

A Light-Weight Client Mobility Approach for Infrastructure Mesh Networks

Ryan Wishart, Asad Amir Pirzada

Queensland Research Laboratory

NICTA*

Brisbane, Australia

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

Marius Portmann*

School of Information Technology and Electrical Engineering

The University of Queensland

Brisbane, Australia

marius@itee.uq.edu.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 external networks. To use the mesh network, a client must route its outbound traffic via one of the mesh routers in the infrastructure mesh. As the clients are mobile, they may move out of range of the mesh router they were using and need to associate with another. Client handoff mechanisms enable this change in mesh routers to occur in a manner that limits disruption to any transport or application layer sessions the client may be running. In this paper we present an extremely light-weight handoff approach for clients that relies on gratuitous ARP messages broadcast at regular intervals from mesh routers within the infrastructure mesh. An evaluation of our approach using a 5 node testbed has shown that client handoffs can be conducted quickly, and with minimal loss of packets for both TCP and UDP traffic.

Keywords: infrastructure mesh networks, routing, mobile client handoffs

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 clients, and the others are used for backhaul communication. This is in contrast to client or hybrid mesh networks, where clients also run a routing protocol and take part in the forwarding of packets [1].

In infrastructure mesh networks, mesh routers therefore serve as wireless access points for the clients within one-hop radio range, which means that all traffic to and from a client will go via the corresponding mesh router. Note that our use of the term “access point” does not imply use of

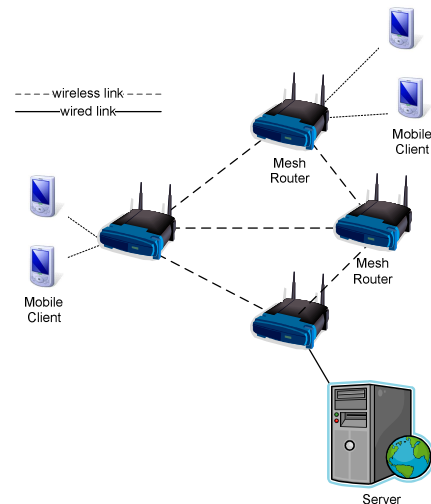


Fig. 1. Infrastructure Mesh Network

IEEE 802.11 infrastructure mode for the wireless interfaces concerned. Indeed, in our approach all interfaces are operated in IEEE 802.11 ad-hoc mode. To avoid any confusion that may arise, we refer to mesh routers that act as access points for clients as “access routers”.

Figure 1 shows a simple example of an infrastructure mesh network. In this example infrastructure mesh network there are four mesh routers, four client devices and a server. All of the mesh routers in the example mesh network can act as access routers for clients and have a dedicated wireless interface for this purpose.

One of the challenges in an infrastructure mesh network is to support seamless mobility for the client devices. These client devices are likely to roam between the transmission ranges of various mesh routers, creating a need for the corresponding communication sessions to be handed off between mesh routers.

In this paper we present a light-weight and simple handoff mechanism, which we refer to as the Light-weight Client Mobility scheme for Infrastructure Mesh networks (LCMIM).

A key advantage of our approach is that no changes to the client devices are required: no special software needs to be installed and no special configuration of the client is needed.

Our approach works with any standard off-the-shelf IEEE 802.11-capable client device.

In our mobility scheme client handoffs are initiated via gratuitous Address Resolution Protocol (ARP) [2] messages, which are broadcast frequently by mesh routers. These gratuitous ARP messages provide an IP to MAC address mapping without the usual preceding ARP request. In our mechanism, these ARP messages can be viewed as a solicitation by the sending mesh router for traffic from any client that receives the gratuitous ARP message. For brevity, we refer to these gratuitous ARP broadcasts as gARPs.

To integrate LCMIM into an existing routing protocol requires only minimal modifications to the protocol. To demonstrate LCMIM, we have integrated it with the Ad-Hoc on Demand Distance Vector (AODV) [3] routing protocol. However, the handoff mechanism can be applied to any reactive ad-hoc or mesh routing protocol. In our modified version of AODV the only state information that needs to be maintained at each mesh router is a list of clients for which that mesh router acts as an access router. In contrast to related work, the cost of updating and maintaining this information during handoffs in our approach is minimal, as we will discuss in more detail.

The key elements of our solution are as follows:

- 1) All client devices are configured to use a well-known IP address as their default gateway. This IP address, referred to as the *virtual gateway address* is used by all mesh routers on the interface they use to communicate with clients.
- 2) Client handoff, and the corresponding updating of ARP entries, is achieved by having all mesh routers periodically broadcast gARPs. Receipt of a gARP message by a client causes that client to map the virtual gateway address IP to the MAC address of the sending mesh router. This mapping occurs in the client's ARP cache.
- 3) Mesh routers perform routing duties, such as route discovery and responding to route requests, on behalf of clients.
- 4) All interfaces on mesh routers are configured to use 802.11 ad hoc mode. This means that clients avoid the overhead of seeking a new 802.11 access point and associating with it.

The remainder of this paper is structured as follows. In Section II we discuss related work in the field. This is followed in Section III with a description of our LCMIM protocol designed to provide light-weight support for client mobility in infrastructure mesh networks. In Section IV we discuss the evaluation of our approach and present experimental results. The paper is concluded in Section V.

II. RELATED WORK

In this section, we examine existing and proposed approaches to client handoff for infrastructure mesh networks.

A. MeshDV

The MeshDV protocol, developed by Iannone and Fdida [4], is designed to provide fast handoffs of clients between

mesh routers in an infrastructure mesh network. The approach assumes that mesh routers allocate one wireless interface for communicating with clients and use a second wireless interface for backhaul communication with other mesh routers. The interface used for communicating with clients is operated in 802.11 infrastructure mode. The second interface used to communicate with other mesh routers is operated in ad-hoc mode.

Whenever a new client associates with a mesh router, the mesh router informs all other mesh routers that all traffic for the client should be tunneled through it. Similarly, whenever a client dissociates with a mesh router, the mesh router is required to inform the other routers in the infrastructure mesh network. This updating represents significant additional overhead in addition to the routing protocol.

Another weakness of this approach is that it requires mesh routers to use 802.11 infrastructure mode to detect when clients associate and disassociate. As discussed by Mishra et al. [5], the time taken by stations to scan for a new 802.11 infrastructure mode access point and re-associate with it can be a significant source of delay during handoffs.

B. LIHP

LIHP (Link-layer Initiated Handoff Protocol), developed by Lin and Rangarajan [6], is a light-weight handoff protocol for mobile devices. To apply LIHP to infrastructure mesh networks, mesh routers need to allocate one of their wireless interfaces to communication with clients. Mesh routers configure this *client interface* in 802.11 infrastructure mode and set it to the same well-known IP address. Similar to our approach, clients are configured to use this well-known virtual gateway IP address as their default gateway address.

When a LIHP mesh router detects that a new client has associated with its client interface, it send a gARP to the client to force it to map its virtual gateway IP address to the mesh router's client interface. The mesh router also sends a notification to the Access Gateway informing it that the client can now be reached via it. This Access Gateway provides connectivity to the Internet and is also responsible for tunneling inbound traffic to clients via the associated mesh router. Outbound traffic from a client is tunneled by that client's mesh router through the mesh to the Access Gateway.

As with MeshDV, the use of 802.11 infrastructure mode introduces additional overhead in terms of access point discovery and association during handoffs between mesh routers. Further, as the Access Gateway is the only entity in the network that knows which mesh routers are associated with which mobile clients all traffic must be tunneled via it, i.e. the Access Gateway acts as a potential bottleneck and single point of failure for the network.

C. SMesh

The SMesh protocol by Amir et al. [7] has specifically been designed to support seamless mobility in infrastructure mesh networks.

Mesh routers running the SMesh protocol use one of their interfaces exclusively for communicating with clients. As with LHIP, all mesh routers configure their client interface with the same well-known IP address. Clients then use this well-known IP address as their default gateway.

In SMesh all mesh routers run a DHCP server which supplies configuration information to clients. By setting each client's IP lease time to 90 seconds, clients are forced to regularly broadcast DHCP requests to mesh routers to renew their IP address. These broadcasts are used by mesh routers to (1) detect the presence of clients and (2) determine which mesh routers have the best link to a client.

Mesh routers that have a high link quality to a client (where link quality is calculated using the number of DHCP requests received from that client in a given time) add themselves to a multicast group associated with that client. Entry to multicast groups is restricted so that only mesh routers with a link quality higher than that of the group's current members are permitted entry. Mesh routers send unicast gARP messages to the client when they join the client's multicast group and also (on a regular basis) when they consider themselves to have the best link to the client.

When the client enters into a situation where it does not have a good quality link to any particular mesh router, more mesh routers will be able to join that client's multicast group. All the members of the client's multicast group forward traffic to the client. This means that the client may receive duplicate IP packets when there is more than one mesh router in its multicast group. However, once a mesh router with a good quality link to the client is found, the multicast group for the client will quickly shrink to just the mesh router with the best quality link.

As a client receives duplicate IP packets from all members of its multicast group, SMesh has the ability to flood a noisy channel with a large number of duplicate packets sent by multicast group members. The use of multicast also introduces extra overhead in that the routing protocol (Spines [8]) must manage multicast groups for all client devices.

SMesh's dependence on DHCP for configuring and discovering clients has drawbacks. The cost of reconfiguring a device via DHCP is platform-dependant and can potentially be very high. As mentioned in [9], DHCP can incur a delay of up to 5 seconds, even if the client's IP address is not changed.

III. LIGHT-WEIGHT CLIENT MOBILITY IN INFRASTRUCTURE MESH NETWORKS

In this section we present the LCMIM protocol which is designed to support mobile clients in an infrastructure mesh network. There are two main components of the LCMIM protocol. The first part involves controlling the handoff between mesh routers, while the second addresses routing in the infrastructure mesh network.

A. Client Handoff

In our approach, mesh routers actively solicit traffic from clients within their transmission range by regularly broadcasting gARP messages. While gARPs have been used to perform

handoffs between mesh routers in existing client mobility protocols (e.g., SMesh [7] and LHIP [6]), our use differs in that gARPs are *broadcast* by *all* mesh routers at regular intervals.

Our approach requires that mesh routers dedicate one wireless interface to serve clients. This interface is set to the same 802.11 channel and ESSID on all mesh routers. It is from this interface that the mesh routers broadcast their gARPs.

The IP address of each mesh router's client interface is set to the virtual gateway address. Clients are configured with a subnet mask of 255.255.255.255 and use the virtual gateway address as their default gateway. This forces all outbound traffic from the client to be routed via the virtual gateway address. The problem of client configuration (in terms of IP address) is beyond the scope of this paper. However, we assume that once chosen, a client's IP address remains static.

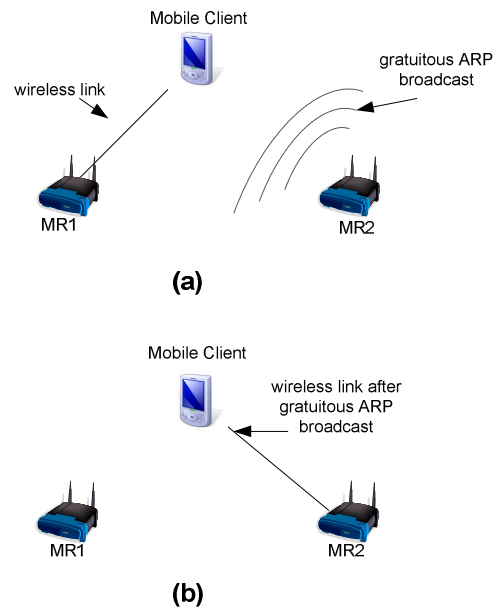


Fig. 2. Example Handoff Scenario

Figure 2 shows an example handoff conducted using our LCMIM protocol. In this example, a client (MC1) is within transmission range of two mesh routers (referred to as MR1 and MR2). In the initial scenario, shown in Figure 2 (a), the client uses MR1 as its access router. This means that the virtual gateway address is mapped to the MAC address of MR1 in MC1's ARP table.

After the client MC1 receives a gARP broadcast from mesh router MR2, it automatically updates its ARP cache. This update forces MC1's virtual gateway address to be mapped to MR2's MAC address. This causes all outgoing traffic from MC1 to be routed via MR2, as shown in Figure 2 (b).

In the LCMIM protocol, the access router used by a client will correspond to the mesh router from which the client most recently received a gARP.

When clients receive gARPs from multiple mesh routers, as can be expected in networks with dense mesh router deployments, a client's virtual gateway address to MAC address mapping can oscillate between the mesh routers within the

client's transmission range. This oscillation is an obvious problem and is unavoidable in our simplistic and extremely light-weight mobility approach. However, our evaluation in Section IV shows that it does not significantly impact on the performance.

In general, the frequency of this oscillation will be dependent on the frequency of gARP broadcasts from the various mesh routers. Mesh routers that send gARPs more frequently will thus be more likely to be used as a client's access router.

The operating system used by the client may also affect the oscillation rate. In Debian Linux (kernel 2.6), used for our testbed, this oscillation effect is dampened by a *locktime* timer which limits updates of ARP cache entries to at most once per second, by default.

The locktime timer is set whenever an ARP cache entry is modified and prevents any modification of the entry until the timer expires. The timer is reset each time a frame is received from the MAC address/IP address combination in the locked ARP cache entry. The effects of this locktime timer are discussed further in Section IV.

It should be noted that updates to unlocked ARP cache entries are near instantaneous [7], resulting in clients switching access routers with no (or very low) packet loss.

B. Routing to and from Clients

The LCMIM handoff mechanism enables the outgoing traffic from clients to be rapidly handed off to new mesh routers. However, changing the access router used by a client creates two routing problems. Firstly, the new access router needs to establish a route to the destination of the traffic sent by the client. Secondly, other nodes in the network must be able to find a route to the client via the client's new access router.

We address these two routing problems through simple modifications to the Ad-Hoc on Demand Distance Vector (AODV) [3] routing protocol.

The standard AODV routing protocol provides reactive routing, meaning that routes to a particular destination are discovered on demand. Route discovery is initiated when a mesh router has no route to the destination to which it wants to send a packet. This results in the mesh router broadcasting a Route Request message (RREQ) to all neighbouring mesh routers. This RREQ message is propagated through the network using a controlled flooding approach until either the destination node, or an intermediary node that has a fresh route to the destination, is reached.

This node sends a Route Reply (RREP) message back along the reverse path via which the corresponding RREQ was received.

In our modified AODV implementation, a mesh router maintains a list of clients (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, it adds this client to its client list. In case the mesh router does not have a route to the destination of the packets received from the client, it will find a route on behalf of the client by initiating an AODV route discovery.

If a mesh router does not receive any data from a client 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, indicating that it has a route to the destination node.

The problem with this approach is that a client 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 clients.

We solve this problem with another 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 happens to be within range, it will respond, upon which the mesh router adds the client to its client list and responds to the RREQ with a corresponding RREP message.

Should a client 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 to arrive at the originator of the route request is chosen and the corresponding route is used.

IV. EVALUATION

We evaluated our LCMIM protocol using a small wireless mesh network testbed of five nodes consisting of a client, three mesh routers (MR1, MR2 and MR3) and a server.

In our testbed configuration (see Figure 3), the three mesh routers had two wireless interfaces each, one to communicate with clients, and one for communication with the other mesh routers and the server. All the wireless interfaces used were IEEE 802.11g. The interface allocated for communicating with clients on each of the mesh routers was set to the virtual gateway address and a common channel and ESSID.

The client was configured with a subnet mask of 255.255.255.255 and used the virtual gateway address as its default gateway.

gARP broadcasts were generated by the mesh routers using the open source utility *garp* [10]. All of the mesh routers ran an open source version of the AODV routing protocol (AODV-UU [11]) with our modifications to provide support for client mobility, as discussed in Section III.

Two different scenarios were run on the testbed to evaluate the performance of our LHICM handoff approach.

A. Scenario 1

In the first scenario we attempted to determine (1) the maximum rate at which clients will oscillate between access routers, and (2) the effects of this oscillation on throughput and latency for a communication session between a static client and a server.

Figure 3 shows the network topology used for the first scenario. Each of the mesh routers (MR1, MR2 and MR3)

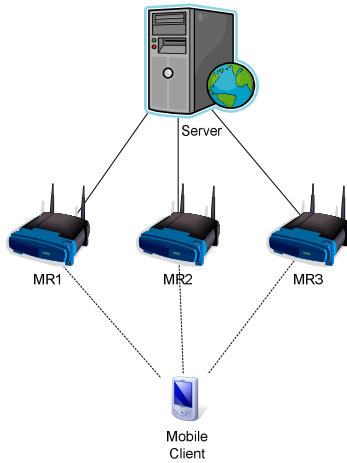


Fig. 3. Network topology used for Scenario 1

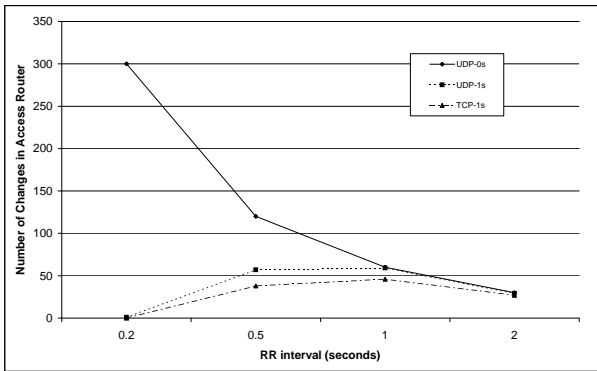


Fig. 4. Number of Access Routers used by the Client

broadcast gARPs at regular intervals from their client interface. These broadcasts were coordinated using a Round Robin (RR) scheme such that MR1 sent a gARP, followed by MR2, followed by MR3, again followed by MR1, and so forth.

A unicast UDP flow was first established between the client and the server via one of the mesh routers. Four separate tests were conducted for UDP using RR intervals of 0.2s, 0.5s, 1s and 2s. Note that by RR interval we mean the time between gARP broadcasts of consecutive mesh routers in the RR scheme.

Each of these tests lasted 60 seconds. For the UDP tests, datagrams were 512 bytes in length and were transmitted at a constant rate of 1Mbps.

The performance of LCMIM was then evaluated with TCP. To achieve this the four previously described UDP tests were repeated using a TCP session instead of UDP.

The *iperf* utility was used to generate the UDP and TCP traffic. UDP traffic was limited to 1 Mbps, whereas TCP traffic was sent at the maximum possible rate that the data path could sustain.

Figure 4 plots the number of access routers used by a client during the TCP and UDP tests versus the Round Robin interval (0.2s, 0.5s, 1s or 2s) used in the test.

Three data series are shown in the Figure: one for UDP

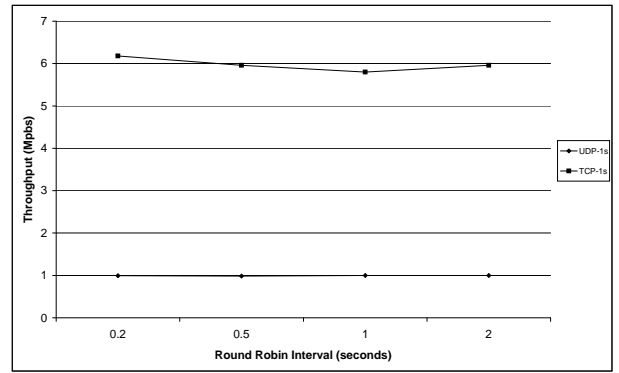


Fig. 5. UDP and TCP session Throughput.

with a locktime of 1 second (UDP-1s), one for TCP with a locktime of 1 second (TCP-1s), and another for UDP with a locktime of 0 seconds (UDP-0s).

The data series UDP-0s shows the number of access routers used by a client during a UDP session when the Linux ARP locktime timer was set to zero seconds. This meant that the client changed the mesh router used as its access router every time it received a gARP in the RR scheme. We use this plot as a reference for the number of gARP messages sent for the four different RR intervals.

As can be seen in Figure 4, for RR intervals of 2 seconds and 1 second the number of access routers used by the client in UDP-1s and TCP-1s closely matched the number of gARPs sent (indicated by UDP-0s).

For RR intervals of 0.5 seconds, the clients conducting TCP (or UDP) sessions experienced roughly half as many changes in access router as there were gARPs sent.

When the RR interval was reduced further to 0.2 seconds, the TCP-1s and UDP-1s plots indicate that the client only used a single access router for the duration of the 60 second test.

A comparison of UDP-0s, representing a UDP session in which the client's locktime timer was set to zero seconds, and UDP-1s (where the client's locktimer is set to 1 second) indicates that the large difference in access router changes for low RR intervals (i.e. 0.2s and 0.5s) was due to the locktime timer.

As previously, discussed, this timer locked entries in the ARP cache for 1 second after they were updated. This meant that the virtual gateway address entry in the ARP cache could not be modified faster than 1 second. Receipt of a gARP from the same IP/MAC address as a locked ARP cache entry reset the timer for that entry. In our scenario this meant that if the client started using MR2 as its access router, and MR2 continuously sent gARP messages at intervals of less than 1 second, the client would never change its access router. This can be observed for RR intervals of 0.2s (where each mesh router sends a gARP every 0.6 seconds).

Figure 5 plots the average throughput over a 60 second period for the UDP and TCP tests described above. As can be seen in the Figure, the throughput recorded for a TCP session between the client and the server was 6Mbps, regardless

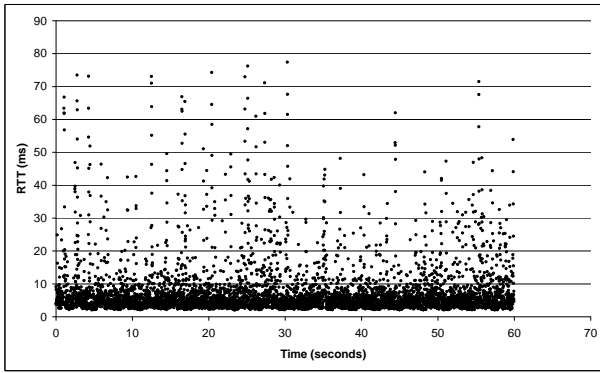


Fig. 6. Round Trip Time (RTT) of UDP datagrams

of the gARP rate and subsequently the number of handoffs performed. Similarly, the throughput recorded by the server for a UDP session between the client and the server remained constant at 1Mbps regardless of the rate at which the mesh routers broadcast gARP messages and the subsequent number of times the client changed its access router (i.e. handoffs were performed). This suggests that our LCMIM protocol is able to sustain high bandwidth connections irrespective of the number of times that client changes its access router.

As with UDP in the first scenario, the handoff time was the amount of time to update the ARP cache entry once a gARP was received. This was near instantaneous, provided the ARP cache entry was not locked.

It should be noted that during the UDP tests, no datagrams were lost due to handoffs. Similarly TCP did not experience any loss of segments as a result of the handoff procedure.

Figure 6 shows the Round Trip Time (RTT) for UDP traffic sent between the client and the server in Scenario 1. The data was obtained from a 60 second test in which the RR interval was set to 1 second. During the test, the client changed its access router 59 times (approximately once per second). As can be seen in the Figure, the RTT for most UDP packets is less than 10ms. The peaks in the RTT observable in the Figure do not correspond with handoffs, but rather are due to natural variation in the wireless channel used. This suggests that our LCMIM protocol offers low-latency handoffs for mobile client in infrastructure mesh networks.

B. Scenario 2

The second scenario in our evaluation examined the performance of a TCP session conducted between the client and the server via one of the mesh routers. The client was assumed to be mobile and moved in and out of range of mesh routers.

To conduct this part of the evaluation, the three mesh routers (MR1 MR2 and MR3) were arranged in a straight line such that the broadcast ranges of MR1 and MR2 overlapped, as did the broadcast ranges of MR2 and MR3. This created five regions consisting of: three regions in which the client received gratuitous ARPs from only one mesh router, and two regions where the client was able to receive gratuitous ARPs from two mesh routers (either MR1 and MR2 or MR2 and MR3).

As our testbed setup was static, mobility was simulated using MAC filtering on the client using the Linux arptables utility.

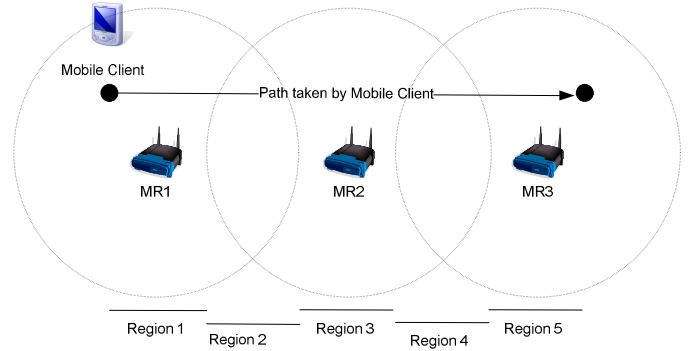


Fig. 7. Configuration for Scenario 2

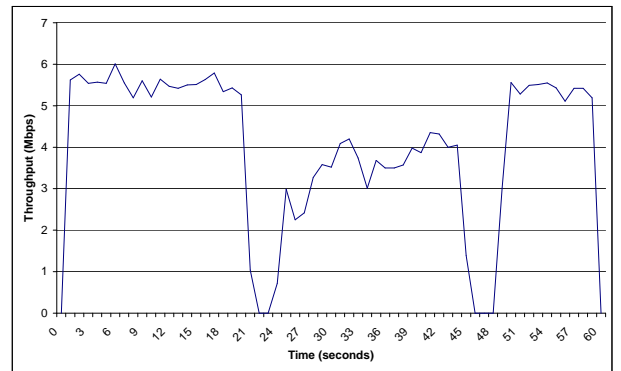


Fig. 8. Scenario 2 TCP Throughput

Figure 7 shows this configuration and also indicates the path traveled by the client. During the scenario there were two hard handoffs that had to occur. The first when the client moved from region 2 to region 3 (it had to change its access router from MR1 to MR2) and the second when the client moved from region 4 to region 5 (the client had to change its access router from MR2 to MR3). These handoffs were “hard” in the sense that upon moving into the new region, the client lost contact with its previous access router. Note that the existence of hard handoffs within this scenario marks a difference with the tests conducted in Scenario 1 where the client sent and received traffic from its access router, but was also able to receive traffic from the other two mesh routers (i.e. those not currently used as the client’s access router).

The test conducted for the scenario used a locktime value of 0 seconds (in order to remove any affects associated with the locktime timer), and a gratuitous ARP broadcast rate of 0.2 seconds for all three mesh routers. Unlike in the previous scenario, no Round Robin scheme was used. A 60 second TCP session was established between the client and the server for the test. This TCP traffic was generated using the Linux iperf utility.

The observed throughput for the test is shown in Figure 8. As can be seen in the Figure, two dips occurred in the TCP

throughput at approximately 24 seconds and 48 seconds into the test. These correspond to the two hard handoffs that must occur when the client moves into region 3 and into region 5. In both cases TCP takes approximately 1.7 seconds to recover.

This time is due to an interaction of our mesh routing protocol with the TCP protocol. To explain this interaction we use the handoff that occurs upon the client's transition into region 3 (i.e. at time 24 seconds) as an example.

When the client is in region 1, it uses MR1 as its access router; the route to the server passes via MR1, and the return route from the server passes through MR1. When the client then moves into region 2, it is able to hear gARP messages from both MR1 and MR2. As the locktime value is set to zero for the test, the client switches its access router to the mesh router (either MR1 or MR2) that most recently sent a gARP. The first time MR2 is used as the access router, a new route must be discovered from MR2 to the server. At this stage both MR1 and MR2 have routes to the server for the client. As the server already has a route to the client (via MR1) it continues using this route to send to the client (irrespective of whether MR1 or MR2 are used as the client's access router). When the client then transitions into region 3, a hard handoff occurs and the client is no longer able to send or receive traffic from MR1. As such, the client uses MR2 as its access router for the duration of its time in region 3.

Following this hard handoff, the server is not aware that the client can no longer be contacted via MR1 and so continues to send to the client using its route that passes through MR1. In our current implementation, if MR1 does not receive traffic from the client within a 900ms period, it decides that the client is out of range and removes the client from its client list. After this has occurred any traffic sent to MR1 for the client will be dropped and an error message sent to the source of the traffic (i.e. the server). When the server receives this error message, it then initiates route discovery to discover a new route to the client (via MR2).

As there is a 900ms lag between the client entering region 3 and MR1 deciding it can no longer contact the client, 900ms worth of TCP acknowledgements sent by the server to the client will be lost (as it is sent via MR1). Loss of acknowledgements causes TCP to drastically throttle the number of segments sent, leading to the drop in throughput observed in Figure 8 at time 24 seconds. The client then starts retransmitting the unacknowledged segments at ever increasing intervals. In our testing, the client resent segments at 0.25s, 0.75s and 1.75 seconds after the segment was initially sent. The first two resends are lost (as they fall within the 900ms time frame). The retransmission at 1.75 seconds after the segment was first sent lies outside this 900ms time frame and is able to reach the client (since the server has a newly discovered route to the client via MR2). TCP on the client then recovers and the throughput rises again.

It should be noted that in Figure 8, the stepped nature of the throughput graph is due to differences in the capacity of the routes (i.e. via MR1, MR2 or MR3) and is not a reflection on our handoff protocol.

V. CONCLUSION

In this paper we presented LCMIM, a light-weight approach to support handoff of clients between the mesh routers of an infrastructure mesh network. Client handoffs were controlled using gratuitous ARP messages broadcast by mesh routers at regular intervals. Any client that received one of these gratuitous ARP messages from a mesh router redirected its outbound traffic towards that mesh router.

An evaluation of our approach found that access router oscillations (where the client frequently changes its access router as might occur when a client is surrounded by many mesh routers) have little effect on TCP and UDP traffic in terms of throughput, latency and packet loss. In situations where the client is mobile, LCMIM was able to support both UDP and TCP sessions. Testing revealed that the performance of LCMIM, particularly with relation to TCP sessions initiated by clients, was greatly affected by three factors: (1) the Linux ARP cache locktime timer value, (2) the TCP backoff behaviour for retransmitting lost segments or acknowledgements and (3) the time that the infrastructure mesh routing protocol waits before it deletes inactive routes to clients.

ACKNOWLEDGEMENTS

NICTA is funded by the Australian Governments Department of Communications, Information Technology, and the Arts; the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs; 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] D. Plummer, "An Ethernet Address Resolution Protocol," RFC 826, 1982.
- [3] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, 2003.
- [4] L. Iannone, K. Kabassanov, and S. Fdida, "The real gain of cross-layer routing in wireless mesh networks," in *Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN'06)*.
- [5] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM Computer Communications Review (CCR)*, vol. 33, no. 2, pp. 93-102, April 2003.
- [6] J. Lin and S. Rangarajan, "LIHP: A Low Latency Layer-3 Handoff Scheme for 802.11 Wireless Networks," in *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006)*, 2006, pp. 401-409.
- [7] Y. Amir, C. Danilov, M. Hilsdale, R. Musaloiu-Elefteri, and N. Rivera, "Fast Handoff for Seamless Wireless Mesh Networks," in *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys 2006)*.
- [8] Y. Amir and C. Danilov, "Reliable Communication in Overlay Networks," in *Proceedings of the IEEE International Conference on Dependable Systems and Networks DSN 2003*, 2003, pp. 511-520.
- [9] 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.
- [10] A. Cassen, "garp utility." [Online]. Available: <http://www.linuxvirtualserver.org/~acassen/>
- [11] E. Nordstrom, "University of Upsala open source implementation of AODV." [Online]. Available: <http://www.docs.uu.se/docs/research/projects/scanet/aodv/aodvuu.shtml>