

# Design and Evaluation of a Metropolitan Area Multitier Wireless Ad Hoc Network Architecture

Jorjeta G. Jetcheva Yih-Chun Hu  
Carnegie Mellon University  
{jorjeta,yihchun}@cs.cmu.edu

Santashil PalChaudhuri Amit Kumar Saha David B. Johnson  
Rice University  
{santa,amsaha,dbj}@cs.rice.edu

## Abstract

*Few real-world applications of mobile ad hoc networks have been developed or deployed outside the military environment, and no traces of actual node movement in a real ad hoc network have been available. In this paper, we propose a novel commercial application of ad hoc networking, we describe and evaluate a multitier ad hoc network architecture and routing protocol for this system, and we document a new source of real mobility traces to support detailed simulation of ad hoc network applications on a large scale. The proposed system, which we call Ad Hoc City, is a multitier wireless ad hoc network routing architecture for general-purpose wide-area communication. The backbone network in this architecture is itself also a mobile multihop network, composed of wireless devices mounted on mobile fleets such as city buses or delivery vehicles. We evaluate our proposed design through simulation based on traces of the actual movement of the fleet of city buses in the Seattle, Washington metropolitan area, on their normal routes providing passenger bus service throughout the city.*

## 1. Introduction

In an ad hoc network, mobile nodes self-organize to form a network without the need for infrastructure such as base stations or access points. Each mobile node acts as a router, forwarding packets on behalf of other nodes, creating “multihop” paths that allow nodes beyond direct wireless transmission range of each other to communicate. Routing protocols for ad hoc networks must discover such paths and maintain connectivity when links between nodes in these paths break due to factors such as node motion or wireless propagation and interference changes.

Few real-world applications of mobile ad hoc networks have been developed or deployed to date outside the military environment, and as a result, no traces of actual node movement in a real ad hoc network have been available for ad hoc network routing protocol simulation studies. The vast majority of performance evaluations of ad hoc network routing protocols have used randomized node mobility models (e.g., [1–3, 12, 13, 22, 23, 27, 34]).

In this paper, we make three primary contributions:

- We propose a novel commercial application of ad hoc networking, which we call *Ad Hoc City*. Ad Hoc City is a scalable, multitier ad hoc network architecture, consisting of a small number of wired base stations plus a mobile multihop wireless backbone, serving mobile users over a metropolitan area; the backbone nodes are implemented on mobile fleets, such as city buses and delivery vehicles, which already cover the area of a city in both space and time.
- We document a new source of real node mobility traces to support large-scale realistic simulations of ad hoc network routing protocols. We acquired several week-long traces of the movement of the fleet of city buses in Seattle, Washington, on their normal routes providing passenger bus service throughout the city.
- We describe a routing protocol for the Ad Hoc City architecture, which we evaluate using these real mobility traces which include 750 to 850 mobile nodes. Our routing protocol, called C-DSR (Cellular DSR), is an extension of the Dynamic Source Routing protocol (DSR) [23].

The original DSR protocol is an *on-demand* routing protocol for ad hoc networks that has been shown through earlier simulations to perform well [2], but DSR was not designed for the hierarchical, multitier architecture of the Ad Hoc City system and has not been used with networks of the size used in our simulations based on the city bus fleet of Seattle. We have modified the operation of DSR to support the scalable on-demand routing needed in our system. C-DSR is consistent with the design philosophy of DSR and operates *entirely* on-demand.

We simulate the performance of the Ad Hoc City network mobile backbone with C-DSR using our movement traces of the Seattle city buses. In our simulations, the buses represent mobile nodes forming the wireless mobile backbone of the Ad Hoc City architecture. The number of network nodes, including mobile nodes (buses) and 8 base stations, present in our simulations varies between 750 and 850. This network provides wireless service to an area of approximately 5000 square kilometers (2000 square miles). Our simulations show that the system performs well; C-DSR successfully delivers over 96% of the originated data packets, with average delivery latency of around 86 ms.

Section 2 of this paper describes related work. In Section 3, we present an overview of our Ad Hoc City

---

This work was supported in part by NSF under grant CCR-0209204, by NASA under grant NAG3-2534, and by a gift from Schlumberger to Rice University. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of NSF, NASA, Schlumberger, Rice University, Carnegie Mellon University, or the U.S. Government or any of its agencies.

network architecture. Section 4 summarizes the standard operation of the Dynamic Source Routing protocol (DSR), and Section 5 presents the design of C-DSR. In Section 6, we describe the real movement traces that we collected to enable evaluation of Ad Hoc City. We discuss our evaluation methodology in Section 7 and results in Section 8. Finally, we conclude with Section 9.

## 2. Related Work

Qiao et al. [29] presented an architecture for enhancing cellular networks called iCar, in which wireless relay stations are placed on the borders between cells and are used to improve the load balancing of the traffic among the cells and to decrease call blocking. Hsieh et al. [15] also proposed a system for enhancing a cellular network with ad hoc network routing, in which nodes use ad hoc routing to reach the base station along multiple hops and switch to cellular operation when the bandwidth available in ad hoc mode is lower than that achievable in cellular mode. Unlike these proposals, the system we present in this paper does not use the traditional cellular protocols but instead utilizes an ad hoc network backbone itself composed of mobile wireless nodes.

There have been several proposals for a hybrid cellular and ad hoc networking infrastructure in which nodes within a cell use ad hoc network routing to reach the base station responsible for the cell [14, 16, 17, 25]. These proposals focus on the design and performance of the hybrid system within a single base station cell, however, and do not discuss the routing mechanisms for roaming between different cells.

A number of approaches have been proposed for connecting an ad hoc network to the Internet (e.g., [26, 31–33]). For example, Maltz et al [26] described an implementation connecting an ad hoc network running the Dynamic Source Routing protocol (DSR) [23] to the Internet, using Mobile IP. Their approach allows for roaming of nodes between different ad hoc network clouds and between an ad hoc network cloud and the Internet, and uses subnetting to distinguish between nodes in different ad hoc network clouds. This addressing scheme is more suitable when mobile nodes roam between ad hoc network clouds only occasionally and spend most of their time in their “home cell.” In this paper, on the other hand, we assume that nodes may roam between different cells on a regular basis, and that they do not have a “home cell.” In addition, though our architecture supports communication between an ad hoc network and the Internet, our emphasis is on on-demand routing within the ad hoc network.

The most commonly used mobility models proposed for use in simulations of routing in ad hoc networks are based on random mobility patterns (e.g., [1–3, 12, 13, 22, 23, 27, 34]). These models rely on probability distributions to determine the location, speed, and direction of each mobile

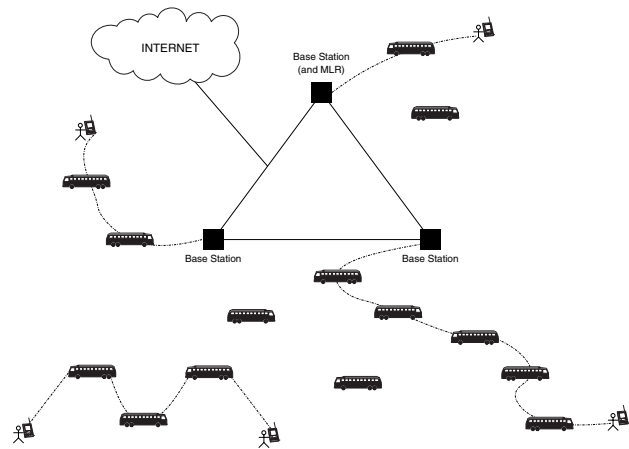


Figure 1: Example of the Ad Hoc City Architecture

node or groups of nodes in the network. Some models of vehicular motion have also been proposed in the literature to model the movement of cars on highways based on *driver behavior models* (e.g., [4, 10]). Though such movement models are important for routing protocol performance analysis, they may not accurately represent real mobility of individual nodes. The movement scenarios presented in this paper, on the other hand, are based on real traces of the motion of buses on their regular routes in Seattle, Washington. These traces allow us observe protocol performance in a more realistic setting.

## 3. Ad Hoc City Overview

Ad Hoc City is a multitier wireless ad hoc network routing architecture for general-purpose wide-area communication. The backbone network in our architecture is composed of wireless devices mounted on mobile fleets, such as the city buses or delivery vehicles within a city. These vehicles cover the area of the city in both space and time, and can be organized in a multihop wireless ad hoc network backbone that provides network access and general communication services throughout the city. In addition, in the Ad Hoc City architecture, several fixed base stations, with which nodes can communicate over multihop paths, are sited throughout the metropolitan area; these base stations improve ad hoc network routing scalability and provide Internet connectivity. Individual mobile users within the city can communicate over this architecture using devices such as laptops, PDAs, or future cellular phones, to connect to the Internet or to communicate with each other. An example of the Ad Hoc City architecture is illustrated in Figure 1, with the routes shown between several mobile users.

By using mobile fleets as a network backbone, deployment of the Ad Hoc City infrastructure is greatly simplified. For example, no fixed sites for these backbone nodes need to be purchased or leased; maintenance is simplified, since the vehicles naturally return to a base or headquarters

as part of their normal route, thus obviating the need for a technician to be dispatched to a remote site and avoiding the need for equipment to access places that are difficult to reach such as poll tops. In addition, the effect of physical failure of any backbone node is lessened, as it is averaged over the area of the vehicle's route, and other vehicles in the mobile fleet will naturally fill in most coverage gaps caused by the failure. The only fixed part of the network infrastructure is a small number of base station nodes, and due to the multihop routing capability of the ad hoc network mobile backbone nodes, placement of the fixed base station nodes is significantly simplified over traditional architectures such as cellular systems; exact placement, for example involving topographical surveys, is not necessary. The use of ad hoc network routing contributes to the robustness and adaptiveness of the system relative to traditional cellular systems, as the routing protocol is able to adapt to changes in network topology and node failures, as well as to route around congested areas of the network.

#### 4. DSR Overview

The Dynamic Source Routing protocol (DSR) [23] for wireless ad hoc networks operates entirely *on-demand*. DSR transmits routing packets only when a node tries to send packets to another node for which it does not have a route in its Route Cache, or when a route that a node is using breaks while the node is actively using this route to forward packets. As a result, DSR only reacts to topology changes that affect routes that are actively being used, and only incurs overhead in response to actual communication demands in the network.

DSR does not create any forwarding state in the network; instead, the complete sequence of hops along the path to a destination is stored as a *source route* in the routing header of each packet. Source routing provides three main benefits: packets easily always follow a loop-free path, forwarding behavior is extremely simple, and each node forwarding or overhearing the packet can harvest routing information from the packet's source route.

DSR employs two basic mechanisms, *Route Discovery* and *Route Maintenance*. To discover a route to the destination of a new packet for which the sender does not currently have a route, the sender initiates Route Discovery; to detect that a route being used for a packet no longer works, Route Maintenance is used. During Route Discovery, in its simplest form, the source node floods a ROUTE REQUEST packet. When the REQUEST reaches the destination for which the route is needed, the destination returns the discovered route back to the source in a ROUTE REPLY packet, along the reverse path traversed by the REQUEST. The returned route is then used by the source node to source route subsequent data packets to the destination. Route Maintenance uses hop-by-hop (or passive) acknowledgements for data packets in order to detect that a link along

the route to the destination is broken. When a node along the route detects a broken link, it sends a ROUTE ERROR to the data source, indicating the broken link to remove from its Route Cache.

The DSR specification includes a number of optimizations to the protocol in addition to the basic mechanisms described here. An extension to DSR to eliminate the per-packet overhead of source routing has also been proposed [19]. Incorporating this extension into C-DSR is a topic of future work.

#### 5. Network Architecture Design

Ad Hoc City is a multitier ad hoc network architecture that enables general-purpose wide-area communication services, such as in a metropolitan area, as shown in Figure 1. There are two types of mobile nodes in the complete Ad Hoc City architecture: *personal mobile nodes* such as laptops and PDAs, which are owned by end-users, and *network mobile nodes*, which are mounted on vehicles in mobile fleets and form the mobile backbone of the networking infrastructure. The network thus forms a hierarchical, multitier structure. Network mobile nodes route packets between personal mobile nodes within the Ad Hoc City network, and between personal mobile nodes and the Internet, through a small set of fixed base stations.

Each base station serves the mobile nodes within a *topological cell* around the base station. The coverage of this cell is determined not by the wireless transmission range of the base station but by a distance in wireless hops from it; that is, all nodes within some number of wireless hops of the base station use that base station as a point of connection to the Internet and to nodes in other topological cells within the Ad Hoc City network. Each network mobile node in turn serves a varying set of *personal mobile nodes* that are currently located within direct transmission range of it. Each packet is routed along a possibly multihop route between network mobile nodes to the base station nearest the source node; this base station then routes the packet into the Internet or to the base station closest to the destination mobile node, which then forwards the packet to the destination. If the source and destination nodes are both within the Ad Hoc City network, when it is more efficient to communicate directly rather than through a base station, packets between the source and destination are multihop routed between the network mobile node closest to the source and the network mobile node closest to the destination, bypassing any base stations.

In our design, personal mobile nodes do not participate in data packet forwarding, as they typically would have limited energy resources and since such forwarding would complicate network management in terms of, for example, security and topology control.

Personal mobile nodes may use a protocol such as Mobile IP [28] to associate with a network mobile node.

Since mechanisms for providing this association between personal mobile nodes and network mobile nodes have already been extensively explored in the literature (Section 2), we focus on the design of the mobile backbone of our network; in particular, we focus here on the protocol allowing a network mobile node to communicate with other network mobile nodes and with base stations, as it moves within a topological cell or between cells. In the rest of this paper, we use the term *mobile node* to refer to a network mobile node.

We do not discuss security for Ad Hoc City in this paper. A proposal to secure DSR is presented in [20] and can be applied to our architecture with some modifications. In addition, a scheme for charging nodes for network usage in a multihop cellular network running a source routing protocol is discussed in [30] and may also be used in the context of Ad Hoc City.

### 5.1. Mobile Location Register

One of the base station nodes, or possibly a dedicated machine, called the *Mobile Location Register*, keeps track of the location of each network mobile node, i.e., which base station the node is associated with. This information is indexed by mobile node address and is kept in a *registration table* at the Mobile Location Register. Each base station keeps a *registration table cache* in which it caches registration table entries so that it does not have to consult the Mobile Location Register each time it needs to forward a packet.

In our simulations, we assume a reliable, wired link with unlimited bandwidth between each pair of base stations and also to the Mobile Location Register. Though our protocol does not depend on in-order or reliable packet delivery for correct operation, a mechanism for retransmitting packets lost over such links may be necessary for improved performance in a real implementation.

In our design, data can originate from a mobile node within the Ad Hoc City network, or from an Internet node. We assume that the Mobile Location Register can determine whether a destination is in the ad hoc network or in the Internet, and as a result Route Discovery is not performed for destinations that are not within the ad hoc network. For example, if the ad hoc network were allocated a CIDR block [11], the Mobile Location Register could use the address prefix to decide whether or not a node is in the ad hoc network.

### 5.2. Location Registration

Our extensions to the DSR protocol are entirely on-demand, as routing control functions are performed only when there is a need to discover a communication path between a pair of nodes. The locations of mobile nodes

are discovered through monitoring of data packets and control packets sent as part of the Route Discovery and Route Maintenance procedures initiated by C-DSR.

A network mobile node may associate with a base station when the base station hears a ROUTE REQUEST originated by that node, or when the node replies to a ROUTE REQUEST from a Route Discovery initiated by the base station. If no entry in the registration table exists for a node when another node attempts to send packets to it, the Mobile Location Register initiates *paging* to discover the location of the destination node and a route to it. Paging involves performing Route Discovery from one or more base stations in an attempt to discover a route to a mobile node, as described in Section 5.7.

Each time a base station hears a ROUTE REQUEST packet initiated by a mobile node, it forwards it to the Mobile Location Register. To ensure that the freshest information is kept in the registration table, the registration entry is updated based on both the hop count and the ROUTE REQUEST identifier: more recent ROUTE REQUEST identifiers always take precedence, and between two ROUTE REQUESTS with the same identifier value, the one with the shorter hop count takes precedence. If the source mobile node of this ROUTE REQUEST is not currently registered in the registration table at the Mobile Location Register, or if the new ROUTE REQUEST takes precedence over the current registration for that mobile node, then the Mobile Location Register updates its registration table to record this base station as the current base station serving that mobile node.

In order to also register the target node of a ROUTE REQUEST initiated by a base station (e.g., as in paging), the target includes an additional field in its ROUTE REPLY to carry what would have been that node's next ROUTE REQUEST identifier. This identifier allows the ROUTE REPLY to be ordered relative to ROUTE REQUESTS initiated by the mobile node; the Mobile Location Register is thus able to choose the freshest information for registration; stale ROUTE REPLYs do not modify the registration table, but fresh REPLYs can override existing registrations.

The Mobile Location Register can also use the source route of ROUTE REQUESTS and ROUTE REPLYs to create *implicit registrations* for the nodes listed in it. In particular, each REQUEST and REPLY contains a source route ending at some base station; the Mobile Location Register can implicitly register each node in that source route with that base station. These implicit registrations *cannot* be used to update the registration table entries, as their freshness cannot be determined, but they can be cached and used to optimize paging, as described in Section 5.7.

As mentioned in Section 5.1, each base station keeps a cache of registration table entries that it has used recently. When a base station receives a data packet for for-

warding and has no cached registration table entry for the destination of the packet, it queries the Mobile Location Register in order to update its cache and to determine how to route the packet. This procedure is described more fully in Sections 5.4 and 5.5.

### 5.3. Route Discovery

When a mobile node **S** wishes to send a data packet to a destination **D**, it checks its Route Cache for a route either to **D** or to any base station. A route to a base station will only be selected if that route is of length less than or equal to  $h_b$ , where  $h_b$  is a network-wide parameter that represents the maximum number of hops that a node can be away from a base station. This value depends on the placement of base stations in the network, as described in Section 7.1.

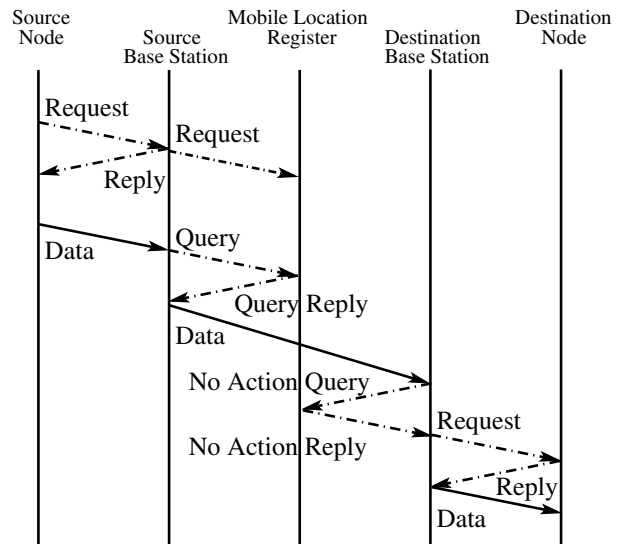
If the mobile node does not have a route to the destination in its Route Cache or the route is not acceptable because its length exceeds  $h_b$ , the node buffers the data packet and initiates a Route Discovery by flooding a hop-limited ROUTE REQUEST packet for the destination. The hop-limit in this ROUTE REQUEST packet is set to the maximum number of hops a node can be from a base station,  $h_b$ . As in the base DSR protocol (Section 4), non-duplicate copies of the ROUTE REQUEST packet are forwarded by each node within the specified hop limit.

When a base station receives a ROUTE REQUEST for a target node **D**, it reverses the list of nodes accumulated in the source route of the ROUTE REQUEST packet and includes this list in the header of a ROUTE REPLY packet that it sends back to the source of the REQUEST along this reverse path. In addition, the base station forwards the ROUTE REQUEST to the Mobile Location Register for registration purposes, as described in Section 5.2.

The destination node **D** may be within the hop-limit of the ROUTE REQUEST flood, in which case **D** will also receive the REQUEST packet. As in the base DSR protocol, in this case **D** will return a ROUTE REPLY in response to the REQUEST.

As in the base DSR protocol, if the source node does not receive a ROUTE REPLY for its Route Discovery, it will continue to repeat the Route Discovery, with an exponentially increasing interval between Discovery attempts, up to a maximum, or until all packets it has buffered to this destination time out.

When a source node **S** sending a packet to a destination node **D** has a route in its Route Cache to either a base station or to **D** within hop limit  $h_b$ , then **S** sends the packet for **D** along the best route in its Cache. Given a choice between more than one base station route, the sender **S** will choose the base station that is the fewest number of hops away; given a choice between a base station route and a direct route to **D**, then **S** will choose the shorter route, and if the routes are the same length, it will choose the direct route.



**Figure 2:** Route Discovery and base station processing when the destination is still in the cell of the base station recorded in the Mobile Location Register's registration table. This figure shows the control packet exchange among a data source, the base stations, and the Mobile Location Register, when the source has a packet to send to a destination but does not have a route to either the destination or any base station in its Route Cache. The dotted lines indicate messages sent only when necessary. For example, a Route Discovery will not be initiated if the appropriate route is in the node's Route Cache; instead, the data will be forwarded immediately.

When sending a data packet directly to the destination, the source can place a source route in the packet, as in the base DSR protocol. When sending a data packet to a destination using a base station route, the source adds the (source) base station address to the end of the source route it includes in the packet's header. The Route Discovery procedure as well as the processing of data packets at the source and destination base stations (Section 5.4 and 5.5) are illustrated in Figure 2.

### 5.4. Data Packet Processing at the Source Base Station

When a base station receives a data packet forwarded to it by a mobile node (as opposed to by another base station), it checks its registration table cache for an entry for the destination of the packet. If such an entry exists, the base station removes the source route from the data packet and forwards the packet to the destination base station. If the source base station does not have an entry for the destination node, it buffers the packet, creates an empty entry for that destination, and queries the Mobile Location Register for the base station associated with the destination of the packet. Packets received while the base station is waiting for a response from the Mobile Location Register, indicated by an existing but empty registration table cache entry, are buffered by the base station.

The base station queries the Mobile Location Register by sending it a QUERY packet. When the Mobile Location Register receives a QUERY, it checks its registration table for an entry for the destination node specified in the QUERY. If an entry exists, the Mobile Location Register returns a QUERY REPLY with the address of the destination base station in whose cell the destination node was last registered. If no such entry exists, the Mobile Location Register initiates paging for the destination, as described in Section 5.7. When a base station receives a QUERY REPLY, it updates its registration table cache entry for the destination in the QUERY REPLY, then checks if it has any packets buffered for the destination specified in the QUERY REPLY; if so, it forwards the buffered data packets to the destination base station specified in the QUERY REPLY. Figure 2 shows the processing steps at a source base station.

If the source base station and the destination base station are the same, i.e., the source and the destination are in the same topological cell, the source base station will be listed in the registration table cache as the destination base station for that destination node. No special processing is required in this case, as the source base station assumes the role of a destination base station, and performs the processing described in Section 5.5.

### 5.5. Data Packet Processing at the Destination Base Station

When the source base station sends a packet to another base station, as described in Section 5.4, the receiving base station, which we call the *destination base station*, checks its Route Cache for a route to the destination of the packet. If the destination base station finds such a route in its Cache, it places this route in the source route of the packet and sends the packet along that route. Otherwise, it buffers the packet and sends a NO ACTION QUERY to the Mobile Location Register.

The Mobile Location Register does not initiate paging in response to a NO ACTION QUERY. The purpose of this QUERY is to allow a base station that no longer has a route to the destination to check whether or not it should initiate a Route Discovery for this mobile node, or whether that node has already registered elsewhere.

When the Mobile Location Register receives a NO ACTION QUERY, it checks its registration table for an entry for the destination node specified in the QUERY. If the Mobile Location Register has such an entry, it returns a NO ACTION REPLY specifying the base station currently associated with the destination. If no entry for the destination exists, the Mobile Location Register instead returns a NO ENTRY REPLY; the Mobile Location Register may not have an entry for a destination node even when such an entry was cached at the source base station because the Mobile Location Register may have rebooted recently.

When the base station receives a NO ACTION REPLY from the Mobile Location Register indicating that the mobile node is now associated with a different base station, it caches the new registration for the destination in its registration table cache, and then returns the buffered packets for this destination to the source base station from which it received them. Similarly, when the base station receives a NO ENTRY REPLY from the Mobile Location Register, it returns all buffered packets for that destination back to the base station from which it received them.

Before a packet is returned to the source base station in this way, however, a “Stale Flag” is set in the header of the packet, allowing the receiving base station (i.e., the packet’s source base station) to identify it as a stale packet. When a base station  $BS_1$  receives a packet from some other base station  $BS_2$  with the Stale Flag set, then  $BS_1$  checks if its registration table cache entry for the destination of the packet points to  $BS_2$  as the destination base station. If so,  $BS_1$  removes that entry from its registration table cache. Then  $BS_1$  clears the Stale Flag and processes the packet as if it was just received from a mobile node inside its topological cell, as described in Section 5.4.

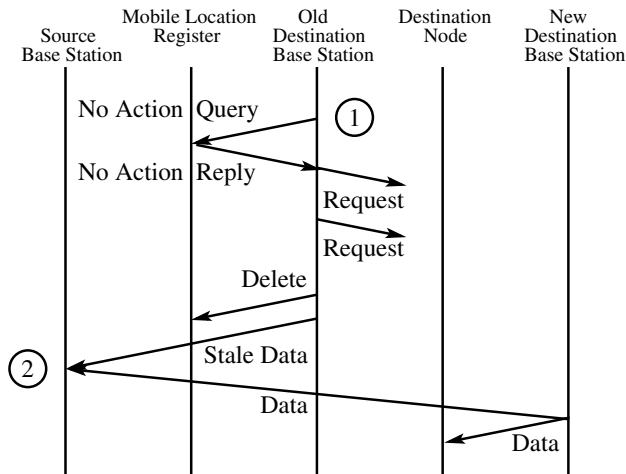
If, after sending a NO ACTION QUERY, a base station receives a NO ACTION REPLY indicating that it is the current base station for the destination node, it concludes that the mobile node is still in its topological cell, and initiates Base Station Local Route Repair, as described in Section 5.6.

### 5.6. Base Station Local Route Repair

When a base station receives a NO ACTION REPLY from the Mobile Location Register indicating that it is the destination base station for some destination node for which it does not currently have a route, it performs a local route repair procedure; the base station also performs local route repair when, while forwarding a packet to some destination, the base station receives a ROUTE ERROR for that destination. While a base station is performing local route repair for a destination, it buffers any new packet it receives for that destination.

In local route repair, a base station performs a Route Discovery for that destination in its topological cell, and reattempts that Route Discovery if necessary at exponentially increasing intervals, up to some maximum number of Discovery attempts. Each Route Discovery flood is hop limited to the size of the cell,  $h_b$ . If the destination base station receives a ROUTE REPLY in response to its Route Discovery, it updates its Route Cache, and forwards the buffered data packets along the new route to the destination.

If the destination base station does not receive a ROUTE REPLY after the maximum number of Route Discoveries has been attempted, the base station returns the packets to the source base station, setting the “Stale Flag” in the header of each packet as described in Section 5.5.



**Figure 3:** Base station processing when the destination is no longer in the cell recorded in the Mobile Location Register's registration table. At point (1), at the destination base station, a data packet arrives from another base station, and the base station does not have a route to the packet's destination. At point (2), the source base station receives stale data packets from the old base station; it queries the Mobile Location Register for the latest registration information for the destination. The Mobile Location Register may initiate paging, if a DELETE caused it to delete its entry for this mobile node; otherwise, it responds with the address of the current base station with which the destination is associated, allowing the source base station to send its buffered data packets to the new destination base station.

The base station also sends a DELETE packet to the Mobile Location Register, indicating that the destination is not in its topological cell. When the Mobile Location Register receives a DELETE packet, it checks its registration table to ensure that the DELETE was received from the base station with which the destination is currently registered. If the message is indeed from the current base station, the Mobile Location Register deletes the entry for that destination from its registration table; otherwise it ignores the DELETE, as it must be stale. The base station local route repair procedure is illustrated in Figure 3.

When a local repair fails, or when a source base station sends packets to a base station with which the destination is no longer registered (Section 5.5), the packets are returned to the source base station rather than being forwarded to a new destination base station. This mechanism was designed for simplicity and correctness purposes as it ensures that only one base station is responsible for discovering the destination base station for packets sent by a mobile node in its cell, and that routing loops which may otherwise result from stale registration information would not occur. In particular, packets may get passed from base station to base station in a cycle as a result of such stale information.

## 5.7. Paging Mechanism

Paging is used to locate a destination mobile node when it is not currently registered with any base station. The Mobile Location Register can implement any paging policy using the information in its registration table, including the implicit registration information described in Section 5.2. In particular, any optimization used in traditional cellular systems can also be used in the Ad Hoc City architecture.

We use a simple policy to improve paging based on implicit registration information. If the Mobile Location Register has no existing registration, including implicit registration, for the destination specified in a QUERY packet, the Mobile Location Register sends a PAGE packet to all base stations; each base station then initiates a Route Discovery for the target destination node, indicating in the ROUTE REQUEST that this REQUEST is a page. Otherwise, if the Mobile Location Register has an existing implicit registration for the destination, it first sends a PAGE only to the base station with which the destination was last implicitly registered. That base station then initiates a Route Discovery page as described above. If the Mobile Location Register does not receive a new registration for the destination within a timeout, it sends a PAGE to all base stations. If the Mobile Location Register does not receive a new registration for the destination within a further timeout, it continues to page all base stations, up to MAX\_PAGES times, with an exponentially increasing timeout between each new attempt.

## 6. Bus Movement Traces

To evaluate the Ad Hoc City architecture, we used traces of the actual movement of buses in the Seattle, Washington area King County Metro bus system. We chose the King County bus system because the data was available on-line, and because bus movement patterns closely match other vehicular traffic patterns; Seattle also provides a topographically challenging routing environment, created by a 35 square mile lake in the middle of the city (Figure 6).

The King County bus system is composed of over 1200 vehicles covering a 5100 square kilometer area. An Automated Vehicle Location (AVL) system tracks each bus using a combination of odometry and signpost transmitters [7]. Internet users can monitor the location of each bus in real-time using the Busview software [6–8]. We developed mechanisms for recording the data being delivered to Busview clients, and for converting the recorded data into movement patterns suitable for use in the ns-2 network simulator [9], which we use to evaluate our system (Section 7).

Figure 4(a) shows the number of buses in the traces we collected over a two week period starting on Saturday, November 17, 2001. The number of buses has a very predictable day-of-the-week pattern: on weekdays, the number of buses reaches about 1200, three times as many as

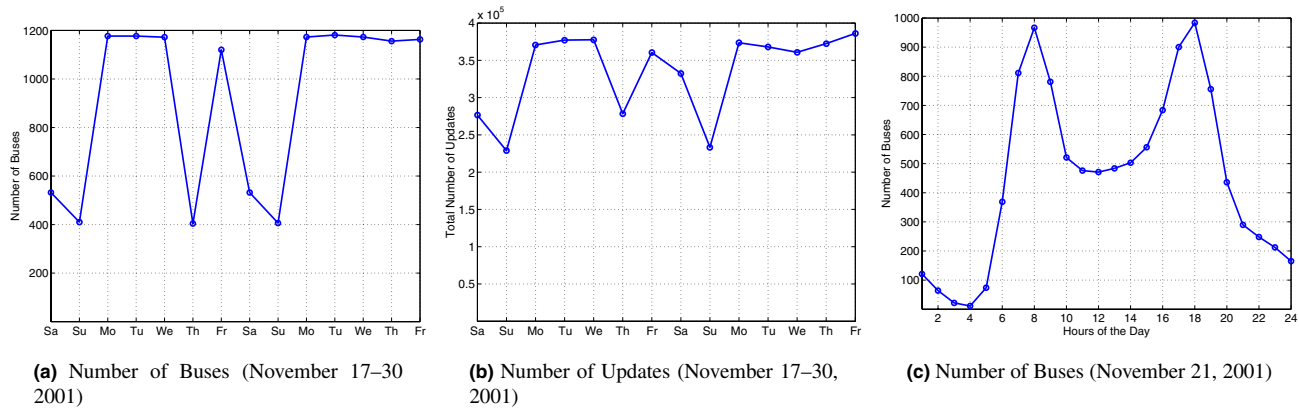


Figure 4: Bus Trace Characteristics

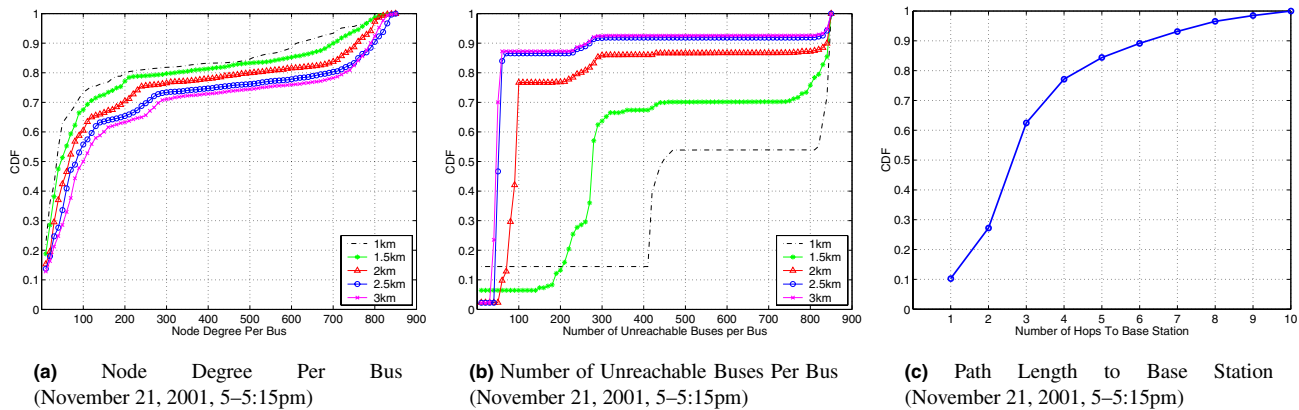


Figure 5: Movement Scenario Routing Characteristics

on Sundays and holidays and slightly more than twice as many as on Saturdays. The dip on Thursday, November 22 corresponds to Thanksgiving Day, on which buses ran their Sunday schedules.

The traces are based on location update messages sent by each bus; the number of updates that are sent on each day, shown in Figure 4(b), follows a similar weekly pattern as the number of buses. Between 230,000 and 380,000 updates are sent per day, which represents between 317 and 525 updates per bus per day.

The bus movement traces also show a very pronounced diurnal cycle, shown in Figure 4(c), with clearly defined morning and evening rush hour spikes; the number of buses during rush hour is higher than the number of buses during other daytime periods by up to a factor of 2. Topology control algorithms could be used to exploit this known and clearly defined cycle to increase or decrease the transmission range of the nodes as necessary to maintain network connectivity. For example, such an algorithm could reduce transmission range during rush hour to increase total network throughput, and increase range at other times to maintain connectivity.

Per-bus inter-update times on a weekday between a rush hour period and a non-rush hour period are similar, with about 70% of the updates being 12 minutes apart or less, and about 90% being 20 minutes apart or less. The speeds reported by the buses at rush hour and non-rush hour time are also similar, with 90% of the average speeds being less than 10 miles an hour and 97% being less than 20 miles an hour.

To better understand the routing characteristics of our scenarios, we analyzed the topology created by the bus movement traces. As nodes in the network are assigned longer wireless transmission ranges, the node degree of each bus increases, as expected (Figure 5(a)). In our traces, when the wireless range is 3 km, median node degree is 100, whereas at a range of 1 km, median node degree is 30. Though relatively low node degree is desirable for improved scalability, a 1 km-range results in substantial network partitions (Figure 5(b)). To balance between network connectivity and scalability, in our simulations we chose a range of 1.5 km (Sections 7 and 8). We then chose a link bandwidth of 2 Mbps, which is consistent with bandwidth utilization in CDMA2000 [24]. We use a fixed nominal transmission



range for simplicity; a node in an actual deployment could dynamically choose a wireless range based on its location, congestion level, base station affiliation, and time.

## 7. Evaluation Methodology

To evaluate the Ad Hoc City architecture and C-DSR protocol, we simulated it using the *ns-2* network simulator [9] with the Monarch wireless extensions [2]. The simulator incorporates models of signal strength, radio propagation, wireless medium contention, capture effect, and arbitrary continuous node mobility. The standard radio model in *ns-2* is based on the Lucent Technologies WaveLAN 802.11 product [21], which provides a 2 Mbps transmission rate and a nominal transmission range of 250 m. As described in Section 6, we modified the physical layer from the base *ns-2* distribution to increase the nominal range from 250 meters to 1.5 km. Our implementation of C-DSR is based on the version of DSR that uses the *Link-MaxLife* Route Cache caching strategy [18].

In our simulations evaluating Ad Hoc City and C-DSR, we used six distinct 15-minute segments from our bus movement traces from Monday, November 19, 2001, 7–8 am and Wednesday, November 21, 2001, 5–6 pm. The number of buses in these movement scenarios varies between 750 and 850. Running longer simulations was not possible due to resource limitations.

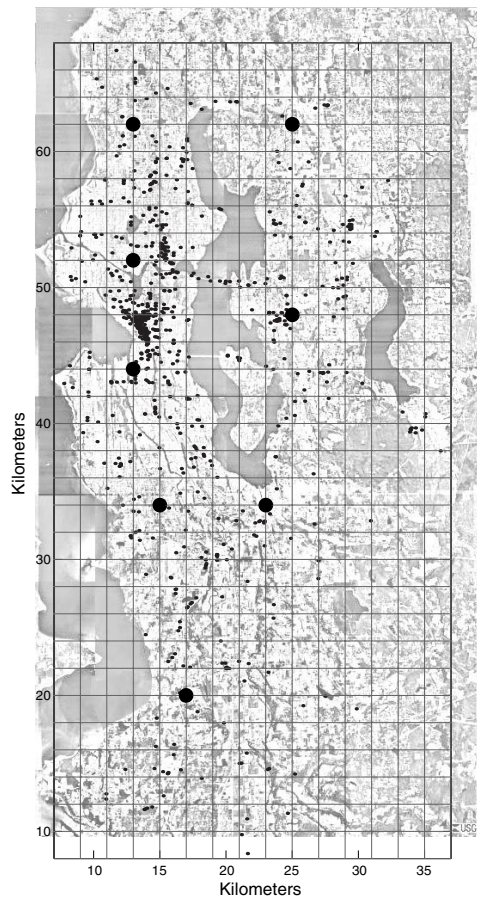
### 7.1. Base Station Placement

To improve the scalability of our system and to enable Internet connectivity, we added a small number of fixed base stations. Base station placement in traditional cellular networks is a long, painstaking, and expensive process in which an optimal location is chosen for each base station, using signal strength measurements and cellular usage prediction techniques. In our Ad Hoc City architecture, such precision in placing the base stations is not necessary; the ad hoc network is not as sensitive to propagation conditions and obstacles, because the routing protocol adapts to the dynamic topology of the network and is able to use multihop routes.

In our experiments, we chose the base station locations to satisfy the following criteria:

- Each mobile node should generally be within a specified maximum number of wireless hops,  $h_b$ , from a base station. We chose this maximum number of hops in our simulations to be 8.
- Areas with high node density are served by more base stations in order to improve scalability.

We placed a total of 8 base stations covering the Seattle area. The locations of these 8 base stations are shown as large black dots in Figure 6, which is an approximate superposition of the base station positions on a slightly modified USGS aerial photograph of Seattle from Microsoft



**Figure 6:** Locations of the 8 Ad Hoc City base stations in Seattle, Washington in our simulations. The smaller marked points show an example snapshot of the locations of the buses at one point in the bus movement traces.

TerraServer [5]. The smaller marked points show the current position of each bus in the city at one specific time, to illustrate the typical distribution of buses within the city and around each base station. The coverage of a base station is the set of mobile nodes located within the specified  $h_b$  maximum number of wireless hops (e.g., 8) from that base station.

Without the use of base stations, with a transmission range of 1.5 km, the maximum path length is up to 48 wireless hops. With the 8 base stations added to the scenario, the maximum path length is at most 20 wireless hops (Figure 5(c)). In addition, 90% of the nodes are within 6 hops of a base station, yielding a maximum path length of 12 wireless hops. By imposing a limit of 8 wireless hops as a maximum distance to a base station we exclude a small number of nodes that are sometimes not within any base station's topological cell. Had we not imposed this limitation, these nodes would often be partitioned, since they are in a very sparse area of the network, such as the location 30 km east and 18 km north of the origin in Figure 6.

The number of buses in each topological cell varies based on the density of the cells. With the current choice of base station locations and topological cell size, the median cell population is 50 buses, and 7 of the cells contain 250 buses or less. Over a period of 15 minutes, 58% of the buses cross over into neighboring cells; 90% of the buses that cross over into other cells perform fewer than 7 crossovers, and 90% of the transitions between cells are within 150 seconds of each other. These fast transitions indicate that most crossovers are caused by buses moving along the boundary between a pair of neighboring cells.

## 7.2. Communication Patterns

Each of our communication patterns consists of 200 unique flows, each of which generates four 64-byte packets per second. Flows arrive following a Poisson distribution at a load of 20 Erlang (i.e., there are 20 simultaneous flows on average), with a fixed flow holding time of 90 seconds. We chose this model because it represents communication patterns typical of small wireless devices, such as cellular phones and personal digital assistants (PDAs). Each pair of communicating nodes was chosen randomly across all connected pairs of mobile nodes.

## 7.3. Performance Metrics

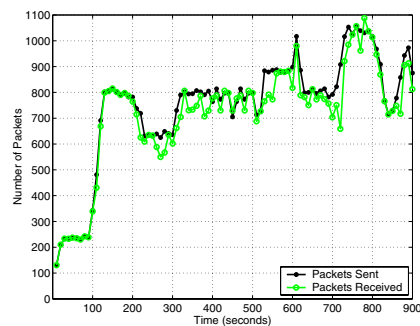
To evaluate the performance of the Ad Hoc City architecture and C-DSR, we used four metrics: packet delivery ratio, packet overhead, packet latency, and path length.

*Packet Delivery Ratio*, or PDR, is the fraction of data packets originated by the application layer on a source node that are received by the application layer at the corresponding destination node. *Packet Overhead* is the number of control packets sent by the routing protocol. For each successfully delivered packet, we compute *Path Length* and *Packet Latency*. Path Length is the number of times a packet was transmitted over the wireless medium. Packet Latency is the time it takes for a packet to be delivered to its destination once it is generated by the source application.

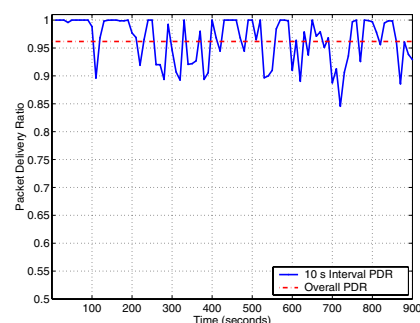
## 8. Results

We performed 6 simulation runs, described in Section 7, each for 900 seconds of simulated time. In these scenarios, packet delivery ratio (PDR) ranged between 92% and 97%, median latency was 42.52 ms, and average overhead was 222 overhead packet transmissions per network node. Because the results for our six simulation runs exhibit comparable performance, we discuss the results of one such run in detail in this section. In particular, we discuss the results from the evening rush hour on Wednesday, November 21, 2001, between 5:00 pm and 5:15 pm PST.

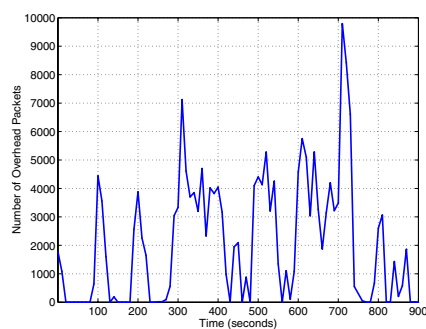
We divided the 900-second simulation time into 10-second intervals. Figure 7(a) shows the number of packets sent and received during each interval. The variation in sent packets is due to the varying number of flows active at each



(a) Sent and Received Packets



(b) Packet Delivery Ratio



(c) Number of Overhead Packets

Figure 7: Performance Metrics

point in time (Section 7.2). Overall packet delivery ratio (PDR) for this experiment was 96.17%. Figure 7(b) shows the overall PDR and the PDR within each 10-second interval. The traffic load and PDR are closely correlated; each time a new flow begins, the PDR declines as a result of the overhead (Figure 7(c)) of searching for a new route, which may include paging.

Packet overhead rises rapidly when new flows are introduced in the network. For example, at times 310 and 710 seconds, 5 to 6 new flows entered the network, resulting in a substantial increase in packet overhead. In fact, the latter 10-second interval represents 10% of the overhead over the whole 900-second run. In this simulation, a total of 199,933 overhead packets were sent, which represents

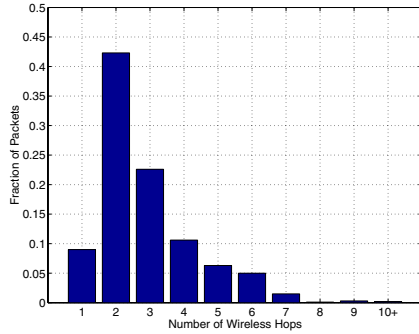


Figure 8: Path Length Distribution (wireless hops only)

an average of 235 overhead packets per network node, over a period of 900 seconds. Even though the the number of overhead packets exceeds the number of data packets by a factor of 10 at certain points in the simulation, this overhead is distributed among a greater number of nodes than is data traffic, and thus its effect on any one node is small.

A total of 21% of packets were able to be delivered without traversing a base station; 57% of these traverse more than one hop. Figure 8 shows the distribution of wireless hop path lengths of delivered packets, i.e., the number of wireless hops each packet traversed to reach its destination. The communicating pairs of nodes in our simulations tend to be relatively close to a base station because a large number of the well-connected nodes are in the very dense downtown areas; as a result, these are the nodes that often get picked when a uniform random selection procedure is used.

Most packets are delivered with low or moderate latency; the 95th percentile latency is 86.57 ms for all packets. For packets traversing the base stations, 95th percentile latency was 119.24 ms, of which 50 ms simulated the wire delay between base stations. For packets not traversing any base station, 95th percentile latency was 9.05 ms. Such latencies should not pose a serious problem for network traffic, as they are typical of current Internet latencies.

Overall, our evaluation demonstrates that the Ad Hoc City architecture is viable and can provide good performance; further optimizations to the protocol will also improve this performance. For example, varying the wireless range of nodes to control the node degree and therefore the level of spatial multiplexing, will lead to increased packet delivery ratios. In addition, a number of improvements, including the use of a better paging policy, modified base station placement, efficient broadcast algorithms, and variable topological cell sizes for more balanced cell loads, can each significantly improve performance.

## 9. Conclusions

Few real-world applications of mobile ad hoc networks have been proposed and studied, and no traces of actual node movement in a real ad hoc network have been available. This paper has proposed a novel commercial appli-

cation of ad hoc networks, for which we designed a multi-tier ad hoc network architecture and routing protocol, which we call *Ad Hoc City*. The Ad Hoc City architecture is designed for general-purpose wide-area communication, such as in a metropolitan area. In addition, we have documented a new source of realistic mobility traces to support detailed simulation of new ad hoc network applications on a large scale.

We evaluated our architecture through detailed simulation, using traces of actual node movement of the fleet of city buses in the Seattle, Washington metropolitan area, on their normal routes providing passenger bus service throughout the city; the number of mobile nodes in our simulations ranges between 750 and 850, providing wireless service to an area of over 5000 square kilometers (2000 square miles). These real-world mobility traces provide a unique and challenging environment for ad hoc network routing protocol performance evaluations. In our simulations of the Ad Hoc City architecture using these mobility traces, our protocol design achieves packet delivery ratios of over 96% with average delivery latency of around 86 ms. Overall, our simulations demonstrate that the Ad Hoc City architecture is viable, and that with further optimizations, could provide good performance in a real deployment.

## Acknowledgements

We would like to thank Shu Du for his assistance with the communication patterns used in our simulations. We would also like to thank the anonymous reviewers, whose comments and suggestions helped to improve the presentation of the paper.

## References

- [1] C. Bettstetter. Smooth is Better Than Sharp: A Random Mobility Model for Simulation of Wireless Networks. In *Proceedings of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'01)*, July 2001.
- [2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. G. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 85–97, October 1998.
- [3] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.
- [4] Z. D. Chen, H.T. Kung, and D. Vlah. Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways. In *Proceedings of the Second ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001) Poster Paper*, October 2001.
- [5] Microsoft Corporation. TerraServer Homepage. Available at <http://terraServer.microsoft.com/>.
- [6] D. Dailey, M. Haselkorn, and D. Meyers. A Structured Approach to Developing Real-Time Distributed Network Applications for ITS Deployment. *The ITS Journal*, 3:163–180, 1997.

- [7] D. J. Dailey. Smart Trek: A Model Deployment Initiative. Technical Report WA-RD 505.1, Washington State Transportation Commission, 2001.
- [8] D.J. Dailey, G. Fisher, and S. Maclean. Busview and Transit Watch: an Update on Two Products from the Seattle Smart Trek Model Deployment Initiative. In *Proceedings of the Sixth Annual World Congress on Intelligent Transport Systems*, November 1999.
- [9] K. Fall and K. Varadhan, editors. *ns Manual*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2002. Available from <http://www-mash.cs.berkeley.edu/ns/>.
- [10] H. Fuessler, M. Mauve, H. Hartenstein, M. Kaesemann, and D. Vollmer. A Comparison of Routing Strategies for Vehicular Ad Hoc Networks. Technical Report TR-02-003, Fakultät für Mathematik und Informatik, Universität Mannheim, March 2002.
- [11] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy. RFC 1519, sep 1993.
- [12] D. Gazis, R. Herman, and R. Rothery. Nonlinear Follow-the-Leader Models of Traffic Flow. *Operations Research*, 9:545, 1961.
- [13] A. Helal, S. Shah, and C. Lee. Mobility Benchmarking In Ad-Hoc Networks. *The Information Processing Society of Japan (IPSI) Journal*, 43(11), June 2002.
- [14] H.-Y. Hsieh and R. Sivakumar. Performance Comparison of Cellular and Multi-hop Wireless Networks: A Quantitative Study. In *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2001.
- [15] H.-Y. Hsieh and R. Sivakumar. A Hybrid Network Model for Cellular Wireless Packet Data Networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, November 2002.
- [16] H.-Y. Hsieh and R. Sivakumar. Internetworking WWANs and WLANs in Next Generation Wireless Data Networks. In *Proceedings of the International Conference on 3G Wireless and Beyond*, May 2002.
- [17] H.-Y. Hsieh and R. Sivakumar. On Using the Ad-hoc Network Model in Wireless Packet Data Networks. In *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, June 2002.
- [18] Y.-C. Hu and D. B. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, August 2000.
- [19] Y.-C. Hu, D. B. Johnson, and D. A. Maltz. Flow State in the Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsrflow-00.txt, February 2001. Work in progress.
- [20] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, pages 12–23, September 2002.
- [21] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, New York, 1997.
- [22] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 195–206, August 1999.
- [23] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.
- [24] D. N. Knisely, S. Kumar, S. Laha, and S. Nanda. Evolution of Wireless Data Services: IS-95 to cdma2000. *IEEE Communications Magazine*, 36(10):140–149, October 1998.
- [25] Y.-D. Lin and Y.-C. Hsu. Multihop Cellular: A New Architecture for Wireless Communications. In *Proceedings of INFOCOM 2000*, March 2000.
- [26] D. A. Maltz, J. Broch, and D. B. Johnson. Quantitative Lessons From a Full-Scale Multi-Hop Wireless Ad Hoc Network Testbed. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 992–997, September 2000.
- [27] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang. A Wireless Hierarchical Routing Protocol with Group Mobility. In *Proceedings of IEEE WCNC'99*, September 1999.
- [28] C. Perkins, editor. IP Mobility Support. RFC 2002, October 1996.
- [29] C. Qiao and H. Wu. iCAR: An Integrated Cellular and Ad-Hoc Relay System. In *Proceedings of the IEEE Conference of Computer Communications and Networks (IC3N)*, pages 154–161, October 2000.
- [30] N. Salem, L. Buttyan, J. Hubaux, and M. Jakobsson. A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks. In *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, June 2003.
- [31] A. Striegel, R. Ramanujan, and J. Bonney. A Protocol Independent Internet Gateway for Ad Hoc Wireless Networks. In *Proceedings of the Proc. of Local Computer Networks (LCN)*, November 2001.
- [32] V. Typpö. *Micro-Mobility within Wireless Ad Hoc Networks: Towards Hybrid Wireless Multihop Networks*. Diploma thesis, Department of Electrical Engineering, University of Oulu, Oulu, Finland, 2001.
- [33] R. Wakikawa, J. T. Malinen, C. E. Perkins, A. Nilsson, and A. J. Tuominen. Global Connectivity for IPv6 Mobile Ad Hoc Networks. IETF Internet-Draft, draft-wakikawa-manet-globalv6.txt, November 2001. Work in progress.
- [34] M. M. Zonoozi and P. Dassanayake. User Mobility Modeling and Characterization of Mobility Patterns. *IEEE Journal on Selected Areas in Communications*, 15(7):1239–1252, September 1997.