

On the Design and Implementation of Infrastructure Mesh Networks

Krishna N. Ramachandran
University of California Santa Barbara
krishna@cs.ucsb.edu

Scott Miller
Lucent Bell Labs
scm@lucent.com

Milind M. Buddhikot, Girish Chandranmenon
Lucent Bell Labs
{mbuddhikot, girishc}@lucent.com

Elizabeth M. Belding-Royer, Kevin C. Almeroth
University of California Santa Barbara
{ebelding,almeroth}@cs.ucsb.edu

Abstract—Wireless mesh networking is an exciting new technology that has applications in defence, metro-area Internet access, and transient networks (e.g. disaster recovery, conventions). In this paper, we describe the design and implementation of a self-configuring, secure *infrastructure mesh network architecture*, called *MeshCluster*, composed using multi-radio network nodes. A subset of radio interfaces on these nodes are used for providing network access to end-devices whereas other radio interface are used for relaying packets to nearest Internet gateway. We identify four key design problems: (1) auto-configuration of MeshCluster nodes and relay infrastructure, (2) single and multipath routing in the relay infrastructure using routing metrics, (3) load balancing in the relay infrastructure, and (4) support for end-device mobility across access interfaces of mesh network. For each of these problems, we describe in detail our design, prototype implementation, and performance results.

I. INTRODUCTION

In the world of ubiquitous mobile wireless networks that is taking shape, wireless mesh networks are emerging as a significant new technology. Their promise of rapid deployability and reconfigurability makes them suitable for important applications such as disaster recovery, homeland security, transient networks in convention centers, hard-to-wire buildings such as museums, unfriendly terrains, and rural areas with high costs of network deployment. They can provide large coverage area, reduce “dead-zones” in wireless coverage, lower costs of back-haul connections for base-stations, and improve aggregate 3G, 802.11 cell throughput and help reduce end-user battery life.

We distinguish two kinds of mesh networks: (a) *Client-mesh* networks [35], [41], [52] wherein end-devices (such as PDAs, laptops) participate in packet forwarding. These networks are *infrastructure less* in the sense, operation of client-mesh is not managed and monitored by a service provider. They are useful for opportunistic[41] or predictable store-and-forward message transport [52]. Alternately, when used only for packet forwarding for multi-radio clients (for example, with 3G and 802.11 interfaces), they improve coverage and data rates for wide area cellular service [35]. (b) *Infrastructure-mesh* networks wherein the end-devices do not participate in the packet relay and the multi-radio relay nodes are part of the network infrastructure. This paper primarily focuses on this kind of mesh networks.

Research Contributions We describe design and implementation of a self-configuring, secure *infrastructure mesh network architecture*, called *MeshCluster*, composed using multi-radio network nodes. A subset of radio interfaces on the relay nodes are used for providing network access to end-devices whereas

other radio interface are used for relaying packets to nearest internet gateway. We identify and solve four key design problems: (1) auto-configuration of MeshCluster nodes and relay infrastructure, (2) single and multipath routing in the relay infrastructure using routing metrics, (3) load balancing in the relay infrastructure to make best use of the channel capacity, interfaces of mesh network. For each of these problems, we present in detail our design, prototype implementation and performance results.

A. Outline of the Paper

Section II describes in detail our *MeshClusters* reference architecture and provides overview of the four design problems we address. Section III describes our secure auto-configuration scheme. Section IV describes design of the routing architecture and packet forwarding components of *MeshCluster*. Specifically, we present a new AODV based routing scheme called AODV-ST that optimizes the common case traffic flow from relays to gateway. We describe our load balancing solution in Section V. In Section VI we describe design of three different schemes for supporting end-device mobility. Section VII describes our prototype implementation of the *MeshCluster* and Section VIII presents various performance results. We review related work in Section IX. Finally, Section X presents our conclusions and on-going work.

II. MeshClusters REFERENCE ARCHITECTURE

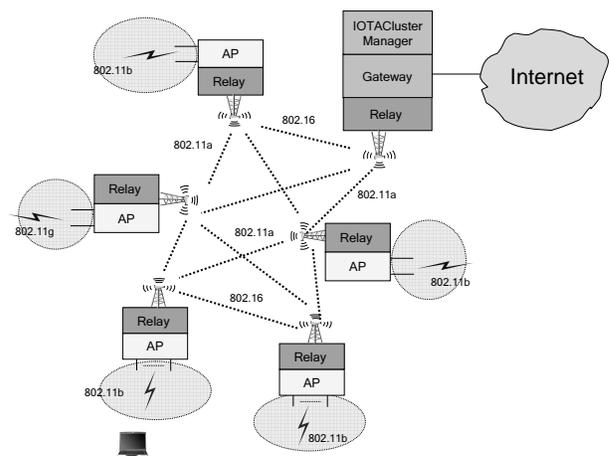


Fig. 1. MeshCluster: reference architecture

The *MeshCluster* architecture illustrated in Figure 1 consists of two new network elements: the relay and the gateway nodes. The relay elements are multi-radio systems that support two kinds of wireless network interfaces: access and relay, whereas gateway elements support: relay and Internet back-haul (up-link) interfaces. The end-user Mobile Nodes (MNs) access network using the access interfaces. The relay interfaces are used to construct a self-configuring, secure, managed, power-adaptive packet forwarding backbone between the relay and gateway nodes. The access links can be based on 3G (e.g:[13]) or 802.11[12] standards, whereas the relay links can be based on 802.16 or 802.11.

The gateways are connected to the Internet via wired (Ethernet) or wireless (1xRTT, EV-DO, 802.16) up-links. The placement of relays and gateway nodes depends on the deployment scenario. For example, in case of municipal metro area network aimed at providing broadband access to end-users, relays can be mounted on poles and the gateway nodes may be located in data centers in one of the downtown buildings. The in-building mesh networks such as enterprise buildings, convention centers, museums may follow similar structured placement. In both these scenarios, relay nodes will be stationary. On the other hand, in applications where transient networks are created such as disaster recovery, outdoor events, relays may be placed arbitrarily and may be quasi-stationary. In some cases such as defence applications, where soldiers in vehicles use relays to communicate to their command-and-control via a remote gateway node, relay mobility may be significant. In our work, we do not account for this scenario.

A *MeshCluster-Manager* entity, optionally co-located with the gateways, implements management and monitoring functions such as power level and frequency assignment for access and relay links, load-balancing in the relay cluster, and mobility and authentication support.

Design Problems: We consider following design problems in the context of MeshCluster architecture with 802.11 based relay network and present our solutions: (1) Robust, secure auto-configuration and associated protocol, (2) Packet routing and forwarding in the relay cluster that adapts to failures and network conditions such as load and interference and optimizes common-case traffic, (3) load balancing in the relay infrastructure, and (4) Seamless end-user mobility across the relay nodes.

Due to space constraints we do not consider the problem of designing interference-aware schemes to assign channels to relay and access interfaces. Our solution for this challenging problem can be found in our paper [45].

III. SECURE AUTO-CONFIGURATION

MeshClusters uses a *secure registration and auto-configuration* protocol to register with the MeshCluster-Manager. This protocol operates at the IP layer and employs well known ideas in the work of the ZeroConf working group [11] and security protocols.

Each relay runs a auto-configuration agent initialized at the boot time. This agent uses one or more of the relay interfaces to listen to *ESSID* broadcasts for all ad hoc networks operating in its area. For each *ESSID*, the agent first joins the ad hoc network using the *BSSID* broadcast. It then picks an IP address from

the zero-configuration address space 169.254.*.* and joins the IP based relay infrastructure. The 16 bits of the selected address can be computed using a truncated hash of the MAC address and time-of-the-day string. Since hash is likely to be unique, probability of the event of multiple nodes booting simultaneously picking the same address is significantly low.

The relay node then listens for the gateway advertisements, periodically received and rebroadcast by the relays already part of the MeshCluster. These advertisements contain gateway capability information such as Internet back-haul link speeds, relay capacity, best path available through the relay that rebroadcast the gateway advertisements etc. The agent begins a *configuration session* with one or more gateways selected based on certain criteria, for example *closest gateway* — gateway which can be reached by a *shortest hop-count* path or based on capabilities such as capacity — *least loaded gateway* or *high capacity gateway*.

The auto-configuration protocol supports optional authentication phase in which the agent performs mutual authentication to the gateway using security credentials such as digital certificates or symmetric key stored in relay in tamperproof hardware. The authentication protocol resembles IEEE 802.11i [14] with the key difference that Extensible Authentication Protocol (EAP) packets are IP-encapsulated instead of Ethernet encapsulation. Any of EAP schemes that support mutual authentication and dynamic session security key derivation such as EAP-TLS, EAP-SIM, EAP-AKA may be employed. Using the derived session keys all packet flow between the relay and gateway can be encrypted. Also, note that we don't focus on the security of the routing protocol in the relay network. We argue that if each relay node is authenticated to the gateway, a common dynamic group key can be securely distributed and used to protect routing protocol messages. Clearly, to achieve this, relay network may operate two *ESSIDs*, one (e.g: *Join_Mesh*) for traffic during authenticate-and-join phase and other (e.g: *Authenticated_Mesh*) for post authentication phase.

The relay agent conveys its capabilities such as number and type of radio interfaces and its observed environment such as visible neighbors in different frequency ranges, observed interference etc. which may be useful to the gateway for frequency assignment. The gateway conveys configuration parameters to the relay such as *ESSID* for access, frequencies used on relay and access interfaces, power levels to use, mobility method to use, addressing schemes, and any path-specific information. After the configuration session is complete, the zeroconf address is relinquished but the security parameters for the session may be preserved for future reconfigurations.

IV. MESHCLUSTER ROUTING ARCHITECTURE

We considered various design options detailed in the following:

- *Layer-2 vs. Layer-3 Routing:* Should the mesh routing solution operate at layer 2 or layer 3? Conceivably, the relay network could employ layer-2 Ethernet bridging and its associated 802.3d spanning tree based forwarding. In this case, the access cloud of all relays appear as a big layer-2 network at the gateway nodes. This has the advantage that no access and relay subnet management is required and layer-3 mobility is rather easy to

support. However, such virtualization comes with the cost of transporting the entire layer-2 packet originating in the access networks to the gateway nodes and a complex virtualizing Ethernet layer. Also, naive use of protocols such as DHCP, ARP, RARP that employ layer-2 broadcast can result in bandwidth wastage in the relays.

On the contrary, a layer 3 solution does not suffer from these drawbacks and also, operates effectively across the different physical layer technologies that may be used in a heterogeneous mesh network deployment. This requirement is especially important with the rapid innovation in physical layer technologies and the increasing availability of them in the market.

- *Leveraging existing Layer-3 wireline routing protocols:* Can we leverage existing wireline routing protocols such as OSPF or RIP for routing within the mesh network? Such an approach if adopted would take advantage of extensively tested and optimized wireline protocols for routing within the mesh. Furthermore, the task of network management would be greatly simplified because of the easy availability of tools that manage and monitor wireline protocols. However, wireline routing protocols oftentimes result in relays exchanging a high volume of periodic control messages, which can be a significant traffic overhead in bandwidth constrained wireless mesh networks. Furthermore, wireline routing protocols typically assume that the relays are static. This assumption fails to hold in a wireless mesh network where relays can be mobile. Wireline protocols can therefore be inefficient in handling network mobility

- *Optimizing for common-case traffic:* In most deployment scenarios of mesh networks, a significant portion of traffic in the relay network is due to end-user access to services such as web servers, VPN gateways, database and file servers in the wired infrastructure such as the Internet or enterprise networks. The data traffic, such as VOIP, multimedia flows, between end devices in access clouds of two different relays will be a small fraction of the total traffic. As such optimizing routing to efficiently support forwarding of the *common case* i.e. the gateway destined traffic can improve performance of the relay infrastructure.

- *Using existing ad hoc routing protocols:* Finally, we considered using existing ad hoc routing solutions, such as AODV [44], DSR [29], and OLSR [20] for routing within the mesh. These protocols inherently support network mobility and are designed to be low-overhead in their operation. These features makes them attractive for use in wireless mesh networks. OLSR is a link state routing protocol, analogous to OSPF and relies on knowledge of complete topology information at all nodes. It is quite efficient if the traffic is distributed equally likely between any two pairs of nodes which is in contrast to our common case traffic argument. As such OLSR overhead and capability may be a overkill. On the contrary, AODV is a simple, low-overhead, reactive routing protocol that is standardized in IETF and has public domain robust implementations [33], [10], [9]. Therefore, we use AODV as a base MeshCluster routing protocol. One can conceivably design a hybrid protocol that reacts to traffic pattern and switches from a AODV based protocol to a OLSR-based protocol in the event traffic distribution becomes more uniform. We do not consider this mode of operation.

A. Design of AODV-ST

We argue that use of AODV “as-is” leads to a poor mesh routing solution due to following operational deficiencies:

1. *AODV lacks support for high throughput routing metrics:* AODV is designed to support the shortest hop count metric. This metric favors long, low-bandwidth links over short, high-bandwidth links. Furthermore, AODV computes the metric using a broadcast discovery mechanism. Broadcast packets are typically sent at the lowest data rate and hence the propagation characteristics of higher data rate unicast packets cannot be accurately predicted using broadcast packets [34]. Because of these reasons, AODV can select routes with poor end-to-end throughput [22].

2. *AODV lacks an efficient route maintenance technique:* A route discovered with AODV may no longer be the optimal route further along in time. This situation can arise because of network congestion or the fluctuating characteristics of wireless links. AODV lacks a provision to re-discover the new optimal route. Several proposed techniques [40], [36] overcome this drawback by discovering multiple routes to a destination. These routes are then individually monitored for their path characteristics. In a large-scale wireless mesh network, the number of paths monitored by the relays can potentially be very large and can result in high control-traffic overhead.

3. *AODV route discovery latency is high:* AODV is a reactive routing protocol. This means that AODV does not discover a route until a flow is initiated. This route discovery latency can be high in large-scale mesh networks.

4. *Large routing table sizes:* AODV is designed for classic ad hoc networks where traffic flows are between nodes or node clusters rather than between nodes and Internet hosts. So simplistic reuse of AODV implementations result in routing table entries at relay nodes for all Internet hosts accessed by end devices in the access clouds. As such the routing tables can become unnecessarily large. AODV must be augmented with appropriate tunneling mechanisms to optimize routing table size for common case traffic.

Our enhanced *AODV-Spanning Tree* (AODV-ST) protocol eliminates above limitations as follows: First, it supports high throughput metrics, such as ETX [21] and ETT [24]. Our current implementation supports the ETT metric [24] although other metrics can be easily supported. Second, it proactively maintains of *spanning trees* whose roots are the gateways in the mesh network to significantly reduce route discovery latency and achieve lightweight, soft state route maintenance. Last, it employs IP-in-IP tunnels to reduce the routing table at relays to sum total of number of relays and access subnets.

Figure 2 illustrates the concept of AODV-ST spanning tree for a sample network of seven relays and two gateways. Each relay in the network lies on two spanning trees ST-1 (shown by solid lines) and ST-2 (shown by dashed lines). The gateways initiate the creation of the spanning trees by emanating periodic control messages that are selectively broadcasted in the network. Each spanning tree is created such that a relay node on a tree lies on the optimal path to the gateway corresponding to that tree. The route maintenance overhead is kept to a minimum because the paths to all relays on the spanning trees are proactively maintained. Furthermore, the route discovery latency is eliminated

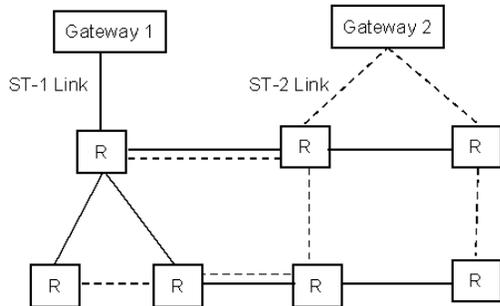


Fig. 2. Gateway specific spanning trees.

as each relay in the network is aware of an optimal path to its default gateway. A relay chooses the gateway with which it can achieve the highest capacity (as determined by the routing metric) as its default gateway.

For relay-to-relay communication, AODV-ST relies on the reactive route discovery strategy utilized in AODV. Conceptually, AODV-ST is a *hybrid routing protocol*: it uses a proactive strategy to discover routes between commonly used end-points (relay-to-gateway) and uses a reactive strategy for routes between less-commonly used end-points (relay-to-relay).

In the following, we first provide a brief overview of AODV and then discuss the specifics of protocol operation.

B. AODV Overview

AODV is an on-demand ad hoc routing protocol[44]. For neighbor detection, AODV can use either broadcast HELLOs or link layer feedback. Route discovery is based on a <route request, route reply> cycle. Route discovery begins with a broadcast Route Request (RREQ) message containing the destination address for the requested route and a RREQ sequence number that guarantees loop-free operation. As the RREQ is propagated throughout the network, each intermediate node creates a reverse route entry towards the originator (source) of the RREQ. An intermediate node forwards only the first RREQ it receives from the originator. If the destination-only flag is set in the RREQ message, only the destination is allowed to issue a Route Reply (RREP). If the destination-only flag is not set in the RREQ, an intermediate node is allowed to issue an RREP provided it has an active route towards the destination. The RREP message is unicast towards the source along the reverse route setup during RREQ propagation. As the RREP is propagated, intermediate nodes on the reverse route create a forward route entry for the destination node in their respective route tables. When an active route breaks, the node in the route that detects the break has the option of doing a local repair by finding another route towards the destination, or sending a Route Error (RERR) message towards the source to notify it of the break.

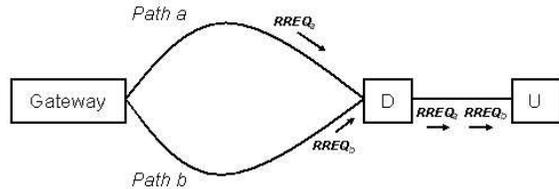


Fig. 3. Duplicate route request is rebroadcasted if it is received on a better path.

C. The AODV-ST Routing Protocol

In AODV-ST, the gateways periodically broadcast RREQ messages to initiate the creation of spanning trees. Before a RREQ is broadcasted, a gateway sets the *destination-only* flag in the RREQ and sets the RREQ destination address to the network-wide broadcast address. These settings differentiate normal route discovery RREQs from the RREQs for spanning tree creation. A RREQ also contains a *metric* field which is set to zero by the gateway. When an intermediate relay receives an RREQ, it checks if the RREQ is a gateway-initiated RREQ. If the condition is satisfied, it creates a reverse route to the gateway *provided* the RREQ is received on the best known path to the gateway. The relay can make this determination because of the metric field contained in the RREQ. This field is updated by each intermediate relay to represent the characteristics of the path it has traversed. The specific handling of the field at each relay is dependent on the path metric being used. To simplify the explanation, we postpone the discussion on metric handling to the next subsection. Once a relay creates a reverse route entry for the gateway, it sends a gratuitous RREP back to that gateway. This gratuitous RREP also has a metric field that is set to zero initially. The field is updated at every intermediate relay on the path to the gateway. When an intermediate relay receives the gratuitous RREP, it creates a forward route to the originating relay. It updates the path metric to the originating relay with the metric value contained in the gratuitous RREP.

A relay re-broadcasts a gateway initiated RREQ only if the path traversed by the RREQ is the best known path to the relay. Note that an intermediate relay does not wait until it receives all RREQs before picking the best one to rebroadcast. This is required to reduce the route discovery latency. This would mean that an upstream relay could receive a duplicate RREQ from the same downstream relay if the duplicate RREQ represents a better reverse path. This mode of operation is illustrated in Figure 3. Relay D in the figure receives two RREQs from the gateway G that traverse two different paths a and b where a is better than b . Assume that the $RREQ_a$ is slightly delayed with respect to $RREQ_b$. When D receives $RREQ_b$, it rebroadcasts it as it arrived on the first path known to it. However, when the delayed $RREQ_a$ is received, D rebroadcasts it because it arrived on the better path. Relay U therefore receives two duplicate RREQs from D .

As the RREQ is broadcasted hop-by-hop throughout the mesh network, the spanning tree is implicitly formed through the cre-

ation of reverse routes to the gateway at the relays. The time interval between successive gateway-initiated RREQs is set to ten seconds in our implementation. We empirically determined this interval to be a good setting. Each relay on receiving the successive RREQs updates its reverse routes based on the metric field contained in them.

For relay-to-relay communication, a relay node initiates a RREQ with the destination flag set and the destination address set to the address of the node to be reached. The destination flag is set because the most up-to-date path information is required at the source during path selection. The handling of the RREQs at the intermediate nodes is similar to the procedure described above.

D. Routing Metric Support in AODV-ST

A routing metric used with AODV-ST must satisfy two requirements: First, the metric must increase in value with increasing hopcount. This is required to prevent loop-free path selection. Second, it must be a *bi-directional* metric, i.e., the metric must give equal weightage to a path's performance in the forward and reverse directions. This is necessary for two reasons. First, TCP flows are bi-directional in nature. Therefore, both directions of a path must be considered during route selection. Second, AODV-ST creates a reverse route to a gateway upon receiving a RREQ that traverses in the forward direction from the gateway to the relays. Therefore, the metric must represent a path's performance in both directions, otherwise AODV-ST can select uni-directional paths.

Our current implementation of AODV-ST supports the Expected Transmission Time (ETT) metric [24]. ETT is a measure of the expected time needed to successfully transmit a packet of a fixed length, s , on a link. It yields high throughput paths because it selects a path with the least delay.

ETT is given as $(etx * s/b)$ where etx is the expected number of transmissions necessary to send a packet on the link [21]; s is the size of the packet (set to 1024 bytes in our implementation); and b is the bandwidth of the link. etx is computed by issuing periodic broadcast probe messages (sent every second in our implementation) in the forward and reverse directions and by measuring the corresponding forward delivery ratio (d_f) and the reverse delivery ratio (d_r) for a predetermined time interval. This time interval is set to ten seconds in our implementation. The etx for the link is then given as $etx = 1/(d_f * d_r)$. The link bandwidth, b , is determined using feedback from the radio driver. We modified the hostap driver [2] used in our implementation to support the feedback of the link data rate every second. The driver computes per-second link data rates by averaging the data rates of packets that traverse a link in the one second intervals. Where a driver does not provide rate feedback, we rely on packet-pair probing [32] to estimate bandwidth. In our implementation of this technique, a pair of packets, one small (134 bytes) and the other large (1200 bytes), are sent back-to-back every minute for ten minutes in both directions of the link. As soon as the smaller size packet is received, a timer is started to measure the delay incurred in receiving the larger packet. We choose the minimum of ten delay samples to estimate the link bandwidth. The link bandwidth then is simply the ratio of packet size and minimum delay. We use the minimum delay sample to

reduce any adverse impact queuing delays have on the transmission of the packet pairs.

The ETT metric, however, is not an optimal choice in a multi-radio wireless mesh network because it does not consider the frequency diversification of a path during path selection [24]. This can lead to sub-optimal routing in our current implementation. We are exploring various techniques to enhance the ETT metric to account for frequency diversification. One possible approach is to use the Weighted Cumulative Expected Transmission Time (WCETT) metric [24]. WCETT requires knowledge about each link in the path, such as the link's delay and its assigned frequency. This requirement can be easily satisfied by using a link-state routing protocol such as OLSR [20] or OSPF [37]. On the other hand, AODV-ST is a distance-vector routing protocol in which link-level information is not disseminated by design. This complicates the support of WCETT in AODV-ST. As future work, we plan to incorporate in AODV-ST a simple, low-overhead scheme to accumulate link-level information so that a metric like WCETT can be easily supported. We are currently evaluating a link-level accumulation scheme similar to the one used by AODV-bis [43].

V. MESHCLUSTER LOAD BALANCING

A. Load Balancing Defined

Load balancing is a desirable feature to have in a wireless mesh deployment. It reduces congestion in the network, increases network throughput, and prevents service disruption in case of failure. Load balancing in wireless mesh networks can be defined in the following two ways:

- *Path load balancing*: Path load balancing can improve network performance and reliability by distributing traffic among a set of diverse paths. There are proposals to achieve path load balancing in wireline networks [31], [19] and multi-hop wireless networks [38]. Pearlman et. al. [42] show that path load balancing provides negligible performance improvement in wireless multi-hop networks because of route coupling of candidate paths between common endpoints. Route coupling is a result of the geographic proximity of the candidate paths. This can lead to self-interference between those paths and can therefore adversely impact performance.
- *Gateway load balancing*: In this interpretation of load balancing, traffic is distributed among a set of gateways in the wireless mesh network, i.e., one of several gateways is chosen as the egress point for flows originating from the network. We believe that the performance improvement with gateway load balancing will be greater than with path load balancing because route coupling of paths to different gateways from an endpoint in the mesh is expected to be less in a well-planned deployment. For this reason, the MeshCluster architecture supports gateway load balancing. To the best of our knowledge, there is no prior work on gateway load balancing for wireless mesh networks.

B. Gateway Load Balancing Protocol

This subsection provides an overview of the gateway load balancing solution supported by the MeshCluster architecture.

An access relay (relay that is also an access point) lies on the spanning trees corresponding to the gateways in the network.

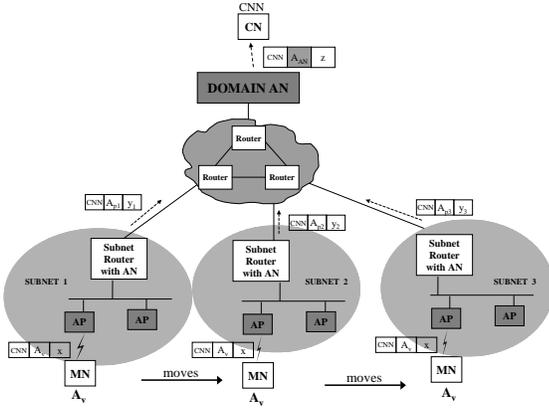


Fig. 5. MobileNAT based mobility technique

lays. To understand this, consider a TCP flow to correspondent node (CN) $cnn.com$. The relay node NATs the traffic with $(SA = A_v, DA = CNN, SP = x)$ to $(SA = A_{p1}, DA = CNN, SP = y_1)$ using a rule $(A_{p1} \mapsto A_v, x \mapsto y_1)$ and tunnels to AN using $(SA = A_{p1}, DA = AN)$ tunnel header. The AN NATs this further to $(SA = AN, DA = CNN, SP = z)$ with a rule $(A_{p1} \mapsto AN, y_1 \mapsto z)$. When the MN moves to a new relay node with external IP address A_{p2} , the mapping at AN is changed $(A_{p2} \mapsto AN, y_2 \mapsto z)$ and a new mapping $(A_{p1} \mapsto A_v, x \mapsto y_2)$ at the relay. The change of mapping rules at the relay and AN are signaling path operations are carried out by mobility agent software running at the AN and relays. This software also detects arrival new “visiting” nodes at the relay by performing IP-level packet filtering of packets with missing NAT rules. Note that the scheme has several advantages: (1) no client side software is required. (2) The scheme is agnostic to routing protocol in the relay network. (3) The access networks of relays can be managed as separate subnets or as part of a large subnet. (4) Addresses visible in the relay network are that of the externally visible A_{pi} addresses of the relays. None of the A_v addresses of the MNs are visible, keeping the routing tables quite compact.

Simple DHCP based mobility

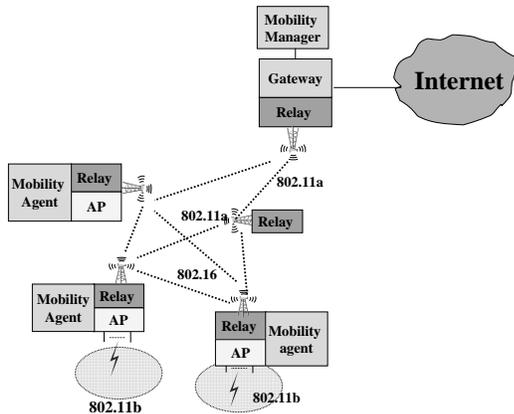


Fig. 6. Simple DHCP based technique

Figure 6 illustrates a simpler mobility scheme which relies on DHCP, AODV and monitoring of layer-2 events in the access

networks of relay nodes. Much like MobileNAT scheme above, it allows client to acquire a dynamic IP address and maintain that address as it moves across multiple relays. Also, it relies on a mobility manager (MM) at the gateway nodes and a mobility agent (MA) on the relays. The MA in the relays monitor changes in the layer-2 802.11 associations to detect new visiting mobile endpoints and propagate routes for the corresponding IP address in the AODV based relay network. If the MN has an address x , for the traffic destined $cnn.com$, the packets $(SA = x, DA = CNN)$ tunneled to the gateway with a tunnel header $(SA = x, DA = GW)$. The reverse traffic to x does not need to be tunneled and carries $(SA = cnn.com, DA = x)$. Whenever, AODV route request are launched for x , the most recent relay with x , responds with a router reply. The artifact of this is that the forwarding entry for address x appear in AODV routing tables at the relay nodes. The MAs in different relays in the mesh can allow proactive updates amongst themselves on detection or loss of MNs to help make AODV state update fast. This can help handoff performance and also, help track mobility of the enduser across multiple relays in the network.

One important issue that affects the performance of this technique is how well host operating system on the MN reacts to switching between two access points in the mesh network. If the ESSID for access clouds on all relays is identical, the host OS only requires completion of layer-2 association with new AP; it does not require an additional DHCP request and response to configure its interface IP address. As a result the handoff is much faster. On the other hand, if each relay access ESSID is different, in absence of any out-of-band or pre-configured information, the OS may assume the worst and restart DHCP transactions. Even though same IP address may be returned, if the protocol stack associated with the interface is always brought down during this process to account for the worst case of obtaining a different IP address, all flows are broken. Therefore, we would recommend keeping the ESSID the same for all access points in one mesh network. The only easy alternative is to use a mobile IP client, which masks of such disconnects by design. A proof-of-concept implementation of our scheme has been completed.

There are two main drawbacks of this schemes: (1) It relies on the proactive discovery of new location of an MN via route discovery mechanism of AODV and as such is tied to the relay routing protocol. (2) The routing table size in the relay network increases linearly with the number of mobile nodes. In presence of large number of mobile nodes, the route discovery and table size can prove to be a prohibitive penalty.

VII. PROTOTYPE IMPLEMENTATION

Our prototype implementation of *MeshCluster* nodes employs a small form factor single-board computer with 2 PCMCIA slots, one micro-PCI slot and built in 100 Mbps ethernet interface.[6]. As such, the relay nodes in our prototype can have upto three 802.11 interface cards, whereas gateway nodes can have one wireline and three wireless interfaces. The PCMCIA interfaces on the gateway node can be populated using Sierra Wireless/Airprime[5] 1XRTT or EV-DO wide-area wireless cards. The *MeshCluster* software runs Linux 2.4 and kernel and implements a broad set of functionalities in the form of



b) IOTA Gateway

Fig. 7. MeshCluster node

modules, such as AODV-ST relay module based on the NIST AODV kernel implementation, an enhanced 802.11 hostap access point, IP services modules (QoS, DHCP, NAT, Dynamic Firewall), Security and Accounting (RADIUS server/client), web services (web portal, caching, web-based configuration), and mobility support (Mobile IP, and MobileNAT variant).

We also used a mobility client software that manages multiple network interfaces to support automated interface selection and seamless mobility handoff using MobileIP. The mobility client, developed as a part of our research on 802.11 and 3G integration, includes a complete Mobile-IP stack that supports per-subnet FA [50] and co-located FA [50] modes and performs most of the mobility management[17]. Current version of mobility client does not integrate the new mobility technique described in Section VI.

VIII. PERFORMANCE EVALUATION

This section presents results from our evaluation of the MeshCluster architecture. We first present our goals and the evaluation methodology.

A. Goals and Evaluation Methodology

Our goal is to evaluate the individual components of the architecture such as the mesh auto-configuration scheme, the routing solution, the load balancing scheme, and the mobility management schemes. Due to space constraints, we omit results from our evaluation of the load balancing scheme. We plan to present those results in a forthcoming paper.

For the auto-configuration scheme, we are interested in analyzing the time it takes for all devices in the mesh network to join the network. We vary the number of interfering networks in the vicinity of the mesh to study the impact of such networks on the time taken for auto-configuration to complete. We rely on a simulation-based evaluation because it allows us to easily control the number of interfering devices in our analysis. Achieving this objective in a testbed environment is challenging. We utilize the Qualnet simulator for this evaluation.

To gauge the performance improvements with the MeshCluster routing solution, we compare AODV-ST against AODV in terms of end-to-end TCP throughput on the UCSB Meshnet [8], a twenty-five node wireless multi-hop testbed deployed in a five-floored office building on the campus of UC Santa Barbara. We

utilize twenty of them that are deployed on the first four floors in our comparison. Figure 8 shows a map of the floors and the location of the testbed devices on the various floors. The testbed devices are of two types. The ones indicated by squares in the figure are small form-factor desktop computers running Linux 2.4.27. Each of these computers is equipped with a EnGenius 2511-CD IEEE 802.11b radio. The ones indicated by circles are Linksys WRT54G wireless routers, each equipped with a Broadcom IEEE 802.11b/g radio. They are installed with the OpenWRT Linux distribution and utilize Linux 2.4.20 as the kernel.

Finally, we evaluate the mobility management schemes by utilizing a smaller testbed consisting of four nodes in a line topology A - B - C - D. Nodes A, B, C, and D, are IBM Thinkpad T21 laptops running Linux 2.4.27. Nodes A and B are equipped with a single radio whereas C and D are equipped with two radios each. A is designated as the gateway, B as a relay, and C and D as access relays. All the radios used in this setup are EnGenius 2511-CD IEEE 802.11b radios. Our goal for this evaluation is to compare the two mobility management schemes in terms of the delay associated in handing off from one access relay to another.

B. Results

Auto-Configuration Scheme: Our simulation environment is a network topology consisting of 30 mesh relays randomly distributed in a terrain of 1000x1000m. All relays are equipped with a single IEEE 802.11a radio. At the start of a simulation, each relay randomly picks a channel to operate on. One of the relay devices is designated as a gateway. The gateway initiates periodic advertisements every second to indicate its presence. An advertisement is embedded in a AODV Hello message and is propagated hop-by-hop throughout the network as Hello messages are exchanged between relays that have joined the mesh network. We implemented a layer 2 beaconing mechanism that a relay uses to issue periodic link-layer frames (every 100 msec) that contain the ESSID (name) of the network it has joined. This link-layer frame resembles IEEE management frames that are exchanged between stations that are part of a IEEE 802.11 ad hoc network. At the start of the simulation, only the gateway is assigned an ESSID. A relay listens for 200 milliseconds on each supported channel (12 in all in our simulation) in order to receive ESSID advertisements. It then joins each discovered network for a period of three seconds to listen to gateway advertisements. If a relay receives gateway advertisements on a particular ESSID, it attempts to register with the gateway. If the registration is successful, the relay is a part of the mesh network. The relay then propagates the gateway advertisements to its neighbors using AODV Hello messages.

Using the above described simulation setup, we measured the time taken for all the relays to join the mesh network in presence of varying degrees of interference. We introduced interference by varying the number of devices external to the mesh network from zero to fifty in increments of ten. Each interfering device at the start of a simulation randomly selects a channel and ESSID name to operate on.

Figure 9 represents the time (on Y axis) taken by the thirty devices (on X axis) to join the mesh network. In the absence of interfering devices, the mesh relays join the network within 20

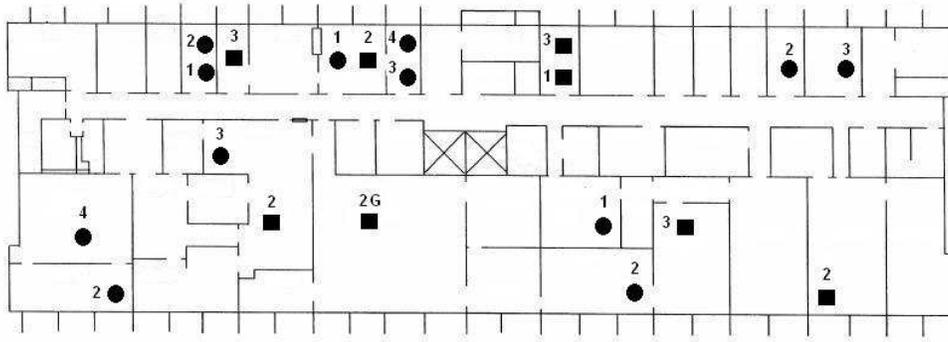


Fig. 8. Map of twenty nodes that are part of the UCSB Meshnet. The squares indicate small form-factor desktop devices equipped with wireless radios. The circles indicate Linksys WRT54G wireless routers. The number next to a device indicates the floor of the building the device is on. The gateway is marked with G.

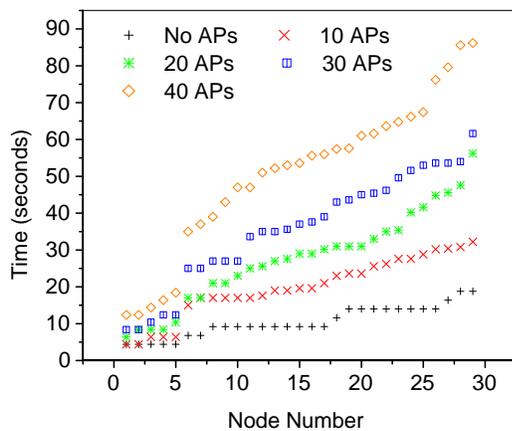


Fig. 9. Bootstrap time with varying interference

seconds of startup. A breakup of the join time is as follows. The relays that are immediate neighbors of the gateway take 2400 milliseconds (12 channels * 200 milliseconds per channel) to scan all channels for ESSIDS. Since there are no interfering networks, they discover the only ESSID that is advertised by the gateway. The relays then join the discovered ESSID for upto three seconds in order to listen to gateway advertisements. After receiving the advertisements, they join the mesh network. With increasing distance from the gateway, the mesh join time increases. This is because relays close to the gateway do not forward gateway advertisements until they join the mesh network. As the number of interfering devices increases, we see that the join time also increases. This is because the relays can join the network advertised by the interfering devices before discovering the mesh network. Note, however, that the time taken to join the network does not increase by a significant amount. For instance, even with fifty interfering devices in the vicinity of the mesh network, more than 80% of all nodes join the network under one minute, and all devices join the network within one and half minutes of bootup. Our results indicate that the MeshCluster auto-configuration scheme can reduce network management overhead during mesh deployment because of its capability to quickly bootstrap a mesh network without manual intervention.

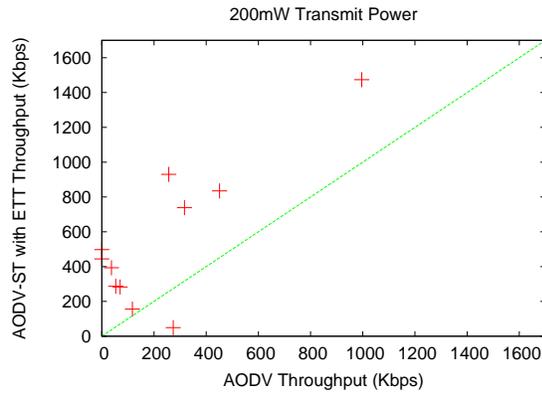


Fig. 10. Performance comparison of AODV against AODV-ST. All radios are configured to transmit at 200mW.

AODV-ST Routing Performance In our comparison of AODV-ST against AODV, we set all radios on the Meshnet to use auto-rate feedback [30] and set the number of link-layer re-transmissions to eight. We designated the node indicated by the solid square in figure 8 to be the gateway. We set the route cache timeout value for AODV-ST to AODV's default timeout value of three seconds. We conducted two sets of experiments by varying the transmission power of radios in the MeshNet. We varied the transmission power to gauge its impact on routing performance. In the first set of experiments, all nodes were configured to transmit at 200mW. In the second set, the nodes were configured to transmit at 100mW. For an experiment set, all nodes except the gateway send thirty seconds worth of TCP traffic to the gateway one at a time with a gap of five seconds between each transfer. The five second delay allows for any state in the route cache from the previous transfer to be timed out. The collection of thirty second transfers is performed in succession for AODV and AODV-ST. This arrangement ensures that the results obtained using the two routing protocols are comparable, since the TCP transfers are run within a few minutes of each other. Finally, we repeat the set of transfers three times for each routing protocol to average out any variance in measurements.

Figure 10 shows results from our comparison for the first set of experiments. The X and Y axes indicate throughput in Kbps attained with AODV and AODV-ST respectively. The average

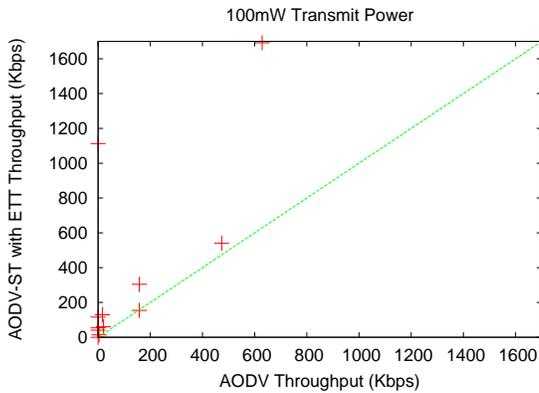


Fig. 11. Performance comparison of AODV against AODV-ST. All radios are configured to transmit at 100mW.

throughput of the *good* paths comprising of only one hop on the meshnet is greater than 4000 Kbps. AODV-ST offers no performance improvement over AODV because both protocols choose the one hop path for the data transfer. Five such data points are not depicted in the graph in order to improve the clarity of the graph in the lower throughput ranges. From the figure, it is clear that AODV-ST offers significant throughput improvements over AODV. Note the region of the graph near zero on the X axis. Several data points in this region indicate that AODV-ST offers greater than 250 Kbps throughput, whereas with AODV the attained throughput is less than 70 Kbps. This is because AODV uses the shortest hop count metric which typically selects paths comprised of long, low-bandwidth links; AODV-ST on the other hand utilizes the ETT metric which takes into account link reliability and link bandwidth during path selection, which results in AODV-ST selecting a least-delay path. The ETT metric, however, can result in AODV-ST selecting higher hop-count paths than paths selected using the shortest hop-count metric as with AODV. We verified that this is true for the data points in the figure. In a majority of the cases, the hop-count with AODV-ST is one more than with AODV; for the data points close to zero on the X axis, the hop-count with AODV-ST is often two more than with AODV.

Figure 11 shows results from our comparison with the radio transmit power set to 100mW. Again we omit five data points with throughput above 4000 Kbps to improve the clarity of the figure. AODV-ST results in improved performance over AODV. However, the performance improvements are not as significant as in the above case. This is because of the effect transmit power has on the neighbor connectivity of the meshnet nodes. Because of the reduced transmit power, the neighbor connectivity is sparser than in the 200mW case. As a result, the number of candidate paths available from a source to a destination is smaller and is of higher hop-count. As a result, there is a higher likelihood that AODV and AODV-ST choose paths that overlap with each other. We validated this to be the case by comparing the paths selected using AODV and AODV-ST and found an increase in path overlap. We believe that the increased overlap results in comparable throughputs with the two protocols.

Evaluation of Mobility Schemes: We implemented both mobile IP based mobility and DHCP based mobility. As discussed

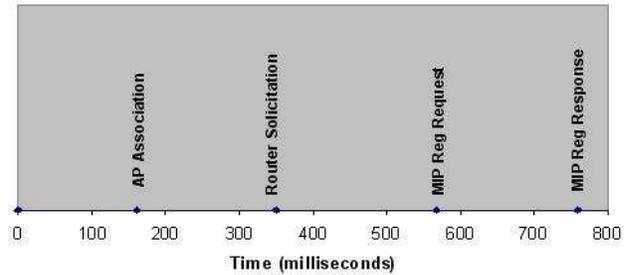


Fig. 12. Mobile-IP Latency

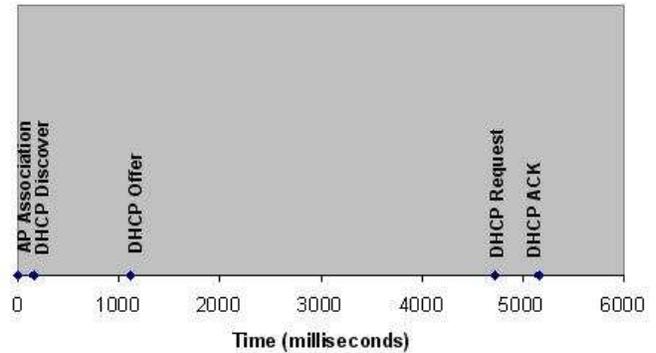


Fig. 13. Simple-IP Latency

in Section VI, we deployed a foreign agent in each of the access nodes, and a home agent in the gateway node. In the case of simple IP based mobility, we implemented a mobility agent in the access nodes. This agent gets indication from layer-2 that a new node has joined. Figure 12 shows the switching delay for mobile IP based scheme, and Figure 13 shows the delay for DHCP based mobility.

In mobile IP based mobility, the client resumed its network connection in 760ms, and in simple IP mobility, the client resumed its network connection in the new network in 5.2 seconds. The individual components of latency are as shown in Figure 12 and Figure 13. As seen in the figure, the layer-2 association is accomplished in less than 200ms, however, the layer-3 attachment takes much longer.

Although we expected better latency in simple IP based scheme, we did not observe this in practice. This is due to the specific way Windows DHCP client handles a new attach. In Windows, the client restarts the entire DHCP process if the wireless network card changes point of attachment. As a result, the Windows client starts off a DHCP DISCOVER, OFFER, REQUEST, ACK sequence. We can attempt to cut down the delay between DISCOVER and OFFER, however, the delay between OFFER and REQUEST is internal to Windows. If it could change the process, and simply verify that it is attached to the same IP subnet and maintain the connections, it would enable a simple seamless mobility with low latency. We intend to explore these options for improving DHCP based mobility, especially since it does not require an added client in the enduser device.

IX. RELATED WORK

There exists a lot of related work that is in some way concerned with wireless mesh networks. While it is not possible to summarize all the proposals because of space constraints, we present a representative sample below.

Considerable research has addressed the problem of routing in wireless multi-hop networks [20], [26], [29], [44]. Earlier works focused on wireless ad hoc networks where energy limitation and mobility are major constraints. These proposals utilize the shortest hop count metric as the path selection metric. This metric has been shown to result in poor network throughput because it favors long, low-bandwidth links over short, high-bandwidth links [22], [34]. More recent proposals aim to improve routing performance by utilizing route selection metrics [21], [23], [24].

Wireless multi-hop networks suffer from serious capacity degradation due to the half-duplex nature of the wireless medium [27]. Several proposals aim to alleviate this capacity problem by revamping the MAC layer to support the intelligent selection of a wireless channel during packet transmission [15], [28], [39], [49]. Other proposals aim to improve the capacity by equipping relays with multiple radios [24], [45], [47], [51]. Some wireless mesh hardware vendors also offer multi-radio mesh routers that utilize proprietary channel assignment schemes [1], [3], [7].

Little work has focused on developing system architectures for wireless mesh networks. Bicket et. al. evaluate the MIT Roofnet architecture [16] and find that their Cambridge Roofnet deployment can provide users of the network with usable performance despite lack of careful planning in deployment. The MeshCluster architecture is similar in spirit to the MIT Roofnet architecture. The key distinguishing aspect between the two architectures is that the Roofnet architecture is specifically targeted for community networks where relays are expected to be static and end-user mobility is minimal; the MeshCluster architecture is also well-suited for deployments where relays and end-users are mobile. Examples of such deployments include transient networks deployed for search-and-rescue and military operations.

There have been a number of testbed multi-hop wireless network deployments [4], [8], [24], [48]. A majority of such deployments are intended for conducting research on multi-hop wireless networks. There is also a growing deployment of commercial wireless mesh networks around the world. Such deployments support last-mile broadband connectivity, emergency services, and remote monitoring applications. Some of the hardware vendors that offer deployment services include Tropo Networks, Strix Systems, and Firetide Inc.

X. CONCLUSIONS

Wireless mesh networking has emerged as a promising new technology for the rapid deployment of wireless networks for applications such as search and rescue, homeland security, and metro-scale broadband connectivity. Although several mesh networking hardware vendors are offering services for the deployment of wireless mesh networks, the research community in general has paid little attention to the design of architectures

and systems that can fulfill the promise of this new technology. The MeshCluster architecture described in this paper is our attempt at fulfilling this gap.

We identified the following critical challenges that must be addressed for mesh networking to reach its full potential: (1) mesh network auto-configuration, (2) high-throughput packet routing, (3) mesh network load balancing, and (4) a mobility management framework that can ensure seamless user mobility in the mesh network. We described in detail the design, implementation, and evaluation of components of the MeshCluster architecture that address these challenges.

As future work, we are investigating the use of multiple radios to improve the capacity of the MeshCluster architecture. Currently, we are implementing an interference-aware channel assignment scheme that takes into account interference from co-located wireless networks during channel assignment. We are also developing a routing solution that is optimized for multi-radio, multi-channel wireless mesh networks. Our initial analysis indicates that reactive routing protocols such as AODV and DSR may suffer from control packet flooding problems in a multi-radio, multi-channel network because of the multitude of paths that are available in such networks. We are also integrating mesh network management and monitoring utilities such as DAMON [46] into the MeshCluster architecture to assist in the management and monitoring of wireless mesh networks. Our goal is to offer the MeshCluster architecture for download to the research community so that it can leverage this architecture for research and development purposes.

REFERENCES

- [1] BelAir Networks. <http://www.belairnetworks.com>.
- [2] HostAP Driver. <http://hostap.epitest.fi>.
- [3] MeshDynamics. <http://www.meshdynamics.com>.
- [4] MIT roofnet. <http://www.pdos.lcs.mit.edu/roofnet/>.
- [5] Sierra wireless (<http://www.sierrawireless.com>).
- [6] Soekris, inc. (<http://www.soekris.com>).
- [7] StrixSystems. <http://www.strixsystems.com>.
- [8] UCSB MeshNet Project. <http://moment.cs.ucsb.edu/meshnet>.
- [9] UoB AODV Implementation. <http://www.aodv.org>.
- [10] Uppsala University AODV Implementation. <http://core.it.uu.se/AdHoc>.
- [11] Zero configuration working group (<http://www.zeroconf.org/>).
- [12] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.
- [13] TIA/EIA/IS-835B - cdma2000 Wireless IP Network Standard. , Third Generation Partnership Program 2 (3GPP2), 2000.
- [14] Part 11: Wireless MAC and physical layer specifications: Specification of Enhanced Security. Technical report, IEEE P802.11i, November 2002.
- [15] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping For Capacity Improvement in IEEE 802.11 Ad Hoc Wireless Networks. In *ACM MobiCom*, Philadelphia, PA, September 2004.
- [16] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *ACM Mobicom*, Cologne, Germany, September 2005.
- [17] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller, and L. Salgarelli. Integration of 802.11 and Third Generation Wireless Data Networks. In *IEEE INFOCOM 2003*, April 2003.
- [18] M. Buddhikot, A. Hari, K. Singh, and S. Miller. MobileNAT: A New Technique for Mobility Across Heterogeneous Address Spaces. *Special Issue of ACM/Kluwer Journal on Mobile Networks*.
- [19] Robert Carter and Mark Crovella. Server Selection Using Dynamic Path Characterization in Wide-Area Networks. In *IEEE Infocom*, Kobe, Japan, April 1997.
- [20] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). Internet Engineering Task Force (IETF), rfc3626.txt, October 2003.
- [21] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *Pro-*

- ceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California, September 2003.
- [22] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: Shortest path is not enough. In *Proceedings of the First Workshop on Hot Topics in Networks*, Princeton, New Jersey, October 2002. ACM SIGCOMM.
 - [23] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-hop Wireless Networks. In *ACM Sigcomm*, Portland, OR, August 2004.
 - [24] R. Draves, J. Padhye, and B. Zill. Routing in Multi-radio, Multi-hop Wireless Mesh Networks. In *ACM MobiCom*, Philadelphia, PA, September 2004.
 - [25] C. Perkins (Editor). IP Mobility Support for IPv4. RFC 3220, IETF, January 2002.
 - [26] R. 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 *ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, October 2004.
 - [27] P. Gupta and P. Kumar. Capacity of Wireless Networks. In *IEEE Transactions on Information Theory*, volume 46, pages 388–404, March 2000.
 - [28] N. Jain, S. Das, and A. Nasipuri. A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks. In *IEEE International Conference on Computer Communications and Networks*, Scottsdale, AZ, October 2001.
 - [29] David Johnson, David Maltz, and Yih-Chun Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR). Internet Engineering Task Force (IETF), draft-ietf-manet-dsr-09.txt, April 2003.
 - [30] A. Kamerman and L. Monteban. WaveLAN 2: A high-performance wireless LAN for the unlicensed band. In *Bell Labs Technical Journal*, Summer 1997.
 - [31] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the Tightrope: Responsive Yet Stable Traffic Engineering. In *ACM Sigcomm*, Philadelphia, PA, August 2005.
 - [32] S. Keshav. A Control-Theoretic Approach to Flow Control. In *ACM Sigcomm*, Zurich, Switzerland, September 1991.
 - [33] Luke Klein-Berndt. Nist kernel aodv implementation (http://w3.antd.nist.gov/wctg/aodv_kernel/).
 - [34] Henrik Lundgren, Erik Nordstrom, and Christian Tschudin. Coping with Communication Gray Zones in IEEE 802.11b based Ad Hoc Networks. In *ACM International Workshop on Wireless Mobile Multimedia*, Atlanta, GA, September 2002.
 - [35] Haiyun Luo, Ram Ramjee, Prasun Sinha, and Li (erran) Li. Ucan: Unifi ed cellular and ad-hoc network. In *ACM Mobicom*, Sept 2003.
 - [36] Mahesh Marina and Samir Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of the 9th IEEE International Conference on Network Protocols*, pages 14–23, Riverside, USA, November 2001.
 - [37] J. Moy. OSPF specification. RFC 1131, Internet Engineering Task Force, October 1989.
 - [38] A. Nasipuri and S. R. Das. On-Demand Multipath Routing for Mobile Ad Hoc Networks. In *IEEE International Conference on Computer Communications and Networks*, Boston, MA, October 1999.
 - [39] A. Nasipuri and S. R. Das. MultiChannel CSMA with Signal Power-Based Channel Selection for Multihop Wireless Networks. In *IEEE Vehicular Technology Conference*, Boston, MA, September 2000.
 - [40] E. Belding-Royer P. Sambasivam, A. Murthy. Dynamically adaptive multipath routing based on aodv. In *Mediterranean Ad hoc Networking Workshop*, Bodrum, Turkey, June 2004.
 - [41] M. Papadopouli and H. Schulzrinne. A Performance Analysis of 7DS a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users. In *Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest*, Ewing, NJ, March 2001.
 - [42] M. Pearlman, Z. Haas, P. Sholander, and S. Tabrizi. On the Impact of Alternate Path Routing for Load Balancing in Mobile Adhoc Networks. In *ACM Mobicom*, Boston, MA, August 2000.
 - [43] C. Perkins, E. Belding-Royer, and I. Chakeres. Ad Hoc On Demand Distance Vector Routing. Internet Engineering Task Force (IETF), draft-perkins-manet-aodvbis-02.txt, July 2004.
 - [44] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. Internet Engineering Task Force (IETF), RFC 3561, July 2003.
 - [45] K. Ramachandran, K. Almeroth, E. Belding-Royer, and M. Buddhikot. Interference Aware Channel Assignment in Multi-Radio Wireless Mesh Networks. In *Dept. of Computer Science Technical Report, University of California, Santa Barbara*, Santa Barbara, CA, July 2004.
 - [46] Krishna Ramachandran, Elizabeth Belding-Royer, and Kevin Almeroth. DAMON: A Distributed Architecture for Monitoring Multi-hop Mobile Networks. In *IEEE International Conference on Sensor and Ad hoc Communications and Networks*, Santa Clara, CA, October 2004.
 - [47] A. Raniwala and T. Chiu. Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network. In *IEEE Infocom*, Miami, FL, March 2005.
 - [48] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In *Wireless Communications and Networking Conference*, New Orleans, LA, March 2005.
 - [49] J. So and N. H. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver. In *ACM MobiHoc*, Tokyo, Japan, May 2004.
 - [50] P. Srisuresh and K. Egevang. Traditional IP network address translator (traditional NAT). RFC 3022, Internet Engineering Task Force, January 2001.
 - [51] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Mobile Ad Hoc Networks. In *International Symposium on Parallel Architectures, Algorithms and Networks*, Dallas, TX, December 2000.
 - [52] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *Proceedings of ACM Mobihoc 2004*, Tokyo, Japan, May 2004.