

HOVER: Hybrid On-demand Distance Vector Routing for Wireless Mesh Networks

Stephan Mir[†]

Télécom Paris, ENST,
Paris, France.

Ecole Polytechnique, Palaiseau, France.
stephan.mir@polytechnique.org

Asad Amir Pirzada

[†]Queensland Research Laboratory,
National ICT Australia Limited,
Brisbane, QLD 4000, Australia.
asad.pirzada@nicta.com.au

Marius Portmann[†]

School of ITEE,
The University of Queensland,
Brisbane, QLD 4072, Australia.
marius@itee.uq.edu.au

Abstract—Hybrid Wireless Mesh Networks are a combination of mobile ad hoc networks and infrastructure wireless mesh networks, consisting of two types of nodes: Mobile Mesh Clients and static Mesh Routers. Mesh Routers, which are typically equipped with multiple radios, provide a wireless multi-hop backhaul. The more resource constrained Mesh Clients also participate in the routing and forwarding of packets, to extend the reach of the network. Current ad-hoc routing protocols have been designed for relatively homogeneous networks and do not perform well in Hybrid Wireless Mesh Networks. In this paper, we present HOVER (Hybrid On-demand Distance Vector Routing), a modified version of the AODV routing protocol, that achieves significant performance improvements in terms of packet delivery and latency in Hybrid Wireless Mesh Networks. Our modifications include a link quality estimation technique based on HELLO packets, a new routing metric that differentiates between node types, and a channel selection scheme that minimizes interferences in multi-radio mesh networks. We present an evaluation of our improvements via extensive simulations. We further show the practicality of the protocol via a prototype implementation and provide measurement results obtained from a small test-bed.

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

I. INTRODUCTION

Wireless Mesh Networks (WMN) have recently gained considerable popularity owing to their properties such as self-configuration, self-optimizing and self-healing capabilities. The low cost of commodity IEEE 802.11 wireless hardware, on which most WMNs are based, further adds to the appeal of the technology. WMNs offer an attractive platform for a wide range of applications, such as public safety and emergency response communications, intelligent transportation systems, and community networks.

A WMN consist of two types of wireless nodes:

Mesh Routers and Mesh Clients. The Mesh Routers have improved computational, communication and power resources as compared to Mesh Clients. Mesh Routers are generally static and form the multi-hop backhaul network. A subset of Mesh Routers can also provide gateway functionality and connect the WMN to external networks. Mesh Routers are also typically equipped with multiple wireless network interfaces and are therefore able to establish high capacity connections by using multiple orthogonal channels. Mesh Clients are mobile devices, which take advantage of the existing communication infrastructure provided by the Mesh Routers. In Hybrid WMNs [1], both Mesh Clients and Mesh Routers are actively involved in routing and forwarding of packets, and Mesh Clients can access the wireless backhaul network via multiple client hops.

Current WMNs suffer from the problem of performance degradation with increasing number of wireless hops. One of the reasons is the limitation of the IEEE 802.11 MAC layer based on CSMA/CA, which has not been designed for multi-hop networks. A major problem is co-channel interference, where nodes within interference range are transmitting simultaneously on the same channel, resulting in collisions, reduced throughput and increased communication delays [2]. A further limiting factor is the half-duplex nature of IEEE 802.11 network interface cards, which does not allow simultaneous sending and receiving.

Even though some of these problems can be mitigated by providing nodes with multiple network interfaces [3] [4], the problem of finding optimal routes in WMNs, and in Hybrid WMNs in particular, is largely an open research issue. Challenges are mobility, heterogeneity and the problem of channel selection with the goal of minimizing interference.

The Ad-hoc On-demand Distance Vector (AODV)

protocol [5] is one of the predominant reactive ad-hoc routing protocols. AODV was originally developed for homogenous mobile ad-hoc networks, where nodes typically have a single wireless network interface and have comparable computational and communication resources. Consequently, AODV has some shortcomings when applied to Hybrid WMNs. First of all, with hop-count as its routing metric, it lacks the ability to differentiate between node types, i.e. Mesh Routers and Mesh Clients. It is therefore unable to exploit the heterogeneity in the network. Furthermore, AODV is not aware of the wireless channels used for the network interfaces, and therefore it cannot minimize co-channel interference and maximize the use of multiple orthogonal channels between node pairs.

In this paper, we present three modifications to AODV to address these shortcomings of AODV, and improve its performance for Hybrid WMNs. HOVER (Hybrid On-demand Distance Vector Routing), our modified version of AODV, outperforms standard AODV by a considerable margin, as we will show in Section IV and V. Our proposed modifications, and also our key contributions, are as follows:

- We propose a modification to AODV's route discovery mechanism that maximizes the involvement of Mesh Routers in the establishment of end-to-end paths. It is achieved by introducing a new routing metric that differentiates between Mesh Routers and Mesh Clients.
- In a multi-radio WMN, neighboring nodes can be connected via multiple links on orthogonal channels. We integrated a channel selection scheme into AODV's route discover mechanism that can select the "best" of these available links for each hop of the path.
- We implemented a link quality estimation scheme based on HELLO packets in AODV. HELLO packets are broadcast regularly by each node to monitor link connectivity and to detect link breaks. Hence, no additional overhead is introduced in the network.

The rest of the paper is organized as follows. Section II discusses relevant related work. The HOVER protocol is discussed in detail in Section III. Simulation results and their analysis are discussed in Section IV. A prototype implementation of HOVER and its evaluation is presented in Section V. Section VI concludes the paper.

II. RELATED WORK

AODV-ST [6] is a Hybrid routing protocol developed specifically for infrastructure mesh networks. The protocol has been designed with the aim of providing Internet access to Mesh Clients with the help of one or more gateways. AODV-ST uses a proactive strategy to discover routes between the Mesh Routers and the gateways, and a reactive strategy to find routes between Mesh Routers. In the proactive case, the gateways periodically broadcast special RREQ to initiate the creation of spanning trees. All subsequent RREQ packets with a better routing metric are used to update the existing reverse route to the gateway. AODV-ST uses the Expected Transmission Time (ETT) [7] routing metric, which measures the expected time needed to successfully transmit a fixed-size packet on a link. AODV-ST has primarily been designed for single-radio wireless nodes and hence cannot exploit the full potential of multi-radio nodes in the network. Neither does it differentiate between different types of nodes in Hybrid WMNs.

Hyacinth [3] is a multi-channel static wireless mesh network protocol that uses multiple radios and channels to improve the network performance. It implements a routing protocol and supports a fully distributed channel assignment algorithm, which can dynamically adapt to varying traffic loads. Hyacinth's channel assignment algorithm breaks a single-channel collision domain into multiple collision domains, each operating on a different frequency.

The Multi-Radio Link Quality Source Routing (MR-LQSR) [7] protocol has been developed for static community wireless networks. The protocol works in conjunction with the Mesh Connectivity Layer (MCL). The protocol identifies all nodes in the wireless mesh network and assigns weights to all possible links. The link information including channel assignment, bandwidth and loss rates are propagated to all nodes in the network. The Expected Transmission Time (ETT) on each link is computed using the Expected Transmission Count (ETX), bandwidth and packet loss. The ETT metric is further used to compute the Weighted Cumulative Expected Transmission Time (WCETT), which is a routing metric that also takes into consideration channel diversity.

The Multi-Channel Routing (MCR) protocol [4] has been developed for dynamic WMNs, where nodes have multiple wireless interfaces, each supporting multiple channels. The protocol makes use of an interface switching mechanism to assign interfaces to channels. Two types of interfaces are assumed: fixed and switchable.

Switching is carried out depending upon the maximum number of data packets queued for a single channel. The switching mechanism assists the MCR protocol in finding routes over multiple channels. MCR uses a new routing metric, which is computed as a function of channel diversity, interface switching cost and hop-counts. The diversity cost is assigned according to the least number of channels used in a route. Thus, a route with a larger number of distinct channels is considered to be having a lower diversity cost. The switching cost is used to minimize the frequent switching of wireless interfaces.

Some of the discussed works use dynamic channel allocation and switching mechanisms, combined with the routing protocol to improve the performance of the network. However the accurate and synchronized execution of these mechanisms in a mobile network requires the availability of a virtual switching protocol and incurs switching delays [8], [7]. Our protocol avoids this complexity by assuming a static allocation of channels to interfaces, while still achieving significant performance improvements.

Some related works introduce new routing metrics to improve the quality of end-to-end paths. However, to the best of our knowledge, our proposal is the first protocol that specifically addresses problem of routing in Hybrid WMNs, by differentiating between different node types. A further original contribution of this paper is the integration of an intelligent channel selection mechanism with the route discovery mechanism of a reactive routing protocol.

III. HYBRID ON-DEMAND DISTANCE VECTOR ROUTING (HOVER) PROTOCOL

Our contributions presented in this paper consist of a number of modifications to AODV. We therefore first provide a brief overview of AODV's basic route discovery mechanism before we discuss our extensions.

A. AODV

The AODV routing protocol is a distance vector routing protocol that has been optimized for ad-hoc wireless networks. It is an on demand or reactive protocol, as it finds the routes only when required. AODV borrows basic route establishment and maintenance mechanisms from the DSR protocol [9], and hop to hop routing vectors from the Destination-Sequenced Distance-Vector (DSDV) routing protocol [10]. 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 (RREQ) packet. Each RREQ contains a unique ID field. Each recipient of the RREQ that has not seen a RREQ with the same source IP and RREQ ID pair, or does not maintain a fresher (with larger sequence number) route to the destination, rebroadcasts the same packet. Such intermediate nodes also create a reverse route to the source node.

When the RREQ reaches the destination node, or a intermediate node that has a fresher route to the destination, a Route Reply (RREP) packet is generated and unicast back to the originator of the RREQ. Each RREP 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 RREP increments the hop-count and establishes a forward route to the source of the packet. Finally, the RREP is forwarded towards the originator of the RREQ using the reverse route that was established when forwarding the corresponding RREQ.

To maintain connectivity information and to detect link breaks to immediate neighbors, nodes can either use link-layer feedback or periodic Hello messages. In case a link break is detected for a next hop of an active route, a Route Error (RERR) packet is sent to the active neighbors that were using that particular link.

When using AODV on a multi-radio node, each RREQ is broadcast on all interfaces. Intermediate nodes with one or more interfaces operating on a common channel receive the RREQ and create a Reverse Route that points towards the source node. If the RREQ is a duplicate, it is simply dropped. The first RREQ received by the destination or any intermediary node is selected, and all other RREQs belonging to the same route discovery are discarded. The RREP is generated in response to the selected RREQ, and is sent back to the source node on the established Reverse Route [11].

B. HOVER

HOVER uses the same basic route discovery mechanism as standard AODV. However, in order to guarantee that routes are preferentially established via Mesh Routers, and to provide optimal link selection when multiple links exist between immediate neighbors, it implements the following three additional mechanisms:

- Node-type Aware Routing
- Link Quality Estimation
- Optimal Link Selection

1) *Node-type aware routing*: Hop-count is the default routing metric used by AODV. However, this metric does not guarantee the selection of optimal paths. A longer path consisting of high quality links can perform much better than a shorter path consisting of low quality links. Since Mesh Routers are more static, have multiple radios and are typically equipped with higher gain antennas, they are expected to maintain higher quality links with their neighbors than Mesh Clients. A longer path comprised of Mesh Routers can therefore be expected to perform better than a shorter path comprised of Mesh Clients. The basic idea is to preferentially involve Mesh Routers in the creation of end-to-end paths. This not only improves the performance of the path, but also minimizes the unnecessary draining of the batteries of the more resource constrained mobile devices.

We therefore introduce a new simple routing metric. This new metric or path cost is computed as follows:

$$Cost = MR_COUNT * MR_COST + MC_COUNT * MC_COST$$

The parameter *MR_COUNT* represents the number of Mesh Routers in the path and *MC_COUNT* stands for the number of Mesh Clients. *MR_COST* and *MC_COST* are the relative weights (or costs) associated with each type of node. In our implementation, we chose the following values for the weights: *MR_COST* = 1, *MC_COST* = 4.

The path cost is computed by accumulating the individual link costs during the route discovery process. The path cost metric is stored in a 8-bit field in an AODV extension header of RREQ and RREP packets.

During the propagation of RREQs, this cost is used for the creation of reverse routes, from destination and intermediary nodes back to the originator of the RREQ packet. When forwarding the Route Reply packets, the cost is associated with the forward routes. This mechanism is explained in more detail below.

2) *Link quality estimation*: Each node running HOVER maintains a set of links to its adjacent nodes. Since we consider multi-radio nodes, multiple links can exist between a pair of nodes. The maximum number of links is limited by the number of available interfaces on the nodes. HOVER keeps track of the quality of these links by listening to Hello packets from its neighbors. When a node receives a Hello packet from a neighbor, it marks that link, identified by the receiving interface and the IP address of the sender, as valid.

When no Hello packets are received for a given period of time (*ALLOWED_HELLO_LOSS* *

HELLO_INTERVAL), the corresponding link is marked as invalid. If the invalidated link is part of an active route, HOVER simply switches the route to use another valid link to the same neighbor. This repair mechanism is extremely efficient and quick, since it is done locally and does not involve sending any routing control packets, in contrast to AODV's local route repair mechanism.

To keep track of the quality of the links between two nodes, we count the number of Hello packets that are received and lost over a period of time. In our implementation, we set this time window to $10 * HELLO_INTERVAL$ seconds. Individual Hello counters are maintained for each neighbor and interface pair. These counters allow us to find the parameters d_r and d_f , which are the delivery ratios of Hello packets for the reverse and forward direction of a link respectively.

The parameter d_r is simply the number of Hello packets that were actually received in a given time window, divided by the number of expected Hello packets. This parameter therefore indicates the link quality for incoming traffic, i.e. the reverse route direction. Since wireless links are often asymmetric, this quality is not necessarily the same for traffic flowing in the other direction.

The parameter d_r is communicated back to the corresponding neighbor by piggybacking it on the RREP packets. For the receiving node, the received parameter d_r corresponds to its parameter d_f , i.e. the delivery ratio on the forward path. With both of these parameters, a node can compute the Expected Transmission Count (ETX) [12] parameter.

$$ETX = 1/(d_f * d_r)$$

We use ETX as a simple link quality metric for all the links that a node maintains with its immediate neighbors. This link quality metric can be used to select the best link between two nodes during route creation.

3) *Optimal link selection*: During the route discovery phase, an end-to-end path is established via a concatenation of individual links. The nodes involved in the path are determined by our node-type aware routing metric. However, if multiple links exist between two neighboring nodes in the path, we need to select one of these links to be used for the route. Our goal is to choose the link that provides the best performance for the path.

During each path discovery process, every node considers the quality of its links to the relevant neighbor and computes a preference or grading value for each link. In

addition to the ETX link quality metric, a node can take into consideration other factors for this, such as channel diversity or current channel use, e.g. determined by the number of active flows using that channel [13].

When the RREP packets are forwarded and the forward path is created, each node recommends the grading of each link on the reverse path. The grading value is piggybacked onto the RREP packets and helps the RREP recipient in determining the optimal link to be used for communication with the RREP sender.

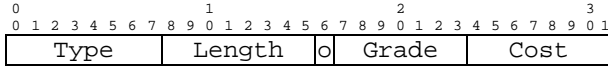


Fig. 1. HOVER Packet Extension

To convey the path cost and link grading, we add an AODV extension header (shown in Fig. 1) to all RREQ and Route Replies. The extension header further contains an optimization flag (o-flag), which will be discussed below.

Path cost and link grades are expressed as 8-bit and 7-bit fields respectively. The o-flag is used to differentiate between the non-optimal and optimal RREP packets. The extension is compliant to the AODV RFC [5] and will be ignored by nodes that do not understand the extension. Alternately, the existing reserved fields in the AODV RREQ and RREP packets can be used to convey the cost, grading and o-flag. Only the cost field is used for RREQ packets, and all three fields are used for RREP packets.

C. Path Discovery in HOVER

The basic idea of HOVER's path discovery mechanisms is to establish a potentially non-optimal path as quickly as possible, in order to minimize the path creation delay. At this stage, the path discovery mechanism is not finished, but continues to look for a more optimal path. When the optimal path has been found, the route is changed over to the new path.

This is implemented as a two phase optimization process. The first phase is carried out during the propagation of the RREQ packets, while the second phase is carried out during the propagation of the RREP packets. Phase I and II of this process are depicted in Figures 2 and 3 respectively. The behavior of standard AODV is indicated with solid lines. Our extensions are shown via shapes drawn with dashed lines, and with text below a dashed line.

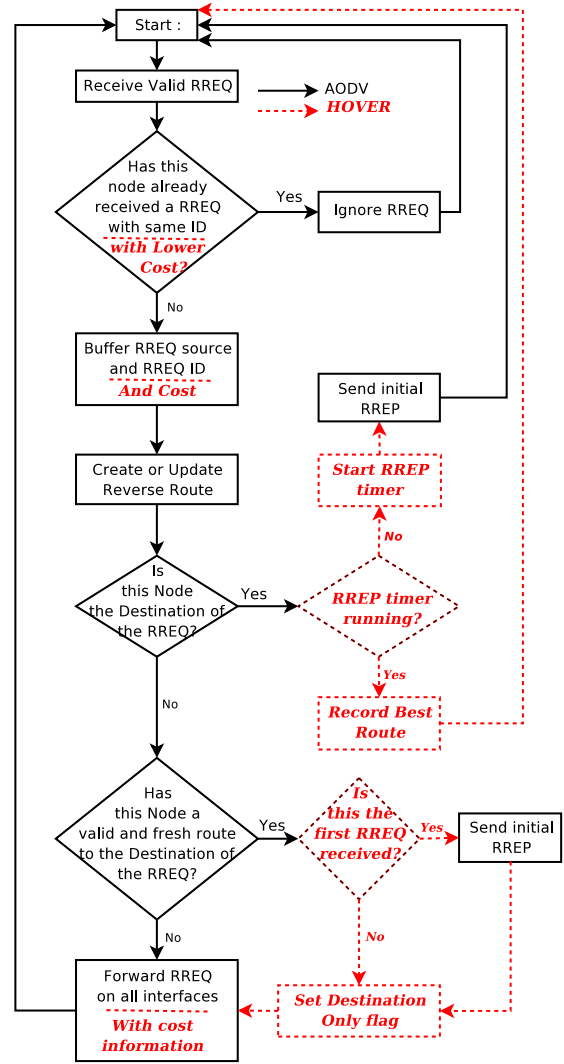


Fig. 2. Phase I flow Chart

During Phase I, each intermediary node that receives a RREQ packet first checks whether it is a duplicate or not. If this is not the case and the packet is the first RREQ with a particular ID, the node stores information about this RREQ, including its cost metric and its ID. A reverse route is then created to the source of the RREQ.

If the RREQ is a duplicate, it is only accepted if it has a lower path cost. Thus, an intermediate node may forward multiple copies of the same RREQ if the path cost of subsequently received RREQs is lower than the previously received copy. This is in contrast to AODV, where only the initially received RREQ is forwarded.

If an intermediate node has a route to the destination, it can only send a single RREP packet to the first received RREQ with the same ID. This condition is imposed in order to avoid intermediate nodes from replying to all

copies of the same RREQ, which would result in significant overhead. For all subsequent copies of the same RREQ, intermediate nodes simply set the Destination Only flag¹ in the RREQ and forward it [14].

In order to establish an initial route as quickly as possible, the destination responds immediately to this first RREQ received. To indicate that this is a temporary and possibly non-optimal route, the o-flag in the RREQ header is set to false.

This non-optimal RREP is used to establish a path from the source to the destination, which may be sub-optimal, both in terms of Mesh Client involvement and link selection. This copies the behavior of standard AODV.

In HOVER, once an initial non-optimal RREP has been sent, a destination node starts a RREP optimization timer. Until the timer expires, the destination node continues to receive and buffer RREP packets with the same ID. When the timer expires, the destination node selects the optimal RREP, i.e. the one with the lowest path cost according to our metric, and replies with a corresponding optimal RREP.

Both the non-optimal and the optimal RREP follow the Phase II process, as shown in Fig. 3. In contrast to the non-optimal RREP, the optimal RREP is sent via all wireless interfaces. Before sending the optimal RREP, the responding node sets the o-flag to true and initializes the path cost field to zero. The node further sets the link priority or link grading, based on the link quality metric or other factors such as channel diversity. This allows the node to recommend a link for the forward route to the next node.

Sending of multiple RREP packets for a single route discovery is one of the key differences to standard AODV, where the RREP is only sent on the single wireless interface on which the first RREQ was received. The primary reason for sending multiple RREPs is to give the receiving node the opportunity to select the optimal link for the forward route. This also avoids the problem of uni-directional links being created as a result of Hello packets having a longer range than the RREP packets [15], since they are sent via broadcast and therefore via a lower rate than unicast packets.

The non-optimal RREP is treated like a normal RREP, with the only difference that the path cost field is updated according to the same rule used during the forwarding of RREQ packets. If the corresponding forward route

¹The Destination Only flag indicates that only the destination can respond to this RREQ.

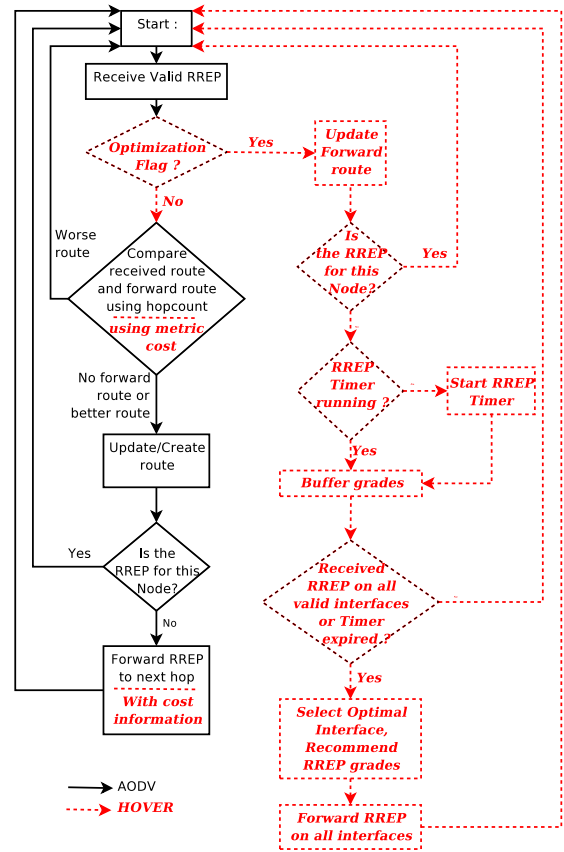


Fig. 3. Phase II flow Chart

already exists at the node that receives the RREP, the cost of the existing route is compared with the cost field of the RREP. In case the path cost value in the RREP is lower than the cost of the existing forward route, the route is updated and the RREP is forwarded. Otherwise, the RREP with a higher cost is simply discarded.

The optimal RREP is transmitted on all wireless interfaces by the destination or the intermediate nodes. Upon receipt of a RREP by the originator of the corresponding RREQ or an intermediary node, the o-flag is checked. If the flag is set, and we therefore have an optimal reply, the RREP is buffered and a RREP optimization timer is started. The node continues to receive and buffer RREP packets (including link grades) with the same ID and the o-flag set until either the optimization time expires or optimal RREP packets have been received via all the node's interfaces.

If multiple optimal RREPs have been received, the receiving node needs to select one for creating or updating the forward route to the sender of the RREP. The chosen RREP is also further sent along the reverse path if the receiving node is an intermediary node.

TABLE I
SIMULATION PARAMETERS

Examined protocols	AODV & HOVER
Simulation time	900 seconds
Simulation area	1000 x 1000 m
Propagation model	Two-ray Ground Reflection
Mobility model for Mesh Clients	Random waypoint
Maximum Speed of Mesh Clients	20 m/s
Transmission range	250 m
Number of Connections	30
Traffic type	CBR (UDP)
Packet Size	512 bytes
Packet Rate	32 pkts/sec
Transmission Rate	128 kbps/flow
Number of Mesh Routers	25
Number 802.11b radios in Mesh Router	6
Number of Mesh Clients	50
Number 802.11b radios in Mesh Client	1
RREQ Optimization Timer	1 seconds
RREP Optimization Timer	20 milliseconds

The selection of RREP can either be based on the link grading or locally available information, such as link usage or channel diversity. The main advantage of sending the link priorities or grades in the optimal RREP is that it provides the node with link quality information from the other end of the link.

This process continues until the optimal RREP is received by the source that initiated the route discovery. The source now switches from the earlier created non-optimal route to the newly formed optimized forward route.

IV. SIMULATION RESULTS AND ANALYSIS

A. Simulation Environment

We evaluated the efficiency of the HOVER protocol through extensive simulations in NS-2 [16], using the Extended Network Simulator (ENS) extensions [17]. A dense WMN covering an area of 1 square km is established using 25 Mesh Routers arranged in a regular 5x5 grid, with a distance of 176 meters between immediate neighbors. The network further consists of 50 randomly placed Mesh Clients. Concurrent UDP flows are established between randomly selected source and destination Mesh Client pairs.

The following two simulations were conducted to evaluate the performance of the HOVER protocol under varying mobility and traffic load conditions:

- Test 1: Varying Mesh Client speeds
- Test 2: Varying traffic load

The performance metrics are obtained by ensemble averaging the results from over 50 individual simulation runs for each test [18]. The parameters common to the two simulations are listed in Table I.

The simulations provide the following performance metrics:

- Aggregate Goodput: The total number of application layer data bits successfully transmitted in the network per second.
- Routing Packet Overhead: The ratio of the total number of control packets generated to the total number of data packets that are successfully received.
- Average Latency: The mean time (in seconds) taken by the data packets to reach their respective destinations.

B. Test 1 : Varying the Mesh Client Speeds

In Test 1, we have varied the speed of the Mesh Clients from 0 to 20 m/s. The results of Test 1 are shown in Fig. 4. AODV endeavors to create the shortest path (in terms of number of hops) between a source and destination node pair. Thus it essentially ignores whether a node is a Mesh Router or a Mesh Client, and focuses on rapid route creation. The rationale behind the rapid route creation is that the quickest path is most likely the optimal path. Thus paths created by AODV may or may not consist of Mesh Routers. Thus, once the traffic starts to flow over the paths, we observe a high packet loss primarily due to the interference observed in the dense network. As the number of flows is relatively high, each flow contends with other flows to gain access to the medium. Furthermore, we also have contention for the shared channel within a single flow, if a common channel is used for multiple hops.

Each flow created using HOVER initially goes over non-optimal path, just like in the case of standard AODV. However, upon expiry of the RREQ optimization timer (1 second), the flow is switched to the optimal route comprising of mostly Mesh Routers and with optimal link selection. As a consequence, HOVER achieves a significantly higher goodput compared to AODV.

The node-type aware routing mechanism allows HOVER to route most of the traffic via the Mesh Routers. Due to the availability of multiple links and the optimal link selection mechanism, HOVER is able to forward packets via paths with less contention and interference. As a result, the packet delays are significantly reduced.

The absolute number of control packets of HOVER

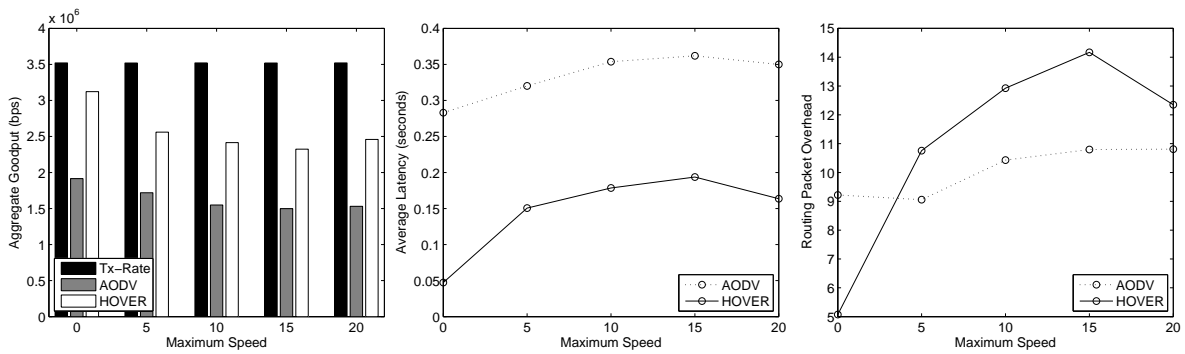


Fig. 4. Test 1 Results : Varying the mesh client speeds

is always higher than for AODV, due to the fact that multiple copies of RREQs are forwarded. However, at zero Mesh Client speed, HOVER achieves a lower routing overhead than AODV, even though it generates a higher number of control packets. This is due to the fact that the routing overhead metric is computed as the ratio of control packets to successfully received data packets, which is higher in the case of HOVER.

Once the client speed is higher than zero, HOVER has a higher routing overhead. In this case routes are less stable and need to be re-established more often, and the overhead of sending multiple copies of RREQs and RREPs starts to become more evident.

C. Test 2 : Varying the Traffic Load

The results of Test 2 are depicted in Fig. 5. In this test, we increased the number of simultaneous 128 kbps flows from 10 to 50.

In this case, the aggregate application layer data rate is more than 5Mbps (40 x 128kbps). This essentially saturates the physical layer, which has a rough effective throughput of no more than 5 Mbps [19]. However, HOVER achieves a significantly lower packet loss rate when the number of flows is less than 40.

The aggregate goodput achieved by HOVER remains higher than that of the standard AODV due to reduced packet losses at lower traffic loads. The packet delivery ratio for both protocols remains at almost 100% when the number of flows is set to 10. However, HOVER shows an improvement of around 25% in the packet delivery ratio over AODV when the number of flows is set to 20 or 30.

The latency of the packets using HOVER remains more than 100 ms lower than that observed using AODV, with 20 or more simultaneous data flows. The routing overhead of HOVER remains lower than that of the

AODV when the number of flows remains below 40. This is essentially due to the low speed of the Mesh Clients (1 m/s) maintained in Test 2.

V. PROTOTYPE IMPLEMENTATION AND EVALUATION

A. Prototype Implementation

We implemented a prototype of HOVER based on AODV-UU (version 0.9.3)². Some of the key changes to the code that were required to implement our extension are summarized in the following:

- Contrary to claims, the current version of AODV-UU (0.9.3) does not correctly support multiple network interfaces per node. We made a number of modifications to fix this problem.
- We added a mechanism to keep track of multiple links to neighbors and their respective quality metric. This was achieved by extending AODV's routing table.
- An optimization timer has been associated with each routing table entry. This timer is started when the first RREQ is received, and the optimal RREP is sent when the timer expires.
- AODV's Hello mechanism has been modified to compute quality metrics for each link.

B. Testbed Set-up

To further evaluate the efficacy of HOVER, we implemented a small testbed comprised of nine wireless nodes as shown in Fig. 6. The hardware and software setup are explained in the following sub-sections.

1) *Hardware*: The testbed consists of four Mesh Routers shown at the bottom of Fig. 6. Mesh Routers are implemented as standard PCs (AMD Sempron 2800+ processor and 512MB of RAM), running Linux kernel

²<http://core.it.uu.se/AdHoc/AodvUUImpl>

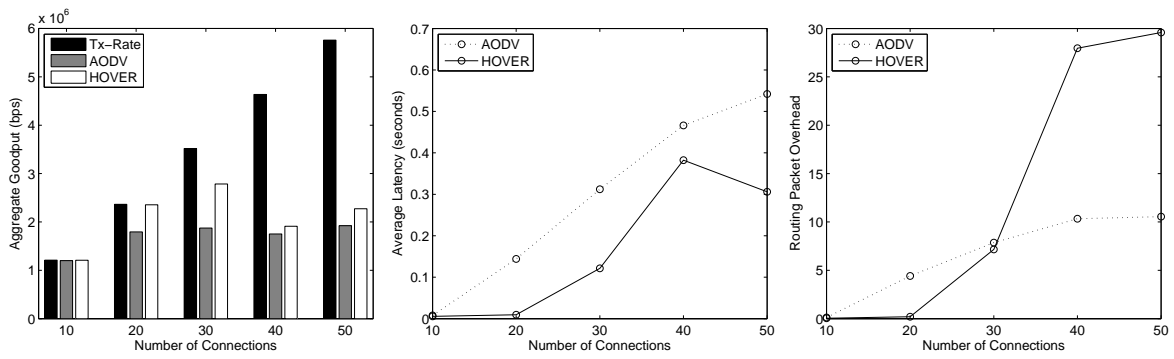


Fig. 5. Test 2 Results : Varying the traffic load

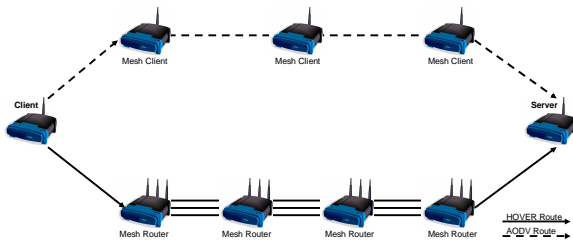


Fig. 6. Hybrid Mesh Network Testbed

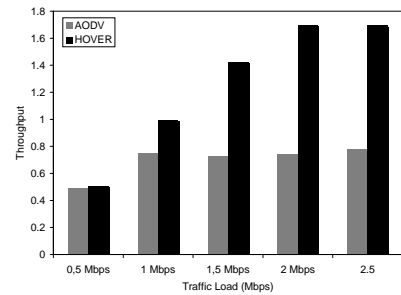


Fig. 7. Throughput results under varying traffic loads

version 2.6.8. Each Mesh Router is equipped with three mini-PCI CM9 (AR5212 802.11a/b/g) wireless interfaces each, using the Madwifi driver SVN version r1639. One interface is tuned to 802.11b channel 1, and the remaining two are tuned to 802.11a channel 42 and 160 respectively. Linksys WRT54GL wireless routers were used as Mesh Clients and were running the openWRT Linux distribution. The WRT54GLs are equipped with one 802.11b radio only, tuned to 802.11b channel 1.

2) *Software*: We used Iperf³ is to measure the network throughput and the latency was measured using ping. Since all nodes are located within close proximity of each other in our lab, and therefore are within one-hop range of each other, we implemented virtual topology control via MAC layer filtering using iptables⁴. The resulting topology is illustrated in Fig. 6. There exist two possible paths between the client and the server, a four hop path via Mesh Clients only, and a five hop path via Mesh Routers.

C. Prototype Evaluation

A UDP session was established between the iperf server and the client, both implemented as Mesh Clients

³<http://dast.nlanr.net/Projects/Iperf/>

⁴<http://www.netfilter.org>

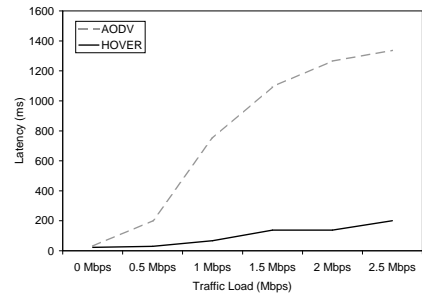


Fig. 8. Latency results under varying traffic loads

and situated at the far right and far left in Fig. 6. The offered load of each session was increased from 500 kbps to 2.5 Mbps. All results have been averaged over twenty individual test runs.

The results, shown in Fig. 7, indicate that AODV always selects the first path, due to its smaller hop-count. HOVER also initially selects the first path, but after the optimization timeout of a second, switches over to the second, more optimal path. HOVER efficiently uses the multiple links that are available between nodes in the second path to significantly increase the throughput.

The latency results, shown in Fig 8, also confirm minimal contention for the physical wireless medium

due to more optimal channel selection. The results are in line with our simulation results, in which HOVER shows a notable improvement over AODV in terms of latency. However, due to the absence of any inter-flow interference [20] in the testbed, the improvement in latency is notably higher in our testbed compared to our simulation results.

VI. CONCLUSIONS

Hybrid Wireless Mesh Networks are composed of a combination of static Mesh Routers and mobile Mesh Clients. The Mesh Routers have significantly higher computation and communication resources in comparison to the Mesh Clients. However, current routing protocols do not discriminate between the two types of nodes and are therefore not able to exploit this heterogeneity. In this paper, we presented a number of extensions to the AODV protocol that significantly improves its performance in terms of throughput and latency in Hybrid WMNs. Our extensions allow differentiating between node types and make optimal use of the multiple links that are available at Mesh Routers. We implemented three basic methods to achieve this: node-type aware routing, link quality estimation, and optimal link selection. We have demonstrated the superior performance of HOVER over AODV with the help of extensive simulations. We also demonstrated the practicality of our concept via an actual implementation, and provided further evaluations via testbed experiments.

ACKNOWLEDGEMENTS

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Research Center of Excellence programs and the Queensland Government.

REFERENCES

- [1] I. F. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, 2005.
- [2] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [3] A. Raniwala and T. C. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3. IEEE Press, 2005, pp. 2223–2234.
- [4] P. Kyasanur and N. H. Vaidya, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, 2006.
- [5] C. Perkins, E. M. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC 3561*, 2003.
- [6] K. Ramachandran, M. Buddhikot, G. Chandranmenon, 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.
- [7] 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.
- [8] R. Chandra and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks using a Single Wireless Card," in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 2. IEEE Press, 2004, pp. 882–893.
- [9] 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.
- [10] 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.
- [11] A. A. Pirzada, M. Portmann, and J. Indulska, "Evaluation of MultiRadio Extensions to AODV for Wireless Mesh Networks," *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access (MobiWac)*, pp. 45–51, 2006.
- [12] D. S. J. 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.
- [13] B. J. Ko, J. Padhye, V. Misra, and D. Rubenstein, "Distributed Channel Assignment in Multi-radio 802.11 Mesh Networks.," Columbia University, Technical report, 2005.
- [14] C. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.
- [15] I. Chakeres and E. Belding-Royer, "The utility of hello messages for determining link connectivity," in *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, 2002, pp. 504–508.
- [16] NS, "The Network Simulator," <http://www.isi.edu/nsnam/ns/>, 1989.
- [17] 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.
- [18] A. A. Pirzada, C. McDonald, and A. Datta, "Performance Comparison of Trust-Based Reactive Routing Protocols," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 695–710, 2006.
- [19] J. K. Anderson and N. Youell, "A Closer Look at WLAN Throughput and Performance," *Bechtel Telecommunications Technical Journal*, vol. 1, no. 1, pp. 86–94, 2002.
- [20] 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.