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
Email:{vnavda,anand,samir}@cs.sunysb.edu

***Abstract*— Wireless mesh networks are multihop networks of wireless routers typically used for wireless coverage over a large community. Applications include community-scale peer-to-peer networking or sharing of a limited number of high-bandwidth wired gateways. In this paper, 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 as layer-3 handoffs. We describe the design rationale, including address assignment, hand-off and routing techniques. We also describe a testbed implementation of *iMesh* and analyze the handoff performance, as well as UDP and TCP performance when frequent handoffs are present. The performance 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. Overall, the *iMesh* architecture demonstrates the feasibility of supporting seamless mobility in a wireless mesh network even in presence of frequent handoffs.**

I. Introduction

In this paper, we investigate a mesh networking architecture as an alternative to wireless LANs based on IEEE Standard 802.11 [1]. 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 networking interfaces as well as access point/router platforms, mesh networking will become as ubiquitous as wireless LANs, and will "blanket" communities with wireless coverage at a low cost. Several usage scenarios have been envisioned [2]. Examples include (i) broadband connection sharing for "last-mile" access; (ii) neighborhood or community mesh networking, where a mesh network parallel to the Internet is used for applications of local relevance, such as sharing data or multimedia, or collaborative backup; (iii) community-wide or metropolitan-area wireless networks specifically used for niche applications, such as law-enforcement, emergency management or traffic systems; (iv) any application where rapid deployment of a wireless network is desired over a wide area. Noting these usage scenarios and their potential economic advantages, several companies are exploring commercialization of mesh networking, such as Tropos [3], Packethop [4], Meshnetworks [5], Firetide [6] etc. There are several community initiatives as well using commodity 802.11-based hardware platforms (see, for example, [7], [8], [9]).

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

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

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 [10] like solution called *Transparent Mobile IP* [11] 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 and their impact on transport protocol performance.

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

II. System Architecture

For providing a 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 [12], [13], [14]. 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. Note that ad hoc network routing implementations use such a mechanism to detect route breaks. For example, the existing link with the old access point can be considered broken when ACKs do not come back in the MAC layer of 802.11 even after repeated transmission attempts. Similarly, layer 3 beacon/hello messages can be used. While these may be straightforward to accomplish, the approaches need the client device configured with appropriate software.

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 are now made of a wireless backbone network or *Wireless Distribution System* (WDS) [1]. The network interfaces at the neighboring access points use WDS links to communicate between them. A neighbor discovery protocol listens to the layer-2 beacon messages that a 802.11 interface emits in the infrastructure mode in order to configure the WDS links to the neighboring APs. A multihop routing protocol maintains end-to-end connectivity. When a mobile client moves and reassociates with a new AP, the reassociation triggers routing updates in the network so that the packets destined for the mobile can be delivered to this new AP for transmitting to the mobile. The *iMesh* network architecture is shown in Figure 1.

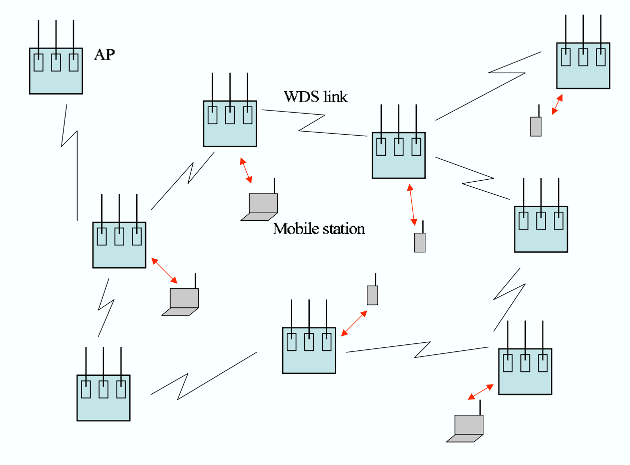

Fig. 1. The *iMesh* architecture. Each AP can have multiple wireless interfaces (three shown) tuned to different bands/channels. These interfaces form a wireless backbone network using WDS links. The mobile stations associate with a nearby AP (links shown using straight arrows) as in a regular wireless LAN unaware of the mesh routing architecture.

The access points are not mobile and are typically powered from a power outlet [2]. This brings about a couple of important design choices. First, the routing protocol can be proactive, such as based on traditional *link-state* or *distance-vector* approaches rather than on-demand approaches studied in connection with mobile ad hoc networks [15]. This is because the mesh network topology will be stable over longer time scales. Second, it is possible now to configure multiple wireless interfaces on each access points, as optimizing power consumption on access points is no longer an important design goal. The interfaces could now be tuned to different bands (say, 802.11b, g or a) and channels within the same band for bandwidth aggregation and/or to exploit channel diversity in various interesting ways. Use of multiple interfaces on access points, however, gives rise to an interesting channel assignment problem.

A WDS link between two neighboring access points can only exist when they have at least one interface on a common channel. Our design accommodates use of multiple interfaces per access point. However, a solution of the channel assignment problem is beyond the scope of this article. Some solutions are available in current literature [16], [2].

An important component of the *iMesh* architecture is how handoffs are handled. This is critical to support seamless mobility. In the following, we describe the approaches in two parts – link layer (layer-2) and network-layer (layer-3) handoffs.

A. 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. For example, a client can initiate a handoff when it fails to communicate with the AP it is currently associated with. Or, the handoff initiation can be more proactive. For example, the client can continuously do signal strength measurements for the beacon messages from APs that it is hearing on the current channel. If the signal strength of the AP it is currently associated with falls below a threshold and the signal strength from another AP is sufficiently higher, the client may trigger a handoff to the second AP. The second condition avoids ping-ponging between two APs due to slight fluctuation of signal strengths.

Handoff is often associated with *probing*. Probing proactively seeks APs to associate with instead of waiting to hear beacon signals. This is because beacon intervals can often be too high (e.g., more than 100 ms). Also, there may not be any AP to associate to in the current channel. 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.

Several research studies have investigated link layer handoff latency in 802.11-based wireless LAN and various optimizations [17], [18], [19]. Our work has benefited a lot from these experiences. It turns out that the major factor in the handoff delay is the time spent in probing and waiting for probe responses. Since probe responses may come back at different times (as they go through backoffs in the MAC layer to avoid collisions) too small a *probe-wait time* may miss important probe responses. Also, it is possible that the best possible AP to handoff to is on a different channel than the mobile station is on currently. Thus, the probing must be done in different channels in a sequential fashion. In each step, channel switching also adds to the delay. The probing can be optimized by only probing a small set of channels by exploiting prior knowledge [18], [19]. This has been shown to substantially minimize link layer handoff latency.

B. Network Layer Handoff

The original IEEE 802.11 standard [1] specified only the MAC and PHY layers of a WLAN system and defined the basic architecture, including the concepts of APs and DSs. However, the inter-connection and inter-operation of different APs in a DS was left as an implementation choice. Later, as 802.11 systems grew in popularity, certain DS related functions in the APs (particularly related to handoff and state exchanges between the old and the new APs during handoff) were specified as an extension – *Inter-Access Point Protocol or IAPP (802.11f)* [20]. The goal was to make APs from different vendors inter-operate across a DS. A very generic DS architecture is assumed, and the inter-AP communication was assumed to run over TCP/UDP/IP. An example of such inter-AP communication relevant to our work is the *move request* message from the new to the old AP after reassociation and a corresponding *move confirm* message from the old to the new AP with any context information (e.g., related to security) to be transferred. However, IAPP does not assume any particular architecture on the part of the DS, and does not specify how the inter-AP communications should be routed or even how the APs should get an IP address for such IP-based exchange to work. These are left as implementation choices.

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

[2] Note that it is possible to do the routing using purely MAC addresses and using ARP and proxy ARP techniques intelligently [12].

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 [10], 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* [11]. 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 mobile 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 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 are “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, as 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 [21]. Link state based protocols have the advantage that loop-freedom is easy to derive. Also, having a complete link state database in each node in the network makes it easy in future to use complex routing policies and metrics. As will be discussed in the related work section, several mesh networking initiatives also do use link-state protocols.

III. Implementation Details

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 [22] and an embedded version of Linux are used to obtain a software-based AP functionality. In HostAP driver, there is an AP mode, other than the more usual client mode. In the AP mode various AP functions are supported in software – such as authentication (and deauthentication), association (reassociation, and disassociation), power saving (PS) mode, signaling and frame buffering for PS mode, support for WDS etc. The driver has also various features for development, debugging and researching IEEE 802.11 environments like access to hardware configuration records, I/O registers, and frames with 802.11 headers. Presumably, other software-based AP architectures such as Microsoft’s native WiFi [23] can be used to build *iMesh.*

In the following, we elaborate on four relevant implementation details for *iMesh*. They are related to (i) auto-configuration of APs at startup, (ii) triggering of network layer handoff, (iii) implementation of the routing protocol, and (iv) packet buffering during handoff to improve performance. A block diagram of the AP protocol stack is shown in Figure 2. This figure will be occasionally referred to in the following sub-sections.

A. Auto-Configuration at Start-up

At start-up, the APs self-configure automatically to join the mesh backbone. The goal here is to automatically discover and create high quality symmetric WDS links with the mesh neighbors. The steps taken are as follows.

1) Configure the wireless interfaces to run as AP (i.e., in Infrastructure Mode) on a pre-defined channel and set the ESSid to a predefined string. This defines the identity of the ESS (extended service set) to be used by the mesh network.
2) Listen for beacons from neighboring APs for a configurable amount of time.
3) On hearing a beacon from an AP, check if this AP belongs to the same ESS by looking at the ESSid field of the beacon frame. If it does, then check the signal strength of the received beacon

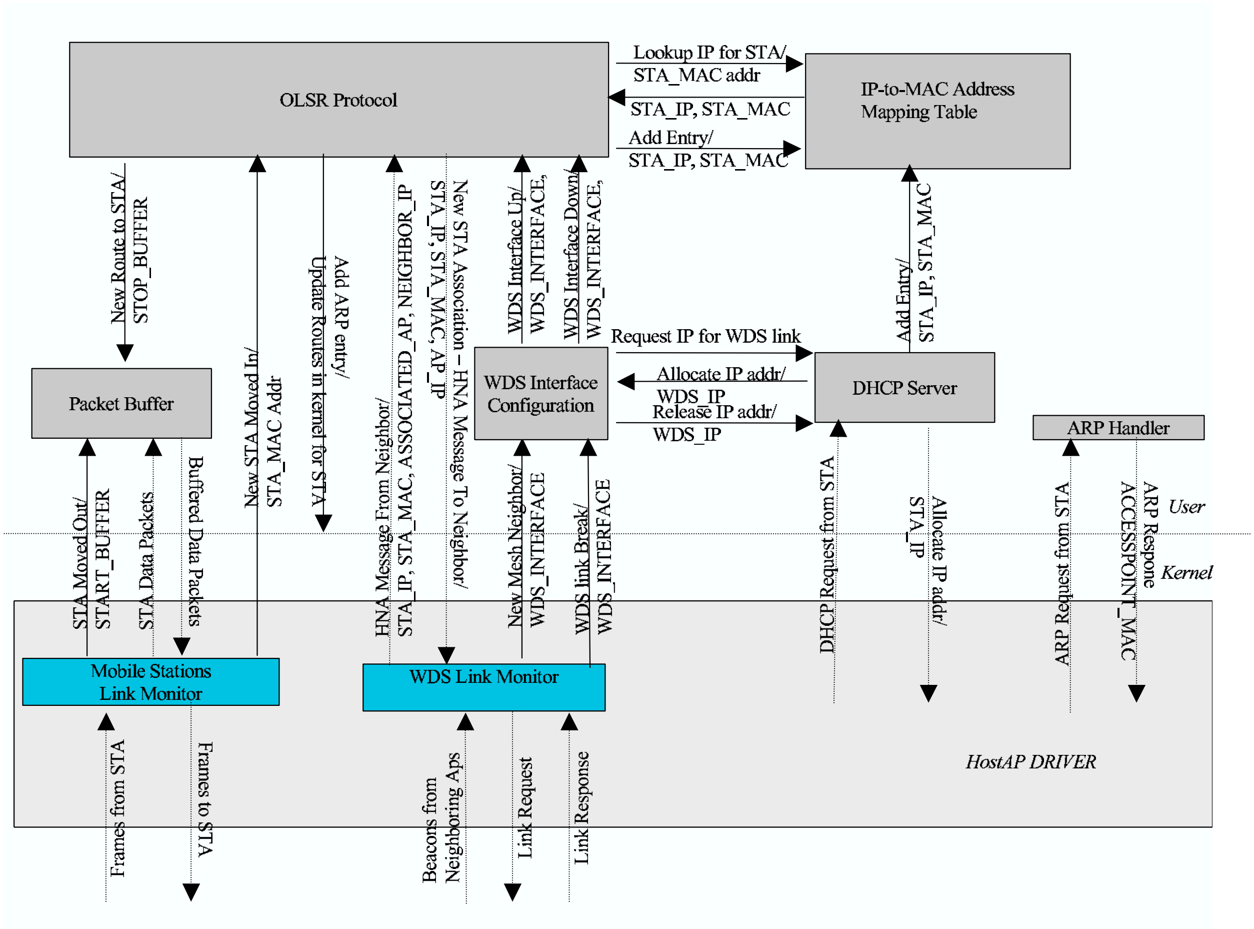

Fig. 2. Block diagram of major software components in an *iMesh* access point.

frame. If it is above a certain threshold value, then initiate the creation of a new WDS link with this AP. This involves a message handshake between the two APs. Send out a *Link Request* message unicast to the remote AP. This is a normal data frame using WDS addressing with a specific tag in the ethernet header so that the receiver can identify this as a handshake frame.

At the Remote AP side, on receiving the *Link Request* frame, check again if the quality of the signal for the received frame is above the threshold. If the signal quality criterion is satisfied, the remote AP responds with a *Link Response* frame. At the end of a successful handshake, each of the APs create a new WDS link. This process involves creation of a logical network interface by the HostAP driver that is exposed as a new interface to the upper layers (see the module *WDS Interface Configuration* in Figure 2). Upper layer protocols running on the AP can use a WDS interface (note that this is a logical interface) to communicate with a peer AP. For a packet sent on such a logical interface by upper layers, the HostAP driver encapsulates the packet with the four address WDS header with destination address set to the WDS peer for the link.

4) Before a WDS interface can be used, it has to be assigned an IP address. We use a *user-level* DHCP server module running on each AP. Each of these servers are allocated a unique pool of IP addresses to allocate to the WDS links as well as to the mobile stations. When a new WDS link is created, the *WDS Interface Configuration* module delivers a *Request IP for WDS Link* event to the DHCP server for it to allocate an address. See Figure 2.
5) As soon as an IP address is assigned, a trigger is passed on to the routing module to include the interface in the routing protocol (*WDS Interface Up* in Figure 2).

B. Triggering Network Layer Handoff

The access points are implemented by suitably modifying the HostAP driver. The default handoff functionality in HostAP is switched off and handoff is implemented using *iMesh*-specific software implementation.

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. 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 this new route 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.

This IP-to-MAC mapping is required to be distributed for a reason. Recall that our design goal is transparency to the mobile station. When the mobile hands off to another AP, this new AP must initiate routing updates so that the packets destined for mobile station can now be delivered via the new AP (say, AP2). However, since routing uses IP addresses, AP2 must learn the mobile's IP address right after reassociation.

Consider the scenario where the mobile station has an active connection to another host in the mesh network or a host in the Internet via a gateway AP. Initially, it was associated with AP1 as described before. Now assume that it reassociates to AP2. The HostAP driver learns about this new association and notifies the OLSR protocol (*New STA Moved In* in Figure 2), which in turn determines the mobile's IP address by looking up the *IP-to-MAC address mapping table*. Since a mobile station can be associated with only one AP at one time, AP2 deletes the existing route of the mobile (that would be using AP1 as the last hop node) and adds a new one-hop route. Then it triggers a link-state update. The OSLR protocol takes care of propagating the update network-wide and route recalculations at every other AP in the network.

C. Routing

The OLSR protocol runs on all WDS interfaces at every AP. Note again 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 [21] 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 (*New STA Moved in* in Figure 2) 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.

As mentioned before, the IP-to-MAC layer mapping for an associated station needs to be propagated across the network too. The IP-to-MAC mapping for that station is piggybacked on the route update. To accomplish this, we change the HNA message header (destination IP address and subnet mask) to contain an extra field for the MAC address of the destination, which in this case is the mobile station.

D. Packet Buffering

When a mobile station initiates a link layer handoff, it loses connectivity with the old AP and establishes connectivity with the new AP. As a result, when handoff is in progress, packets sent out by the old AP to the mobile station are lost. Buffering can be used to alleviate the packet losses during handoff.

We have implemented a packet buffer in user space using the *netfilter* framework [24] provided in the Linux kernel. When the AP receives an explicit deauthentication frame from an associated station or when successive frames sent to the station are unacknowledged,[3] it assumes that the station has switched to another AP. Now, using a hook into the pre-routing stage of the IP stack, all incoming IP packets are examined. If the destination IP address of a packet matches with the IP address of the mobile station that has lost association with the current AP, the IP packet is queued in the buffer. Buffering continues until the new location of the mobile is learnt through the OLSR protocol and the route is appropriately changed. All buffered packets destined to this mobile are now re-injected back into the IP stack. These packets are now transmitted using the new route.

[3] A timeout mechanism is used to take care of false alarms, i.e., situations when buffering is started due to packet losses caused by transient channel errors and not due to the movement of the mobile.

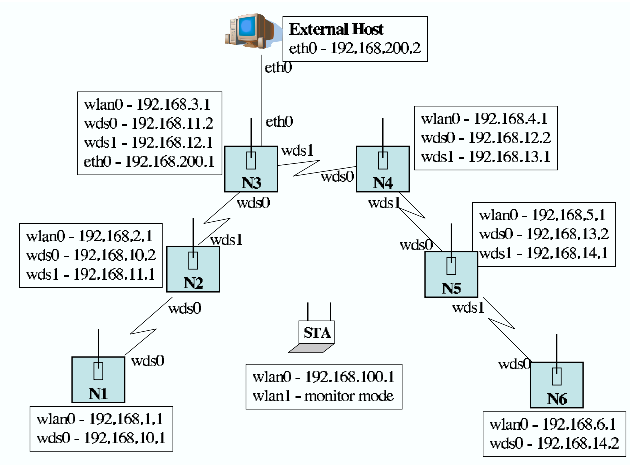

Fig. 3. *iMesh* testbed used for performance evaluation. $N1$ to $N6$ are the APs with WDS links forming a linear topology. $N3$ is also connected to an external host via the Ethernet interface. STA is the mobile node. For each node in the network, the logical interfaces (`wds0`, `wlan0` etc.) are shown. Each interface has an IP address. The routing protocol operates on the `wds` interfaces. The `wlan` interfaces are for AP to mobile communication.

IV. Performance Evaluation

A. Description of Testbed

Our *iMesh* testbed uses Soekris Engineering net4521 [25] processor boards, even though only one interface has been used for the work reported here. This compact, low-power, low-cost, advanced communication computer is based on a 133 Mhz 486 class processor. It has two 10/100 Mbit ethernet ports, up to 64 Mbyte SDRAM main memory and uses a 128 MB CompactFlash module for program and data storage. The board was expanded using a MiniPCI type III 802.11b interface and two PC-Card/Cardbus 802.11b interfaces. The interfaces are based on Intersil Prism 2.5 chipsets. The cards are connected to external rubber duck antennas via a pigtail. The processor boards run Pebble Linux V41 distribution [26] (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. Four of the APs and the mobile station are IBM Thinkpad R-series laptops running Redhat 8 distribution with linux-2.4.22 kernel. The remaining two APs are Soekris boards as described before. Each AP is equipped with a single Prism chipset based 802.11b PCMCIA card. We used two types of cards in our testbed, manufactured by EnGenius Technologies 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 this version 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 (*master mode*). All nodes operate on the same channel (channel 1) and they are assigned the same ESSid.

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 instead of using the auto-configuration protocol. We set up a linear topology as shown in Figure 3. The mobile station 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.[4] 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 [19], [18], [27], [28] 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 channel 1. 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 [29]. The airjack driver is capable of sending and receiving management frames in software. More will be said about the client-side drivers momentarily.

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

[4] We are working on extending the testbed to a wider geographic region so that true multihop links can be formed and using a robotic platform to move a mobile station to get repeatable measurements.

also ported Transparent Mobile IP [11] 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. In the following subsections, we report four different types of experiments: handoff latency, round-trip time (RTT) and impact of mobility over constant-bit-rate UDP traffic and TCP traffic.

B. 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 is complete when STA re-establishes connectivity to the network via newAP. Handoff begins when the STA loses its association with oldAP. This is indicated by either the receipt of a deauthentication frame at oldAP from STA, or repeated failure of packet transmissions from oldAP to STA, or the transmission of probe request frame from STA. Since handoffs are forced in our experiments via a script on the client side we are able to determine the exact start of handoff by noting the timestamp of the deauthentication frame from the client to the oldAP. Layer-2 handoff ends when the client is able to associate with the newAP. At this time the layer-3 handoff starts.

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 3) 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 3) 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 4 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 [17], [19], [18]. The timeline shows about 50ms delay for probing. A single channel is probed. In case the methods used in [19] 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.

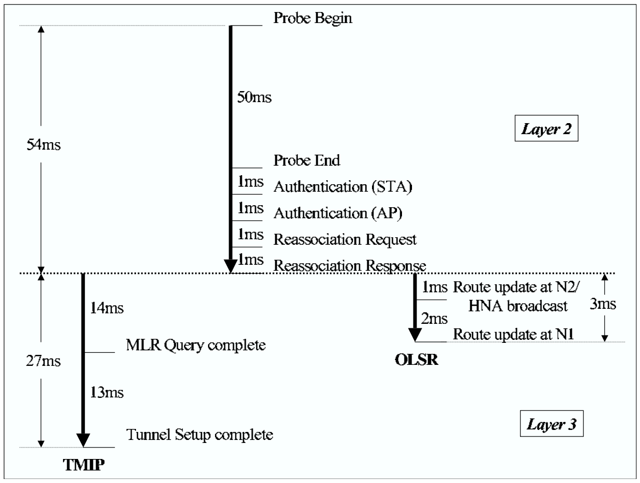

Fig. 4. Timeline describing a typical fast one-hop handoff from $N1$ to $N2$ for a mobile station with *iMesh* running OLSR and TMIP. The probing delay during layer-2 handoff is optimized by scanning only a single channel. Layer-3 handoff for OLSR is around 3ms while TMIP incurs an average of 27ms. Note that the timeline is not to scale.

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 coverage holes. This is because typically the distance between the newAP and oldAP will be less than twice the radio range.

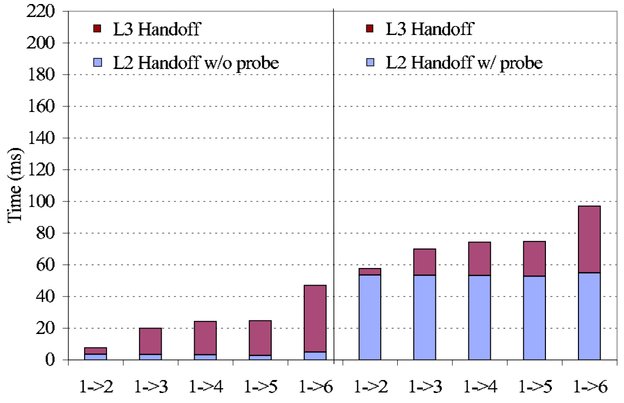

Fig. 5. Handoff latencies for *iMesh* using OLSR routing without probing and with probing on a single channel. The notation $i \rightarrow j$ indicates hand-off from Ni directly to Nj.

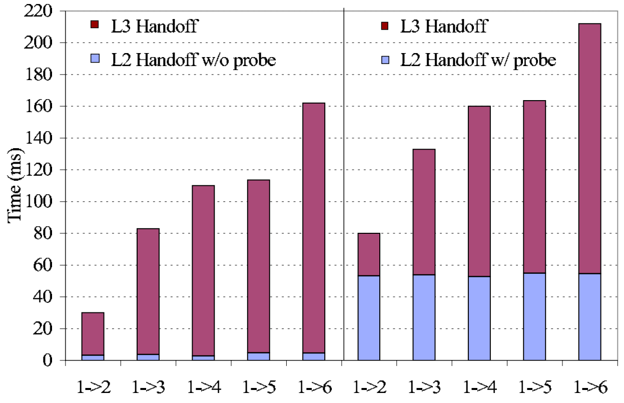

Fig. 6. Handoff latencies for TMIP without probing and with probing on a single channel. The notation $i \rightarrow j$ indicates handoff from Ni directly to Nj.

This makes the shortest path length in hops between these two APs very small. While route updates may be transmitted network-wide in OLSR, actual route changes will be limited to within a small neighborhood of newAP, because of the stated closeness of newAP and oldAP,

Two sets of handoff results are presented in Figures 5 and 6. 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 [29] 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 [19], [18] 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.

Looking at Figures 5 and 6 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 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 [19], [18], [28].

C. *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 and the remaining experiments in the following section 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 measure the RTT values for each ping. Figure 7 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 $N2$

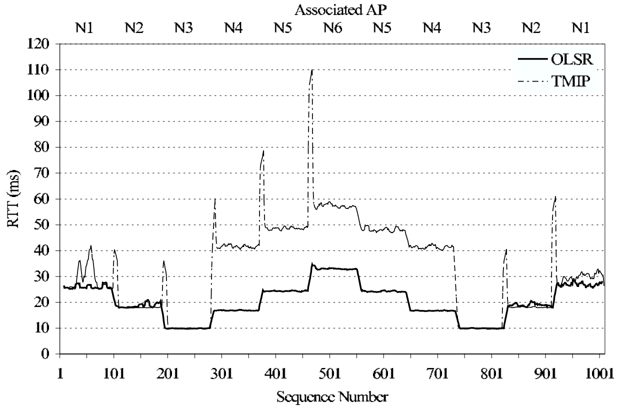

Fig. 7. RTT Measurements for pings for *iMesh* with OLSR and TMIP with handoffs at intervals of 10 sec.

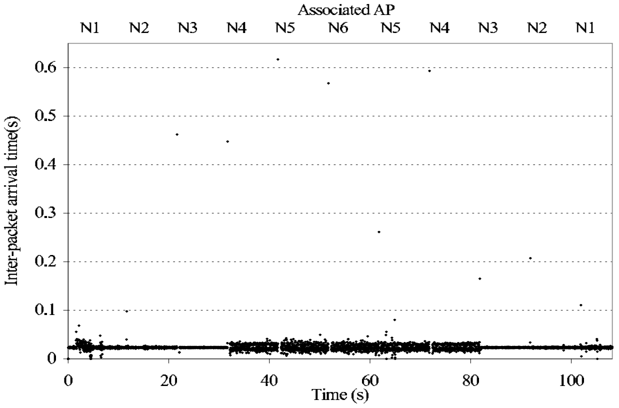

Fig. 9. Inter-arrival times for 500Kbps CBR UDP traffic for TMIP with handoffs at intervals of 10 sec.

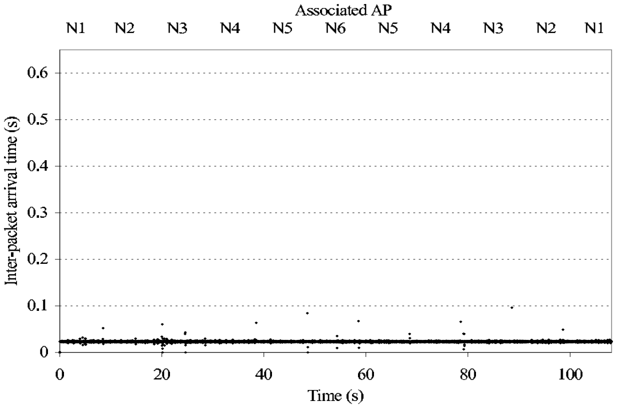

Fig. 8. Inter-arrival times for 500Kbps CBR UDP traffic for *iMesh* with OLSR with handoffs at intervals of 10 sec.

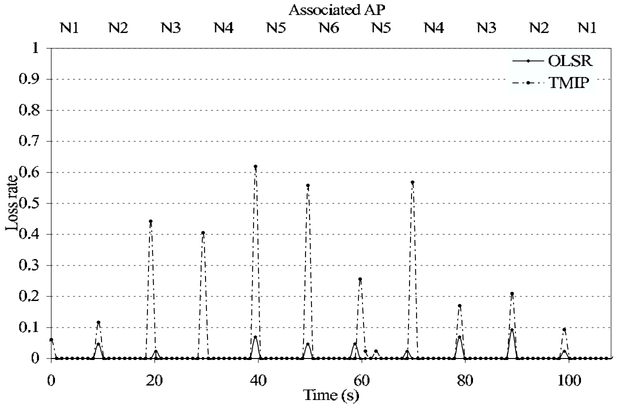

Fig. 10. Average loss rates for 500Kbps CBR UDP traffic for *iMesh* with OLSR and TMIP with handoffs at intervals of 10 sec.

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 tho 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.

D. Inter-arrival Measurements for CBR Traffic

We measure the inter-arrival times experienced at the mobile station for a 500Kbps CBR (constant-bit-rate) UDP packets sent from the external host. Figures 8 and Figure 9 plot the inter-arrival times for each packet received at the station. The same “walk” described in the previous subsection is also used here. Note substantially reduced variations in the times for OLSR routing. Figure 10 plots the average loss rates of 1 second intervals for both OLSR and TMIP. Note significantly higher losses for TMIP at the instants of handoff.

E. TCP Throughput Experiments

We also study the impact of frequent handoffs on the performance of a long-lived TCP connection. The same “walk” is used again. The mobile station starts a TCP transfer from the external host at the start of the walk. Figure 11 depicts the instantaneous throughput at the mobile node during the walk for both OLSR and TMIP. Note that in TMIP, the instantaneous throughput sometimes drops to zero, likely due to larger handoff latency. TMIP’s throughput also suffers significantly in the situations where its path length is larger than OLSR’s path length. (during the middle of the walk). Overall, our measurement shows that OLSR achieves around 42% improvement in the aggregate TCP throughput over TMIP during the entire transfer.

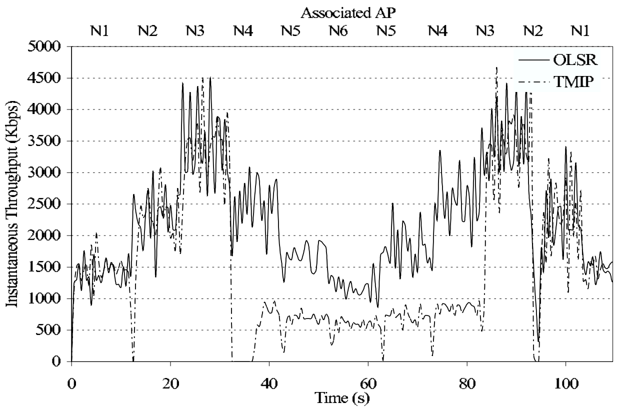

Fig. 11. Instantaneous TCP throughput for OLSR and TMIP.

V. 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, little public information is available about the architectural choices and performance. The information we could gather from the white papers are quite limited. For example, Firetide [6] provides a complete routing solution on a mesh routing backbone using the TBRPF routing protocol [30] used for mobile ad hoc network routing. However, here the routers do not serve as APs. APs must be "connected" to the routers using a separate wired ethernet link. Vivato [31] uses two radios on the APs, but only one of them is used in infrastructre mode; the other is used to bridge to similar other radios on other APs. The goal is to fill coverage "holes" in a WLAN deployment. Tropos [3] and Strix Systems [32] appear to provide a similar architecture as we report here; however details are unclear. In our knowledge there is no reported performance evaluations about hand-off performance.

There have been several community initiatives for mesh networking. Examples include Locust World [33] (uses transparent mobile IP [11]), Bay Area Research Wireless Network [8] (routing solution unknown), Seattle Wireless Network [7] (can use Internet routing protocols such as OSPF or RIP), and Champaign-Urbana Community Wireless Network [9] (uses *Hazy Sighted Link State* routing protocol [34]). However, none of such initiatives report any performance data or experience report.

On the other hand, there is some documentation on building and using mesh networking concepts in the research community. The Roofnet project at MIT [35] and mesh networking research in Microsoft Research [36] are the most prominent examples of work in this area. 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 [37], that is based on the well known Dynamic Source Routing (DSR) protocol [38]. 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 [39] and use of multiple radios on each node [40]. Recently, Raniwala and Chiueh considered routing and channel assignment algorithms for multi-radio mesh networks with a similar testbed [16], [41]. However, 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 hand off to. Shin, Mishra and Arbaugh have proposed schemes to reduce probing latency by scanning only a subset of channels. The probe wait time is optimized by exploiting the knowledge of operating channels of APs in the neighborhood [19]. Shin, Rawat and Schulzrinne have proposed a new handoff procedure which reduces the probe time by using a selective scanning algorithm and a caching mechanism [18]. Cleyn et. al. [27] and Sharma et. al. [28] independently have considered efficient layer-3 handoffs for wireless LANs, but only in the context of standard-compliant mobile IP [10]. Our layer-3 handoff latencies are much superior relative to these reported studies. This is in spite of the fact that our APs are connected via a wireless network while wired links are considered in these studies. However, it is fair to state that conformance to standard mobile IP imparts some inefficiency to these implementations. This also happens with transparent mobile IP in our work.

Recently, the IEEE 802.11 committee has started a new working group (802.11s) for the so-called *ESS mesh* networking [42]. 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.

VI. 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. This measurement includes certain layer-2 optimizations reported in literature. We consider this an excellent performance relative to recent studies on wireless LAN. We have also showed experimentally that the improvement in handoff latency over transparent mobile IP translates to a superior transport protocol (UDP and TCP) performance in terms of packet losses, delay and throughput.

References

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

[2] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *Proc. of Broadnets*, 2004.

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

[4] Packethop. http://www.packethop.com.

[5] MeshNetworks. http://www.meshnetworks.com.

[6] Firetide, "Wireless instant networks." http://www.firetide.com.

[7] Seattle Wireless. http://www.seattlewireless.net.

[8] Bay Area Research Wireless Network. http://www.barwn.org.

[9] Champaign-Urbana Community Wireless Network. http://www.cuwireless.net.

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

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

[12] S. Desilva and S. R. Das, "Experimental evaluation of a wireless ad hoc network," in *Proc. of IEEE IC3N*, pp. 528–534, Oct. 2000.

[13] D. Maltz, J. Broch, and D. Johnson, "Lessons from a full-scale multihop wireless ad hoc network testbed," *IEEE Personal Communications Magazine*, vol. 8, pp. 8–15, Feb. 2001.

[14] 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*, pp. 220–229, October 2004.

[15] C. Perkins, ed., *Ad Hoc Networking*. Addison Wesley, 2001.

[16] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2004.

[17] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, 2003.

[18] 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*, pp. 19–26, ACM Press, 2004.

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

[20] "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation." http://csl.ee.iastate.edu/ cpre543/80211f.pdf.

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

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

[23] Microsoft Corp., "Native 802.11 framework for IEEE 802.11 networks." http://www.microsoft.com.

[24] "The netfilter/iptables project: the Linux 2.4.x/2.5.x firewalling subsystem." http://www.netfilter.org.

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

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

[27] 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*, vol. 45, no. 3, pp. 345–361, 2004.

[28] 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.

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

[30] R. Ogier, F. Templin, B. Bellur, and M. Lewis, "Topology broadcast based on reverse-path forwarding (TBRPF)." IETF Draft, October 2003. Work in progress.

[31] Vivato. http://www.vivato.net.

[32] Strix Systems. www.strixsystems.com.

[33] Locust World. http://www.locustworld.com.

[34] C. Santivanez and R. Ramanathan, "Hazy sighted link state (HSLS) routing: A scalable link state algorithm," tech. rep., BBN technical memo BBN-TM-I30I, BBN technologies, Cambridge, MA, 2001. Available at http://www.ir.bbn.com/documents/techmemos.

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

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

[37] D. Aguayo, J. Bicket, and R. Morris, "SrcRR: A high throughput routing protocol for 802.11 mesh networks." Technical Report, 2004.

[38] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing* (T. Imielinski and H. Korth, eds.), ch. 5, Kluwer Academic, 1996.

[39] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 133–144, 2004.

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

[41] A. Raniwala and T. Chiueh, "Evaluation of a wireless enterprise backbone network architecture," in *Proc. of Hot Interconnects*, 2004.

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