

# Efficient Paging Mechanisms in LUMNets

Stephan Bohacek Vinay Sridhara Lei Luo  
{vsridhar, bohacek}@udel.edu, shin@eecis.udel.edu

University of Delaware  
Newark, DE 19716  
Technical Report

**Abstract**—The task of forming a route can be broken into two parts, finding any route to the desired host and refining that route so it is suitable for the connection. This first task is referred to as paging and consists of simply delivering a single packet to the destination. Traditional MANET routing protocols perform this task via flooding. However, when the number of nodes ranges into the thousands, flooding becomes highly inefficient. In the case of cellular networks, paging is performed with one of more base stations broadcasting a page message. However, in large-scale urban networks, nodes are not necessarily in direct contact with the infrastructure, hence such techniques are not applicable. In this paper, the topology and architecture of a large-scale urban mesh network is examined and a paging scheme is developed. This scheme is designed for and takes advantage of the topology, mobility, and architecture found in urban mesh networks. The resulting scheme is highly efficient in that it generates little overhead and is scalable.

## I. INTRODUCTION

Large-scale urban mesh networks (LUMNets) are poised to become next generation of networks to provide ubiquitous internet access to mobile users. In early 2006 the city of Philadelphia [1] finalized partnership to begin the construction of an mesh network that will cover the 135 sq. mile city. The City of San Francisco has recently received proposals for a LUMNet in that city [2]. Taipei is establishing wireless access in 5% of the city, which includes the city's most-populated area, and within a year expects to cover around 90% of the city [3]. Several other cities such as Minneapolis [4], Anaheim, California, and Tempe, Arizona are planning similar LUMNet deployments. These deployments are meant to enhance city and emergency services communication as well as to provide city-wide, low-cost, ubiquitous Internet access for residents and visitors.

While these networks are being deployed, there are several challenges that remain unsolved. Perhaps the most pressing issues are related to scalability and coverage. Consider, for example, that during the lunch time, downtown Manhattan has approximately 10,000 people outdoors in 1 km<sup>2</sup> [5]. Similarly, combination office usage surveys [6] and maps of downtown areas [7] [8] indicates that urban core may have office worker

densities that exceed 40,000 people per km<sup>2</sup> in certain city sections.

Another important issue is coverage. In the case of Philadelphia, the initial plan was that there would be only one hop from the infrastructure nodes (either wired or wireless infrastructure) to the client host (or subscriber host). Due to the administrative and financial difficulties with the city owned networking including infrastructure nodes that are inside of private buildings, the infrastructure nodes are restricted to be outdoors. However, the 802.11 physical layer is not able to easily communicate from indoors to outdoors. As a result, it is not possible for the mesh network to provide one hop connectivity to all locations inside the buildings. For this reason, the initial requirements in Philadelphia and San Francisco are that the network should have at least 90% coverage outdoors and that there will be *some* location in 90% of the buildings that can connect to the wireless network. Recognizing the limitation that this puts on the utility of the network, the planners in Philadelphia considered consumer owned relays that could extend the coverage to inside of buildings. That is, the planners of these networks recognized the possibility that the urban mesh network might include not only the infrastructure and client nodes, but also client nodes that act as relays. In terms of topology, such LUMNets then resemble hybrid networks, which are essentially cellular networks that allow mobile to mobile relaying to enhance performance. While there has been some attention to developing protocols for hybrid networks, hybrid network topologies for LUMNets is an area that has not been well explored.

This paper examines paging in LUMNets. By paging, we mean the task of finding a client node in a LUMNet, or more specifically, delivering a single packet to a client node. Paging is the first step in forming a connection with a client. In on-demand routing protocols for MANETs, paging is integrated into the route request phase (e.g., see DSR [9], AODV [10]). Such protocols use flooding to find the destination. Once a route is found, some routing protocols will use the route traveled by the search message, while others will only use this path as an initial path and then expand the search to

find back-up routes and/or higher quality routes [11] [12] [13]. Thus, the problem of paging a destination and the problem of constructing a route to a destination are not necessarily the same. The routing aspects are not addressed in this paper.

While LUMNets such as those described above can be viewed as a mixture of infrastructure and ad hoc networks, the flooding techniques used in MANET routing protocols are not appropriate. For example, in the case of Philadelphia, it is expected that the network will have 200,000 users [1]. It is unreasonable to flood a network of 200,000 nodes every time a path to a destination is required. This paper presents a technique that is suitable for the types of LUMNet planned for Philadelphia, San Francisco, etc. It is important to note that since these LUMNets include both infrastructure and mobility relaying, traditional paging techniques are not applicable. A key aspect of the scheme developed here is that it takes advantage of the unique topologies that arise in LUMNets.

This paper proceeds as follows. In the next section the structure of LUMNets is investigated. Section III then describes the paging problem. Section IV describes the scalable paging technique. Section V then provides analytic performance analysis of the protocol while Section VI presents simulation-based performance analysis. And finally, concluding remarks can be found in Section VII.

## II. THE TOPOLOGY OF LUMNETS

1) *Coverage provided by the infrastructure:* The propagation of wireless signals in urban areas results in a specific structure of the LUMNet. Specifically, the outdoor communications occurs over good propagation environments, while indoor communications suffers from poor propagation environments. Thus, the outdoor area is well connected. For example, a relatively small number of hops are required to travel great distances. As a result, LUMNets are expected to provide very good coverage outdoors. On the other hand, since the infrastructure is restricted to be outdoors, client nodes within buildings are not well connected. They are mostly only connected to client nodes on the same floor or, at most, a few floors away. For example, consider Figure 1. This figure is based on propagation found from a 3-D ray-tracing experiment on a nine block region of downtown Chicago where the infrastructure nodes were approximately 50 m. apart. The left-hand frame shows the fraction of all outdoor and indoor location that are reachable by the infrastructure at different channel gains. As expected, outdoor locations are well connected and can support high bit-rate communication (i.e., they are connected with low path loss), while even if low bit-rate communication is used (e.g., a 70 dB path loss threshold is used), only a small fraction of indoor locations are

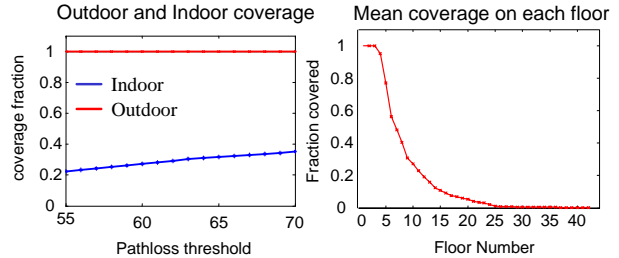


Fig. 1. (a) The figure illustrates the disparity in coverage of indoor and outdoor locations. (b) The figure illustrates the coverage on different floors. The results obtained are from a simulation carried out for a section of downtown Chicago using raytracing mechanism. 27 infrastructures were placed outdoors on lamp posts (3.5m) with a separation of 50m between them.

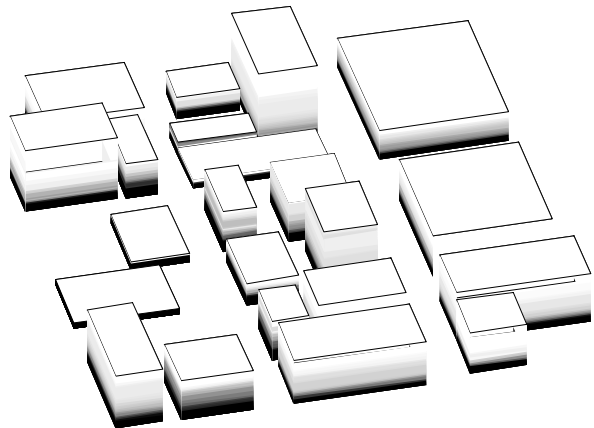


Fig. 2. The figure indicates the coverage inside of buildings due to the infrastructure nodes placed outdoors and separated by 50m. Darker color indicates more coverage. It can be observed that there is no coverage on upper floors of tall buildings.

reachable from the infrastructure. The right-hand frame of Figure 1 provides further insight into the problem with indoor coverage. This figure shows how the coverage varies as a function of the floor (here it is assumed that a path loss of 70 or less will support communication). Clearly, while good coverage is provided at the lower floors, the outdoor radios are not able to penetrate the upper floors. If the mesh network is composed of roof top infrastructure, then the figure would be flipped, with good coverage on the upper floors, and poor coverage on the lower floors.

To understand the topology further, consider Figure 2. This figure indicates the coverage of an urban area. The area under consideration in Figure 2 is based on a section of downtown Chicago and the propagation was determined with a 3-D ray-tracing. The coverage of a building is indicated by the color. Specifically, the darker the color, the higher the fraction of the locations on that floor of the building are covered by the

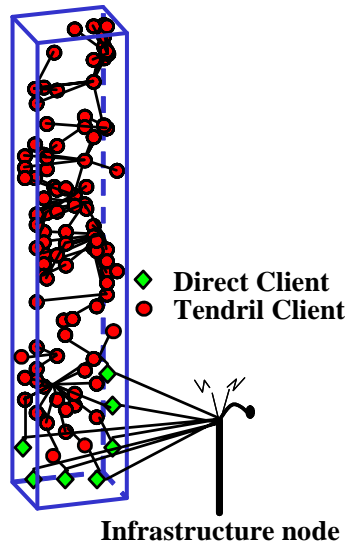


Fig. 3. The figure depicts the structure of multihop communication in a building. This topology resembles that of a tendrill.

infrastructure. In Figure 2, the infrastructure is designed so that each location outdoors is covered by at least one infrastructure node. However, as clearly shown, in Figures 2 and 1, a significant part of the indoor area is not within range to the infrastructure. Specifically, the white colored floors on the upper floors indicate that no locations on these floors can communicate directly with the infrastructure. While one solution is to abandon the hope of ubiquitous communication and insist that indoor communication is provided by the individuals with administrative control of the buildings, or parts of the buildings, we seek protocols that support the goal of city-wide ubiquitous communication.

2) *Topologies with mobile relaying:* If multi-hop communication is permitted, then the coverage of the LUMNet is greatly extended. Indeed, if the subscriber density is high enough, then all regions of the city will be able to communicate with the infrastructure. However, due to the structure of propagation (i.e., good outdoor propagation, and bad indoor propagation), the topology of the LUMNet takes a particular form with a highly connected core that is composed of outdoor nodes and weakly connected tendrills that are composed of client nodes within buildings. Figure 3 shows an example of a tendrill that resulted from a ray-tracing experiment.

We use the following nomenclature to describe the clients within the LUMNet topology (also, see Figure 4). If a client node is directly reachable by the infrastructure, then the client is referred to as a *direct client* (DC). While most, or even all, outdoor nodes will be DCs, some DCs will be indoors as well. We refer to tendrill clients (TC) as client nodes that are not directly reachable from the

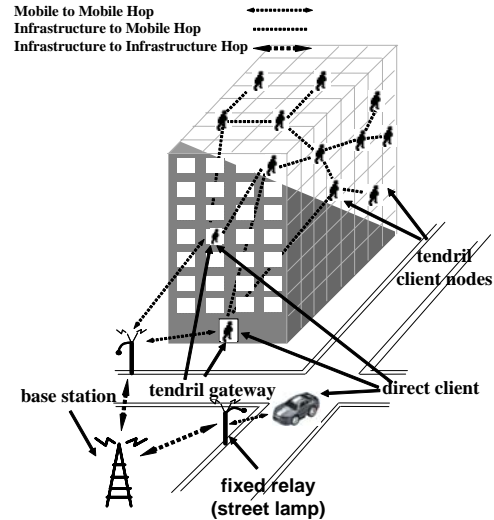


Fig. 4. A conceptual visualization of a mesh networks. The picture above highlights the different elements of multihop wireless mesh networks.

infrastructure, but are reachable when client nodes are used as relays. Of the DCs, some can act as a gateway to the tendrill in that they have infrastructure nodes as neighbors and TCs as neighbors. Such DCs are referred to as tendrill gateways (TGWs). Of course, as client nodes move, their status may change.

Besides the client nodes, there are two types of infrastructure nodes, namely base stations that act as gateways between the wireless network and wired Internet, and fixed wireless relays. We make no distinction between these types of infrastructure nodes, and simply refer to them as WFIN (Wi-Fi infrastructure nodes).

One type of node not discussed here are permanent consumer owned devices. Since these nodes are not owned that the mesh network provider, they cannot be full-fledged WFINs. However, it may be possible to have such nodes play an elevated role in relaying as compared to the typical mobile clients. This can be done in several ways. For example, as described below, efficient flooding within a tendrill is used to find client nodes. It is straightforward to implement efficient flooding to increase the role of some nodes. Furthermore, if the permanent client node is a TGW, then, as described below, it would advertise a long duration connection with the WFIN. The mobile clients make use of such information, and hence, are more apt to rely on such permanent clients as TGWs. However the details of taking advantage of fixed consumer owned devices is left for future work.

### III. PAGING IN WIRELESS NETWORKS

As mentioned in Section I, this paper focuses on paging, that is, the problem of finding a destination. In Section I, we distinguished paging from constructing a route. Thus, the only objective here is to efficiently find client nodes. It is important to note that we only consider the problem of the infrastructure finding a client, and not the problem of the client finding the infrastructure. However, since the infrastructure is large (it is made up of many nodes) and a particular client is small (it is only one node), this second problem is considerably more simple than the one examined here.

Paging is an integral part of mobility management and there has been extensive work on paging, especially in the cellular phone scenario [14], [15], [16], [17], [18], [19]. Paging has also been integrating into mobility management for data networks (e.g., [20], [21]). However, in these cases, the clients are within one hop of the infrastructure, and hence these techniques cannot be applied for paging TCs. Nonetheless, the basic ideas are common to all paging algorithms, including the one developed here. Specifically, each client node will occasionally register with the infrastructure. This registration will contain various information and allows the infrastructure to learn the current location of the client. One approach is for clients to periodically register with the infrastructure, while another is for nodes to register when an event occurs. For example, a client might register when it can no longer "hear" its last WFIN or it can not hear any node from a set of WFINs that was determined during the last registration. Alternatively, a client might register when it has changed WFINs a specific number of times [16]. Besides periodic registration, these techniques require that the client is within communication range of a WFIN. While periodic registration could be applied to the LUMNets considered here, this approach has the drawback that clients that are not moving register as frequently as clients that are not moving. Hence, event-triggered registration where the events are related to mobility is preferable to periodic registration. The scheme presented here utilizes event triggered registration.

The goal of client registration is to reduce the size of the region that must be searched when the client is paged. For example, if a DC registers whenever it moves out of range of the last WFIN with which it registered, it is clear which WFIN it is reachable through. However, when a TC registers, the information about which WFIN the client is reachable from is not sufficient to reduce the region searched. To see this, one must consider the structure of the propagation discussed above, i.e.,

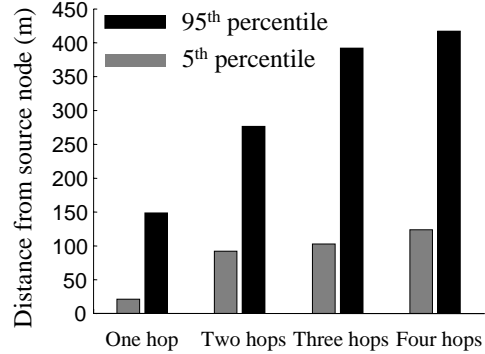


Fig. 5. Heterogenous Propagation. The 5th and 95th percentile of the distances from the WFIN to nodes that are exactly 1,2, 3, and 4 hops away from the WFIN.

the good<sup>1</sup> propagation outdoors and bad propagation indoors. This structure means that distant parts of the city can be reached in a few hops, while the upper floors of a building requires a large number of hops. Consider, for example, a TC that has recently registered with a WFIN and has provided further information that it is 10 hops from the WFIN (such values are common when the TC is on the upper floors of a building). Thus, if the WFIN desires to reach the TC, it could, perhaps, broadcast a paging message that is to be relayed no more than  $10 + K$  times, where  $K$  allows for a few extra hops in case the TC has moved. However, due to the good propagation conditions outdoors, such a message would be propagated to the far reaches of the city. Furthermore, the message needs to be relayed into all buildings near the WFIN. Consequently, the fixed radius flood would utilize bandwidth in a large part of the city.

This effect can also be seen in Figure 5. This figure shows the 5th and 95th percentile of the distance from the WFIN to nodes that are exactly 1, 2, 3, and 4 hops away from the WFIN. Note that as the number of hops increases, the 5th percentile does not change. These nodes are within buildings. However, the 95th percentile increases with the number of hops. As a result, specifying that a flood only spread  $K$  hops will have little impact of the geographic area the flood covers. The scheme presented next does not use hop count to control the region where the searched is performed

### IV. SCALABLE PAGING FOR LUMNETS

#### A. Overview

The basic idea of the scalable paging approach developed here is that WFIN should limit the search of

<sup>1</sup>Due to the straight streets and buildings lining the streets, it is common that when propagating down a street, the signal strength decays like  $d^{-\alpha}$ , where  $\alpha < 2$ . This effect is known as propagation in urban canyons.

the TC to the tendril where TC is located. Thus, the infrastructure must keep track of which tendril a node resides in and TCs within the tendril must also know which tendril they reside in. If these objectives are met, then the WFIN simply broadcasts a page message with the correct tendril identification and all nodes in the identified tendril forward the page message according to an efficient flooding scheme (a few possible efficient flooding schemes can be found in [22] and [23]). Hence, the overhead for the search is the overhead to efficiently flood a single tendril. A tendril may be a single building as shown in Figure 3, or may have some other structure. Regardless, the overhead required to efficiently flood a tendril is far less than the overhead required to efficiently flood the entire network. Furthermore, overhead is independent of the size of the network and only depends on the size of the tendrils within the network.

A tendril is identified by the TGW nodes that provide connectivity between the tendril and the infrastructure. In order for the TC to know which TGWs serve the tendril, the TGW floods beacons through the tendril (here, and in all situations where flooding is used, it is assumed that some sort of efficient flooding scheme is used). By examining how many hops a beacon has traveled, and by comparing this hop count to the hops traveled by beacons from other recently heard TGWs, the TC decides whether to propagate the TGW beacon. In this way, TCs can determine a set of TGWs that are between it and the WFIN. When a TC sends a registration packet to the infrastructure, it includes a list of all TGWs that it has recently heard and that provide high quality communication between the WFIN and the TC (these TGWs are called *active TGWs*). The set of TCs that share the same active TGWs constitute a tendril. Thus, as long as the tendril information is up-to-date in both the infrastructure and the tendril nodes, then the region of the network that must propagate the page message is greatly limited.

An important challenge is to keep the tendril information up-to-date without requiring excessive overhead due to registrations. The next section describes a method that reduces the overhead due to registrations by allowing the registration of a single TC to be gratuitously applied to all the TCs within the tendril. One complication arises when a client moves into a new tendril. In this case, the TC's registrations may be gratuitously applied to all the TCs within last tendril it was in, and the registrations by TCs within the new tendril will not be gratuitously applied to TC that just entered the tendril. This complication can be eliminated by a client rapidly detecting that it has changed tendrils. The algorithm for this detection is developed in Section IV-C.

## B. Similarity and Gratuitous Registration

While restricting the page to remain within a specific tendril greatly reduces the overhead involved in a page, it requires that tendril clients keep the WFINs informed about the TGW that they are reachable through. This requirement can lead to excessive registration overhead when TGWs move or go off/online. For example, when a DC moves close to a building and is able to communicate with nodes within the building, it becomes a TGW for the tendril within the building. Similarly, when a node inside the building moves toward the edge of the building, it may be able to communicate with the infrastructure and become a TGW. When either of these events occurs, there is a risk that every node within the tendril will try to register with the information that it is reachable through the new TGW.

To reduce the number of registrations when the TGW change, the similarity of nearby nodes is exploited, where two nodes are *similar* if they share the same set of active TGWs (active TGW will be defined shortly). The idea is that when a registration from a tendril client is received, the WFIN applies any changes in the registration to all nodes that are similar to the registering node. Thus, one registration will suffice for the entire set of similar tendril clients. To implement this idea, the tendril nodes must be able to determine whether a node has registered and whether the WFIN will apply this registration to it. The algorithm that implements this scheme is discussed after the section IV-B.0.a where active TGWs are defined.

a) *Active TGW*: A TC's *active TGWs* are those TGWs that offer good connectivity between the tendril and infrastructure. Specifically, an active TGW must have offered connectivity from itself to the tendril client for a sufficiently long time and with high enough quality and offer connectivity between itself and the infrastructure for a sufficiently long time and with high enough quality. Here we specify that communication quality is the fraction of successful transmissions. While not investigated here, quality could also include other metrics such as SNR and interference.

To determine which TGWs are active TGWs, the TGW beacon contains a beacon number, the beaconing period, and a measure of the quality of communication from the TGW's "best" WFIN to the TGW. The "best" WFIN is the one that has high enough quality of communication to the TGW for the longest duration. By including a beacon number, the TC is able to determine if any beacons have been missed. The beaconing period allows the tendril client to determine whether and how many beacons have been missed since the last one was received. Together, these allow the TC to determine the probability of packet delivery from the TGW to the TC. The WFIN beacons also include beacon numbers and

beaconing period and allow the TGW to compute the communication quality from the WFINs. It is possible to include other measures within the TGW beacon. For example, as the beacon is being flooded through the tendrils, each node could append the received signal strength. On the other hand, the nodes could append whether the node is a fully mobile node (e.g., mobile VoIP hand-set), or semi-permanent (e.g., a desktop or consumer owned device that is specifically designed to act as a relay). Such techniques are not investigated here.

As TGW beacons are received, the TC is able to determine whether the TGW is stable (i.e. the number of beacon received exceeds a threshold) and whether the communication quality from the TGW is sufficient (e.g., the probability of a beacon being successfully received exceeds a threshold). If both of these requirements are met, then the TGW becomes an active TGW. On the other hand, a TGW is removed from the list of active TGWs when the communication quality falls below a threshold or when a sufficient number of consecutive TGW beacons have not been received. It is important to note that it takes time for a TGW to become an active TGW. Thus, a mobile node that happens to momentarily be near a building will not become an active TGW. Furthermore, hysteresis is included so that there is limited "flap" in the list of active TGWs.

*b) Gratuitous Registration:* In order to support similarity-based implicit registration, for each TC, three lists of active TGWs are maintained. First, the TC maintains a list of currently active TGWs. Second, the WFIN that the TC registers with maintains an estimate of the TC's active TGWs. Third, the TC maintains an estimate of the WFIN's estimate of its list of active TGWs. In general, when the TC detects that the WFIN's estimate of active TGW differs from the TC actual list of active TGWs, the TC sends a registration. After the WFIN receives a registration, the WFIN's estimate of the TC's list of active TGWs matches the TGWs list. And when the acknowledgement of the registration is received by the TC, all three lists are identical.

The following scheme is used to reduce registration overhead. This scheme is also depicted in Figures 6 and 7. When a TC receives a TGW beacon from a TGW, it determines its list of active TGWs (perhaps adding the TGW whose beacon just arrived and perhaps deleting any TGW from which beacons have not recently arrived). The TC then compares this updated list of active TGWs to the TC's estimate of the WFIN's list of active TGW for this node. If these lists are different, then the node will perform two tasks; first it will prepare to register with the WFIN, and second, it appends its active TGW information on the TGW beacon and propagates the TGW beacon. As is the case with all flooding, the propagation of the TGW beacon follows an efficient

flooding algorithm.

Now consider the propagation of the TGW beacon that has been appended with the list of active TGWs. We will refer to the TC that appended the TGW information as the *registering TC* and any TC that receives the appended TGW beacon as a *receiving TC*. The appended active TGW information includes both the registering TC's updated list of active TGWs and its estimate of the WFIN's list of active TGW for the registering client. Upon receiving this appended TGW beacon, the receiving TC first confirms that this beacon has not travel too many hops as compared to other TGW beacons that arrive from other active TGWs. If so, the TGW beacon is dropped. If not, the receiving TC compares its estimate of the WFIN's list of active TGW to the estimate that is included in the beacon. If these two lists match, then the registering TC and the receiving TC are similar. In this case, when the WFIN receives the registration from the registering TC, it will apply the updated active TGW list to all similar clients, which includes the receiving TC. Thus, if the registering TC and the receiving TC are similar, then the receiving TC updates its estimate of the WFIN's list of active TGW for it to match the list of active TGWs included in the beacon. The receiving TC then updates its actual list of active TGW according to the received TGW beacon (i.e., to reflect the new active TGW represented by the reception of the TGW beacon and/or to reflect the TGW deleted due to missed beacons). In the typical case, the actual list and the newly updated estimate of the WFIN's list match, and hence the receiving node does not prepare to send a registration packet. Rather, it updates the number of hops that the TGW has traveled and propagates the TGW beacon (following the rules of the efficient flooding algorithm). On the other hand, if the receiving TC is not similar to the registering TC or if the receiving TC's estimate of the WFIN's list of active TGWs for it does not match the receiving TC actual list of active TGWs, then the receiving TC prepares to register, and append its estimate of the WFIN's list of active TGW for the it along with its actual list of active TGWs. This extended TGW beacons is then propagated. In this way, a single client registration will serve to register a large number of clients greatly reducing the number of registrations. But, if this registration does not suffice for some nodes, then these nodes will register as well.

The number of registrations is further reduced as follows. When a registering node prepares to register, it first listens for a random duration for other registrations. If it overhears another node transmit a registration and these two nodes are similar (i.e., they have their estimate of the WFIN's estimate of active TGWs is the same), then the node that has not yet transmitted its registration will terminate the registration process.

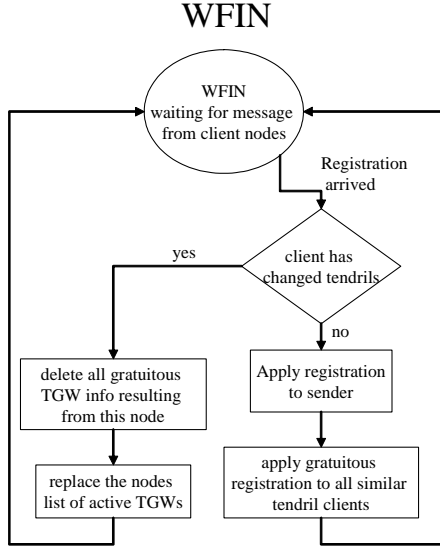


Fig. 6. Flowchart describing the functionality of a WFIN on reception of a registration from a client.

While this scheme reduces the number of registration messages, it increases the impact of a dropped or lost registration packet. Hence, a node that transmits a registration packet will repeat the transmission periodically until an acknowledgement from the WFIN is received.

### C. Detecting a Change of Tendril and Appropriation of Neighbors' Active TGW Lists

1) *Detecting a Change of Tendril*: One problem with the scheme above arises when a client node changes tendrils. For example, a TC may be similar to a number of TCs within a tendril. Thus, any registration sent by this TC will be applied to these other TC. However, when this TC moves to another tendril, it will detect new TGWs and hence will eventually register with this new TGW information. Upon receiving such registrations, the WFIN will update the list of active TGWs for all the TCs that are similar to the registering node, which includes TC in the old tendril. Similarly, and more importantly, when a node changes tendrils, it will eventually remove TGWs from its active TGW list. When this TC sends registrations, the registrations will result in the incorrect deletion of the TGW from similar nodes. While the spurious assignment of additional TGWs to the WFIN's estimate of a TC's list of active TGWs will result in page messages being propagated through additional tendrils, the spurious deletion of TGWs could result in page messages not being propagated through the tendril where the TC resides.

In order to reduce such spurious updating of active TGW list, the WFIN will mark all TGW information it has applied to a client as gratuitous and it will record

which node or nodes are responsible for the gratuitous registrations. Furthermore, each TC will attempt to detect when it changes tendrils. When a change of tendril is detected, the node will send a message to the WFIN which will delete any gratuitous TGW information that resulted from this node.

Furthermore, when a client node detects that it has changed tendrils, it will also appropriate the list of active TGW from its new neighbors. This list of TGWs is sent in a registration message to the WFIN. The WFIN then not only deletes any gratuitous TGW information due to this node, but also replaces the node's list of active TGWs with this new list. In this way, the client becomes instantly similar to the other nodes within the new tendril. These steps used to detect a change of tendril are described next. This algorithm is depicted in Figures 6 and 7. Furthermore, the fine grain details of this scheme are discussed in the analysis of the algorithm in Section V.

As mentioned above, in support of efficient flooding, TCs will periodically broadcast neighborhood information. For example, it is typical to exchange 1-hop neighbor information so that each client is able to construct its 2-hop neighborhood. Piggy-backed on the neighborhood broadcasts, TCs include their list of active TGWs. Thus, a node can easily determine whether it is similar to its neighbors. If a client is not similar to its neighbors, then it is likely that one or more nodes moved. But a difference in neighboring TCs' list of active TGWs only indicates that some client has moved, it does not indicate which client moved. However, once a TGW beacon is flooded through the tendril, the TGW beacon will clarify that some active TGWs are correct and others are incorrect. When a sufficient number of TGW beacons confirm that which of the list of active TGWs is correct, the TC or TCs with an incorrect list of active TGWs will appropriate the list of active TGW from the neighbors with correct a list. This TC (or TCs) will send a special registration message indicating that the TC has changed tendrils and the new list of active TGWs will be added.

Note that the appropriating of neighbor's list of active TGWs greatly speeds up the generation of the list of active TGWs. More specifically, as mentioned above, to reduce the number of registrations and changes in active TGWs, it is important that the active TGW lists not include transitory TGWs (i.e., client nodes that are TGW for only a short time). This can easily be accomplished by only making a TGW active once a TGW when a large number of TGW beacons have arrived. A drawback of this approach is that it increases the amount of time that a TC must remain in a new tendril before it makes the TGWs that serve the tendril into active TGWs. However, by appropriating the list of active TGWs from its neighbors, a TC can determine its active TGW quickly and register with the WFIN, but still reduce the impact

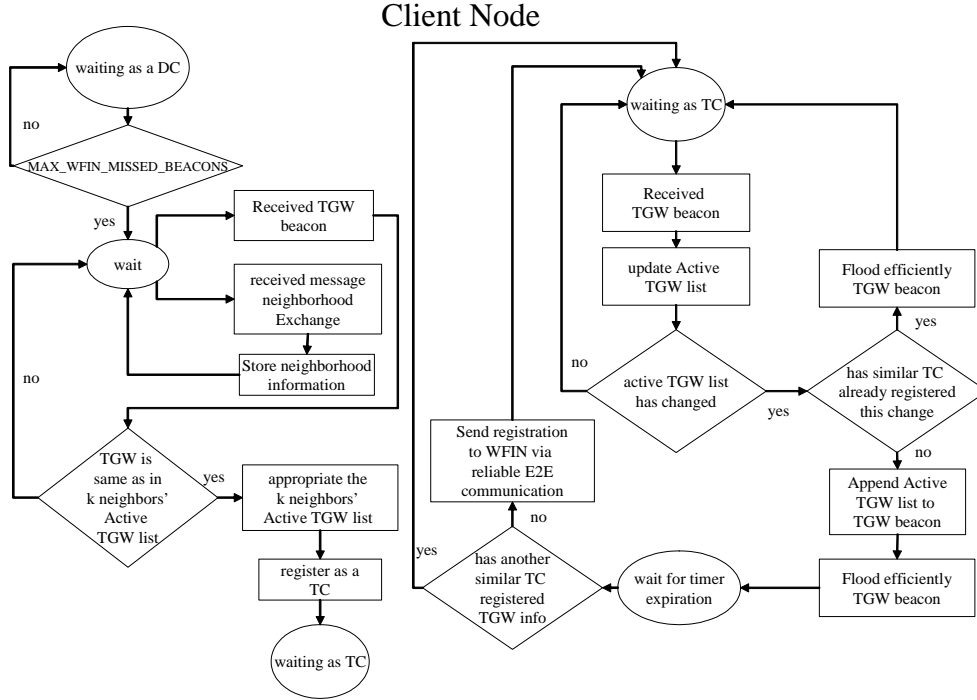


Fig. 7. The figure shows the functionality of a client node as a flowchart. The left hand flowchart depicts the functionality of a Tendril Gateway or Direct Client. The right hand flowchart shows the functionality of a Tendril Client

of transitory TGWs.

2) *Detecting Entry into a Tendril:* When a client moves within a large building it may move to a location where there are different active TGWs (i.e., two tendrils are overlapped). The scheme discussed above will detect such changes. A second way in which a client changes tendrils is when it moves from being an DC to a TC. It is important to distinguish the change of tendril problem and the change from a client becoming a TC. In the first scenario typically occurs when tendrils overlap. Hence, while the TC is in the area of overlap, it is reachable via both tendrils (i.e., both sets of TGWs). However, when a client becomes a TC, it means that it is no longer a DC and hence is no longer reachable by the infrastructure. Reachability will only be restored when the TC registers. The duration of the period until the TC become reachable is an important part of the performance of the paging scheme. As will be shown in Section V, there is a trade-off between the average length of this duration and the overhead generated by the scheme.

The algorithm to detect which tendril a client has entered is similar to the scheme to detect that a TC has changed tendrils. We briefly describe the algorithm here. Further details are included in Section V where the trade-off between the duration of unreachability and overhead is examined. Also, Figures 6 and 7 depict the essential parts of this algorithm.

Briefly, the a client becomes aware that it is no longer a DC when a sufficient number of WFIN beacons have been missed. The client then loses its status as a DC and becomes a TC. This event triggers the broadcast of its neighborhood information to support efficient flooding. This broadcast induces the client's new neighbors to broadcast their neighborhood information. This appended on to the neighborhood information are the TCs' list of active TGWs. The new TC records the received TGWs. When a TGW beacon is flooded, the new TC is able to confirm which of the neighbors' active TGWs are valid. The new TC then appropriates the active TGW from these TCs and registers with the WFIN.

#### D. Other issues

While there are many minor aspects to the paging scheme, some of the more important ones are as follows. While the registration process discussed above is sufficient for a node to always be reachable, the tendril nodes will periodically send registrations to the WFIN. These registration are useful to confirm gratuitous registrations and provide an upper bound on the worst-case duration that a client is not able to be paged due to incorrect or corrupted registrations (e.g., infrastructure node failure/rebooting). However, rebooting can be accommodated with specialized beacons that request registrations at a



particular hop distance. Hence, the WFIN could request registrations in an expanding ring fashion.

In spite of the mechanisms described above, in many cases, when a new TGW becomes an active TGW, several nodes in the tendril will register. For example, if two nodes are within range of a newly active TGW but not within range of each other, they will both register that they are reachable via the newly active TGW. While this slightly increases the overhead due to registration, it also provides robustness in that it reduces the impact of any single TC's registration.

Registrations only occur shortly after a TGW beacon is received from an active TGW. The registration follows the reverse path that the beacon took. Thus, the path is likely to exist. Furthermore, when a TC or DC forwards the registration, it appends its own registration information. Moreover, if a TC desires to make a connection with the WFIN or some node not in the mesh network, the TC can initiate a connection to the base station by using the reverse path followed by the most recently received TGW beacon.

The TGWs do not send TGW beacons into the tendril. Rather, they periodically register with the WFIN. TCs overhear the registration and flood it into the tendril as a TGW beacon.

There are several aspects of this scheme that are not discussed, as they can be achieved using techniques commonly used in mobility management and wireless networking. For example, the infrastructure must maintain and distribute information about which WFIN each client is reachable from. Also, deciding when a client is a DC, and allowing a TGW to become active more quickly when the TC and its neighbors do not have any active TGWs can be attacked using standard techniques that employ thresholds etc.

## V. ANALYSIS OF PERFORMANCE

One of the main performance aspects of the scalable paging algorithm described above is the overhead generated. This overhead is incurred when paging a node. Further, overhead is generated to maintain the tendril and direct client information. This second type of overhead includes the registrations by tendril and direct clients, the propagation of tendril gateway beacons, the periodic broadcast of neighborhood information, and the periodic beaconing by WFINs.

The main objective of this work is to reduce the overhead involved in paging a node. Since this scalable paging algorithm restricts the paging to the tendril where the destination is located, the overhead incurred to page a node is greatly reduced. However, besides reducing the overhead when paging a node, it is also important that clients are able to be paged, i.e., the infrastructure is aware which tendril a client resides in. In general,

as long as the network remains connected, nodes are reachable, except for a short time when the node moves from being a DC to become TC. As will be shown, there is a trade-off between the overhead generated to maintain the client information, and the duration when the node is unreachable. In this section we study this trade-off analytically. Section VI studies this trade-off via simulations.

The TC information is maintained in three ways. First, the TCs will periodically register. However, due to the efficiency of the similarity technique, the periodic registration can be made to occur infrequently, and hence, has little impact on the overhead. The second way in which client information is maintained is by TCs registering when a change in the direct clients for the tendril is observed. Recall (see left hand flowchart of Figure 7) that when a TC detects a change in the active TGW, it propagates the TGW beacon that resulted in the detection of the change with information so that other TCs in the tendril do not register with the same information. In theory, only a single TC is required to register with the new tendril gateway information. In practice, a few TCs will register. In any case, only a small fraction of the nodes within the tendril register when the list of active TGW changes. Furthermore, this registration only occurs when the tendril gateways have changed. A node will not become a tendril gateway unless it is reasonably stable. For example, in our simulations, a direct client had to provide a stable connection for about five minutes before becoming a TGW. Thus, nodes that are currently in motion, will not become TGW. On the other hand, from studies of mobility [24], it is known that office workers are at their desk for an average of 18 minutes between trips. Thus, once a node becomes a TGW, it will stay as a TGW for an average of 13 minutes. As a result, the registration due to a change in the list of active TGWs for a tendril is rare, and does not result in significant overhead.

A third way in which the client information is maintained is by a mobile node detecting that it has entered a new tendril and then registering with the infrastructure. The registration itself results in little overhead. However, allowing the node to quickly detect that it has entered a new tendril results in significant overhead. Recall (refer right hand flowchart of Figure 7) that a node detects that it has entered a new tendril with two mechanisms; the TGW beacons that the node receives are from TGWs that are not in its current list of active TGWs, and the node's own set of active TGWs differ from its neighboring nodes' set of active TGWs. The node detects that its own set of active TGWs are different from its neighbors' by comparing its active TGWs to those within the neighborhood information packets that nodes periodically broadcast. This detection process takes some

time and begins only after the client is no longer able to communicate with any WFIN. Thus, during the detection process, the node is not able to be paged. Thus, we closely examine this scenario by detailing the part of the algorithm that detects that a client is no longer a direct client and to determine which tendril it has entered. Once the node makes this determination, it registers with the WFIN (and includes its new tendril information) and is again reachable.

The first step in this process is when a DC detects that it is no longer within reach of the WFIN and occurs when a  $\text{MAX\_WFIN\_MISSED\_BEACONS}$  beacons are missed. This takes  $T_{WFIN} \times \text{MAX\_WFIN\_MISSED\_BEACONS}$ , where  $T_{WFIN}$  is the time between WFIN beacons. Once this occurs, the client node begins to search for TCs. Specifically, it transmits its neighborhood information. Typically, the neighborhood information includes the sender's list of active TGWs along with a list of the sender's neighboring clients. Of course, at this point, the client is not aware of these neighbors and has no active TGWs. Hence, two empty lists are included in the neighborhood information broadcast.

The neighboring TCs detect that a new node is in the vicinity and increase the rate at which they broadcast their neighborhood information and list of active TGWs. Specifically, these nodes will divide the time until they are scheduled to broadcast their neighborhood information by two. Thus, the time from when the client detects that the WFIN is no longer reachable until the time when a particular neighbor broadcasts its neighborhood information is uniformly distributed between 0 and  $T_{NB}/2$ , where  $T_{NB}$  is the average time between a node's neighborhood information broadcasts when a new neighbor has not been detected. Applying order statistics, the average time until  $k$  neighbors broadcast their neighborhood information is given by

$$MNB(T_{NB}, \bar{D}, k) := \int_0^{T_{NB}/2} \frac{2\bar{D}}{T_{NB}} \binom{\bar{D}-1}{k-1} (1) \times \left(\frac{2}{T_{NB}}x\right)^{k-1} \left(1 - \frac{2x}{T_{NB}}\right)^{\bar{D}-k} dx,$$

where  $\bar{D}$  is the number of neighbors. It should be noted that the number of neighbors is not fixed, but a random variable. Setting  $\bar{D}$  to be the mean number of neighbors results in an approximation of the average time to receive  $k$  neighborhood information broadcasts. Figure 8 (a) shows this (approximate) average time until 3 such broadcasts are heard for different values of  $T_{NB}$  and  $\bar{D}$ .

The reception of neighborhood information broadcasts alone will not guarantee that a client has changed tendrils. For example, it is possible that a group of nodes has changed tendrils; hence, the neighbors of the node

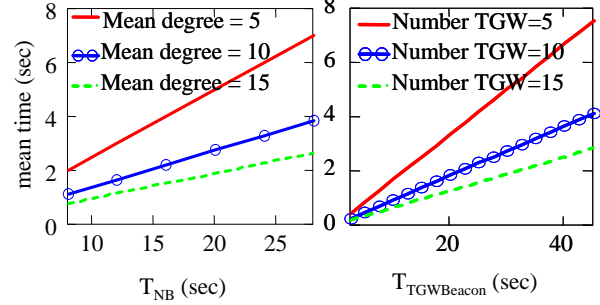


Fig. 8. (a). Mean time to receive three neighborhood broadcasts when a new node broadcasts its TGW history as a function of the total number of neighbors and  $T_{NB}$ , the time between broadcasting neighborhood information. (b) Mean time to receive a TGW beacon when there are different numbers of TGWs. The x-axis is the time between TGW beacon transmissions.

have moved, not the node itself. However, the entrance into a tendril is confirmed by the reception of a TGW beacon from a TGW that is included in  $k$  neighbors' lists of active TGWs (we insist that this TGW beacon is received with sufficient signal strength and reports a sufficiently long duration and high quality connection with the WFIN). The average time for a TGW beacon to arrive is

$$MTGWB(T_{TGW}, \bar{N}) := \int_0^{T_{TGW}} \bar{N} \frac{1}{T_{TGW}} \left(1 - \frac{x}{T_{TGW}}\right)^{\bar{N}-1} dx, \quad (2)$$

where a TGW broadcasts a TGW beacon every  $T_{TGW}$  sec. and where there are  $\bar{N}$  TGWs for this tendril. Again, since the number of TGW is not deterministic, setting  $\bar{N}$  to be the average number of TGWs for each tendril results in an approximation of the average time until a TGW beacon is heard. Figure 8 (b) shows the value of  $MTGWB$  for different scenarios.

Finally, the mean time for a client node to detect that it has entered a new tendril is

$$T_{WFIN} \times \text{MAX\_WFIN\_MISSED\_BEACONS} + \max(MNB(T_{NB}, \bar{D}, k), MTGWB(T_{TGW}, \bar{N})) \quad (3)$$

This function is shown in Figure 9. The mean time to register after entering a tendril was also found from simulations; these results are shown in Figure 11. These figures are in reasonable agreement.

From (1), (2), and (3), it is clear that the duration that a node is unreachable after entering into a tendril depends on the density of nodes (i.e.,  $\bar{D}$  is the average degree of a client that enters the tendril and  $\bar{N}$  is the average number of TGWs), on the rate that beacons are sent and the rate that neighborhood broadcasts are performed, i.e.,  $T_{WFIN}$ ,  $T_{TGW}$ ,  $T_{NB}$ . It appears reasonable that these constants could be made adaptive. This tactic will be investigated in future efforts.

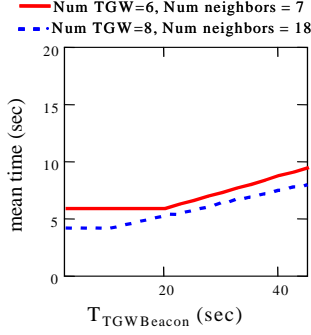


Fig. 9. Mean time that a node is unreachable after it changes from a direct client to a tendrill client. It is assumed that the  $T_{INBeacon} = 0.5\text{sec.}$ ,  $\text{MAX\_IN\_MISSED\_BEACONS} = 6$ ,  $T_{NB} = 15\text{ sec.}$ , and the other parameters are as indicated.

## VI. SIMULATION RESULTS

From the analysis above, approximate parameter values can be found that satisfy the desired trade-off between the duration of unreachability and the overhead. Here we use simulations to perform further analysis. These simulations used the UDeIModels [25] [26] and QualNet. The UDeIModels [27] are a set of simulation tools that provide realistic mobility and propagation in urban areas. Several different scenarios were examined, namely, a 1-building/1-WFIN scenarios, a 2-bulding/1-WFIN scenario, a 4-building/2-WFIN scenario, and an 8-building/3-WFIN scenario. All buildings had 10 floors and the node population was approximately  $100 \times$  the number of buildings. It is important to note that 100 nodes/building is considerably fewer than typically occupy a 10 story building. However, as the node density increases, efficient flooding plays an increasingly important role in the performance. While efficient flooding is well studied [22], there has been little work devoted to the specifics of flooding in an urban environment. This paper devoted to paging, and hence, to avoid the impact of the efficient flooding protocol, we restrict our attention to relatively sparse populations.

### A. Client information maintenance

We begin by examining the types of overhead. Figure 10 shows the per node overhead to maintain the client information for  $T_{WFINBeacon} = 0.5\text{ sec.}$ ,  $T_{NB} = 15\text{ sec.}$ ,  $T_{TGWBeacon} = 20\text{ sec.}$  These results were generated from the two building scenario, but provide a good approximation into the other scenarios. As expected, little overhead results from the TC registration, but significant overhead is due to the TGW beacons and neighborhood exchanges.

Figure 11 shows the relationship between several parameters and the mean time for a node to become reachable (i.e., register) after entering into a tendrill from

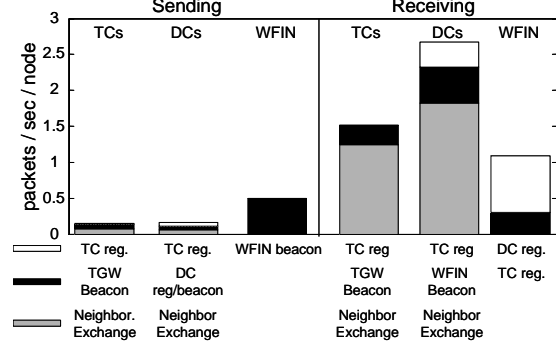


Fig. 10. Rates and types of packets that are sent and received. These packet types are used to maintain the client information.

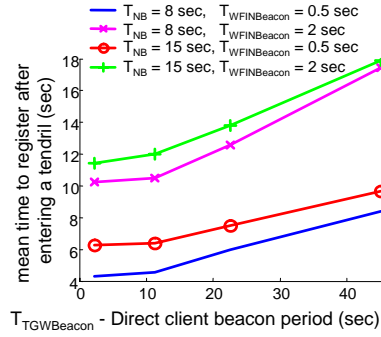


Fig. 11. Mean time for a client node to register after entering a tendrill. In these simulations,  $\text{MAX\_IN\_MISSED\_BEACONS} = 6$  and  $k = 3$ . The values of the other parameters are as shown. The two building scenario was used in these simulations.

being a direct client. The case where  $T_{NB} = 15\text{ sec.}$ ,  $T_{WFINBeacon} = 0.5\text{ sec.}$ , should be compared to the analytical result shown in Figure 9.

The parameter values should be selected to achieve a trade-off between the overhead and the average duration until a node registers after entering a tendrill. It is important to note that nodes move from being a direct client to being a TC relatively infrequently (indeed, once a node becomes a TC, it must remain a TC until either it has verified that it is no longer reachable from its TGW and is reachable from the WFIN, or the node has a strong connection with the WFIN for a sufficiently long time). Thus, during the course of the day, a node is only unreachable for a small fraction of time. Hence, in order to reduce overhead, it may be desirable to allow extended periods where the node is unreachable. Nonetheless, we select  $T_{NB} = 15\text{ sec.}$ ,  $T_{WFINBeacon} = 0.5\text{ sec.}$ ,  $T_{TGWBeacon} = 20\text{ sec.}$ , and  $\text{MAX\_IN\_MISSED\_BEACONS} = 6$ .

### B. Route Search/Paging Overhead

The main objective of the above algorithms is to find clients such that the overhead remains bounded as

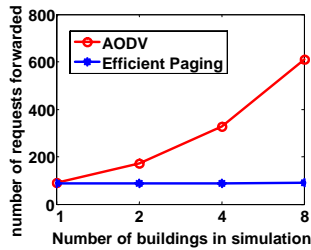


Fig. 12. Comparison of overhead for performing a single page using the efficient paging algorithm and route search technique used in AODV. Each building has 10 floors and is populated with 100 nodes.

the size of the network increases. Figure 12 shows the overhead for performing a single page/route search using the algorithm presented above as compared to the route search technique used in AODV. In the case of AODV, it is assumed that nodes have no prior information as to the location of the desired node. Hence, AODV will flood the network when performing route search.

As shown in Figure 12, the overhead for a single route search grows linearly with the number of buildings when AODV is used, while it remains constant when the scalable paging described here is used. Note that in the case of one building, AODV and the scalable paging algorithm aims to flood a single building. (As mentioned above, the impact of efficient flooding is minimal since the node density is low).

## VII. CONCLUSION

While large-scale urban mesh networks (LUMNets) are being deployed around the world, there remain substantial challenges related to coverage and scalability. Coverage can be enhanced if mobile nodes are permitted to act as relays. In this case, traditional mobility management techniques are not applicable. In particular, paging or searching for a mobile node in such networks has not been addressed. Flooding, which is used in MANET routing protocols cannot be used since it does not scale to the size of the LUMNets being developed. This paper presents a scalable technique to page nodes in such LUMNets. The scheme leverages the topology that arises in urban areas. Specifically, the fact that wireless propagation is good outdoors, but greatly restricted indoors produces a topology with highly connected core that is composed of outdoor nodes, and weakly connected tendrils made up of indoor nodes. The scheme presented restricts the flooding to the tendrils where the mobile node resides.

## REFERENCES

[1] Philadelphia Mayor's Office, "Government leaders teleconference - wireless: The 21stCentury technology," 2004. [Online]. Available: <http://www.phila.gov/mois/press/multimedia/pdfs/WirelessPhiladelphia092004.pdf>

[2] City and C. of San Francisco, "Request for proposals 2005-19 - TechConnect: Community wireless broadband network," Available at <http://www.sfgov.org/site/techconnectindex.asp>, 2005.

[3] T. Times, "Ma says taipei to be wireless in two days," December 19, 2005.

[4] T. C. of Minneapolis, "Request for proposal to provide broadband IP data access services," 2005.

[5] F. Pozzi and C. Small, "Vegetation and population density in urban and suburban areas in the u.s.a." *Presented at the Third International Symposium of Remote Sensing of Urban Areas*, June 2002.

[6] M. Wist, "Office space - how much is enough?" Gerald Eve, Tech. Rep., 2001.

[7] "Geographic information system (GIS)," <http://www.gis.com>.

[8] U. C. Bureau, "Topologically integrated geographic encoding and referencing (TIGER)," Available at <http://www.census.gov/geo/www/tiger/>.

[9] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. kluwer academic publishers, 1996, pp. 153-181.

[10] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," *Proceedings of the 2ns IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[11] M. Marina and S. Das, "Ad hoc on-demand multipath distance vector routing," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2002.

[12] A. C. VALERA, W. K. SEAH, and S. RAO, "CacHing and MultiPath routing protocol," *5th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)*, 2002.

[13] A. Nasipuri and S. Das, "Demand multipath routing for mobile ad hoc networks," 1999.

[14] S. J. Kim and C. Y. Lee, "Modeling and Analysis of the Dynamic Location Registration and Paging in Microcellular Systems," in *IEEE Transactions on Vehicular Technology*, vol. 45, no. 1, Feb 1996, pp. 82-90.

[15] C. Rose and R. Yates, "Minimizing the Average Cost of Paging Under Delay Constraints," in *Wireless Networks*, 1995.

[16] I. Akyildiz, J. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCSnetworks," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629-638, 1996.

[17] A. Hac and Z. Zhou, "Locating Strategies for Personal Communication Networks: A Novel Tracking Strategy," in *IEEE Journal on Selected Areas in Communication*, vol. 15, no. 8, Oct. 1997, pp. 1425-36.

[18] A. Bar-Noy and I. Kessler, "Tracking Mobile Users in Wireless Networks," in *IEEE Transactions on Information Theory*, vol. 39, 1993, pp. 1877-1886.

[19] A. Bar-Noy, I. Kessler, and M. Naghshineh, "Topology-based Tracking Strategies for Personal Communication Networks," in *Mobile Networks and Applications*, vol. 1, no. 1, August 1996, pp. 49-96.

[20] X. Zhang, J. G. Castellanos, and A. T. Campbell, "P-MIP: Paging in mobile IP," in *Proceedings of the 4th ACM International Workshop on Wireless Mobile Multimedia*, 2001.

[21] S. K. Das and S. K. Sen, "A New Location Update Strategy for Cellular Networks and Its Implementation Using a Genetic Algorithm," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM 1997)*, September 1997, pp. 185-194.

[22] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2002, pp. 194-205. [Online]. Available: [citeseer.ist.psu.edu/williams02comparison.html](http://citeseer.ist.psu.edu/williams02comparison.html)

[23] Y. Yi, M. Gerla, and T.-J. Kwon, "Efficient flooding in ad hoc networks using on-demand (passive) cluster formation," *Mobihoc*, 2003.

[24] R. R. Panko and S. T. Kinney, "Meeting profiles: Size, dura-

tion, and location,” in *Proceedings of The 28th Annual Hawaii International Conference on Systems Science*, 1995.

- [25] J. Kim, V. Sridhara, and S. Bohacek, “Realistic simulation of urban mesh networks - part i: Urban mobility,” *Submitted*, 2006.
- [26] V. Sridhara and S. Bohacek, “Realistic simulation of urban mesh networks - part II: Urban propagation,” *Submitted*, 2006, <http://udelmodels.eecis.udel.edu>.
- [27] S. Bohacek and V. Sridhara, “UDelModels,” 2004, <http://udelmodels.eecis.udel.edu>.