Design and Evaluation of iMesh: an Infrastructure-mode Wireless Mesh Network

Vishnu Navda, Anand Kashyap and Samir R. Das
Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY 11794-4400, U.S.A.
Email:{`vnavda,anand,samir`}`@cs.sunysb.edu`

Abstract

We design and evaluate iMesh, an infrastructure-mode 802.11-based mesh network. Here, 802.11 access points double as routers making the network architecture completely transparent to mobile clients, who view the network as a conventional wireless LAN. Layer-2 handoffs between access points trigger routing activities inside the network, which can be thought of as layer-3 handoffs. We describe the design rationale, and a testbed implementation of iMesh. We present results related to the handoff performance. The results demonstrate excellent handoff performance, the overall latency varying between 50-100ms depending on different layer-2 techniques, even when a five-hop long route update is needed. Various performance measurements also demonstrate the clear superiority of a flat routing scheme relative to a more traditional, mobile IP-like scheme to handle layer-3 handoff.

1 Introduction

In this paper, we investigate a mesh networking architecture as an alternative to wireless LANs based on IEEE Standard 802.11 [2]. In a mesh network, *wireless access routers* are deployed to cover a region where wireless access is desired, much like the way access points are deployed in a traditional wireless LAN. However, unlike access points in a wireless LAN, the access routers are not connected to a wired infrastructure. They are rather interconnected via wireless links to form a backbone wireless network. The mobile client nodes (e.g., laptops and palmtops) still associate with one nearby access router, oblivious of the nature of the backbone connectivity. This method of eliminating the "wires" from the wireless LAN provides a significant deployment advantage. It is envisioned that with plummeting cost of IEEE 802.11-based hardware, mesh networking will become as ubiquitous as wireless LANs, and will "blanket" communities with wireless coverage at a low cost.

The goal of this paper is to design and evaluate a wireless mesh network architecture for community networking applications. The goal is to be able to provide seamless networking services to the mobiles both for last mile access and peer-to-peer access. Our architecture uses 802.11b-based access points (AP) that also double as routers thus providing the service of a wireless access router, following the terminology used before. We will refer to them as APs, or access routers, or mesh routers. The fundamental design goal that we pursue is *client side transparency*. The client mobile stations[1] are unaware of the mesh networking backbone. They view the network as a conventional wireless LAN spread out over an extended geographic area. Thus the clients still associate with an AP using a traditional association mechanism in wireless LANs. When the client moves and re-associates with a different AP, a layer-2 handoff event occurs that in turn triggers appropriate routing updates in the mesh network backbone. Thus, the handoff process involves both layer-2 and layer-3 procedures. We describe how the layer-2 and layer-3 handoffs work together efficiently, and present design choices for the layer-3 handoff process – one using a mobile IP [19] like solution called *Transparent Mobile IP* [25] and the other using a "flat" routing protocol based on link-state routing. We also present a detailed performance evaluation of the handoff latencies in both layers.

The rest of the paper is organized as follows. In the following section we develop our system architecture. In Section 3, we describe the implementation details and in Section 4 we present the performance results. Section 5 describes the related work. The conclusions are presented in Section 6.

---

[1] We will use the words "client," "mobile" and "station" interchangeably in this article.

2. Design and Implementation

For providing complete client-side transparency, our design uses 802.11-based access points operated in the *infrastructure* mode. Thus, we refer to our architecture as *iMesh*. This is a departure from the common use of *ad hoc* mode in the experimental study of multihop wireless networks using 802.11-based radios (for example, used in experimental studies reported in [7, 14, 11]. This choice enables us to design the system without any specialized software on the mobiles. If the ad hoc mode were used, when a mobile moves away from an AP, it must find another appropriate "next hop" AP for forwarding its packets. To do this, it must possess the ability to discover its neighborhood via some layer-2 or layer-3 functionality that typical clients do not implement in the ad hoc mode. Thus, use of ad hoc mode will need client device configuration with appropriate software.

A mesh network could be built simply by using *bridging.* Learning bridges run a spanning tree protocol to create a forwarding table for nodes by observing traffic at different links [20]. Bridging is a layer-2 alternative to routing; but it has some inherent problems. Bridges are unable to handle hierarchies in the network and hence do not scale well. Bridges use broadcast until they learn a route and are thus slow to converge. In addition, they are not suitable to handle client-side transparency because the AP will not learn about a client until the client sends a packet to it. Due to these limitations, we propose a layer-3 solution for the mesh network.

This requirement forces us to use infrastructure mode of operation on the AP. It is now sufficient to use the underlying "handoff" capability of the 802.11 client devices to handle mobility. Thus, the client's view of the network is still that of a wireless LAN, while the distribution system (DS) connecting the access points is now made of a wireless backbone network or *Wireless Distribution System* (WDS) [2].

Use of a software-based AP is vital to our design – so that the functionality modifications for the AP and provisioning of the layer-3 handoff process (including the routing protocol) can be done easily. The HostAP device driver [13] and an embedded version of Linux are used to obtain a software-based AP functionality. Here, various AP functions such as authentication (and deauthentication), association (reassociation, and disassociation), support for WDS etc., are supported as a part of the device driver software.

2.1. Link Layer Handoff

When a station moves out of range of an AP, it triggers a link layer handoff to search for and reassociate with a new AP. The exact condition that triggers a handoff is implementation specific. Handoff is often associated with *probing*. Probing proactively seeks APs to associate with instead of waiting to hear beacon signals from AP. In probing, the client broadcasts a *probe request* frame. APs on the same channel respond with *probe response* frames. The client waits for certain amount of time (*probe-wait time*) to collect all the probe responses. Then, the client can switch to other channels to probe. After probing a set of channels (possibly all available channels), the client selects one AP with the best SINR (signal-to-noise ratio) based on the probe responses. After probing is complete, the station authenticates with the new AP. Following successful authentication, the station initiates *reassociation* with the new AP to exchange information about the connection such as transmission rates, beacon intervals, etc. It sends a *reassociation request* frame to the AP that responds with a *reassociation response* frame. At this point, the link-layer handoff completes. It turns out that the major factor in the handoff delay is the time spent in probing and waiting for probe responses. Several research studies have investigated link layer handoff latency in 802.11-based wireless LAN and various optimizations [15, 23, 22]. Our work has benefited a lot from these experiences.

2.2. Network Layer Handoff

In the *iMesh* architecture the APs form a multihop network routable at the IP layer.[2] This gives rise to a mobility management problem – how to deliver frames destined to a station when its point of attachment to the mesh network (i.e., the AP) has changed. Two broad approaches are possible that we both implement in our testbed and compare. The first approach uses a technique similar to mobile IP [19], where each station has a unique "home" location or a home AP. The network implementing the DS keeps track of the mobile stations. Packets destined for the station is still delivered to the "home" AP for propagating to the mobile station. It is now the home AP's responsibility to forward the packet to the AP the mobile station is currently "visiting." This is achieved by a protocol called *Transparent Mobile IP* or *TMIP* [25]. The significant difference from the standards-compliant Mobile IP is that the mobile station does not need to implement any specific protocol. This preserves the transparency we desire. There is a centralized server in the network called *Mobile Location Register (MLR)* which keeps the information about the "home" AP for every mobile station. When the mobile hands off to any "foreign" AP, the foreign AP sends a query to the MLR to find out about its "home" AP. The foreign AP then notifies the home AP about the new endpoint of the mobile with a message handshake, and adds a new, one-hop route to the

---

[2] Note that it is possible to do the routing using purely MAC addresses and using ARP and proxy ARP techniques [7].

mobile. It also sends a gratis ARP response to the mobile so that the mobile updates its MAC address for its default gateway (which is still the home AP) and makes it the same as the foreign AP. Beyond this point, packets directed to the mobile are intercepted by the home AP and "tunneled" (using IP-in-IP encapsulation) to the foreign AP. The communication from mobile AP, on the other hand, can proceed in the normal fashion without involving the home AP. Note that TMIP makes it possible that the mobile station keeps the original IP address even in its "foreign" location.

While the transparent mobile IP approach is straightforward, the forwarding path for the mobile is clearly not optimal due to the so-called "triangular routing" scenario. The approach that we promote in this paper is to use a full-fledged multi-hop routing infrastructure in the network of APs in the DS. The routing infrastructure is "flat"; the routing tables in the APs contain the IP addresses of the all the mobile stations in the system. Optimizations are possible for very large-scale networks to limit the size of the routing tables, though we do not discuss these here. The basic idea of maintaining the routing infrastructure is to use any handoff as a trigger to generate and propagate necessary routing updates. Thus, the network layer handoff consists of completion of notifications for the Transparent Mobile IP case, and convergence of routing updates for the flat routing case.

In the *iMesh* testbed we have chosen a link-state based routing protocol, called Optimized Link State Routing or OLSR [5]. The OLSR protocol runs on all WDS interfaces at every AP. Note that separate *logical* WDS interfaces are created for each neighboring AP. The AP does not run OLSR on its client side interface (the logical interface the client associates to – typically `wlan0`) as the client is unaware of the routing. The link between the AP and mobile station is treated as an *external route* to the mesh network. The OLSR protocol advertises such external routes via the so-called HNA (Host and Network Association) messages [5] designed specifically to inject external routes to the mesh network.

Whenever a mobile station associates with an AP, the HostAP driver sends an association signal to the OLSR daemon, which deletes all pre-existing routes to this station and adds a "direct" route to the client via its `wlan0` interface. This "external" route information is encoded as an HNA message and broadcasted in the network via the OLSR protocol. All APs, on receiving the HNA message, delete all pre-existing routes to this station and add a new route via the AP to which it is currently associated. Also, on receiving HNA, the AP deletes the information about this station from its local database of external routes.

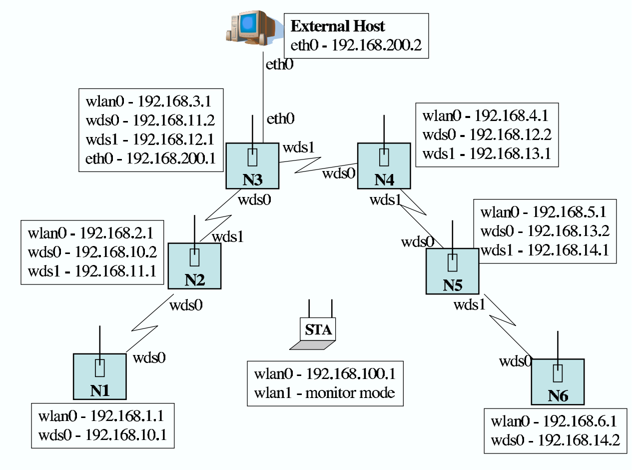

Figure 1. ***iMesh*** **testbed used for performance evaluation.**

A few words are due here about address assignment and how it is related to triggering network layer handoff. When a new mobile station joins the network, say by booting up and associating with one of the APs, the station acquires an IP address via DHCP from the address pool of this AP (each AP maintains an independent address pool). Let us denote this AP by AP1. The mobile uses AP1 as its default gateway. AP1 maintains a mapping of IP address to MAC address of mobile stations in an *IP-to-MAC address mapping table*. AP1 then adds a host-specific route to this mobile in its kernel routing table. At this stage, the mobile station has complete uplink connectivity. AP1 then advertises the new route to the mobile to all other APs in the mesh network through a link state update via the OLSR protocol. The MAC address of the mobile is included in the update message so that all APs in the mesh network can add the IP-to-MAC mapping in their own IP-to-MAC address mapping table. Downlink connectivity is available to the mobile at the end of the link state update, as at this point all APs in the network have a host-specific route to the mobile station with AP1 as the last hop node in the route.

The IP-to-MAC mapping is required to be distributed for a reason. When the mobile reassociates to another AP in future (say, AP2), AP2 must be able to determine its IP address so that it can send out routing updates appropriately. For this to happen without any involvement from the mobile (this is required for transparency), AP2 must know the mapping in advance.

3. Testbed and Performance Evaluation

Our *iMesh* testbed uses Soekris net4521 [24] boards running on Pebble Linux V41 distribution [18] (a small Linux kernel suited for embedded devices) with the Linux-2.4.26 kernel. Our testbed uses six APs and one mobile station. For programming and debugging convenience we have not used Soekris boards in all the APs for the experiments reported in this paper. Four of the APs and the mobile station are IBM Thinkpad R-series laptops running Redhat 8 distribution with Linux-2.4.22 kernel. Each AP is equipped with a single Prism 2.5 chipset based 802.11b PCMCIA card (Netgate and US Robotics). Each card is connected via a pigtail to an external antenna. We use the new firmware version v1.5.6 for the cards since it supports the 4-address WDS frames. As mentioned before, we have used our customized version of open source HostAP driver (0.1.3 version) on each of the APs. The wireless interface on each AP is configured to work in infrastructure mode.

For ease of performing experiments, all nodes are deployed close to each other in the same laboratory room and WDS links are formed manually to control the topology of the mesh network. We set up a linear topology as shown in Figure 1. $N1$ to $N6$ are the APs with WDS links. $N3$ is also connected to an external host via the Ethernet interface. STA is the mobile station. For each node in the network, the logical interfaces (`wds0`, `wlan0` etc.) are shown with their assigned IP addresses. The routing protocol operates on the `wds` interfaces. The `wlan` interfaces are for AP to mobile communication.

The mobile station STA in the experiment is actually kept stationary, and its movement is “simulated” by changing its association to the APs through a script. This makes the experiments repeatable, and stable performance data could be collected. Note that this puts all nodes in the same collision domain and throughput performance suffers. Thus, we will be at best *underestimating* the performance numbers. The reader will soon note that excellent handoff latencies are obtained nevertheless, which are the main results of this paper. Both layer-2 and layer-3 handoff latencies are equal or better than that recently reported in literature [22, 23, 6, 21] for wireless LAN testbeds where layer-3 procedures, when they exist, are run on “wired” ethernet.

The mobile station has two wireless interfaces: PCMCIA card and MiniPCI card. The PCMCIA card is configured as a client. This interface is used for associating with an AP. The MiniPCI card is configured in the *RF monitor mode* to sniff all packets on the channel. We used this interface to collect packet traces during the experiments, enabling us to measure latencies of layer-2 and layer-3 events by a “post-mortem” analysis of packet traces.

Two flavors of wireless interface driver software are used on the client card – regular HostAP driver working in the client mode and a modified *airjack* driver [4]. The airjack driver is capable of sending and receiving management frames in software. Note that specialized drivers are used only to facilitate experimentation (associations need to be changed under program control) and to analyze layer-2 handoff latencies better. The *iMesh* network operation is independent of any specialized support on the client side. To simplify operation, the authentication mode is configured as open authentication for all cases.

In the following, we report the performance of the *iMesh* architecture with the above setup running the OLSR protocol for routing. For comparison we have also ported Transparent Mobile IP [25] or TMIP to our testbed. Note that while we promote a “flat” routing protocol like OLSR for performance reason, the *iMesh* architecture can easily support other layer-3 schemes such as TMIP.

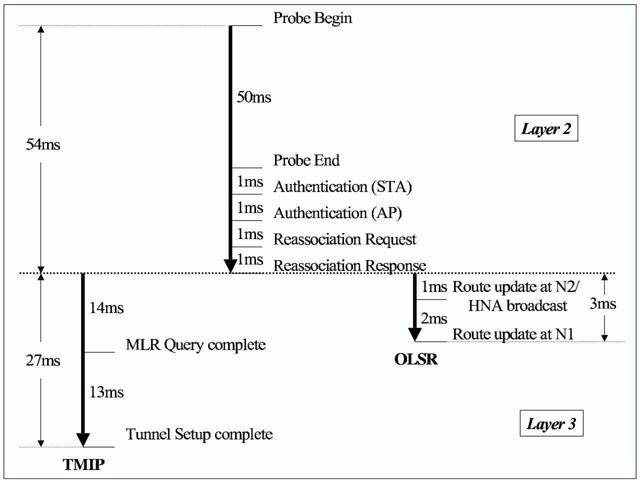

Figure 2. Timeline describing a typical fast one-hop handoff from $N1$ to $N2$ for a mobile station with *iMesh* running OLSR and TMIP.

3.1. Measuring Handoff Latency

In this subsection, we report measurements of handoff latency for a mobile station (STA) as it switches its association from one AP (oldAP) to another AP (newAP). Handoff begins when the STA loses its association with oldAP (in our case, by receiving a deauthentication frame at oldAP) and completes when STA associates with the newAP.

In our *iMesh* architecture with the OLSR protocol, layer-3 handoff starts with the advertisement of a new HNA route to STA by the newAP. The OLSR protocol handles the broadcast of this message in the mesh network. Layer-3 handoff completes when the routes at all APs have been updated to reflect the newAP as the new point of attachment for STA. The handoff delay here depends on the number of route changes and the distance of these changes from the newAP. We have used a linear topology of APs (Figure 1) to experiment with various distances (in number of hops) of these route changes while keeping the size of the testbed reasonably small.

In case of TMIP, the “foreign” AP (i.e., newAP) for the STA first determines the IP address of “home” AP (i.e., oldAP) by querying the central server (which is the AP $N3$ in Figure 1) that implements the *Mobile Location Register* or MLR. It then notifies the home AP so that the home AP can tunnel packets to the foreign AP. Layer-3 handoff completes when this procedure is completed.

The RF monitoring interface on the STA collects a timestamped trace of all exchanges in the wireless channel including the management frames. The trace is later analyzed to determine actual handoff delays in the layer-2 and layer-3. Figure 2 depicts the handoff timeline with a representative set of timing measurements. Note that the layer-2 handoff delay is dominated by the probe delay as explained before and examined critically in recent literature [15, 22, 23]. The timeline shows about 50ms delay for probing. A single channel is probed. In case the methods used in [22] are applied in our testbed, only one channel will be probed. Recall that the entire network operates on the same channel in this set of experiments. If multiple channels need to be probed, the 50ms probe time will be multiplied by the number of channels to be probed, and the handoff delay will increase accordingly. The authentication and reassociation activities are very fast and are done in a fraction of time relative to the probe delay. The layer-2 handoff with a single channel probe completes in about 54ms.

The timeline also shows layer-3 handoff delay for the one hop route change case. This happens, for example, when STA moves from $N1$ to $N2$. In this case the handoff completes when the route is updated at $N1$, as the routes do not change in any other AP. This takes just 3ms. On the other hand, for the case of TMIP, an order of magnitude longer time is taken to complete the MLR query and the notification process with the home AP so that packets can be tunneled.

We now present a set of handoff latency measurements for handoffs for different hop lengths of route changes. For these studies, $N1$ is always the oldAP. The newAP is one of the five remaining nodes thus making up to five hop route changes. We think that our experiments are quite comprehensive, as in a deployed mesh network, route changes are unlikely to be any more than a few hops, if we assume that there are no significant coverage holes.

Three sets of handoff results are presented in Figures 3, 4 and 5. The results for both architectures are presented without probing and with single channel probing. Different wireless device drivers were used to implement these schemes. To implement the no probing case – that provides the fastest layer-2 handoff – the HostAP driver is used in client mode with probing disabled. Thus, the layer-2 handoff in this case involves only authentication and association frame exchanges. To implement the single channel probing case, a modified airjack driver [4] is used. The *MinChannelTime* and *MaxChannelTime* intervals for the *probe-wait time* timer is set to 20ms and 30ms respectively.

Note that while these techniques aggressively reduce layer-2 handoff delay, they are not unreasonable. Research in [22, 23] has shown how probing can be limited using prior knowledge. Such techniques will use only a single channel probe in our testbed. No probing can be reasonable in novel systems where beacons from APs can be used to learn the neighborhood of different APs and then to handoff to another AP in the same channel. Regardless of how layer-2 handoff is designed, our emphasis indeed is on layer-3 handoff in this paper because of the requirement of client-side transparency and the fact that handoff in 802.11 is controlled by the client in the most part.

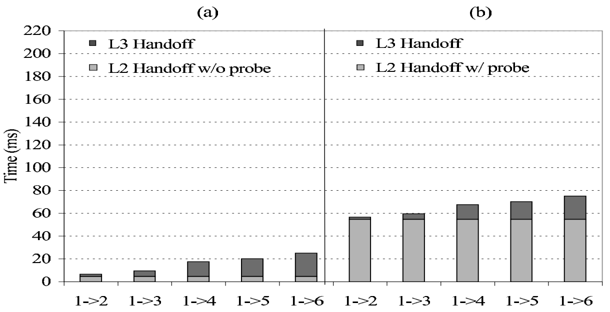

Figure 3. Handoff latencies for *iMesh* using OLSR routing with no background traffic for (a) without probing and (b) with probing on a single channel. The notation $i \to j$ indicates hand-off from Ni directly to Nj.

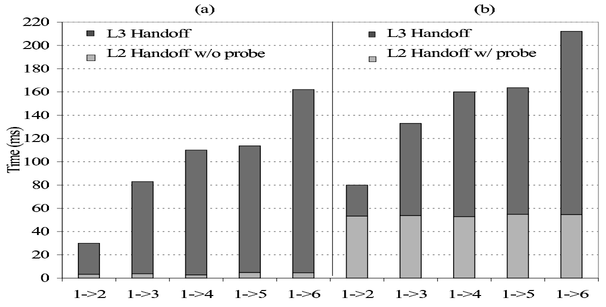

Figure 4. Handoff latencies for TMIP with no background traffic for (a) without probing and (b) with probing on a single channel.

Looking at Figures 3 and 4 layer-2 handoff delays are independent of amount of route changes as expected. Layer-3 delays on the other hand is proportional to the number of nodes along the path between newAP and oldAP in the case of *iMesh* with OLSR. In case of TMIP, latencies are much higher as the messages communicated between newAP and MLR and between newAP and oldAP have to travel over longer paths. Note that the absolute values of handoff delays are excellent when *iMesh* is used with OLSR. The maximum layer-3 latency is noted for a five hop long route change, and that is around 40ms (with less than 100ms total handoff latency), while one hop route change is accomplished within about 3ms (with less than 60ms total handoff

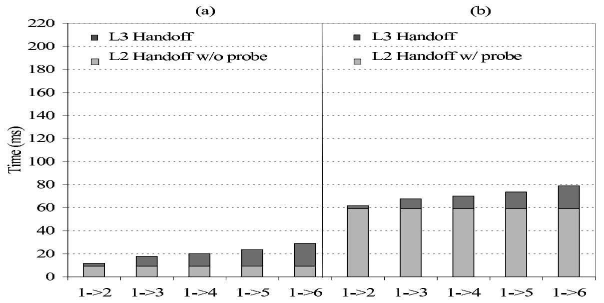

Figure 5. Handoff latencies for *iMesh* using OLSR routing with background traffic of 2.5Mbps for (a) without probing and (b) with probing on a single channel.

latency). When probing is used, layer-3 handoff is actually faster than the layer-2 handoff, indicating that layer-2 handoff optimizations are more important. Note that these handoff latencies are comparable or better than observed on recent handoff studies on wireless LANs [22, 23, 21].

To evaluate handoff latencies in presence of network load, we set up a 500Kbps UDP flow between $N1$ and $N6$ resulting in a total background traffic load of 2.5Mbps. As shown in Figure 5, the performance of handoff latency increases by just a few milliseconds in presence of this traffic.

3.2. Round Trip Time Experiments

We now turn our attention to analyzing the impact of shortest path routing in OLSR vs. use of triangular routing in TMIP by measuring round-trip times. In this experiment a “walk” of the mobile station STA is simulated. Initially, STA is associated with $N1$. At 10 seconds intervals it changes its association to $N2$ through to $N6$, and then retraces its “path” back to $N1$ in the same fashion, changing associations at 10 seconds intervals. The total walk duration is 110 seconds and it involves 11 handoffs.

During the walk, STA continuously pings an external host connected to the gateway $N3$ at 100ms intervals using 1500 byte ICMP packets and measures the RTT value for each ping. Figure 6 depicts the RTT values for TMIP and OLSR-based routing for different packet sequence numbers. For the TMIP case, $N1$ is the home AP for STA and the MLR is running on $N3$. The path length between STA and external host is optimal for iMesh at every instant since shortest path routes are used. With TMIP, when the “foreign” AP is either $N2$ or $N3$, the path lengths are the same as that of OLSR and hence both schemes have similar RTT values. The reason for triangular routes not being used in these cases is that the packets from external host to STA encounter the foreign AP along the path to the home AP and are directly forwarded to the STA instead of being sent to home AP. When foreign AP is one of $N4$, $N5$ or $N6$, outgoing ping request packets from STA choose the shortest route to external host while the ping response packets first reach $N1$ and then are tunneled back to the foreign AP and thus have to travel six more hops than the OLSR case. The spikes at the beginning of each handoff for TMIP occurs when a tunnel to an old foreign AP is deleted and a tunnel to new foreign AP is created. All in-flight ping responses that reach an old foreign AP are forwarded back to the home AP and then tunneled to the new foreign AP.

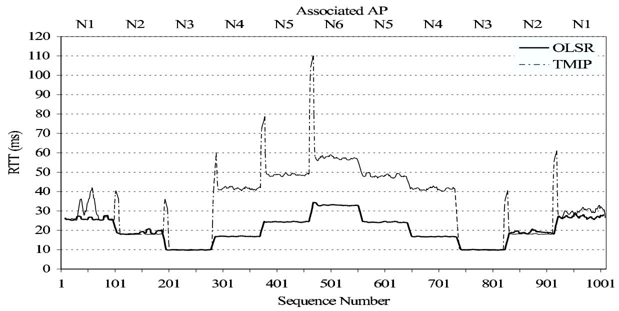

Figure 6. RTT Measurements for ping packets for *iMesh* with OLSR and TMIP. Handoffs occur at intervals of 10 sec.

4. Related Work

There are various industry initiatives to provide mesh networking support either from a service provider or from an equipment manufacturer point of view. However, very little public information is available about the architectural choices and performance [10, 26]. On the other hand, there is some documentation on building and using mesh networking concepts in the research community. See, for example, the Roofnet project at MIT [16] and mesh networking research in Microsoft Research [17]. Roofnet is a collection of wireless routers that are stationary PCs running Linux with 802.11-based interfaces in the ad hoc mode. The network runs a routing protocol, SrcRR [3], that is based on the well known Dynamic Source Routing (DSR) protocol [12]. Mesh networking research in Microsoft Research uses a testbed similar to Roofnet with 802.11 interfaces running in ad hoc mode. They addressed the question of appropriate routing metrics [8] and use of multiple radios on each node [9]. These projects only consider the backbone mesh network, and do not directly provide any support for seamless mobility for mobile stations that is the main focus in our current work.

The wireless LAN research community has addressed the question of fast layer-2 handoffs in wireless LANs. The issue here is to save channel probing times to determine the best AP to handoff to. See, for example, [22, 23]. Cleyn et. al. [6] and Sharma et. al. [21] independently have considered efficient layer-3 hand-offs for wireless LANs, but only in the context of standard-compliant mobile IP [19]. Our layer-3 hand-off latencies are much superior relative to these reported studies.

Recently, the IEEE 802.11 committee has started a new working group (802.11s) for the so-called *ESS mesh* networking [1]. The purpose of this new 802.11s ESS mesh working group is to provide a protocol for auto-configuring paths between APs over self-configuring multi-hop topologies to support both broadcast/multicast and unicast traffic. This solution also uses WDS links.

5. Conclusions

We have presented an 802.11-based "infrastructure-mode" mesh networking architecture called *iMesh*. The goal of our design is client-side transparency, so that existing mobile clients can seamlessly use such a mesh network *in lieu* of a wireless LAN. The fundamental design concept is the use of a flat routing protocol in the mesh network that is triggered by reassociations by a mobile station at wireless access points. This ensures that the optimal path to the mobile can be maintained at all times. We analyzed the performance of the *iMesh* architecture in a six node 802.11b-based mesh network. We presented detailed experimental results involving measurements of handoff latencies at both layer-2 and layer-3. The flat routing demonstrates excellent latency performance relative to a more traditional layer-3 handoff technique using a mobile IP like scheme, called transparent mobile IP. The layer-3 latency for the routing scheme is faster by a factor of about 3–5. If absolute performance is of concern, the routing scheme provides a combined layer-2 and layer-3 handoff latency of less than 50–100ms (depending on the layer-2 technique used) even when the route change involves route updates over five hops and some background traffic is present. This measurement includes certain layer-2 optimizations reported in literature. We consider this an excellent performance relative to recent studies on wireless LAN.

6. Acknowledgments

This work was partially supported by a grant from Computer Associates International, Inc., and by the National Science Foundation (award number ANI-0308631).

References

[1] Amendment to the current 802.11 standard to provide Extended Service Set Mesh Networking. http://standards.ieee.org/board/nes/projects/802-11s.pdf.

[2] IEEE 802.11b/d3.0 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, August, 1999.

[3] D. Aguayo, J. Bicket, S. Biswas, D. S. J. D. Couto, and R. Morris. MIT Roofnet Implementation. http://www.pdos.lcs.mit.edu/roofnet/design/, 2003.

[4] R. Baird and M. Lynn. Airjack Driver, Version 0.6.2-alpha. http://sourceforge.net/projects/airjack.

[5] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, October 2003.

[6] P. D. Cleyn, N. V. den Wijngaert, L. Cerda, and C. Blondia. A smooth handoff scheme using IEEE802.11 triggers: design and implementation. *Computer Networks*, 45(3):345–361, 2004.

[7] S. Desilva and S. R. Das. Experimental evaluation of a wireless ad hoc network. In *Proc. of IEEE IC3N*, pages 528–534, Oct. 2000.

[8] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. *SIGCOMM Comput. Commun. Rev.*, 34(4):133–144, 2004.

[9] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MOBICOM*, pages 114–128. ACM Press, 2004.

[10] Firetide, Inc. http://www.firetide.com.

[11] R. S. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan. Outdoor experimental comparison of four ad hoc routing algorithms. In *Proc. of MSWiM*, pages 220–229, October 2004.

[12] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile computing*, chapter 5. Kluwer Academic, 1996.

[13] J. Malinen. Host AP driver for Intersil Prism2/2.5/3. http://hostap.epitest.fi.

[14] D. Maltz, J. Broch, and D. Johnson. Lessons from a full-scale multihop wireless ad hoc network testbed. *IEEE Personal Communications Magazine*, 8(1):8–15, Feb. 2001.

[15] A. Mishra, M. Shin, and W. Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33(2):93–102, 2003.

[16] MIT Roofnet. http://www.pdos.lcs.mit.edu/roofnet.

[17] Networking Research Group, Microsoft Research. http://research.microsoft.com/mesh.

[18] Pebble Linux. http://www.nycwireless.net/pebble.

[19] C. Perkins. IP mobility support, RFC 2002, October 1996.

[20] R. Perlman. *Interconnections: Bridges, Routers, Switches and Internetworking Protocols*. Addison-Wesley, 1999.

[21] S. Sharma, N. Zhu, and T. cker Chiueh. Low-latency mobile ip handoff for infrastructure-mode wireless lans. *IEEE Journal of Selected Areas in Communications*, 2004.

[22] M. Shin, A. Mishra, and W. A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *Proc. of ACM MOBISYS*, pages 70–83. ACM Press, 2004.

[23] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne. Reducing MAC layer handoff latency in IEEE 802.11 wireless lans. In *Proc. of ACM MOBIWAC*, pages 19–26. ACM Press, 2004.

[24] Soekris Engineering. http://www.soekris.com.

[25] Transparent Mobile IP. http://www.slyware.com/projects_tmip_intro.shtml.

[26] Tropos Networks. http://www.tropos.com.