A Light-Weight Internet Gateway Discovery Scheme For Infrastructure Mesh Networks

Ryan Wishart[1], Asad Amir Pirzada[1], Marius Portmann[1,2], Jadwiga Indulska[1,2]
Queensland Research Laboratory [1]
NICTA
Brisbane, Australia
School of Information Technology and Electrical Engineering[2]
The University of Queensland
Brisbane, Australia
{Ryan.Wishart, Asad.Pirzada, Marius.Portmann, Jadwiga.Indulska}@nicta.com.au

***Abstract*— Infrastructure mesh networks offer a high-capacity wireless backhaul network through which client devices, such as PDAs, can connect to one another or with the Internet. In dynamically deployed mesh networks the routers within the mesh network may be unaware of existing Internet gateways and need to discover them on demand. In this paper we present a light-weight gateway discovery and traffic forwarding approach for discovering the presence of these Internet gateways and managing communication with them. A comparison of our work with existing approaches shows that our method is superior in terms of latency and per packet overhead.**

I. Introduction

Mesh networks are characteristically self-configuring and self-healing wireless multi-hop networks, making them very robust and quick to deploy. These features make wireless mesh networks an interesting technology for a wide range of applications, including public safety and emergency response communications. In this paper we focus on infrastructure wireless mesh networks, in which nodes referred to as *mesh routers* provide a wireless multi-hop backbone network for client devices, which do not actively participate in routing and forwarding of packets.

In a typical infrastructure mesh network, mesh routers are equipped with multiple wireless interfaces. One of these interfaces is normally allocated for communication with client devices (referred to in this paper as the *client interface*), and the others are used for backhaul communication. This is in contrast to client or hybrid mesh networks, where client devices also run a routing protocol and take part in the forwarding of packets [1].

In infrastructure mesh networks, mesh routers serve as wireless access points for the client devices within one-hop radio range. This means that all traffic to and from a client device will go via the corresponding mesh router. Note that here our use of the term "access point" does not imply use of IEEE 802.11 infrastructure mode for the wireless interfaces concerned. For clarity we refer to mesh routers that act as access points for client devices as "access routers".

Figure 1 shows a simple example of an infrastructure mesh network consisting of three mesh routers, four client devices and a gateway mesh router that connects the infrastructure

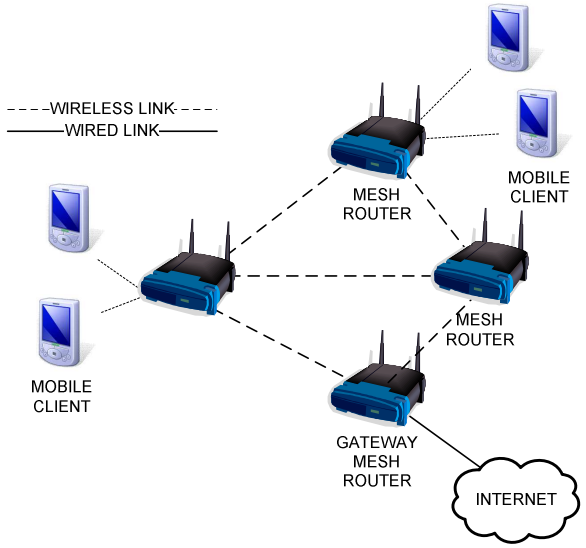

Fig. 1. An Example Infrastructure Mesh Network

mesh network to the Internet. All of the mesh routers in the example mesh network can act as access routers for client devices and have a dedicated wireless interface for this purpose (referred to as a *client interface*).

While our example infrastructure mesh network has only a single gateway mesh router, in a realistic infrastructure mesh network there may be several such gateway mesh routers available.

In this paper we present a new approach to gateway mesh router discovery for infrastructure mesh networks. Our approach requires only minor modifications to the widely used Ad hoc On-Demand Distance Vector routing protocol (AODV) [2] to function. It exhibits low gateway discovery latency and requires fewer gateway discovery messages than many existing approaches. Additionally, tunneling of packets through the mesh is not required.

The remainder of this paper is structured as follows. In Section II we present background information on our approach including an overview of the AODV routing protocol's operation and our work on supporting client devices within

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
 ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3171

infrastructure mesh networks. A critical overview of related work is then performed in Section III before we discuss our approach to gateway mesh router discovery and use in Section IV. The results of a comparison between our novel approach and existing work are then presented and discussed in Section V. In Section VI we conclude the paper.

II. Background Information

A. Overview of AODV

In the standard AODV routing protocol [2], route discovery is initiated when a node has a packet it wants to send but has no route to the packet's destination. This results in the node broadcasting a Route Request message (RREQ) to all its one-hop neighbours.

Nodes that receive a RREQ that are (1) not the requested destination, or (2) do not have a fresh route to the destination forward the RREQ to all their one-hop neighbours. It should be noted that a node only forwards a RREQ if it has not received that RREQ before, or the metric associated with the RREQ is better than the metric it currently has for the route to the RREQ source. In this way the RREQ is flooded through the network with its spread controlled by a time to live field (decremented on each hop). Once this time to live value reaches zero, the RREQ is dropped.

When a RREQ is received by the requested destination node, or a node with a fresh route to the destination, a Route Reply (RREP) message is sent. This RREP travels back along the reverse path over which the corresponding RREQ was received. Each node that receives the RREP creates a route back to the sender of the RREP. When the RREP is received by the source of the RREQ, a bi-directional route exists between it and the requested destination.

B. Our AODV modifications to support client devices

In our modified AODV implementation (presented in [3]), a mesh router maintains a list of client devices (referred to as its *client list*) for which it currently acts as an access router.

When a mesh router starts to receive packets from a previously unknown client device, it adds this client device to its client list. In case the mesh router does not have a route to the destination of the packets received from the client device, it will find a route on behalf of the client device by initiating AODV route discovery.

If a mesh router does not receive any data from a client device for a certain amount of time (900ms in our current implementation), the corresponding client list entry expires and is deleted.

When a mesh router receives a RREQ message, it checks if the destination node is in its client list. If this is the case, the mesh router will reply with a RREP on behalf of the client device, indicating that it has a route to the destination node.

The problem with this approach is that a client device only appears in the client lists of a mesh router if it is actively sending data. This makes it impossible to establish routes to silent client devices.

We solve this problem with a small modification of AODV's route discovery mechanism. Each mesh router that receives a RREQ, and which does not find the destination address in its client list, will send an ICMP ping message to the destination address via its client interface. If the client device happens to be within range, it will respond, upon which the mesh router adds the client device to its client list and responds to the RREQ with a corresponding RREP message.

Should a client device be within range of multiple mesh routers, it will receive and reply to multiple pings. This results in multiple RREP messages being sent to the originator of the route discovery. In our implementation, the first RREP received by the node that initiated route discovery is chosen and the corresponding route is used.

III. Related Work

In this section we provide a critical overview of the related work in the field.

A. Half tunnels and default routes

Nordstrom *et al.* [4], [5] address the problem of gateway discovery and forwarding within a Mobile Ad hoc Network (MANET) using a "half-tunnel" approach. Their work can be applied to an infrastructure mesh network as follows. All mesh routers within the mesh network run the AODV routing protocol. When a client wishes to communicate with an Internet host, its access router launches route discovery using the standard AODV approach described in Section II-A. In the approach only gateway mesh routers can distinguish Internet host addresses from addresses used within the mesh network. Any gateway mesh router that receives a RREQ for an Internet host address generates a RREP message with its address as the RREP source and the IP address of the requested Internet host in a special RREP header extension. This RREP is sent back along the route taken by the RREQ to reach the gateway mesh router.

Upon receipt of this RREP, the client's access router then tunnels all traffic destined for the Internet host via the gateway mesh router. Outbound traffic is Source NATed by the gateway mesh router, while inbound traffic (i.e. destined for a client in the mesh network) is Destination NATed before being routed on the mesh network. Traffic destined for client within the mesh network is not tunneled as the destination address is routable within the mesh network.

The default route approach described by Nordstrom *et al.* in [4] employs the same gateway discovery approach employed in half-tunneling. However, the RREP from the gateway is used differently. Each recipient of a gateway RREP sets its default gateway to be the one hop neighbour that sent the RREP. This means only the mesh router one hop from the gateway mesh router actually knows the gateway's address - all other nodes see the next hop in the path to the gateway as their default gateway.

As Nordstrom *et al.* assume that the mesh network does not use its own address space (e.g., 10.0.0.x), an entry in the routing tables of all nodes on the path to the gateway is required for each Internet host that has been discovered.

A further drawback of these two schemes is that the gateway discovery process is repeated each time a new Internet host needs to be contacted by the client devices.

Additionally, the half-tunneling approach uses IP encapsulation for outbound traffic. This equates to at least 8 bytes of additional header on all outgoing traffic.

B. AODV-ST

Ramachandran *et al.* [6] use a modified version of AODV to proactively establish routes to gateway mesh routers within the infrastructure mesh network. They require each of the gateway mesh routers to regularly send special gateway RREQ messages. These messages have the sending gateway mesh router's address in the source field, and the destination field set to the mesh network's broadcast address (e.g., 10.0.255.255).

Mesh routers receiving these RREQ messages (1) create a reverse route to the sending gateway mesh router, (2) send a gratuitous RREP back to the sending gateway mesh router and (3) forward the RREQ on to its neighbouring mesh routers.

The gratuitous RREP is a RREP sent without a RREQ first being received. As the gratuitous RREP contains the IP address of the sending mesh router, the gateway mesh router is able to determine which mesh routers are downstream of it (and are thus likely to use the gateway mesh router).

It should be noted that the mesh routers only rebroadcast the gateway RREQ messages for the best path to the gateway (determined using a metric in the RREQ header). Each of the mesh routers in the network selects the gateway mesh router with the best routing metric (as provided in the gateway RREQ header) to use as its default gateway. Traffic destined for Internet hosts is then forwarded to this default gateway through the mesh network.

As with the tunneling approach of Nordstrom *et al.*, the gateway mesh routers Source NAT all traffic destined for Internet hosts, and apply Destination NAT to all traffic from the Internet destined for the mesh network.

The downside of this approach is the large cost of proactively maintaining routes (both to the gateway and from the gateway) that may never be used.

C. Mobile NAT

In the Mobile NAT mobility scheme described by Buddhikot *et al.* [7] client devices connect to one another and external networks via access routers. All traffic from clients is sent via their access router which Source NATs the traffic to make itself appear as the origin of the traffic. The access router then tunnels the traffic to the networks single gateway mesh router which again applies Source NAT to the traffic before sending the traffic out onto the Internet. Inbound traffic from the Internet is Destination NATed by the gateway mesh router before being tunneled to the responsible access router who then forwards the packets to the destination client (determined again by Destination NATing the traffic).

As a consequence of using tunneling, the routing tables of the mesh routers are compact (all communication between clients and Internet hosts is tunneled and thus there are no route entries for clients or Internet hosts). However, the reliance on tunneling introduces significant overhead (i.e. at least 8 bytes on all inbound and outbound packets). Furthermore, the approach is designed to support only a single gateway mesh router and it is assumed that all mesh routers are configured with the address of this gateway mesh router leading to a highly static configuration that cannot cope with loss of the gateway.

IV. A Light-Weight Gateway Discovery and Traffic Forwarding Approach

In this section we present our Light-Weight Gateway Discovery Protocol (LGDP) for infrastructure mesh networks. LGDP operates by first discovering a gateway mesh router using a modified version of AODV's route discovery process (using RREQ messages). This gateway mesh router is then set as the default gateway on each mesh router along the path from the RREQ origin to the gateway.

To facilitate our approach we assume that client devices and mesh routers use a non-Internet routable address space (such as 10.0.1.X). As such, mesh routers running LGDP are able to determine if a packet is destined for a mesh network node, or for a node external to the mesh network (i.e. an Internet host).

Our approach operates as follows. Traffic generated by a client and destined for an Internet host is received by that client device's access router. The access router then checks if it already has a default gateway set in its routing table.

Should the access router not have its default gateway set, it creates a special gateway discovery RREQ message. This RREQ differs from a standard AODV RREQ in that (1) the destination IP is a reserved address not used during normal routing (e.g., 255.255.255.255) and (2) an additional flag is set in the RREQ which we refer to as the *gateway discovery flag*. The source of the RREQ is the IP address of the client seeking to contact an Internet host.

In our approach, a RREP can either come from a gateway mesh router, or from a mesh router replying on behalf of a gateway to which it already has a route. This RREP is created with the gateway mesh router's IP address in the source field and then sent over the same path as the RREQ but in the reverse direction. Each mesh router that receives the RREP sets the default gateway in its routing table to the source IP address in the RREP. This RREP is then forwarded onto the next mesh router in the path towards the client device.

When the RREP reaches the access router that initiated the route discovery, each of the mesh routers on the path to the gateway mesh router will have their default gateway set.

If there are multiple gateway mesh routers within the network, the access router that initiated the route discovery may receive multiple RREPs. In our approach the first RREP received is used, subsequent RREP messages are discarded.

Once the access router has its default gateway set it is able to forward traffic from the client device to the gateway mesh router. The gateway mesh router then applies Source NAT to this traffic before sending it out on the Internet.

Traffic received from an Internet host and destined for a mesh network address is first Destination NATed by the gateway mesh router. The gateway mesh router then forwards the traffic onto the mesh network. If the gateway mesh router has no route to the destination within the mesh network, it initiates route discovery (see Section II).

It should be noted that the gateway mesh router may be more than one hop away from the RREQ origin. This runs counter to the usual use of default gateways in routing (which typically are the next hop for packets the router doesn't know how to route). To make the gateway mesh router appear one hop away we insert an entry into the ARP cache (used to store IP to MAC address mappings discovered by the Address Resolution Protocol [8]) of each mesh router on the path to the gateway linking the gateway mesh router's IP address (e.g., 10.0.1.100) with the MAC address of the next hop towards the gateway mesh router. This task is performed when a RREP is received from a gateway mesh router.

The main advantages of our approach are:

- Access routers initiate route discovery to the gateway mesh router only once, not for each client request (as in [4]).
- The reverse route from the gateway mesh router to a client device is only discovered if the client is engaged in bi-directional communication with an Internet host (e.g., a web server).
- Intermediate mesh routers on the path between the gateway mesh router and the access router learn of the gateway mesh router from RREP's they receive. Gateways do not need to proactively announce themselves (as is the case with AODV-ST [6]).
- Mesh routers use the first gateway mesh router that responds to their gateway discovery message. In networks where there are multiple gateways available, we assume that mesh routers will use their closest gateway (as this is likely to be the first to respond to any gateway RREQ message). This ensures outbound traffic is localized to a nearby gateway reducing the overall load on the mesh network.
- Our scheme does not require tunneling of client traffic through the infrastructure mesh to the gateway mesh router. This avoids the overhead associated with tunneling, calculated to be at least 8 bytes of additional header on each IP packet (assuming use of the minimal IP encapsulation scheme described in [9]).

V. Evaluation

In this section of the paper we present an evaluation of LGDP in which we compare it against the half-tunneling approach developed by Nordstrom *et al.* [10], [4]. The half-tunneling approach was chosen for the comparison as (1) it was implemented using the AODV-UU [11] code on which our client-support and LGDP code was developed and (2) it was freely available. So as to make the half-tunneling code comparable to our LGDP we patched it to include our client-support functionality.

Tests for the evaluation were conducted on a testbed of five nodes. All the nodes in the testbed used Ubuntu 7.04 Server Edition (Linux kernel 2.6.20) and were equipped with four Atheros [12] IEEE 802.11 a/b/g wireless cards. The Madwifi [13] wireless network card driver was used on all five machines. A laptop with one 802.11 b/g interface was used as the client device.

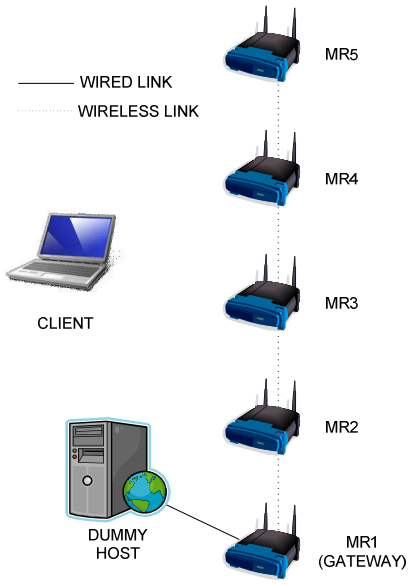

Fig. 2. Testbed chain topology

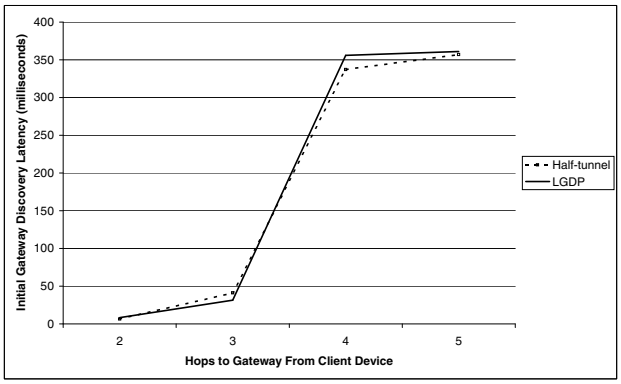

Fig. 3. Gateway discovery latency vs hops to the gateway from the client device

A. Evaluation 1

In the first part of our evaluation we examined the gateway discovery latency for LGDP and Nordstrom's half-tunneling approach. To do this we arranged the five testbed nodes (MR1, MR2, MR3, MR4 and MR5) into a 4-hop chain topology (shown in Figure 2). MR1 was designated as the gateway mesh router GATEWAY. Orthogonal wireless channels were assigned to each link so as to eliminate any possible co-channel interference (i.e. the first link used 802.11b channel 1, the second link 802.11a channel 36, etc.)

The ICMP ping utility was then run on the client device and used to ping the non-mesh address of the gateway node. This triggered the gateway discovery process. By examining the log files produced by the AODV-UU code, we could then determine the amount of time taken to find the gateway mesh router. The test was performed with the client device being 2 hops, 3 hops, 4 hops and 5 hops away from the gateway. That is the client used MR2, MR3, MR4 and then MR5 as its access router. The tests were initially performed with LGDP running on the mesh routers. These were then repeated using the half-tunneling version of the AODV code. The results of

these tests are shown in Figure 3.

As expected, the amount of time to discover the gateway mesh router increases as the number of hops increases. Surprisingly, there was a large jump in the discovery time for paths longer than 3 hops (i.e. more than 2 hops through the mesh). Further investigation ascertained that the effect was due to the expanding ring search technique used by AODV. This technique searches increasingly larger sections of the network, centered on the RREQ initiator, until the destination is found or the maximum ring size is reached. The initial ring diameter in AODV is set to 2 hops meaning RREQ messages can initially only travel 2 hops from the RREQ initiator. If the destination is not discovered within a 2 hop distance, the ring size is increased and a new RREQ message with a larger Time To Live value is sent (after an appropriate timeout period). This timeout period is responsible for the large jump in gateway mesh router discovery time observed in our experiments.

A comparison of the gateway mesh router discovery times for LGDP and half-tunneling (see Figure 3), shows that the two approaches are comparable. The minor variations observed in Figure 3 are attributable to variations in the wireless medium.

B. Evaluation 2

In the second part of the evaluation we looked at the time for LGDP and half-tunneling to discover the gateway mesh router and then access six different Internet hosts. The topology shown in Figure 2 was used again for this evaluation with the client device using MR3 as its access router. This configuration represented the simplest topology in which the access router was not directly connected to the gateway mesh router. This ensured that the ARP cache modification approach described in Section IV was used by LGDP.

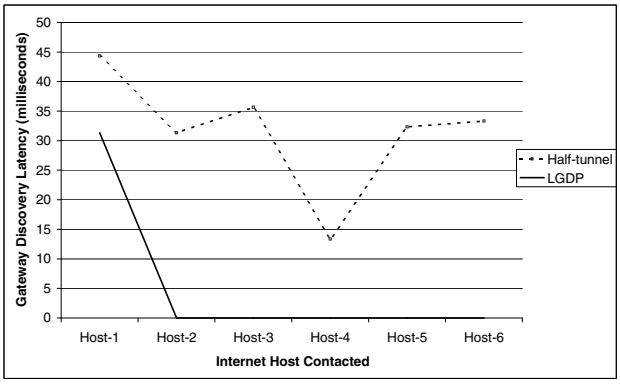

Fig. 4. Latency for Repeated Accesses

In this test, the client-device generated ICMP ping messages to six different Internet hosts (Host-1, Host-2, Host-3, Host-4, Host-5, Host-6). These messages were forwarded by the client to its access router (MR3). The access router was responsible for initiating gateway discovery within the mesh network.

The six different Internet hosts were approximated using a node referred to as "DUMMY HOST". This node had a wired connection to the GATEWAY. The interfaces on the DUMMY HOST were configured using a different subnet to that used by the mesh network. That is, our mesh network operated on the 10.0.1.x subnet, while the DUMMY HOST's network interfaces were set to the 192.168.0.x subnet. Six IP addresses were assigned to the DUMMY HOST to approximate the six Internet hosts for the test.

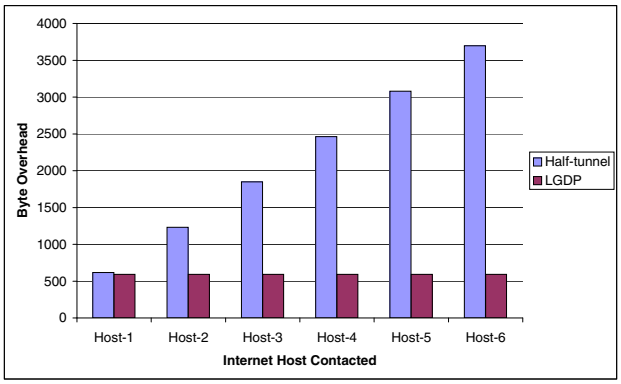

Fig. 5. Control Message Overhead in Bytes

The GATEWAY also had one interface on the 192.168.0.x subnet to enable it to communicate with DUMMY HOST.

The six Internet host addresses were first pinged while the mesh was running our LGDP. This was then repeated with the mesh running Nordstrom's half-tunneling approach. The time to discover the gateway is shown for each of the six Internet hosts in Figure 4.

Figure 4 shows that the gateway discovery latency was approximately 31 milliseconds for LGDP and 45 milliseconds for Nordstrom's half-tunneling approach. As mentioned in Section IV, our LGDP approach performs this discovery process only once while Nordstrom's half-tunneling approach does gateway discovery for *each* Internet host contacted. As a result, LGDP has zero seconds gateway discovery latency for all Internet hosts after the first. In comparison, in Nordstrom's approach the gateway discovery latency remains between 30 and 35 milliseconds for all six Internet hosts with small fluctuations attributable to wireless channel noise.

The overhead (in total number of bytes sent) associated with discovering a gateway in both LGDP and the half-tunnel approach was then compared. In our LGDP implementation gateway discovery RREQ and RREP messages had a combined size of 148 bytes. In the half-tunnel approach the Route Request and Route Reply messages had a combined size of 154 bytes.

In our testbed chain topology, five mesh routers (MR1, MR2, MR3, MR4, MR5) are present. In an ideal case where RREQ TTL issues are ignored, this topology requires at least four RREQ and four RREP messages (i.e. one RREQ and one RREP each hop) to create a route between the client and the gateway via the mesh. For LGDP this amounted to 592 bytes, while for half-tunneling this totalled 616 bytes of control message overhead for gateway discovery.

The total number of bytes sent in the process of discovering gateway mesh routers is plotted in Figure 5. As can be seen, the total number of bytes sent to facilitate gateway mesh router discovery in the half-tunneling approach grows linearly with the number of Internet hosts pinged. Importantly, in LGDP a mesh router discovers the gateway mesh router once.

Subsequent accesses to external address are then forwarded to this gateway mesh router. This is reflected in Figure 5 where the control message overhead for LGDP remains constant at 592 bytes.

VI. Conclusion

In this paper we presented a light-weight gateway discovery protocol, LGDP, for infrastructure mesh networks. In our approach one or more gateway mesh routers with connectivity to the Internet are present within the mesh. LGDP operates by efficiently discovering the closest gateway mesh router (assuming that the closest gateway mesh router is the first to respond to a gateway discovery request). This provides localization of Internet-bound traffic that reduces the load on the mesh network.

In LGDP, each mesh router performs gateway discovery only once. As shown in our evaluation, this provides a significant improvement over existing approaches (such as that of Nordstrom *et al.* [4]) where each different Internet host contacted by the client device requires the gateway discovery process be repeated.

Additionally, intermediate mesh routers on the path between the gateway mesh router and the access router learn of the gateway mesh router from RREP's they receive. This is much more efficient than the AODV-ST approach used by Ramachandran *et al.* [6] where gateways need to proactively announce themselves.

Further performance gains in LGDP are achieved by having the gateway mesh router only discover the reverse route back to the client node if the client is engaged in bi-directional communication with an Internet host.

Unlike many of the existing approaches (e.g., [4], [7]), LGDP does not tunnel traffic through the mesh network. This reduces the per-packet overhead (of at least 8 bytes per IP packet [9]) resulting in higher goodput.

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] I. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, 2005.

[2] C. Perkins, E. Belding-Royer, and S. Das, "RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing," July 2003, status: EXPERIMENTAL. [Online]. Available: http://www.faqs.org/rfcs/rfc3561.html

[3] R. Wishart, A. Pirzada, and M. Portmann, "A Light-Weight Client Mobility Approach for Infrastructure Mesh Networks," in *Proceedings of the 15th IEEE International Conference on Networks (ICON 2007)*, 2007.

[4] E. Nordstrom, P. Gunningberg, and C. Tschudin, "Comparison of Forwarding Strategies in Internet Connected MANETs," *Mobile Computing and Communications Review*, vol. 8, no. 4, pp. 72–76, 2004.

[5] ——, "Comparison of Gateway Forwarding Strategies in Ad hoc Networks," Department of Information Technology, Uppsala University, Tech. Rep. 2004-007, 2004.

[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 WiMesh 2005*, October 2005.

[7] M. Buddhikot, A. Hari, K. Singh, and S. Miller, "MobileNAT: A New Technique for Mobility Across Heterogeneous Address Spaces," *Journal of Mobile Networks and Applications (MONET)*, vol. 10, no. 3, pp. 289–302, 2005.

[8] D. Plummer, "RFC 826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware," Nov. 1982, status: STANDARD. [Online]. Available: ftp://ftp.internic.net/rfc/rfc826.txt, ftp://ftp.math.utah.edu/pub/rfc/rfc826.txt

[9] C. Perkins, "RFC 2004: Minimal encapsulation within IP," Oct. 1996, status: PROPOSED STANDARD. [Online]. Available: ftp://ftp.internic.net/rfc/rfc2004.txt, ftp://ftp.math.utah.edu/pub/rfc/rfc2004.txt

[10] E. Nordstrom and C. Tschudin, "MANET Internet Connectivity with Half Tunnels," in *Proceedings of the 1st Swedish National Computer Networking Workshop (SNCNW 2003)*, 2003.

[11] E. Nordstrom, "University of Uppsala open source implementation of AODV." [Online]. Available: http://www.docs.uu.se/docs/research/projects/scanet/aodv/aodvuu.shtml

[12] Atheros, "Atheros WiFi Chipset." [Online]. Available: http://www.atheros.com

[13] Madwifi, "Madwifi open source driver for the Atheros chipset." [Online]. Available: http://madwifi.org