

ALARM: An Adaptive Load-Aware Routing Metric for Hybrid Wireless Mesh Networks

Asad Amir Pirzada and Ryan Wishart

[†]Queensland Research Laboratory,
National ICT Australia Limited,
Brisbane, QLD 4000, Australia.

Email: {Asad.Pirzada, Ryan.Wishart}@nicta.com.au

Marius Portmann[†] and Jadwiga Indulska[†]

School of Information Technology and Electrical Engineering,
The University of Queensland, Australia,
Brisbane, QLD 4072, Australia.

Email: {Marius, Jaga}@itee.uq.edu.au

Abstract—Hybrid Wireless Mesh Networks (WMN) are generally employed to establish communication during disaster recovery operations. The Hybrid WMN network is formed in a spontaneous manner when wireless nodes belonging to different agencies are operated in the disaster area. These wireless nodes generally have heterogeneous configurations in terms of number of transceivers, computational power, battery resources and mobility pattern. Internet Protocol (IP) acts as the common platform for integrating these heterogeneous devices. Routing protocols are engaged to determine paths between sets of wireless nodes. A number of routing metrics have been developed to achieve performance gains in multi-radio, multi-hop WMNs. However, most of these metrics need access to external information like link quality statistics, channel numbers, etc on a regular basis. This frequent information exchange coupled with the incessant variation in mobility and traffic load conditions causes degraded performance of these metrics in Hybrid WMNs. In this paper, we present a routing metric named ALARM specifically designed for Hybrid WMNs. ALARM is computed using the number of packets queued per wireless interface. This computed value offers an accurate representation of the traffic load, link quality, interference and noise levels. With the help of extensive simulations, we show that our routing metric outperforms well-know routing metrics like ETT and WCETT under varying mobility and traffic load conditions in Hybrid WMNs. We further show the practicality of the metric through a prototype implementation and provide performance results obtained from a small-scale testbed deployment.¹

Keywords: Hybrid, mesh, wireless, network, routing.

I. INTRODUCTION

Wireless Mesh Networks (WMNs) are essentially modified Wireless Local Area Networks (WLANs) with the key difference being that the Access Points (APs) are interconnected wirelessly rather than with wires. In a typical WLAN the APs are responsible for providing connectivity to the mobile clients. In a WMN the Mesh Routers (MR) provide service to the mobile clients. A MR is generally equipped with multiple radios and access to either mains power or batteries offering long battery life. MRs wirelessly establish connections between each other (referred to as the back-haul) which makes WMN deployment easier and cost effective. The mobile clients in a WMN, referred to as Mesh Clients (MC), are single-radio wireless nodes with limited battery power.

¹This paper extends upon our previous ‘work in progress’ paper published in *Mobiquitous* 2007 [1]

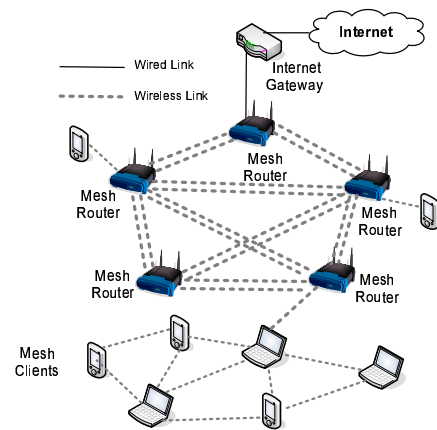


Fig. 1. A Hybrid Wireless Mesh Network

Based upon their deployment pattern, WMN can be categorised into three types: Infrastructure, Client and Hybrid [2]. In infrastructure WMNs, the MCs only communicate with each other via the MRs i.e. all communication flows via the back-haul. A client WMN is a network devoid of MRs and, hence, all communication takes place between the MCs in a purely ad-hoc manner [3] with MCs acting as packet forwarders. A hybrid WMN combines the functionality of both the infrastructure and client WMNs to form a network where the MCs can communicate with each other directly or via the back-haul. The prime advantage of a hybrid WMN is its support for a high level of network connectivity. As both MRs and MCs are able to service MCs, hence, network segregation is minimal in such networks. An illustration of an hybrid WMN is shown in Fig. 1.

All three WMNs encounter a number of challenges when it comes to actual deployment. First and foremost is the interference in the radio spectrum used by IEEE 802.11. This interference occurs due to a number of factors including: co-channel and inter-channel interference, and noise in the ISM bands [4]. Mobility is another complex issue faced by WMNs. Special protocols have to be engaged, which permit seamless roaming in the network. Route discovery is another challenging task especially in hybrid WMNs where three different types of communication modes occur: inter- MR,

inter-MC and between MRs and MCs.

In order to overcome the stated problems, current routing protocols for WMNs use a variety of techniques to minimise the affects of interference, and to support mobility and disperate communication modes. Such routing protocols enable the establishment of routes in a dynamic topology while doing so also seeking to minimize the affects of interference. Routing protocols used in WMNs can be proactive, reactive or a combination of both [5]. In proactive routing protocols the routes are periodically discovered or updated between node pairs using distance vectors or link state packets. While in reactive routing protocols routes are only discovered when required. Hence, these protocols are also known as on-demand routing protocols. Proactive routing protocols reduce route establishment delays at the cost of periodically disseminated control packets. While reactive routing protocols reduce the control packet overhead to save battery power at the cost of higher route establishment delays.

Routing protocols that are employed in WMNs use special metrics to improve upon their routing performance. These routing metrics essentially help in selecting an optimal path in the network based upon certain criteria. These criteria can vary from protocol to protocol and can be based upon hop-count, air-time, link quality, channel diversity, noise factor etc. All routing metrics have their individual merits and demerits. Some metrics are simple to implement but have minimal performance boost, while others may be complex but offer increased performance gains.

Most of the contemporary routing metrics require regular exchange of link quality statistics in order to determine the optimal paths. However under highly dynamic topologies, these metrics do not converge fast enough and discrepancies occur between the time the link statistics are gathered and are computed [6]. In addition, the link quality statistics, used by contemporary routing metrics, are computed in a distributed manner and are disseminated in the network. This process incurs additional packet overhead in the network and aggravates the prevailing interference levels.

Although all routing protocols help in selecting an optimal path during route discovery; these protocols are rarely used to subsequently improve upon the quality of the selected paths. A routing metric essentially provides a snapshot of the optimal path available in the network at a particular time. Thus a path initially selected, based upon the routing metric, remains as it is till the time the path actually breaks or the flow of traffic comes to an end. This implies that the paths selected by routing protocols are also susceptible to the traffic load conditions, interference and noise levels that occur following the initial path selection.

In this paper we present an Adaptive Load-Aware Routing Metric (ALARM). The key benefits of ALARM are that:

- It does not require access to any external information and uses only localised information to discover the least-loaded routes through the network.
- ALARM does not require any link statistics to be shared between nodes and, hence, incurs no additional compu-

tational and packet overhead.

- ALARM uses the interface queue levels to make a decision concerning the aptness of a particular wireless interface. In doing so ALARM automatically captures the combined affects of traffic load, link quality, channel diversity, interference and noise levels.
- In addition ALARM, being dependent only on localised information, permits effective route adaptation to variable traffic load conditions without initiating a new route discovery.

We have applied ALARM to AODV [7], a reactive routing protocol, and compared its performance with existing routing metrics like WCETT and ETT. With the help of results from an actual testbed implementation and network simulations, we show that ALARM outperforms other routing metrics by a significant margin in terms of network performance.

The remainder of the paper is organised as follows. In Section II, we first discuss related work focusing on routing metrics for wireless mesh networks. Our proposed metric ALARM is explained in Section III with its application to the AODV routing protocol discussed in Section IV. Section V provides a performance evaluation of ALARM, and presents and discusses our extensive simulation results. A prototype implementation of ALARM and its testbed-based evaluation are presented in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

A number of routing protocols have been developed for WMNs in recent years. Most of these protocols are based on one of the numerous Mobile Ad-hoc Network (MANET) routing protocols, which have been developed over the past two decades and have been designed for highly dynamic mobile ad-hoc networks. As a result, these protocols have excellent self-healing and self-configuration capabilities. MANET routing protocols typically use hop count as their routing metric, with their main focus on maintaining connectivity. It has been clearly shown that the hop count metric is not able to establish paths with maximum throughput in hybrid or infrastructure mesh networks [8]. This is because hop count does not allow differentiating between high quality and low quality wireless links, and therefore favours shorter paths consisting of low quality links over slightly longer paths with higher quality links. This results in the establishment of paths with non-optimal throughput characteristics.

The Expected Transmission Count (ETX) metric, proposed in [8], attempts to address this problem. ETX is a measure of link quality and it considers the predicted number of times data packets need to be transmitted and re-transmitted at the MAC layer to successfully traverse a link. The corresponding ETX metric for a path is simply the aggregation of the ETX values of the individual links of that path. ETX is defined as follows:

$$ETX = \frac{1}{d_f \times d_r}$$

The parameter d_f is the forward delivery ratio of a link, i.e. the ratio of data frames successfully traversing the link in the forward direction. The parameter d_r is the corresponding parameter for the reverse direction of the link. Both d_f and d_r can be interpreted as the probability of successfully transmitting a data frame. In order for a data frame to be successfully transmitted and acknowledged, a successful transmission in both the forward and the reverse direction is required, with the corresponding probability of $d_f \times d_r$. If attempts to send a data frame are considered as a Bernoulli trial, the ETX value represents the expected number of transmissions, i.e. the inverse of the success probability of a single transmission.

The ETX metric is generally measured using periodic link probe packets. The ratio of successfully received probes from a neighbour provides the reverse link delivery ratio d_r . Similarly, the ratio of successfully received probes by that neighbour indicates the forward link delivery ratio d_f .

The ETX metric has a number of limitations. For example, the assumption that data packets will experience the same loss rates as probe packets does not necessarily hold. Probe packets tend to be very small in size, and since they are sent via link layer broadcast, they are transmitted at the lowest possible data rate in IEEE 802.11 radios. This results in a higher probability of successful transmission and, therefore, a lower ETX value than for data packets. However, a significant shortcoming of ETX is that it does not consider the transmission rate of links. According to ETX, a slow link is considered superior to a significantly faster link if it has a slightly lower ETX value. This is a severe limitation in multi-rate networks consisting of links with varying transmission rates, which is the case for most wireless mesh networks, due to the auto-rate mechanism of IEEE 802.11 radios.

The Expected Transmission Time (ETT) metric [9] is an extension of ETX designed to overcome the problems of ETT with respect to the transmission rate of links. ETT is a measure of the expected time required to successfully transmit a fixed-size data frame on a link. It is defined as follows:

$$ETT = ETX \times \frac{S}{B}$$

S represents the packet size and B represents the bandwidth or capacity of the link. The corresponding ETT path metric is simply the sum of the ETT values of the individual links. By considering the link bandwidth, ETT is able to differentiate between links with varying capacities but similar ETX values.

ETT has significant limitations when applied to multi-radio networks. It is not interference-aware and is not able to establish channel-diverse paths that exploit the availability of multiple inter-node links. Hence, it cannot discover maximum throughput paths.

An example of a WMN routing protocol that employs the ETT routing metric is AODV-ST [10], which has been specifically designed for infrastructure wireless mesh networks. The AODV-ST protocol has been designed with the aim of providing Internet access to Mesh Clients with the help of

one or more gateways. The protocol uses a proactive strategy to discover routes between Mesh Routers and the gateways, and a reactive strategy to find routes between Mesh Routers. AODV-ST has been designed for single-radio networks and, can therefore not exploit the full potential of multi-radio nodes in the network.

The Weighted Cumulative ETT (WCETT) metric [9] has been designed to overcome the key limitation of ETT in the context of multi-radio mesh networks, by specifically considering channel diversity. The WCETT metric of a path p is defined as follows:

$$WCETT = (1 - \alpha) \times \sum ETT + \alpha \times Max X_j$$

X_j is the sum of the ETT values of links which are on channel j in a system which has k orthogonal channels. The first term of the equation simply adds up the individual link ETTs, and therefore generally favours shorter high quality paths. The second term of the equation adds up the ETTs of all links of a given channel, and then takes the maximum over all channels. A path with a large number of links operating on the same channel will, therefore, have a high value. As a result, the second term favours paths with a high level of channel diversity, and ensures low intra-flow interference. α is a tuneable parameter within the bounds $0 \leq \alpha \leq 1$, that enables one to control the preference of path length over channel diversity.

The main disadvantage of WCETT is that it incorrectly penalises channel reuse. For example, any path which reuses a channel on every third hop is considered equivalent to a path in which the channel is reused in three consecutive hops. The former path may not introduce any intra-flow interference because of the spatial separation of the channel reuse, while the latter path may induce excessive channel interference due to the proximity of channel reuse. WCETT, like ETX and ETT, also does not consider inter-flow interference, link load or link congestion when establishing paths.

WCETT has been implemented in the Multi-Radio Link Quality Source Routing (MR-LQSR) protocol developed by Microsoft Corporation. The protocol identifies all nodes in the wireless mesh network and computes ETT for all possible links. The WCETT is then applied to the Link Cache [11] scheme of the Dynamic Source Routing (DSR) [12] protocol. Thus, when the Dijkstra algorithm is executed over the link cache by a source node, the protocol returns the path with the least expected transmission time. As MR-LQSR has been developed for use in static networks only, it cannot be applied directly to hybrid WMNs. Recently, other metrics for WMNs have been proposed [13], [14], but all require some form of information exchange in one form or the other.

A number of load-aware routing protocols have been developed for MANETs [15]-[22]. Some of these protocols take the packet losses into account while others look at the overloading of the interface queues. All protocols have been developed for homogenous ad-hoc networks where all nodes have a single radio each. Consequently, no support is offered for current

```

void DropTail::enqueue(Packet* p)
{
    q_.enqueue(p);
    if (q_.length() >= qlim_) {
        q_.remove(p);
        drop(p);
    }
}

Packet* DropTail::deque()
{
    return (q_.deque());
}

```

Fig. 2. Pseudo-code for Droptail IFQ

multi-rate radios, which adapt their data rates in accordance with the Signal to Noise Ratio (SNR). In addition, as the network topology is considered extremely fluid in MANETs, none of the mentioned protocols offers a mechanism for route adaptation after the initial route development.

III. ADAPTIVE LOAD-AWARE ROUTING METRIC (ALARM)

ALARM uses the Interface Queue (IFQ) length as an indication of link load. The IFQ is a drop-tail buffer, implemented at the MAC layer of 802.11 radios, which contains outbound frames to be transmitted by the physical layer. A build up of frames in the IFQ indicates congestion, either due to traffic load or due to low link quality or interference [23]. The advantage of using IFQ lengths, and hence the level of link load, is that it subsumes a range of other link parameters, such as link quality, link capacity, interference, and noise. Furthermore, the IFQ length is information that is locally available at the data link layer, and does not require communication with other nodes or any expensive operations such as active probing.

The pseudo-code for a standard drop-tail IFQ² is shown in Fig. 2. A packet sent from the higher layers is first en-queued in the Drop-tail IFQ before being passed onto the wireless medium. In case the current IFQ length (q_length) is less than the queue size ($qlim_$), the packet is stored in the queue, else it is simply dropped. Thus by looking at the IFQ length at any instant in time, we can get an accurate estimate of the load on that particular link.

In order to make the IFQ lengths comparable between links with dissimilar data rates, we divide the IFQ length by the current data rate (BW) of a link to compute the estimated time required to empty the queue, which we refer to as the Queue Discharge Interval (QDI).

$$QDI = \frac{IFQ}{BW}$$

The QDI represents the minimum time a packet has to remain in the IFQ before being transmitted on to the physical medium. By normalizing the QDI we ensure that the QDI of different nodes with varying channel bandwidths are made comparable.

²<http://www.isi.edu/nsnam/ns/doc/node68.html>

The Cumulative QDI (CQDI) of a path consisting of n links is the sum of the QDI's of all links forming that particular path.

$$CQDI = \sum_{i=1}^n QDI_i = \sum_{i=1}^n \frac{IFQ_i}{BW_i}$$

Wireless interfaces can be operating at different data rates at any instant of time. Thus a queue attached to a lower data rate interface is likely to get saturated earlier than an equal sized queue attached to a higher data rate interface. QDI thus makes queues and interfaces of different sizes and data rates comparable. As most of the contemporary wireless cards support auto-rate data transmission with wireless card drivers supporting Auto-Rate Fallback (ARF) it is essential that the IFQ length and current data rate be monitored simultaneously. This monitoring can be done in an instantaneous manner or may be averaged over an interval. The window size of the monitoring interval is dependent primarily on the network traffic pattern. A smaller window captures the affects of burst traffic, while a larger window captures long term traffic patterns.

Some other factors that influence the IFQ length are congestion, interference and noise. As the Mesh Clients in Hybrid WMN are constantly on the move, it is very likely that a number of simultaneous flows will be traversing some of the nodes. As the wireless medium is shared, contention takes place when multiple nodes try to access it. This contention causes congestion in the IFQs of contending nodes. The congestion causes building up of the IFQ lengths. Intra-flow and inter-flow interference also induces the same affect on IFQ. This impact is directly related to the incoming and outgoing rate of wireless frames and causes the IFQ to grow and shrink accordingly. Similarly, if there is extensive noise in the wireless medium, the transmitted data packets may need to be retransmitted a number of times (due to checksum errors) before being successfully acknowledged by the recipient node. A low SNR also causes elongation of the IFQs.

IV. EXTENSION OF ALARM TO AODV

A. AODV

The Ad-hoc On Demand Distance Vector (AODV) routing protocol is a reactive distance vector routing protocol that has been optimised for mobile ad-hoc wireless networks [24]. AODV borrows basic route establishment and maintenance mechanisms from the Dynamic Source Routing (DSR) [12] protocol, and hop-to-hop routing vectors from the Destination-Sequenced Distance-Vector (DSDV) routing protocol [25]. To avoid the problem of routing loops, AODV makes extensive use of sequence numbers in control packets.

When a source node intends to communicate with a destination node whose route is not known, it broadcasts a Route Request packet. Each Route Request packet contains an ID; source and the destination node IP addresses and sequence numbers together with a hop-count and control flags.

The ID field uniquely identifies the Route Request packet. The sequence numbers indicate the freshness of control packets, and the hop-count maintains the number of nodes between the source and the destination. Each recipient of the Route Request packet that has not seen the Source IP and Route Request packet ID pair, or does not maintain a fresher (with larger sequence number) route to the destination, rebroadcasts the same packet after incrementing the hop-count. Such intermediate nodes also create a Reverse Route to the source node for a certain interval of time.

When the Route Request packet reaches the destination node, or any node that has a fresher route to the destination, a Route Reply packet is generated and unicast back to the source of the Route Request packet. Each Route Reply packet contains the destination sequence number, the source and the destination IP addresses, route lifetime together with a hop-count and control flags. Each intermediate node that receives the Route Reply packet, increments the hop-count, establishes a Forward Route to the source of the packet and transmits the packet on the Reverse Route.

To detect link breakages, AODV supports both link and network layer notifications. In the former a link layer notification is sent to the network stack when a sent data packet cannot be acknowledged after a number of retries. In the latter, periodic Hello packets are used to convey connectivity information to nodes. Both mechanisms have their individual merits and demerits. The link layer notifications provide a fast way of discovering broken links using the missing acknowledgements of data packets, however, it has no way of pro-actively detecting a broken link if the data flow is intermittent. Network layer notifications can detect link breaks in both cases with the maximum detection window equal to the Hello interval. The downside is the additional routing overhead due to the transmission and reception of periodic Hello packets. With both modes of notifications, if a link break is detected for a next hop of an active route, a Route Error packet is sent to its active neighbours that were using that particular route. In this paper, we have used AODV with Hello packets for two reasons. First we want to use Hello packets to discover multiple links to adjacent nodes and secondly we use the Hello packets to compute the ETT and WCETT routing metrics.

B. AODV-ALARM

When using AODV on a node with multiple radios, each Route Request packet is broadcast on all the node's interfaces [26]. Intermediate nodes with one or more interfaces operating on a common channel, receive the Route Request packet and create a Reverse Route that points towards the source node. If the Route Request packet is a duplicate, it is simply discarded. If we want to use the QDI metric with AODV, this approach does not work, since it does not allow selecting the path with the lowest QDI value. We therefore modified the forwarding behaviour of intermediate nodes. In our version of the protocol, which we refer to as AODV-ALARM, intermediate nodes forward the initial Route Request packet, and establish the corresponding Reverse Route. However, in AODV-ALARM,

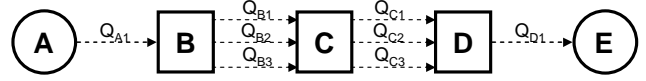


Fig. 3. AODV-ALARM Route Discovery

nodes also forward any subsequently received Route Request packets, if they have a lower CQDI value, and update the Reverse Route accordingly.

We further modified the format of the Route Request header to include a field for the CQDI, which is updated at each hop by adding the link QDI value. The CQDI value is also retained in the routing tables as the cost of the Reverse Route to the source of the route discovery. This value is compared with the QDI value of subsequently received Route Request packets and is updated in case a Route Request with a lower value is received.

When the Route Request reaches the destination, or an intermediary node with a route to the destination, a Route Reply is sent back to the originator of the Route Request. The Route Reply header has also been modified to include a field for the CQDI. This field is initialised to the QDI value of the destination node, and is updated at the intermediate nodes. This gives intermediate nodes information about the path cost to the destination node. Forward Routes are only updated as a result of receiving a new Route Reply, if the corresponding CQDI value is lower than for the current Forward Route, which prevents the problem of route flapping [27]. This is no different from the behaviour of standard AODV, with the only exception that the hop count routing metric is replaced with QDI.

The complete route discovery process of AODV-ALARM is illustrated in the example shown in Fig. 3, where two single-radio Mesh Clients (A & E) are connected via a string of three Mesh Routers (B, C & D) equipped with three radios each. The QDI of a wireless interface is represented by Q_{Ni} . Where N is the node ID and i is the interface number.

When Node A wants to establish a route to Node E, it broadcasts a Route Request with CQDI set to Q_{A1} . Lets assume that Node B receives this Route Request on all three of its wireless interfaces. Node B now creates a Reverse Route to Node A; adds the QDI value of each of its interfaces to the CQDI value in the Route Request header; and broadcasts the Route Request on all three wireless interfaces. Thus the three Route Requests sent by Node B have a CQDI value of $Q_{A1} + Q_{B1}$, $Q_{A1} + Q_{B2}$ and $Q_{A1} + Q_{B3}$ respectively. Node C can now receive one or more Route Request packets from Node B. Node C accepts a duplicate Route Request from Node B only if the CQDI of the received Route Request is lesser than the last seen CQDI value. The accepted Route Request is used to create or update the Reverse Route to Node B. Node C adds the respective QDI values of each of its interfaces to the Route Request and broadcasts the Route Request on all interfaces.

This process continues till the time the Route Request is received by Node E or any node having a prior route to Node E. Lets assume that Node E receives the Route Request packet. It sends a Route Reply to Node D on the previously created

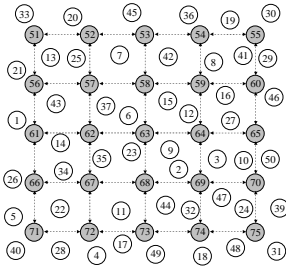


Fig. 4. Structure of the Mesh Network

Reverse Route with the CQDI equal to Q_{E1} . Node **D** adds its own QDI value (Q_{D1}) to the received Route Reply and forwards it. Eventually, the Route Reply reaches Node **A** via a path where the nodes have least saturated interfaces. Thus at the culmination of the route discovery process, a least-loaded Forward and Reverse Route is established between the source and destination nodes in a manner similar to that of standard AODV. The overhead of the route discovery in AODV-ALARM is slightly higher than in standard AODV due to the fact that multiple copies of the Route Request packet are forwarded during route discoveries. However, as our simulation results show, the benefits gained by doing so far outweigh the increased overhead.

C. Route Adaptation

When the network is static with no node mobility and no change in traffic patterns, the mechanisms discussed so far are able to establish high quality routes. However, this is rarely the case in Hybrid WMNs. The selection of links comprising a route might be optimal at the time of route establishment, but this might change drastically during the lifetime of a route, either due a change in traffic load, link quality, or interference levels. We therefore need a mechanism that continually maintains and optimises a route.

The route adaptation mechanism of AODV-ALARM does exactly that. The mechanism is implemented locally at each node, and does not incur any routing overhead. What it does is to periodically check if the QDI of any active link is beyond a given threshold. If this is the case, the node will switch to a less loaded link that is shared with the next hop, if such a link is available. This can be done very rapidly and at virtually no cost. To avoid instabilities and constant switching of links, a hysteresis function is applied. In our simulation, this adaptation process is invoked with a frequency of 1/s. Should a link break occur, the route adaptation mechanism is invoked to locally repair the route by switching to an alternative link. The route adaptation mechanism should not be confused with AODV's local repair feature, which allows a route discovery mechanism to be initiated by an intermediary node in case of a link failure close to the destination node.

V. SIMULATION RESULTS

A. Simulation Environment

We have evaluated the efficiency of ALARM through extensive simulations in NS-2 [28], using the Extended Network

Simulator (ENS) extensions [29]. The simulated network topology, as shown in Fig. 4, consists of 25 static Mesh Routers (Nodes 51 - 75) placed as a regular 5x5 grid in a 1000m x 1000m area. The connectivity between Mesh Routers is indicated via dashed lines. The network further consists of 50 mobile Mesh Clients (Nodes 1 - 50), initially placed uniformly randomly in the simulation area. Each Mesh Client is equipped with one and each Mesh Router is equipped with three radios respectively.

Concurrent Constant Bit Rate (CBR) flows using the UDP protocol are established between uniformly randomly selected source and destination Mesh Client pairs. The performance metrics are obtained by averaging the results from 50 test runs. A total of three simulations were carried out. The default simulation parameters used in all simulations, if not specifically mentioned otherwise, are listed in Table I.

TABLE I
TABLE 1: SIMULATION PARAMETERS

Examined Protocol	AODV, AODV-WCETT, AODV-ETT and AODV-ALARM
Simulation time	900 seconds
Simulation area	1000 x 1000 m
Propagation Model	Two-ray Ground Reflection
Mobility model for Mesh Clients	Random waypoint
Traffic Type	CBR (UDP)
Flow Rate	128 kbps
Packet size	512 bytes
Transmission range	250 m
RTS/CTS	OFF
Physical Data Rate	11 Mbps
QDI Window Interval	100ms

B. Performance Metrics

In each simulation, the following three performance metrics are considered:

- Packet Delivery Ratio:* The ratio between the number of data packets successfully received by destination nodes and the total number of data packets sent by source nodes.
- Routing Packet Overhead:* The ratio of control packets generated to the number of successfully received data packets.
- Latency:* The mean time (in seconds) taken by data packets to reach their respective destinations.

C. Assumptions

The following assumptions have been made in the simulations:

- All radios are statically tuned to a channel.
- All Mesh Clients and Mesh Routers have a radio tuned to a common 802.11b channel.
- The remaining radios on the Mesh Routers are tuned to orthogonal 802.11b channels.
- The transmission and reception ranges of the wireless transceivers are equal.
- All antennas are omni-directional.

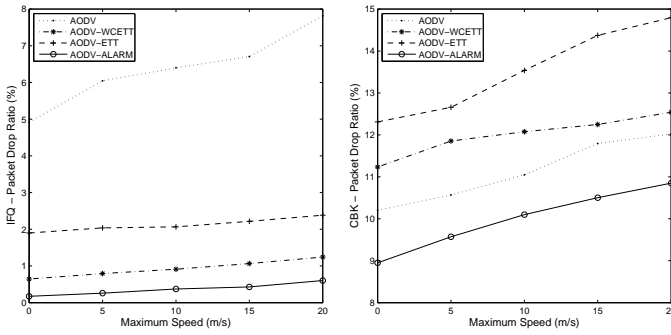


Fig. 5. Packet Drop due to Congestion and Route Unavailability

D. Mobility Model

We use the random waypoint mobility model for the Mesh Clients in our simulation. Mesh Clients first wait for the pause interval of 10 seconds, then move to a randomly chosen position with a velocity chosen randomly between 1 m/s and the maximum speed, wait there for 10 seconds, and then move on to the next random position. A positive minimum speed of 1 m/s has been set as the random waypoint mobility model fails to reach a steady state in terms of instantaneous average node speeds when the minimum speed is set to zero [30].

E. Communication Model

The IEEE 802.11 Distributed Coordination Function (DCF) [31] is used at the MAC layer. All packets are transmitted using the un-slotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) as used by Lucent Technologies WaveLAN-I. In CSMA/CA each broadcasting node waits for a vacant channel by sensing the medium. If the channel is vacant, it makes the transmission. In case of a collision, the colliding stations wait using the Ethernet binary exponential back off algorithm [32]. Virtual Carrier Sensing (RTS/CTS) is disabled during the simulations.

F. Results and Analysis

1) *Measuring Packet Drop Ratio under High Load Conditions:* In Simulation 1, we injected a high traffic load into the network to measure the packet drop ratio under load conditions. A total of fifty 128kbps CBR flows between random Mesh Client pairs are maintained while varying the maximum speed of the Mesh Clients from 0 m/s to 20 m/s. The results of the simulation are shown in Fig. 5.

During this simulation two types of packet drops were monitored: IFQ and CBK. Packet drops can occur in the network primarily due to two reasons. The first being if a route is available but the interface queue (IFQ) is saturated owing to congestion, and the second being when no route is available and a MAC Call-back (CBK) function is invoked³. The results indicate that under high load conditions and variable speeds, AODV-ALARM is able to minimise packet losses. As the routes

created by ALARM traverse interfaces that are the least-loaded, hence, we observe that the overall packet drop due to IFQ saturation is minimal. Similarly, these routes are also least affected by the contention for the physical medium and hence route breaking is minimal. This in turn lowers the number of packets dropped owing to link failures. The route adaptation process assists in selecting the least loaded links after the initial route discovery and consistently lowers the packet drop ratio until the route times-out.

AODV-ETT and AODV-WCETT look at the link qualities using the ETT metric. Although, the ETT metric has a lower IFQ packet drop ratio it has a higher CBK packet drop ratio in comparison to the hop-count metric of the standard AODV routing protocol. This occurrence takes place owing to the fact that the ETT metric selects good quality links only during the route discovery, but once the traffic starts flowing, the quality of the links in the route degrades due to intra- and inter-flow interference. This degradation causes link severing resulting in a higher number of packets being dropped due to MAC call-back. AODV on the other hand does not take the link quality into account while creating routes. Thus the number of packets dropped due to IFQ saturation is very high. This in turn reduces the load on the links resulting in lower packet drops owing to link failures.

2) *Varying the Maximum Mesh Client Speeds:* In Simulation 2, we varied the maximum speed of the Mesh Clients from 0 m/s to 20 m/s, and maintained thirty 128kbps CBR flows between random Mesh Client pairs. The results for AODV, AODV-ETT, AODV-WCETT and AODV-ALARM are shown in Fig. 6.

The results indicate when the speed of the Mesh Clients is zero, AODV achieves a 63% Packet Delivery Ratio (PDR), AODV-ETT 72%, AODV-WCETT 84% while AODV-ALARM achieves almost 90% PDR. However, the PDR starts to decline rapidly as soon as the Mesh Client speeds are increased. The PDR of AODV drops to almost 46% when the speed is increased to 20m/s. The PDR of AODV-ETT, AODV-WCETT and AODV-ALARM drop to 60%, 64% and 67% respectively for a similar increase in Mesh Client speeds.

As the network is considerably congested due to the traffic load as well as the relatively high node density, the routes created using AODV suffer from significant performance degradation. The routes created by AODV do not consider quality of the various links, and therefore establish non optimal paths. AODV-ETT and AODV-WCETT do take into account the link qualities by periodically measuring the ETTs of links. However, at higher speeds the ETT computation of both protocols becomes significantly skewed owing to intermittent receipt of Hello packets [6]. This in turn causes a discrepancy in the time the ETT is computed and the time the route is actually used. As a result, stale routes are selected leading to route severing. Additional routing queries are made to fix the broken routes, augmenting the routing packet overhead. AODV-WCETT also incurs excessive routing overhead, however owing to its naive channel diversification it is able to reduce the routing overhead significantly.

³A link is considered down after seven unsuccessful retries at the IEEE 802.11b data link layer.

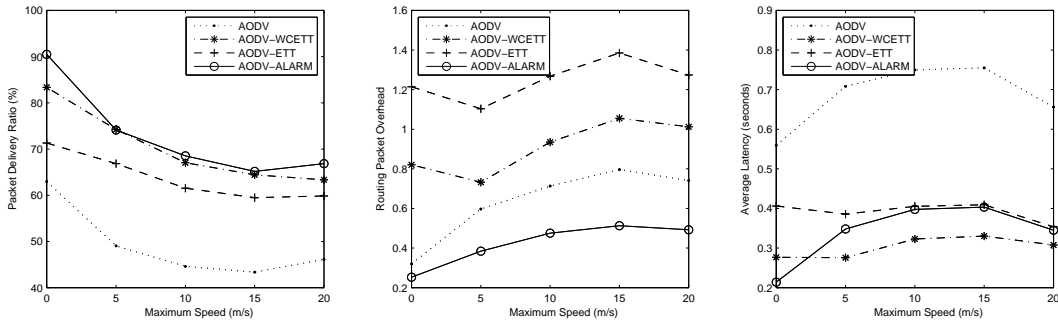


Fig. 6. Varying the Mesh Client Speeds

AODV-ALARM demonstrates a notable reduction in routing overhead compared to all three AODV variants. This relative improvement is maintained over the entire range of maximum client speeds. The main reason for this is the ability of AODV-ALARM to select the least loaded links during route discoveries and its ability to sustain the ensuing traffic through periodic route adaptation. This essentially lowers the requirement for new route discoveries reducing the overall routing overhead.

When multiple flows are directed through a single radio, there is extensive contention for the physical medium. This contention, which occurs at the link layer, causes data packets to be delayed at each hop. AODV-ETT and AODV-WCETT minimise these delays by using links which have better quality and hence offer lower delays. AODV-ALARM lowers these delays by using the least loaded links during route formation and their subsequent usage. The results indicate that as the speed is increased, the latency observed on these routes also starts to increase. This increase is essentially related to the fixed QDI Window Interval that we have used in the simulations, which needs to be dynamically adapted in accordance with the current network speed. At higher speeds the links break very frequently causing fluctuation of load on the links. Thus a shorter QDI Window Interval would capture this effect and help lower the packet latency.

3) *Varying the Number of Flows:* In Simulation 3 we vary the number of simultaneous 128 kbps CBR flows between 10 and 50 and set the maximum speed of the Mesh Clients to 1 m/s. The results of the simulation are shown in Fig. 7.

For a lightly loaded network, with only ten concurrent flows, AODV-ETT, AODV-WCETT and AODV-ALARM perform equally well. However, as soon as the load is increased beyond twenty flows, the improvements of AODV-ALARM become apparent. AODV-ALARM's ability to discover and (once the route is established) to switch flows to links that are least loaded enables it to support higher bandwidths. With an aggregate network load of more than 6 Mbps (50 x 128 kbps) AODV-ALARM is still able to maintain a PDR of almost 65%. However, the PDR of AODV, AODV-ETT and AODV-WCETT starts to degrade rapidly with increasing number of active flows. The poor performance of AODV-ETT and AODV-WCETT is linked directly to the extensive routing overhead generated at higher traffic loads. The ETT

computation is susceptible to both mobility and load [6], and hence causes fallacious route breaks in both AODV-ETT and AODV-WCETT. In addition, the increased contention and interference caused by the increased load results in severe congestion and, as a result, decreases the PDR.

The routing overhead of AODV-ETT and AODV-WCETT increases with the number of flows. This is attributable to the increased number of links which break owing to their inability to gain access to the wireless medium due to the high traffic load. Consequently, AODV-ETT and AODV-WCETT need to frequently initiate new route discoveries and in doing so flood the network with control packets. In comparison, the extensions implemented in AODV-ALARM permit it to keep the routing overhead very low compared with the basic AODV routing protocol.

The results indicate that the packet latency of all four AODV variants increases with the increased load in the network. This increase is essentially related to the high volume of traffic injected in the dense network. The latency of AODV-WCETT and AODV-ALARM remains comparable up to thirty flows. Beyond thirty flows, the PDR of AODV-WCETT falls abruptly lowering the traffic load on some parts of the network. This in turn lowers the overall packet latency as compared to AODV-ALARM, which maintains a relatively higher PDR.

VI. PROTOTYPE IMPLEMENTATION AND EVALUATION

A. Prototype Implementation

We implemented a prototype of ALARM using AODV-UU (version 0.9.5). Some of the key changes to the code that were required to implement our extensions were:

- Contrary to claims, the current version of AODV-UU (0.9.5) does not correctly support multiple network interfaces per node. We made a number of changes to the original AODV-UU code to fix this problem.
- We added a mechanism to keep track of multiple links to neighbours. This was achieved by extending AODV-UU's routing table.
- We extended the AODV-UU header to include the additional information required by ALARM.

B. Testbed Set-up

To evaluate the practical feasibility of ALARM, we implemented it with its complete set of features as discussed pre-

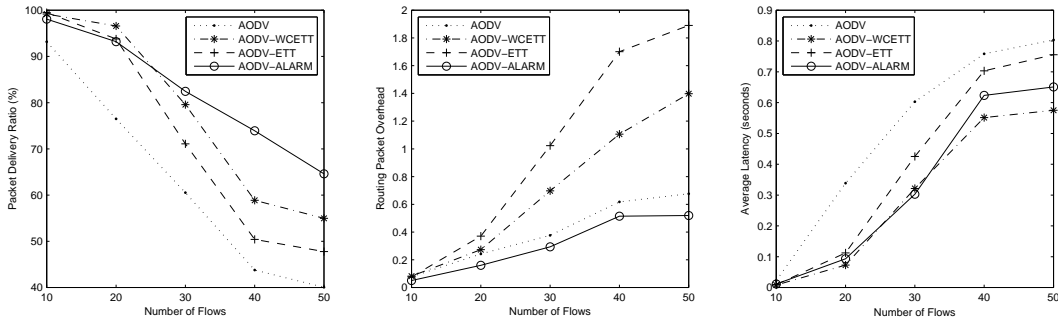


Fig. 7. Varying the Number of Flows

viously. We further carried out performance measurements on a small-scale testbed comprised of seven wireless mesh nodes as shown in Fig. 8. The hardware and software configuration are explained in the following sub-sections.

1) *Hardware*: The testbed was comprised of three multi-radio mesh routers and two single-radio mesh clients. A client and a server were also present. These nodes served as a traffic source and sink, respectively. All seven of the nodes in the testbed were conventional PCs with a VIA C7 1GHz processor and 512MB of RAM. All machines ran Linux kernel version 2.6.20 and used the 0.9.3.3 version of the open source Madwifi driver (SVN version r1639). We patched the standard Madwifi driver to provide access to the current IFQ length, required by ALARM. In our testbed setup, each of the seven machines had one wireless interface tuned to 802.11a Channel 36. The three additional wireless interfaces on the client, server and the mesh routers were tuned to 802.11a Channels 48, 64 and 157, respectively.

2) *Software*: For our tests, we used the Iperf utility to measure throughput, and ICMP Ping to measure latency. As all nodes were within one-hop range of one another in our lab, we implemented virtual topology control via MAC layer filtering using Iptables. This allowed us to establish two possible paths between the client and the server: a short three hop path via three mesh clients and a longer four hop path via the mesh routers.

In our evaluation the client sent CBR UDP traffic to the server at rates of 1Mbps, 5Mbps, 10Mbps, 15Mbps and 20Mbps. These tests were first performed with all seven nodes in the testbed running our multi-interface version of the AODV-UU code. The same tests were then repeated using the ALARM implementation. The Ping utility was also used to

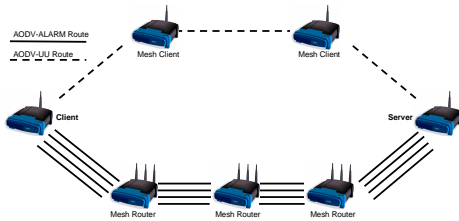


Fig. 8. Hybrid Mesh Network Testbed

generate traffic on Channel 36 across each hop in the topology. This background traffic caused congestion and increased the transmit queues of all wireless interfaces operating on Channel 36. This particularly affected the mesh clients, which only had one interface (which was tuned to Channel 36).

C. Results

Fig. 9 plots the client transmission rate versus the achieved goodput and observed latency. The results show that AODV-ALARM is able to sustain a much higher goodput than AODV-UU. As AODV-UU uses the hop count metric, it selects the shorter route through the mesh clients in our scenario. Since all hops along the route use the same channel, there is considerable co-channel interference as well as increased levels of congestion. On the other hand AODV-ALARM, being inherently load-aware, creates a least congested route via the longer path consisting of mesh routers. This considerably reduces co-channel interference, resulting in increased end-to-end goodput and reduced latency. The route adaptation process further assists AODV-ALARM in selecting the least loaded links after the initial route discovery and consistently improves the goodput and latency for the route lifetime. Even though these performance evaluations were done on a small scale, the results confirm the viability of the proposed metric.

VII. CONCLUSIONS

Hybrid Wireless Mesh Networks present the most versatile form of mesh technology, enabling mobile Mesh Clients to connect to a high speed wireless back-haul network formed

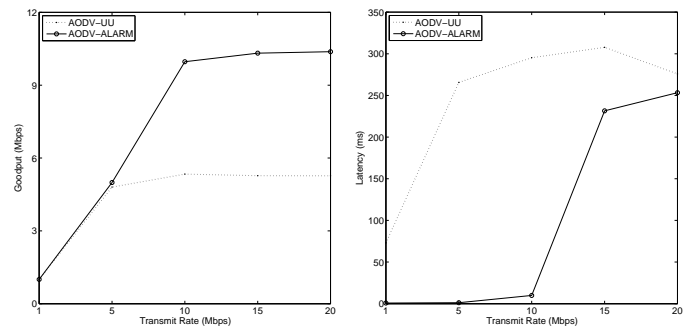


Fig. 9. Throughput and latency results under varying traffic loads

using static Mesh Routers. A major advantage of the Hybrid Mesh Network is its ability to support the back-haul using Mesh Clients in addition to the Mesh Routers. A common problem observed in this network is the performance degradation over multiple wireless hops. This occurs because of problems associated with variable traffic loads, mobility, interference and noise. A number of routing metrics have been developed to overcome these problems. However, most of these metrics require access to external information like link quality statistics, channel numbers and noise levels. This dependence on foreign information, in a dynamic network topology, causes inaccuracies in the metric computation resulting in degraded routing performance. In this paper, we have proposed a routing metric which uses the local interface queue lengths to make a decision concerning the aptness of a particular wireless interface. In doing so the metric automatically captures the combined affects of traffic load, link quality, channel diversity, interference and noise levels. In addition the metric also permits effective post-routing route adaptation to cater for variable traffic load conditions. The simulation results indicate that our routing metric outperforms well know routing metrics under varying mobility and traffic load conditions in Hybrid WMNs. Our implementation and small scale testbed evaluation confirms the practical feasibility of the proposed routing metric.

ACKNOWLEDGEMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program; and the Queensland Government.

REFERENCES

- [1] A. A. Pirzada and M. Portmann, "High Performance AODV Routing Protocol for Hybrid Wireless Mesh Networks," in *Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS)*. IEEE Press, 2007.
- [2] I. F. Akylidiz and X. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, 2005.
- [3] A. A. Pirzada and C. McDonald, "Establishing Trust in Pure Ad-hoc Networks," in *Proceedings of the 27th Australasian Computer Science Conference (ACSC)*, vol. 26. Australian Computer Society, 2004, pp. 47–54.
- [4] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006, pp. 1–12.
- [5] E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 46–55, 1999.
- [6] D. S. J. De Couto, "High-Throughput Routing for Multi-Hop Wireless Networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.
- [7] C. Bettstetter, "Topology properties of Ad hoc Networks with Random Waypoint Mobility," *Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 50–52, 2003.
- [8] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
- [9] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. ACM Press, 2004, pp. 114–128.
- [10] K. Ramachandran, M. Buddhikot, G. Chandramenon, S. Miller, E. Belding-Royer, and K. Almeroth, "On the Design and Implementation of Infrastructure Mesh Networks," in *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*. IEEE Press, 2005, pp. 4–15.
- [11] Y. C. Hu and D. B. Johnson, "Caching Strategies in On-demand Routing Protocols for Wireless Ad hoc Networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM Press, 2000, pp. 231–242.
- [12] D. B. Johnson, D. A. Maltz, and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)," *IETF MANET, Internet Draft*, 2003.
- [13] A. P. Subramanian, M. M. Buddhikot, and S. Miller, "Interference aware routing in multi-radio wireless mesh networks," in *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks*, 2006, pp. 55–63.
- [14] Y. Yang, J. Wang, and R. Kravets, "Designing Routing Metrics for Mesh Networks," in *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*. IEEE Press, 2005.
- [15] J. Gao and L. Zhang, "Load Balanced Short Path Routing in Wireless Networks," in *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2. IEEE Press, 2004, pp. 1098–1107.
- [16] H. Hassanein and A. Zhou, "Routing with Load Balancing in Wireless Ad hoc Networks," in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM Press, 2001, pp. 89–96.
- [17] S. J. Lee and M. Gerla, "Dynamic Load-Aware Routing in Ad hoc Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 10. IEEE Press, 2001, pp. 3206–3210.
- [18] K. Wu and J. Harms, "Load-Sensitive Routing for Mobile Ad hoc Networks," in *Proceedings of the Tenth International Conference on Computer Communications and Networks*. IEEE Press, 2001, pp. 540–546.
- [19] Y. Yuan, H. Chen, and M. Jia, "An adaptive load-balancing approach for ad hoc networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2, 2005, pp. 743–746.
- [20] W. Jiang, "Accurate Queue Length Estimation in Wireless Networks," in *8th International Conference on Passive and Active Measurement (PAM)*, vol. 4427. Springer, 2007, pp. 245–249.
- [21] D. Kliazovitcha and F. Granelli, "Cross-layer congestion control in ad hoc wireless networks," *Ad Hoc Networks*, vol. 4, no. 6, pp. Ad Hoc Networks Volume 4, Issue 6, November 2006, Pages 687–708, 2006.
- [22] L. Ma and M. Denko, "A Routing Metric for Load-Balancing in Wireless Mesh Networks," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, vol. 2, 2007, pp. 409–414.
- [23] A. A. Pirzada, R. Wishart, and M. Portmann, "Congestion Aware Routing in Hybrid Wireless Mesh Networks," in *Proceedings of the IEEE International Conference on Networks*, 2007, pp. 513–518.
- [24] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad hoc Network Routing Protocols," in *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM Press, 1998, pp. 85–97.
- [25] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proceedings of the SIGCOMM Conference on Communications, Architectures, Protocols and Applications*. ACM Press, 1994, pp. 234–244.
- [26] A. A. Pirzada, M. Portmann, and J. Indulka, "Evaluation of MultiRadio Extensions to AODV for Wireless Mesh Networks," in *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access (MobiWac)*, 2006, pp. 45–51.
- [27] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth, "Routing Stability in Static Wireless Mesh Networks," in *Proceedings of the Passive and Active Measurement Conference*, 2007.
- [28] NS, "The Network Simulator," <http://www.isi.edu/nsnam/ns/>, 1989.
- [29] B. Raman and C. Chebrolu, "Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM Press, 2005, pp. 156–169.
- [30] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," in *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2. IEEE Communications Society, 2003, pp. 1312– 1321.
- [31] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 802.11," 1997.
- [32] A. S. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall, 2002.