 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 same virtual cell. We assume that *each MT is supported by exactly one MBS at any time instance.*

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 concept of the new hierarchical multihop unicast routing protocol.

3 The New Multihop Routing Protocol

In dynamic wireless networks, routing cannot be achieved if we fail to (i) locate the destination MT or to (ii) find a feasible path between the source and the destination. In other words, we need a location update and path finding procedures to address these two problems. In this section, we will demonstrate a distributed database structure (Sections 3.1 and 3.2) which allows us to perform the location update (Section 3.3) and path finding (Section 3.4) procedures effectively.

3.1 Partitioning the MBSs

As will be shown in Section 3.2, the distributed database structure requires partitioning of the MBSs within a VLA into disjoint *Breadth First Search* (BFS) trees [3] with diameter at most D and size at most Δ, where $\Delta \geq D$. Each tree is rooted at one unique mobile base station. In brief, if a BFS algorithm wants to add a node to a existing tree, then it will select from those nodes which do not increase the diameter of the existing tree. In other words, if the situation allows the BFS algorithm tends to maintain the maximum distance (hop count) from the root to leaves on the tree. Note that the tree induces a virtual topology since an edge between two bases in the tree corresponds to a directory entry in each end point. In Section 3.4, we show that the virtual topology of the BFS trees allows us to determine the routing cost.

We present the following procedure to partition the MBSs in a VLA:

1. Initially, all the MBSs are unmarked.

2. While there is an unmarked MBS, perform the following steps.

   2.1 Choose a root MBS b from the unmarked MBSs to build a new BFS tree.

   2.2 Expend the BFS tree by adding an unmarked MBS adjacent to that BFS tree.

   2.3 Repeat step 2.2 until the tree is saturated, i.e., at least one of the following conditions is violated.

   - *Condition 1.* The size of the BFS tree cannot exceed Δ.
   - *Condition 2.* The diameter of the BFS tree cannot exceed D.

   2.4 Mark all the MBSs in this tree and go to step 2.

Note that each VLA only invokes the partition algorithm once during the network setup. Therefore, the overhead incurred by this partition algorithm is negligible.

3.2 Distributed Database for Location Tracking

In this section, we demonstrate a distributed database which maintains the location information of the partitioned MBSs and MTs as shown in Figure 2. This distributed database achieves low storage complexity by only requiring involved network entities to store partial topology of the entire network. Furthermore, when combining with the location update and path finding procedures, this distributed database architecture provides a very attractive routing solution to the dynamic wireless network. In the following, we specify the involved network entities and the information they maintain in detail.

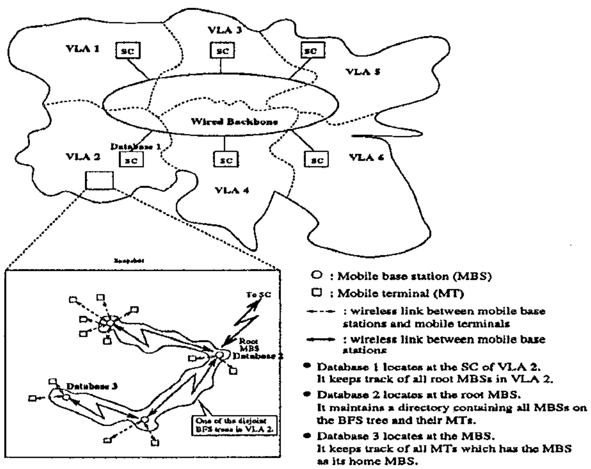

Figure 2: Trees, Directories etc

1. Each switching center (SC) maintains an entry only for the root MBSs in its VLA. The directory size is at most B/D, where B and D the maximum number of MBSs in a VLA and the maximum diameter of a BFS tree, respectively.
2. 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.
3. Each root MBS (selected at step 2 in the partition algorithm) of a BFS tree knows all the MTs covered by the MBSs on this tree. Thus, each root MBS has a directory in the size of $O(T\Delta + \Delta)$.
4. Each MBS in a BFS tree knows the topology of that tree which requires $O(\Delta)$ entries. Thus, the total directory size in an MBS (except the root MBS) is $O(\Delta + T)$ where T is the maximum number of MTs allowed in a virtual cell.

Given the distributed database architecture, we can calculate the total directory size, or storage complexity, in a BFS tree is $O((T\Delta + \Delta - 1) + \Delta - 1(\Delta - 1 + T)) = O(\Delta^2 + T\Delta)$. In addition, the total directory size in a VLA is $O(B/D(1 + \Delta^2 + T\Delta))$.

3.3 Location Update Procedure

The location update procedure is invoked to let the SCs and MBSs to detect and record any change occurring in the virtual topology, e.g., VLAs and virtual cells. In particular, suppose that t is an MT which has its home-MBS b in the home-tree T_i whose root MBS is denoted by r_i. Let b' be the new MBS of the MT t at the time of update. In the following, we consider several possible cases. Note that the proposed algorithm, in its current state, prohibits an MBS or MT leave its own VLA. This is a realistic assumption in various applications. For example, in a digital battlefield scenario [4, 7], a VLA may be a large area. The MBS and MT, corresponding to armored vehicles and foot soldiers, usually stay within their respective VLA.

- *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.

  In Cases 3 and 4 we consider the mobility of the MBS.
- *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.

As explained shortly, the location update procedure responds differently to each one of these cases. However, before we discuss the exact steps how the location update procedure reacts in these different cases, we will determine the size of the update message exchanged among network entities at the update time. An *UPDATE* message is a quadruple: $< type, terminal\_ID, home-MBS\_ID, tree\_ID >$. 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[2].

$$\zeta = 3 + logT + logB + \log(B/D). \quad (1)$$

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

3.3.1 Case 1: Tranquil Home MBS and Moving MT are in the Same Home-Tree

1. Upon moving to a new MBS (i.e., detecting a stronger MBS) b', the MT t sends an $UPDATE(000, t, b, r_i)$ message to b'. The type field of that update message indicates that it is originated from an MT.

[2] We assume that there are, at most, 8 types of update messages. In other words, the type field needs 3 bits to represent the update message types.

2. Upon receiving that update message, the new MBS b' informs the home MBS b of the MT t.

3. Since both b and b' are on the same tree, the MBS b remains to be the home MBS for the MT t.

Cost: Note that 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.

3.3.2 Case 2: Tranquil Home MBS and Moving MT are in Different Trees

1. The MT t, upon moving to a new MBS b', sends an $UPDATE(000, t, b, r_i)$ message to b'.

2. Upon receiving that update message, the MBS 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.

3. The MBS b' requests to ***takeover*** this visiting MT t using the following **reliable** protocol with explicit ACKnowledgment.

   3.1 b' sets the type field to 001 and forwards the $UPDATE(001, t, b, r_i)$ message to r_j. This type field indicates that this is a takeover request from new base b'.

   3.2 Root r_j forwards this message to the SC since it does not maintain routing information for the other trees.

   3.3 SC S forwards the message to the tree rooted at r_i.

   3.4 Root r_i informs the home MBS b.

   3.5 The MBS b *handover* MT t by sending back $UPDATE(011, t, b, r_i)$ message to the new base b' and deletes the entry for t in its directory.

4. Upon receiving the $UPDATE(011, t, b, r_i)$ message, root r_i, root r_j and 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 hand-over 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.

3.3.3 Case 3: Moving Home-MBS and MT in Different Home-Trees

Since our architecture permits MBSs to change their locations, here and also in the next sections we consider the mobility of an MBS. There are important questions which need to be answered first:

- How does an MBS decide which tree it belongs to after it moves?
- What happens to the MTs that it was serving?
- 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. 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.

3. 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).

4. 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. In case that the numbers of two update messages are the same, the moving MBS b randomly selects its new home tree.

5. After deciding which tree it belongs to, the moving MBS b chooses the MBS with the minimum ID (among the MBSs which sent the $UPDATE(110, b', b, r_j)$ message) as its parent in the tree rooted at r_j. The moving MBS b further informs its new parent MBS b' of this decision by sending $UPDATE(111, b', b, r_i)$.

6. 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.

7. Root r_j also informs root r_i via the switching center.

8. 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.)

9. Regarding those MTs that the moving MBS b used to serve, we assume that those MTs are distributed among the neighboring cells. Thus, the root MBS r_i which knows the global view of the tree T_i does it in a centralized way. For example, a tranquil MT used to be 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' the new home MBS of that tranquil MT.

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. Then the moving MBS b 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, then, is 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 is informed and updated by $O(\Delta)$ messages. Thus, total number of messages in this case is still $O(\Delta)$.

3.3.4 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 used to be 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 requires 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 database converges very quickly in response to network topology changes. In other words, the new hierarchical multihop routing protocol can accommodate the very high mobility.

3.4 The Path Finding Procedure

In this section, we present a novel path finding procedure which distributes the complexity and overhead among the MBSs and SCs. In particular, we consider the following four cases.

3.4.1 Case 1: Source and Destination are in the Same Tree

In this case, there are two possibilities: source and destination both reside in (i) the same cell, or (ii) in different cells. Since the first one is trivial, we elaborate the second one. Since the root MBS of a BFS tree knows all the directories of the MBSs on this tree, the MBS of the source routes the packet to the root MBS which in turn forwards it to the MBS of the destination on the tree.
Cost: The routing length will be bounded by twice of the depth of the tree, or, $O(D)$.

3.4.2 Case 2: Source and Destination are in Different Trees but in the Same VLA

Since the information kept in each tree is local to the members of that tree, inter-tree routing operations are performed via the switching center 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 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.

3. The information kept in an SC is only for the root MBSs of the BFS trees, thus that SC does not know the location of the destination MT. Consequently, the SC broadcasts the packet to all the root MBSs in its VLA. Assuming that broadcast is a unicast to each tree, the cost of this broadcast is B/D messages, since there are, at most, that many roots MBS.

4. Only one root MBS will forward the packet to the destination MBS which serves the destination of the packet. This is true since each root MBS knows all MTs residing in this tree.

Cost: In this case, we need two multihop routes[3] and a broadcast between an SC and the root MBSs in its VLA. Therefore, the message complexity is $O(B/D)$ which is primarily contributed by the broadcast overhead.

3.4.3 Case 3: 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 detailed path finding procedure for this case is given in the following.

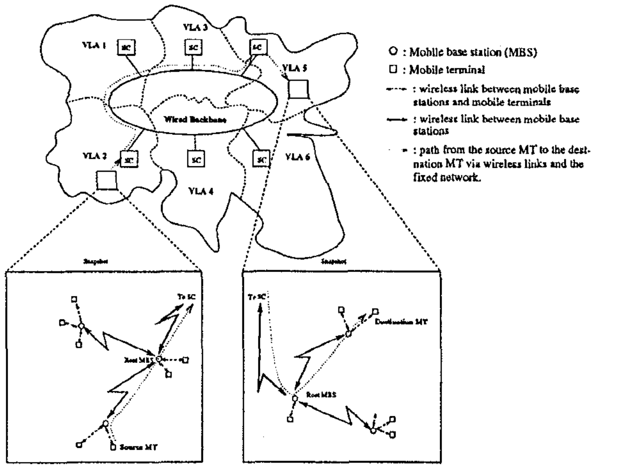

Figure 3: Source and Destination in Different VLAs

The packet from the sources gets routed over the wired-backbone. Upon receiving the packet, an SC performs the following operations.

1. The SC forwards the packet to all the root MBSs in its VLA.

2. Since each root MBS knows the MTs and their bases in its tree, only one root forwards the packet down to the destination tree and, then, to the destination base.

[3](source → source root MBS) and (destination root MBS → destination).

Cost: The cost of this path finding procedure is equal to the sum of that in Case 2 and the cost of routing on the fixed backbone network.

4 Performance Evaluation

Here we evaluate the performance of the proposed algorithm through simulations. In particular, we, first, describe our simulation model including the network topology generation, mobility model, and traffic model. Then we demonstrate and discuss the simulation results.

4.1 Simulation Model

In the simulation, a square-shaped area is divided into nxn unit squares where n is an input parameter and could be any positive integer. We place an SC in the center of each unit square so that the small square becomes the VLA of the SC[4]. Within each VLA, we generate a set of MBSs and MTs based on a uniform distribution. These MBSs and MTs have their own mobility parameters, denoted by m_{MBS} and m_{MT}, respectively. In each time unit, we first determine how many MBSs and MTs move according to their mobility parameters and then generate their movement vectors. A movement vector has x and y elements derived from a uniform distribution between $[0, 1)$. If a moving MBS or MT obtains a movement vector leading outside the VLA from one side of the unit square, we assume that the MBS or MT comes in from the opposite side of the VLA as shown in Figure 4. In this example, a node moves across a VLA boundary, i.e., both its x and y coordinates are larger than 1. Then, according to our mobility model, we reduce both coordinates by 1 so that the new location of the node falls within the VLA area.

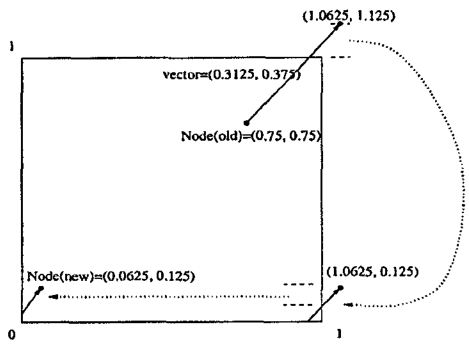

Figure 4: Movement Leading Outside the VLA

In our simulation, we do not assume any particular underlying MAC (medium access control) protocol as well as no specification on the wireless link capacity. Instead, we assume that within each time unit an MBS can transmit a message, if it desires. Thus, we believe that the simulation results will not loose their significance when various network models are considered.

Since an MBS residing at a VLA can directly communicate with the SC of that VLA, we know that the VC of the MBS has a radius of, at least, $\frac{\sqrt{2}}{2}$. Therefore, there exists a wireless link between two MBSs as long as their distance is less than the radius of their VCs, or $\frac{\sqrt{2}}{2}$ in our simulation. However, the MBSs within a VLA are divided into multiple trees using the BFS algorithm introduced in Section 3. An MBS records only its wireless links in the BFS tree. In the beginning of the simulation, an MT may reside in the overlapped region of several VCs. However, the MT selects the nearest MBS as its home MBS. Once the MT selects its home MBS, this MT is considered as being at the same VC until it moves outside of the VC.

[4]In practice, a VLA may have an irregular shape and the location of an SC is not necessarily at the center of its VLA.

Messages generated in the simulation can be divided into two categories, the control and data messages. The control message results from the MBS or MT mobility as described in Section 3. Hence, the control message generation is implied by the mobility model. On the other hand, the data message is generated by the source MT. In our simulation, each MT has probability p to generate a data message destined to an arbitrary MT for each time unit. Both the control and data messages are transferred to their destinations by the MBSs in the network. We assume each MBS has an infinite buffer to accommodate incoming messages and can transmit only one message in each time unit. Furthermore, each MBS will not serve the data messages before it finishes serving all control messages in its queue.

Due to the mobility, an MBS may be isolated because i) it does not have any wireless link to the other MBSs, or ii) it cannot join a BFS tree without violating the size and depth restrictions stated in the BFS algorithm. If an MBS is isolated, it drops all messages it receives. Similarly, an isolated MT can neither receive nor transmit data messages.

4.2 Simulation Results

Here, we start with a single-VLA scenario where all messages are generated by and destined to the MTs in the VLA. In other words, in the single-VLA scenario, only the wireless links are used which either connect the SC and root MBSs or connect the MBSs in the BFS trees. The performance we obtain from the single-VLA scenario is equivalent to a more general nxn-VLA case, if intra-VLA traffic is considered. This feature is a direct result of our proposed routing algorithm, i.e., limiting the mobility of the MBS and MT to their respective VLA. Note that the inter-VLA traffic in the nxn-VLA case affects the average routing cost performance which is also evaluated and will be presented as what in follows.

In the single-VLA scenario, we use the following simulation parameters. We generate 40 MBSs and 200 MTs. Then, we divide these MBSs into several BFS trees according to the algorithm introduced in Section 3.1. We find for each MT its home MBS. The diameter D of these BFS trees is 2 while the size Δ of BFS trees is an input variable. The mobility parameters of the MBS and MT are 1 and 2 per time unit, respectively. The message arrival probability p is another input variable ranging between [0.005, 0.03] message/MT/time unit. Note that the p value could depend on various factors such as the MAC protocols and the link capacity.

In Figure 5 (a), we show the throughput performance λ which is obtained by

$$\lambda = p\,\frac{\alpha}{\beta}. \qquad (2)$$

where α is the number of messages received by their destinations during the simulation time while β is the number of messages generated by the MTs/MBSs during the simulation time.

By inspecting Figure 5 (a), we note that i) the throughput decreases as the offered load, controlled by the parameter p, increases, and ii) the throughput decreases as the size of the BFS trees increases. The first result is obvious but the latter needs an explanation. In our proposed routing algorithm, the root MBSs are responsible for all messages generated by and destined to the MTs stored in its directory. An increasing BFS tree size means that the root MBSs have to maintain more MTs in its directory and handle their according messages. Consequently, the root MBS quickly becomes a bottleneck as the offered load increases. Therefore, we suggest to use a smaller BFS tree size, e.g. $\Delta = 5$ to avoid possible congestion at root MBSs. Furthermore, all three throughput curves in Figure 5 will decrease and, finally, approach zero with increasing p.

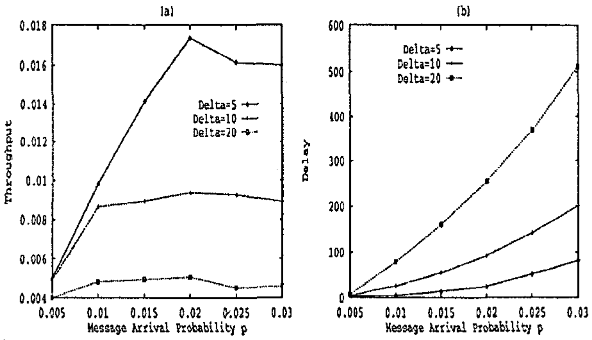

Figure 5: Single-VLA Scenario (40 MBSs, 200 MTs, $D = 2$, MBS mobility=1/time unit, MT mobility=2/time unit) (a)Throughput; (b)Delay

In Figure 5 (b), we show the delay results for three different BFS tree sizes. The delay value takes the average of the delay experienced by all messages transmitted successfully during the simulation period. As expected, the delay performance becomes worse when the BFS tree size increases reflecting the degraded throughput performance.

By assuming that each MBS has a small buffer size for only two messages and using the same simulation parameters as in Figure 5, we show the percentage μ, defined in equation (3), of dropped messages in Figure 6.

$$\mu = \frac{\gamma}{\beta} \qquad (3)$$

where γ is the number of dropped messages and β is the number of messages generated by MBSs/MTs during the simulation time.

It can be observed in Figure 6 that our proposed routing protocol does not discard messages when the offered load is low. However, as the offered load increases, the number of dropped messages of the curve with $\Delta = 20$ increases drastically. The curves with $\Delta = 10$, 5 also increase as the offered load keeps increasing. The three curves in Figure 6 gradually approach 100%, i.e., the root MBS becomes a bottleneck, especially for larger BFS trees.

In Figure 7, we show the average number of MTs which move from their currently residing VC (virtual cell) to another VC in each time unit dependent on the

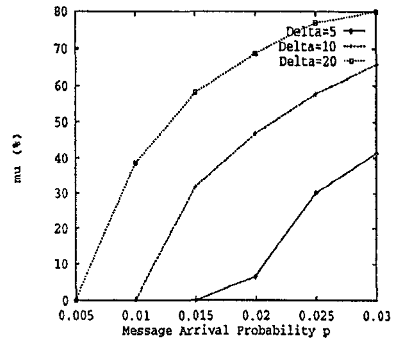

Figure 6: Percentage of Dropped Messages in the Single-VLA Scenario (40 MBSs, 200 MTs, $D = 2$, MBS mobility=1/time unit, MT mobility=2/time unit)

mobility parameter. The average number represents the overhead of control messages caused by MTs changing VCs. The three curves in Figure 7 have MBS mobility m_{MBS} equal to 0, 5, and 10, respectively, while the other simulation parameters remain the same as in the previous experiments. The curve with $m_{MBS} = 0$ (tranquil MBS scenario) increases linearly to the MT mobility m_{MT}. This behavior is expected because, as m_{MT} increases, more MTs can move and, hence, can increase the possibility that they leave their according VCs. In the cases where the curves with m_{MBS} not equal to zero, a tranquil MBS may change to a different VC because the MBS of its currently residing VC moves. In other words, given the same m_{MT}, the curves with higher m_{MBS} values increase faster than those curves with smaller m_{MBS} values. However, as the m_{MT} keeps increasing, the three curves gradually converge to the same value because all MTs move in each time unit regardless of whether their corresponding MBSs are tranquil or moving.

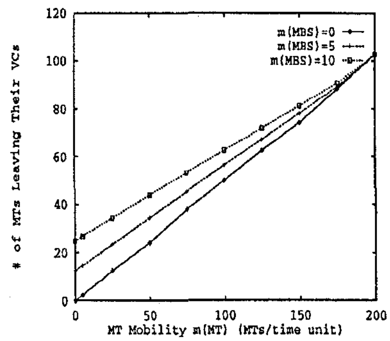

Figure 7: Percentage of Dropped Messages in the Single-VLA Scenario (40 MBSs, 200 MTs, $D = 2$, MBS mobility=1/time unit, MT mobility=2/time unit)

Note that nxn-VLA scenario has same performance behavior as the single VLA scenario. However, the routing costs in nxn-VLA scenario become a significant component. Therefore, we want to compare the average routing costs of our proposed routing protocol and the flooding mechanism[5], denoted by $c_{r,p}$ and $c_{r,f}$, respectively.

First, we define

- c_1 is the cost for an MBS to send a message,
- c_2 is the cost to broadcast a message to all root MBSs via the fixed network in the proposed protocol,
- c_3 is the cost to broadcast a message to all MBSs and MTs using the flooding mechanism, and
- c_4 is the cost of using a wireless link between a MBS and a MT.

In the nxn-VLA scenario, we have

$$c_{r,p} \leq 2(D\ c_1 + c_4) + c_2 \qquad (4)$$

where D is the maximum diameter of a BFS tree. In the worst case, the source and destination MTs are in the leaves in VCs (virtual cell) of two BFS trees which belong to two different VLAs. If the flooding mechanism [6] is used, then we

$$c_{r,f} = c_3 \leq M\ (c_1 + c_4) \qquad (5)$$

where M is the number of MBSs in the network. In the worst case of the flooding mechanism, all MBSs transmit the message once. For example, for a network with a virtual bus topology where $\max c_{r,p} \leq \max c_{r,f}$, then we obtain[6]

$$c_2 \ <\ (M - 2D)c_1 + (M-2)c_4. \qquad (6)$$

Equation (6) must be satisfied to apply our proposed routing protocol. In fact, given M = 160 MBSs distributed in 2x2 VLAs with $D = 2$, c_1=5 , and $c_4 = 1$, the bound for c_2 is 938 according to equation (6) which, we believe, is far larger than the actual c_2 value in the real world dynamic wireless networks. In Table 1, we show our simulation results of the average routing costs for a 2x2-VLA dynamic network with various c_1 and c_2 values while c_4 is set to 1. Each VLA has 40 MBSs and 200 MTs with BFS trees of $D = 2$ and $\Delta = 5$.

In the last column of Table 1, we include the maximum (worst case) value of $c_{r,f}$ for comparison. According to our simulation, the proposed routing protocol achieves one or two orders better routing costs than those obtained by the flooding mechanism.

	$c_2 = 1$	$c_2 = 2$	$c_2 = 5$	$c_2 = 10$	$\max c_{r,f}$
$c_1 = 1$	5.708	6.496	8.85	12.851	320
$c_1 = 2$	8.627	9.418	11.777	15.724	480
$c_1 = 5$	17.375	18.176	20.488	24.474	960
$c_1 = 10$	32.003	32.754	35.175	39.092	1760

Table 1: Average Routing Costs with Various c_1 and c_2 Values

[5]We use the basic flooding mechanism for the worst case comparison.

[6]Equation (6) is derived from equations (4,5) and $c_{r,p} \leq c_{r,f}$.

5 Conclusion and Future Work

We introduced a new multihop routing algorithm for dynamic wireless networks in this paper. This algorithm partitions the MBSs (*mobile base stations*) of a VLA (*virtual location area*) into multiple virtual trees. A message originated from a source MT (*mobile terminal*) is transmitted to its destination via those virtual trees and, sometimes, the fixed network.

Using the concept of the *virtual topology*, network entities store a small fraction of the network topology information. Therefore, if an MBS or MT moves, only a few of involved network entities have to update their directories. The inexpensive location update allows the dynamic wireless network to actually route messages rather than blindly broadcast them. Furthermore, the hierarchical structure of the proposed algorithm effectively shares the load on the multihop links with the fixed network. Roughly speaking, the traffic between remote source MT and destination MT pair involves the fixed network.

Regarding the future research, we plan to relax the constraint stated in Section 3.3 which requires MBSs and MTs to stay in their VLAs. We also plan to improve the survivability of the proposed algorithm by creating a backup scheme for the root MBS which stores the critical information for its BFS (*breadth first search*) tree. In addition, we will experiment with another virtual topology to remove the bottleneck at the root MBS produced by the virtual tree topology.

References

[1] R. E. Bellman, "Dynamic Programming," *Princeton University Press*, Princeton, N.J., 1957.

[2] M. S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM/Baltzer Journal of Wireless Networks*, Vol. 1, pp. 61-81, 1995.

[3] R. K. Ghosh and G. P. Bhattacharjee, "Parallel Breadth-First Search Algorithms for Trees and Graphs," *International Journal of Computer Mathematics*, Vol. 15, No. 3-4, pp. 255-268, Jul. 1984.

[4] R. B. Mouldin and P. A. Young, "Wireless Battlefield Tactical Networking Supporting C2OTM and C4I for the Army Warrior," *Proc. of Milcom '94 Conference*, 1994.

[5] S. Murthy, J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks," *Proc. of ACM Mobicom '95 (Mobile Computing and Networking Conference)*, Berkeley California, Nov. 13-15, Nov. 1995.

[6] W. Stallings, *Data Communications*, Fourth Edition, McMillan Publ. Co., 1995.

[7] T. J. Wheeler, J. D. Richardson, M. Rosenzweig, and D. Cummings, "Designing a Tactical ATM Network Integrating Performance Engineering and Design," *Proc. of Milcom '94 Conference*, 1994.