ABSTRACT

Wireless Mesh Networks (WMN) have been emerging in the last couple of years as a cost effective alternative to traditional wired access networks. IEEE 802.11s is the IEEE standard for Wireless Mesh Networks and is present in the Linux kernel in the form of open80211s (o11s). Currently, in o11s, every mesh node is associated with a maximum of one mesh gate. This type of single association directs the entire traffic from a node towards the mesh gate it is associated with. The gate's available bandwidth is shared by all its clients (nodes), and consequently, the throughput per client is reduced with an increase in the number of clients. By enabling multipoint-to-multipoint associations, this paper aims to improve the utilization of total available bandwidth by using multiple Internet access gateways inside a Wireless Mesh Network. This is achieved in a transparent manner without requiring any changes to applications. The overall load on the network is balanced across multiple gateways by using a traffic sensitive approach for gateway selection. Conventionally, in wireless networks, associating with multiple gateways requires multiple wireless NICs. Our software only approach moves away from this convention and uses a single wireless NIC to associate with multiple internet gateways, thereby reducing hardware costs.

Keywords: Wireless Mesh Network, Multi-gateway, Single radio

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Wireless Mesh Networks (WMNs) have seen tremendous growth in the recent years by reducing the dependency on wired access networks and thereby have become a cost effective solution for large scale deployments. The emergence of community meshes, city wide meshes and rural deployments have demonstrated the benefits of Wireless Mesh Networks over existing networks. In addition to being self-healing and self-organizing like conventional ad-hoc networks, WMNs have superseded these networks by providing the capability of range extension. A typical WMN combines a fixed network (backbone) and a mobile network (backhaul). In a WMN, there are mesh stations (STAs), also called mesh nodes, which are devices that contain an IEEE 802.11-conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM). The mesh stations implement the mesh facility which is a set of enhanced functions, channel access rules, frame formats, mutual authentication methods, and managed objects used to provide data transfer among autonomously operating stations (STAs) that may not be in direct communication with each other over a single instance of the wireless medium. The mesh stations establish peer-to-peer wireless links forming a mesh cloud called the Mesh Basic Service Set (MBSS) and transfer messages mutually. These mesh stations act as routers, forwarding traffic to or from other mesh stations. Some of these mesh stations, called mesh gates, provide access to one or more distribution systems, via the wireless medium for the mesh basic service set (MBSS).

Open80211s is an open-source implementation of the recently ratified IEEE 802.11s wireless mesh standard. Open80211s was included in the linux kernel (release 2.6.26) to bring wireless mesh networking to commodity hardware supported by the kernel. It is the first open implementation of 802.11s to consolidate the numerous non-interoperable mesh protocols into one that is based on a standard specification (IEEE 802.11s). 802.11s based networks use the Hybrid Wireless Mesh Protocol (HWMP) for Layer 2 routing based on MAC addresses. HWMP provides both proactive path selection and reactive path selection. A mesh station that needs to transmit a frame to an unknown destination can dynamically discover the best path to this destination. Mesh stations can also proactively discover the MBSS and determine best paths to any point of the mesh cloud before needing to send any data frame.

1.2 Need of Project

In order to connect to the outside network, the mesh stations in the WMNs form a layer 3 network by associating with a gateway present in the network. Traditionally, each mesh node can only be associated with one of the available gateways. A mesh node associates with a gateway based on its proximity to the gateway. This type of single association directs the entire traffic from a node towards the mesh gate it is associated with. The gate's available bandwidth is shared by all its clients (mesh nodes), and consequently, the throughput per client is reduced with an increase in the number of clients. Additionally, the mesh gate being common to many outbound paths, experiences a reduction in it's effective link capacity due to the multipoint-to-point nature of the associations. Also, there could be additional gateways far from a mesh node which could be used effectively. WMNs are self-healing and self-configuring, but the Layer 3 associations in which mesh nodes are bound to only one gateway are not dynamic. This means that on failure of a mesh gate, the traffic flowing to the gateway through that mesh gate is dropped instead of being re-routed to another gateway which can be reached using the underlying Layer 2 network.

In order to overcome these shortcomings, this project suggests an approach to effectively use the capabilities of 802.11s based wireless mesh networks to efficiently utilize the total available bandwidth by using multiple Internet access gateways present in the network.

1.3 Project Idea

This projects aims to use multiple mesh gateways present in the network to cater to the needs of the mesh clients. By enabling multipoint-to-multipoint associations, the overall load on the network is balanced across multiple gateways by using a traffic sensitive approach for gateway selection. The method routes the traffic on flow basis by mapping every flow to a mesh gate. This maintains the integrity of the flow and avoids the overhead involved in reassembling the packets.

1.4 Applications

WMNs provide a low cost solution for Internet connectivity. By associating the meshclients with multiple mesh gates, the speed of receiving a response will increase. Thiswill be reflected in applications which open multiple network sockets to grab one or more file(s) such as browsing the WWW using a web browser. Using the same for the Bittorrent protocol will increase the speed of acquiring files from the peers as a result of using multiple gates, leading to better utilization of available bandwidth.

Chapter 2

Literature Survey

Reference [1] presents a survey of wireless mesh networks. It discusses challenges that need to be overcome in order to leverage the benefits of this technology. It states that topology control algorithms are required to ensure reliable mesh connectivity and network self-organization. Most applications of WMNs are broadband services with heterogeneous QoS requirements and therefore, in addition to end-to-end transmission delay and fairness, performance metrics such as delay jitter, throughput per node and packet loss ratios must be considered by the communication protocols in use.

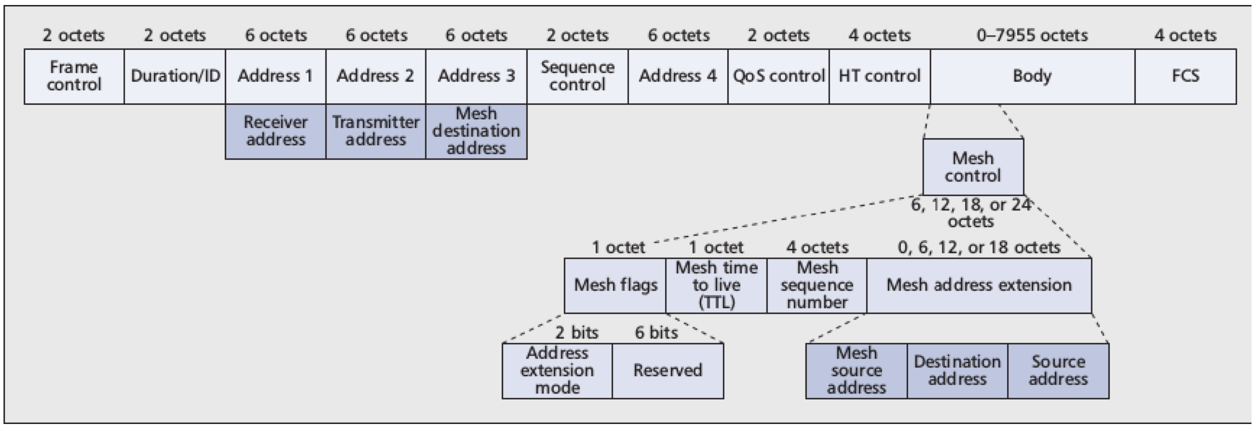


Fig 2.1: 802.11s Frame format

[9] The mesh frame format is slightly different from the standard 802.11 frame format, because a station needs to indicate more parameters. The main difficulty is that the original source of the frame and the final destination of the frame may be non-mesh stations. In order to be able to mention all addresses, the mesh frame contains 6 addresses: up to 4 addresses in the standard 802.11 header, and 1 or 2 addresses in a specific Mesh control field positioned after the HT control field. Non-mesh stations see that Mesh control field as part of the frame body:

- Address 1 is the Receiver Address (RA)
- Address 2 is the Transmitter Address (TA)
- Address 3 is the *mesh* destination address, the target mesh station (which may or may not be the frame's ultimate destination)
- Address 4 is the *mesh* source address, the mesh originator (which may or may not be the frame's ultimate source). When address 4 is not used (e.g. management frames) the mesh source address can also be inside the Mesh Address Extension field.

- Inside the Mesh Address Extension subfield of the Mesh Control field, address 5 is the final destination and address 6 is the original source. One or both of these addresses may be identical to one or 2 of the first 4 addresses if the original source, the final destination, or both are mesh stations. When address 4 is not used, the Mesh Address Extension field can also host the mesh source address

[10] HWMP (Hybrid Wireless Mesh Protocol) is the default routing protocol for WLAN mesh networking. Every IEEE 802.11s compliant device will be capable of using this routing protocol. The foundation of HWMP is an adaptation of reactive routing protocol AODV to layer 2 and radio-aware metrics called radio metric AODV. A mesh node, usually a mesh portal, can be configured to periodically broadcast announcements, which sets up a tree that allows proactive routing towards mesh portal. The reactive part of HWMP follows general concept of AODV. It uses the distance vector method and well-known route discovery process with route request and route reply. Destination sequence numbers are used to recognize old routing information. HWMP uses MAC address as a layer 2 routing protocol instead of IP addresses. Furthermore, HWMP can make use of more sophisticated routing metrics than hop count such as radio-aware metrics. A new path metric field is included in the RREQ/RREP messages that contains the cumulative value of the link metrics of the path so far.

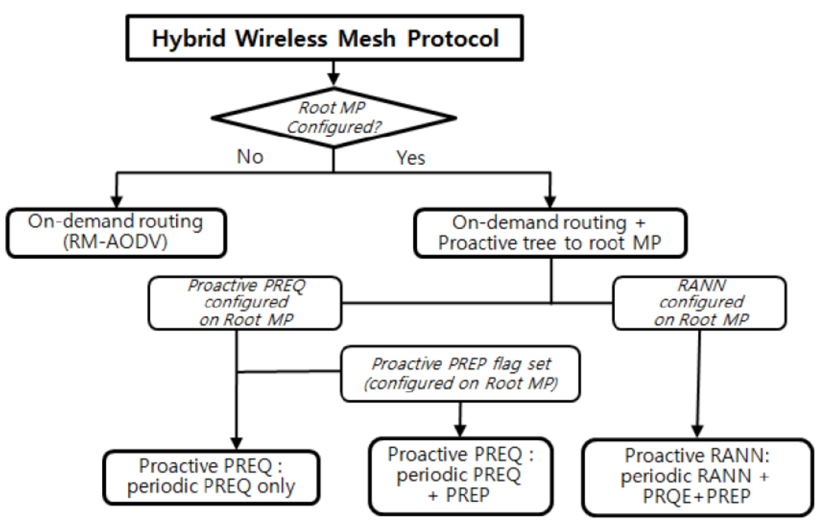


Fig 2.2: Working of HWMP

The overall throughput in a wireless mesh network can be increased by using multiple radios [2]. Each radio is associated with one gateway and channels are selected where the frequency is diversified and high-throughput paths are available. There is a significant improvement in the throughput capacity, however the selection of channels proves to be insufficient and cost intensive.

[8] To improve both capacity and reliability in such networks each node can be equipped with multiple radio interfaces using a set of orthogonal channels. This channel diversity improves the performance of a single-path, while at the same time increases the possibility to create multiple interference-free paths between the sender and the receiver.
When the number of TCP-flows increases, flow-based path allocation becomes superior to using packet-based allocation, regardless if TCP is equipped with mechanisms to mitigate the effects of reordering or not.

[3] suggests the use of RANN frames in order to get all the gateways present in the Mesh Network. RANN frames can only be captured on an interface running in monitor mode as they are layer 2 frames. This leads to the reception of all Layer 2 frames which must then be inspected to filter out RANN frames. This filtering leads to higher use of the processor. The gateway selection process selects one default gateway which routes all the traffic for that node. When the default gateway is overloaded, a Change_Gateway request is sent to one of the associated nodes. The node then selects another gateway as it's default gateway, effectively using only a single gateway.

Chapter 3

Proposed Work

3.1 Problem Statement

To enhance bandwidth utilization in Wireless Mesh Networks using a decentralized approach, by enabling multipoint-to-multipoint associations between mesh nodes and mesh gates.

3.2 Statement of Scope

- Establishing a proof of concept that the suggested approach results in gains in throughput.
- Writing a Node Manager which runs on each mesh node to facilitate mapping every flow to a gateway.
- Choosing the most efficient method to select gateways based on their capabilities.
- Designing an approach to collect the metrics of the gateways and broadcasting them to the clients.
- Making addition and removal of gates seamless
- Keeping active flows consistent with their selected gateway until the end of the flow.

3.3 Features

- Utilizes complete available network bandwidth of mesh gates.
- The routing of traffic is performed on a per-flow basis. Every socket opened on the mesh node is mapped to a mesh gate in the WMN.
- There is no additional hardware cost. A single radio with mesh support is used, as opposed to approaches mentioned in [2], [3] and [8].
- Traffic sensitive routing is performed by the mesh clients by considering the metrics broadcasted by the gates.
- Fluid nature of the mesh gates is transparent to the user. Mesh gates can join and leave the mesh seamlessly.
- Fault tolerant network as a result of using multiple gateways.

3.4 Objectives

- To increase the network throughput and fault resilience of WMNs and use all available Internet bandwidth provided to the network.
- To rectify underutilization of available bandwidth and bridge the gap between radio level metrics and higher layer metrics for effective selection of gateways.

Chapter 4

Research Methodology

4.1 Steps to Acquire and Process

- Collecting the gateway capabilities.
- Broadcasting mesh gate frames.
- Maintaining a list of the mesh gates using the received mesh gate frames.
- Designing the Node Manager which will provide gate selection logic and maintain flow consistency

4.2 Approach

Multipoint association

Conventionally in Wireless Mesh Networks, each mesh node can only be associated with one of the available gateways. This type of single association directs the entire traffic from a node towards the mesh gate it is associated with. The gate's available bandwidth is shared by all its clients (nodes), and consequently, the throughput per client is reduced with an increase in the number of clients. Additionally, the mesh gate being common to many outbound paths, experiences a reduction in it's effective link capacity due to the multipoint-to-point nature of the associations. By enabling multipoint-to-multipoint associations, the overall load on the network is balanced across multiple gateways by using a traffic sensitive approach for gateway selection.

Flow based approach

In order to use multiple gateways, every flow is mapped to a particular gateway. The flow based approach maintains the integrity and hence the end-to-end semantics in case of the TCP traffic. As a result, the overhead involved in reassembling the packets which is required in case of packet based approach is avoided.

Fault tolerant approach

A fault tolerant approach is devised to deal with unresponsive gateways. The entire flow meant for an inactive gateway is routed to an alternate active gateway. This is transparent as the user is not exposed to the failure and recovery of the network except that some delay is experienced.

QoS aware and traffic sensitive routing

The characteristics of all the gateways such as bandwidth, latency, etc. are different. The traffic originating from the clients vary in their QoS requirements. Hence, a mechanism to map the traffic to the best gateway capable of fulfilling the QoS has been incorporated.

4.3 Modules

Module 1: Mesh Gate

Mesh gates collect metrics such as bandwidth, latency. These are broadcasted after a periodic time interval.

Mesh gate bridges the wired Internet connection with the wireless Mesh interface. The gate receives all the requests forwarded to it by the mesh clients, on its mesh interface. The gate then forwards all the request destined for the external network to its wired interface. Packets meant for the gate itself are sent to the kernel for processing.

On reception of responses from the wired interface, the mesh gate forwards them to the corresponding mesh client. Responses meant for itself are sent to the kernel for processing.

Module 2: Mesh Client

Mesh clients process the mesh gate frames and maintain a list of all the gates present in the network. In the case of an inactivity of the gate for an interval of time or an antipong frame sent by the mesh gate, the corresponding entry is discarded.

The mesh client has a KernelTap device interface. This tap device captures all the request packets to be processed by the kernel to the Node Manager.

Node Manager encapsulates the packets in the MAC header with the dst MAC address set to that of the selected gate. The mesh gate is selected for every flow. This flow to gate mapping is maintained in a Flow Cache Table. Every packet received by the Node Manager is checked against the Flow Cache Table to ensure if the flow it belongs to, is already mapped to some gate in which case the packet is encapsulated in the MAC header and forwarded to the mesh interface. In case of absence of an entry in the Flow Cache Table, a gate is selected from the Gate list and an entry is made in the Flow Cache Table. The packet then encapsulated in the MAC header is forwarded to the mesh interface.

Chapter 5

Project Design

5.1 Software Tools

- iw (version 0.9.4 and above) / wpa-supplicant
- click modular router
- tc
- open80211s
- GNU/Linux OS (kernel version 2.6.26 and upwards for mesh support)
- Ubuntu 14.04 LTS

5.2 Hardware Tools

- Wireless Network cards (Qualcomm Atheros AR9285 chipset)
- Routers
- PC, 32/64 bit

5.3 System Architecture

The system architecture consists of two types of nodes:

I. Mesh Nodes
II. Mesh Gates.

The Mesh Gates have a wired interface bridged with the wireless mesh interface. They also have a metric fetch and broadcast system.

The Mesh Nodes have 3 parts:

1. The Gate Info list, a list that maintains information of each mesh gate present in the network
2. Gateway Selector, the core logic of selection of a gateway for a flow resides here.
3. Flow Cache, maintains a mapping of already mapped flows to gateways.

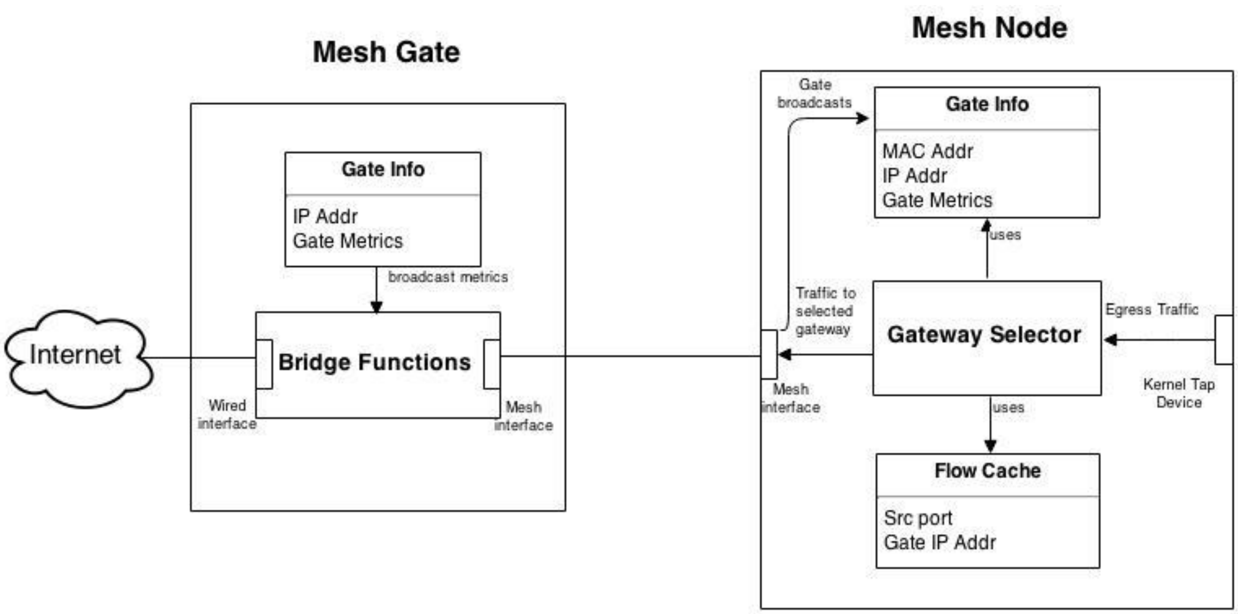


Fig 5.1: Architecture Diagram

Egress traffic from the mesh node arrives at the gateway selector, which selects the best gateway for the flow consulting the Gate Info list and flow cache. The traffic is forwarded to the selected gateway through the internal mesh network and goes out to the external network.

5.4 UML Diagrams

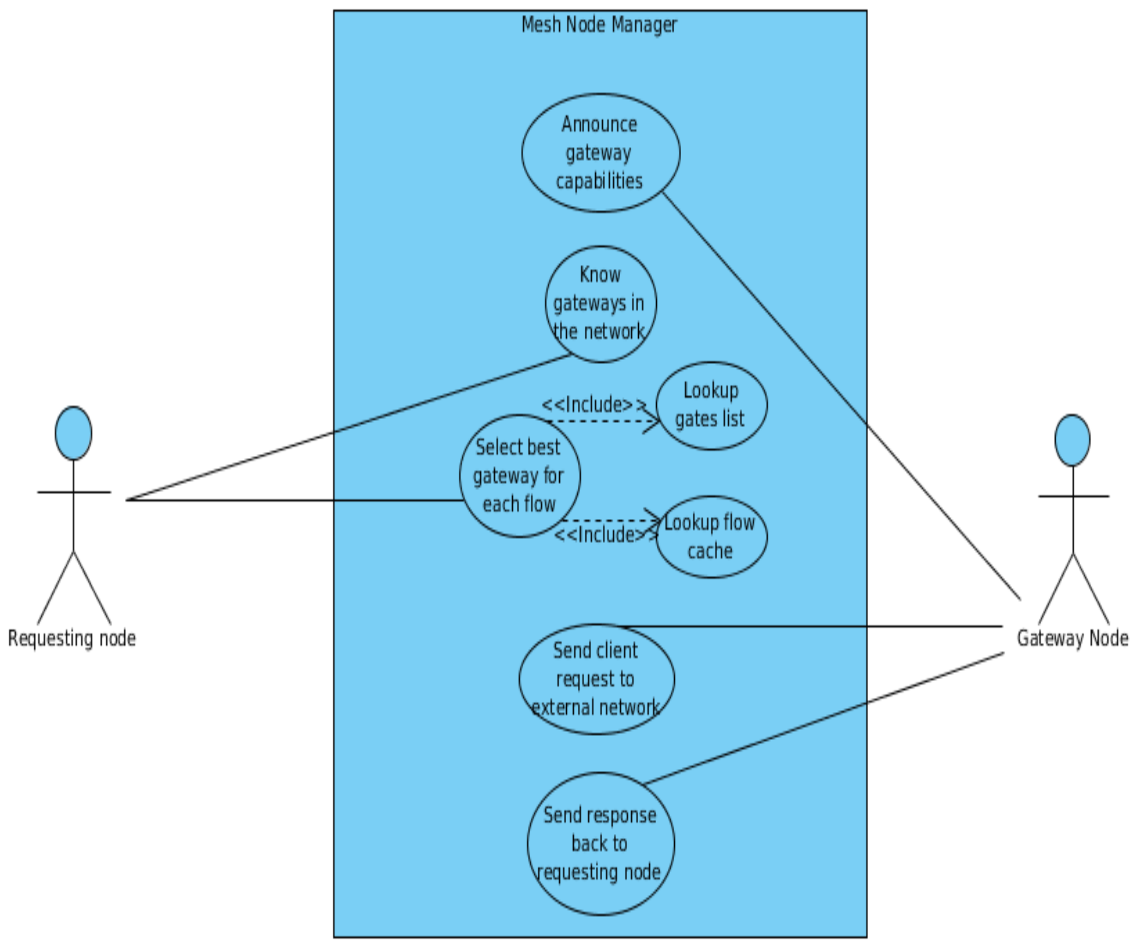


Fig 5.2: Use case Diagram

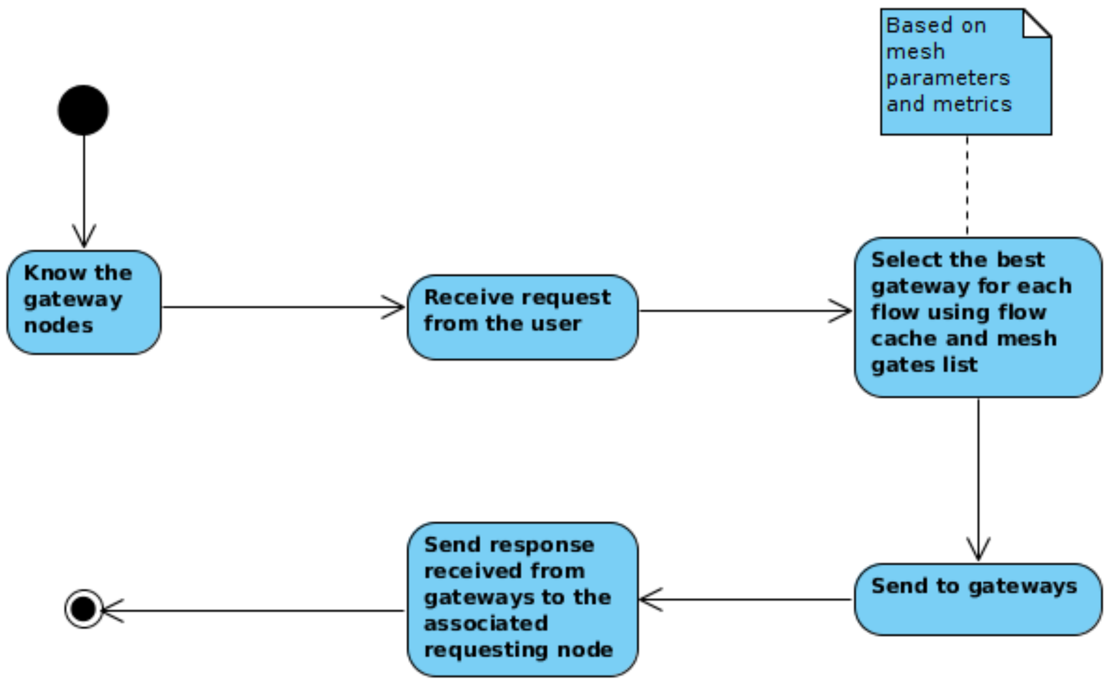


Fig 5.3: Activity Diagram

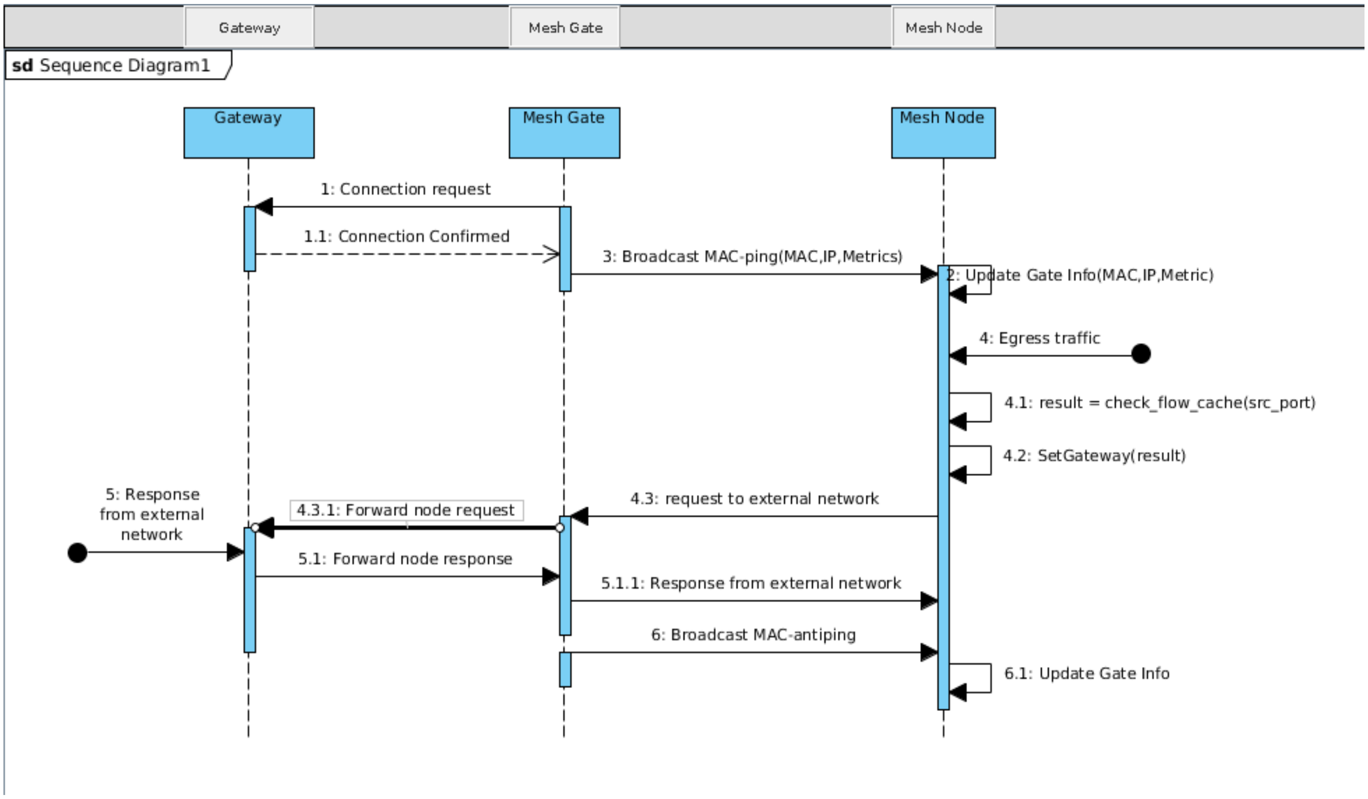


Fig 5.4: Sequence Diagram

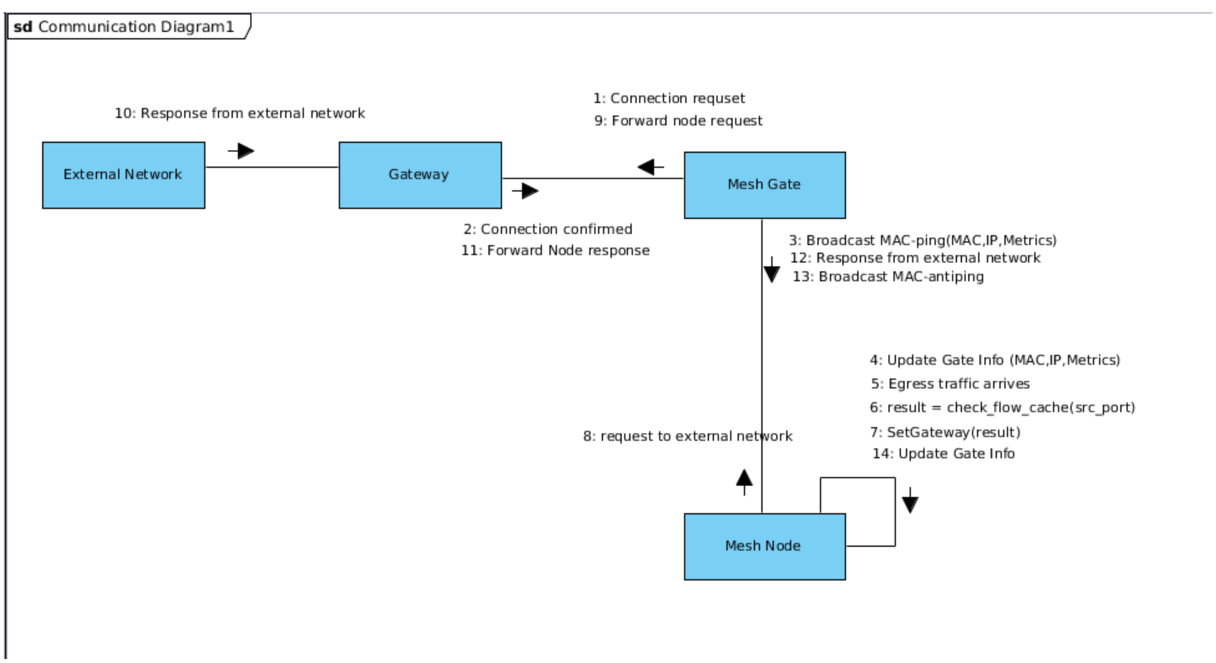


Fig 5.5: Communication Diagram

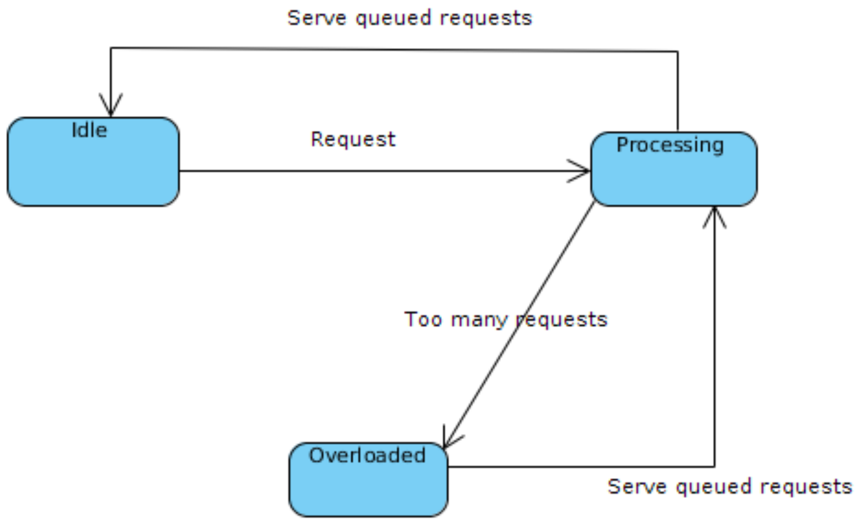


Fig 5.6: State Diagram

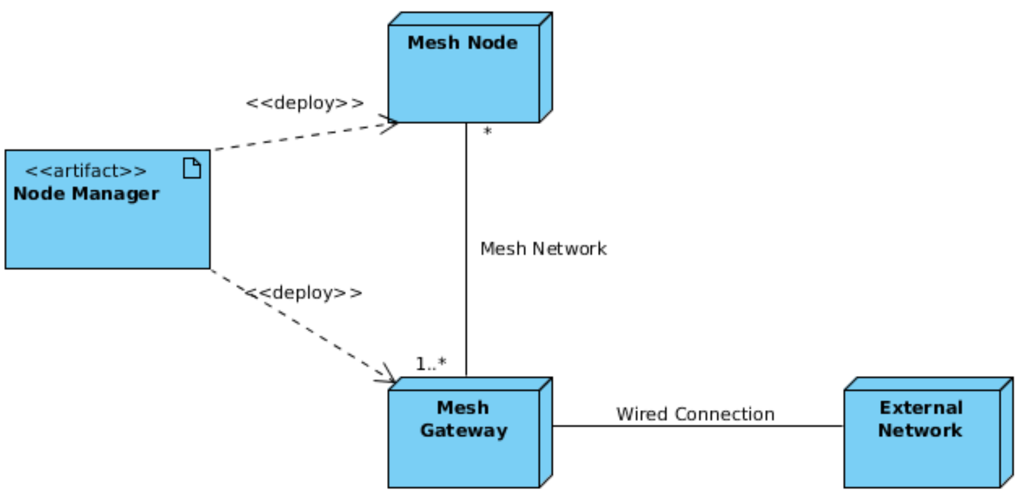


Fig 5.7: Deployment Diagram

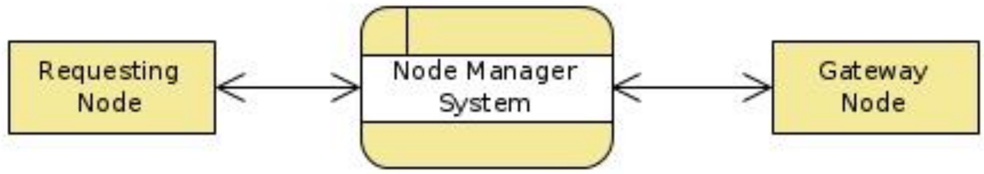


Fig 5.8: DFD Level 1

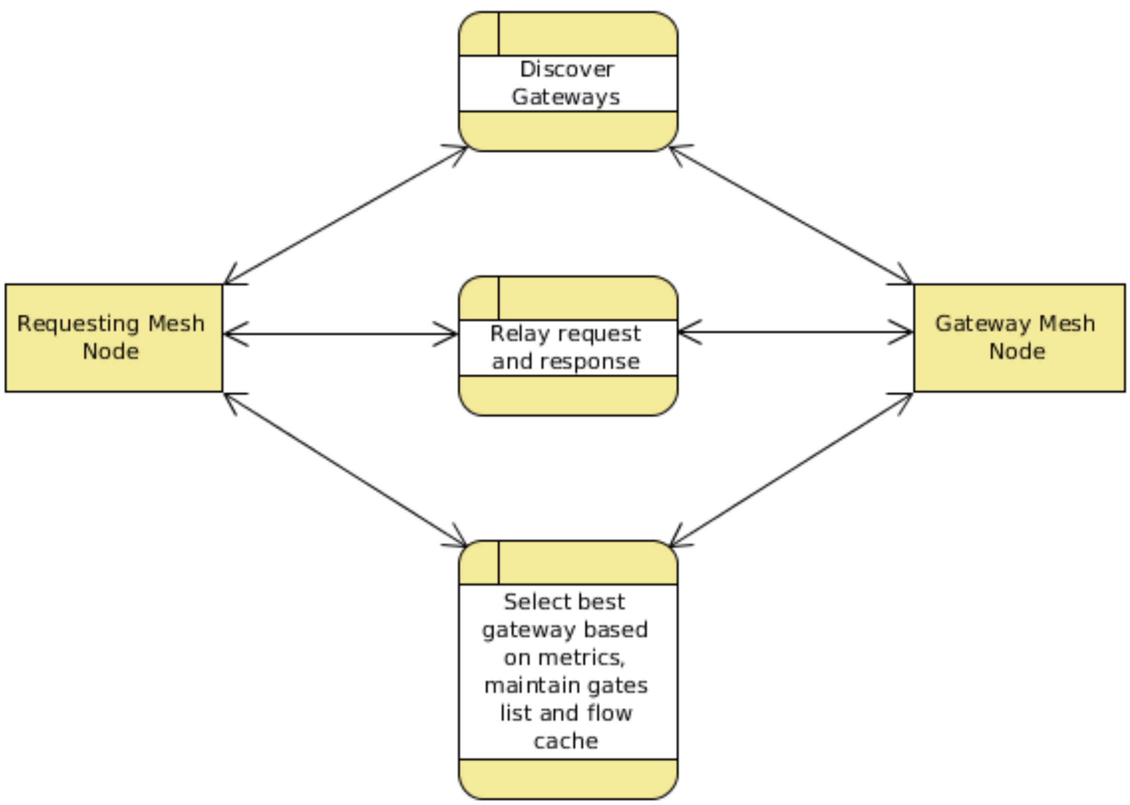


Fig 5.9: DFD Level 2

Chapter 6

Implementation

6.1 Implementation

In order to efficiently utilize all available bandwidth in a WMN, various measures such as gate discovery, multipoint to multipoint associations, dynamic addition and removal of gateways, selection of gateways and fault tolerance need to be considered.

A. *Advertisement of the gateway broadcasted in MAC Frame*

Conventionally in open80211s, the presence of mesh gates is detected by a gate flag in the broadcasted Root Announcement frames. In our solution, the mesh gate timely broadcasts it's IP address and the set of capabilities. This information is used by the mesh nodes to maintain a record for each gateway, which is further used for gateway selection.

B. *Multi-gateway association*

Every time a mesh node wants to start sending a new stream of data (socket creation) to a machine outside the local network, it does so via the best possible gate selected on the basis of the gate metrics and the type of information required to be sent. The node uses the available gateway records to decide the ideal gateway to be selected based on the kind of egress traffic. The gateway annotation on the packets pertaining to this flow is set to the chosen gateway. The flow is bound to the gateway and this mapping exists till the socket is transferring data. In case a gateway fails, all it's mappings are discarded.

C. *Flow Identification*

A unique key is required to differentiate flows from one another. Since a source port on a machine can only be used by one flow at a time, it serves as a unique identifier. The source port number is used to identify which packet belongs to which flow, thereby respecting the requirement of sending all packets of a specific flow through one gateway eliminating the overhead of packet reassembly.

D. *Persistent Mapping*

To maintain the dynamic nature of a mesh network, nodes and gates are allowed to leave and join the mesh network. This highlights the self-healing and self-configuring property of mesh networks.

While the flow of packets are being routed through the specific gateway, more gateways may come up which might provide a better path for the completion of the same request. In such a case, it is important to honour the previous mapping of the source port and the associated gateway as changing gateways will confuse the machine on the other end, which may regard it as a malicious flow of packets. This will result in packet drops at the other end of the socket and therefore needs to be avoided. Hence, a mapping table of which port is bound to which gateway on the local network is maintained. In case the port remains unused for a specific timeout interval, the mapping is then cleared.

In case a gateway goes down, no traffic destined for the outside network should be routed via it. The disappearance of a gate has been accounted for by the missing Layer 2 broadcasts received by the nodes. The nodes wait for a specified time out interval, after which the entry for a non-broadcasting gate is removed. In case the gate comes back up, it will start broadcasting frames again and therefore can be used again for servicing requests.

E. *Interoperability*

The solution has been created with compatibility in mind with the existing o11s. This allows a mix of nodes, both vanilla o11s and the ones based on this solution, to coexist and work without hitches.

6.2 Code Snippets

Packet Processor

```
/*
 * case 0: When a packet destined for the outside network is
 *           received, it is sent out via a gateway. Flow based
 *        approach is used for the same.
 *
 * case 1: Pong beacons are gate advertisement beacons which contain
 *        gateway metrics. The client maintains a list of all the
 *        gateways present in the mesh network          along with their
 *        capabilities.
 *
 * case 2: Antipong beacons are broadcasted by the gateways when they
 *        have no Internet connections. The client discards entry for
 *        this particular gateway.
 */

voidGatewaySelector::push(int port, Packet *p)
{

    Packet *q;

    switch(port)
    {
        case 0:   q = select_gate(p);
                  if(q == NULL)                              // No gate present in the network
                          output(1).push(p);
                  else
                          output(0).push(q);       // Gateway assigned to packet
                 break;
```

```
case 1: process_pong(p);
        p -> kill();
        break;

case 2: process_antipong(p);
        p -> kill();
        break;
   }
}
```

Gateway Selector

```
/*
* port_cache_table - Stores a tuple of src_port and gateway assigned
 *
* Every flow is mapped to one gateway. select_gate procedure
* will consult the port_cache_table to check if the received packet
* is part of a mapped flow.
 *
* If it is mapped, the packet is assigned the mapped gateway
* else a new gate is assigned and cache (port_cache_table) is updated.
 *
* If no gateway is present an error is given to the client
 */

Packet * GatewaySelector::select_gate(Packet *p)
{
    intport_index;
    if(p->has_transport_header())
    {
```

```
uint8_t *ptr = (uint8_t *)p->transport_header();
uint16_t src_port = 0;



src_port += *ptr;          //Extract src port
ptr++;
src_port = src_port<< 8;
src_port += *ptr;

//Check if the flow is mapped to a gateway
port_index = cache_lookup(src_port);

WritablePacket *q = p->push_mac_header(14);
uint8_t *q_ptr = q->data();
uint16_t gates_index;

// Assign mapped gateway
if(port_index != -1)
        gates_index = port_cache_table[port_index].gates_index;

else
{
        // No gateway present
        if(gates.size() == 0)
        {
              gates_index = -1;
              return NULL;
        }
        // Assign a gateway to the new flow and update cache
        else
        {
```

```
                gates_index = src_port % gates.size();
                cache_update(src_port, gates_index);
            }
        }

        // Append the Ethernet header to the packet
        uint8_t type[2] = {0x08, 0x00};
        memcpy(q_ptr, gates[gates_index].mac_address, 6);
        q_ptr+=6;
        memcpy(q_ptr, self_mac_address, 6);
        q_ptr+=6;
        memcpy(q_ptr, type, 2);

        return (Packet *)q;
    }
    else
        return NULL;
}
```

Source Code repository: https://github.com/scar1337/nodeman

6.3 Testing

In order to test the solution, experiments were carried out to emulate the working of this solution in a real network using the Click Modular Router. Set up using o11s as the base network, there are four machines involved. A Client C, two gateways GW1 and GW2, and a server S. The client C cannot directly reach the server S and must communicate via either GW1 or GW2. Files are stored on the server S and the client C wants to download these files.

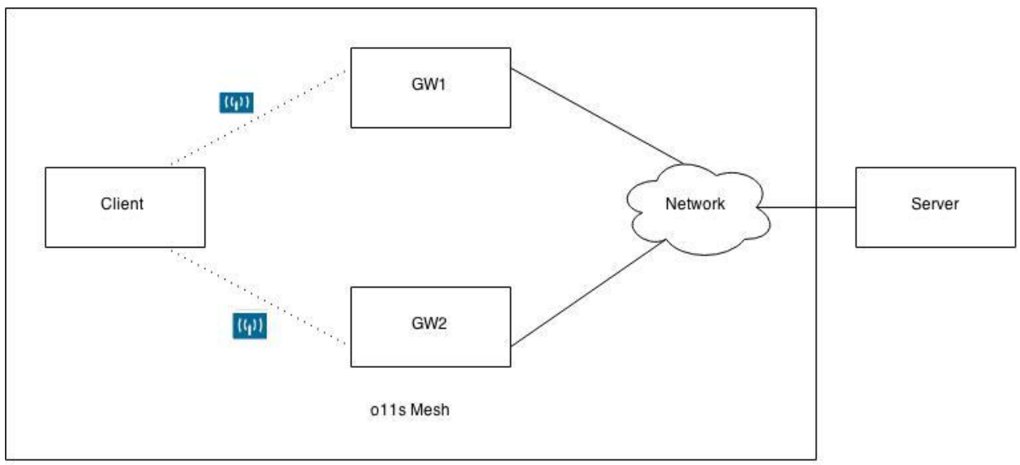
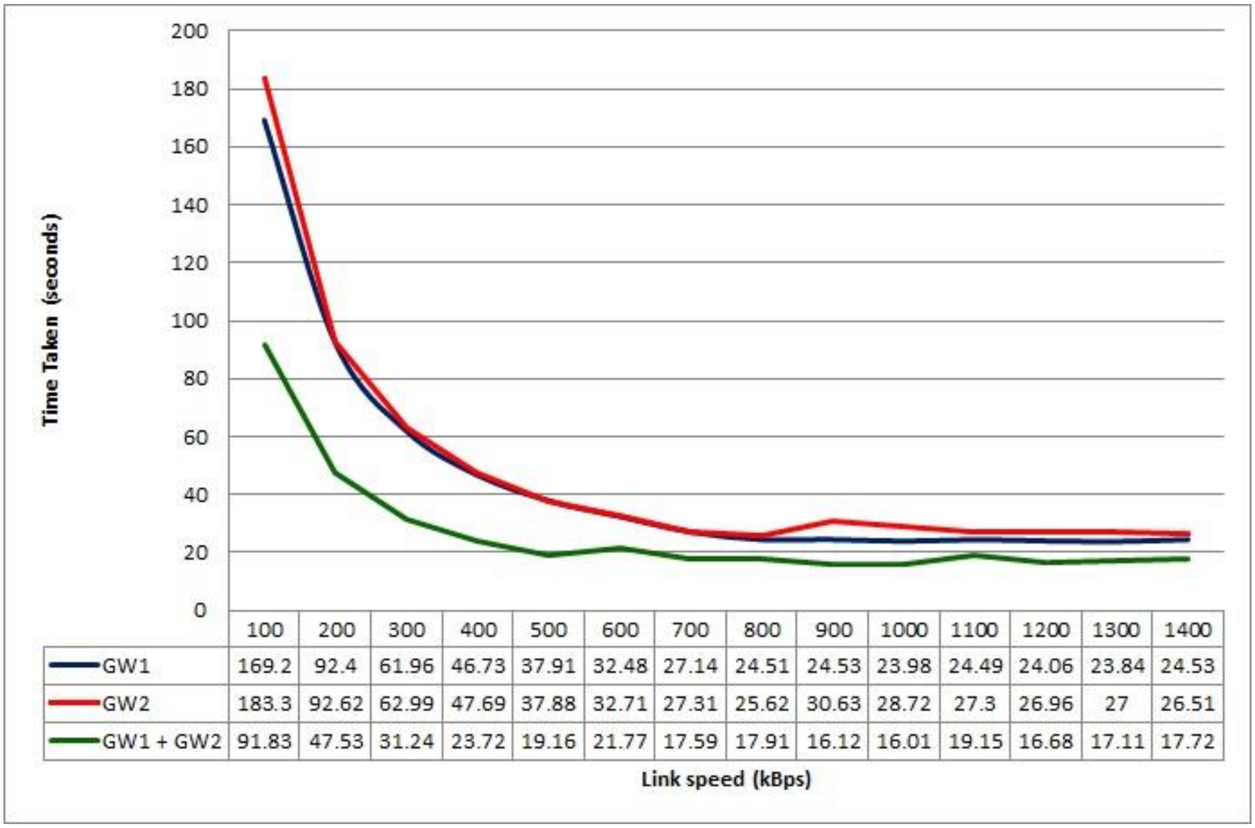


Fig 6.1: Layout of the Experiment

Experiment 1:

Set up a mesh network as shown in Figure x. The link capacities on either sides of both GW1 and GW2 are kept identical and varied from 100 KBPS to 1400 KBPS. Downloads of 16MB of data were performed for each value of the link speed once with GW1, once with GW2 (using 16 sockets) and ultimately with both GW1 and GW2 simultaneously (spitting the sockets evenly between the two gates). In this scenario, the server was a part of the mesh network, hence the wireless channel was utilized by the server leading to higher response time for the client.



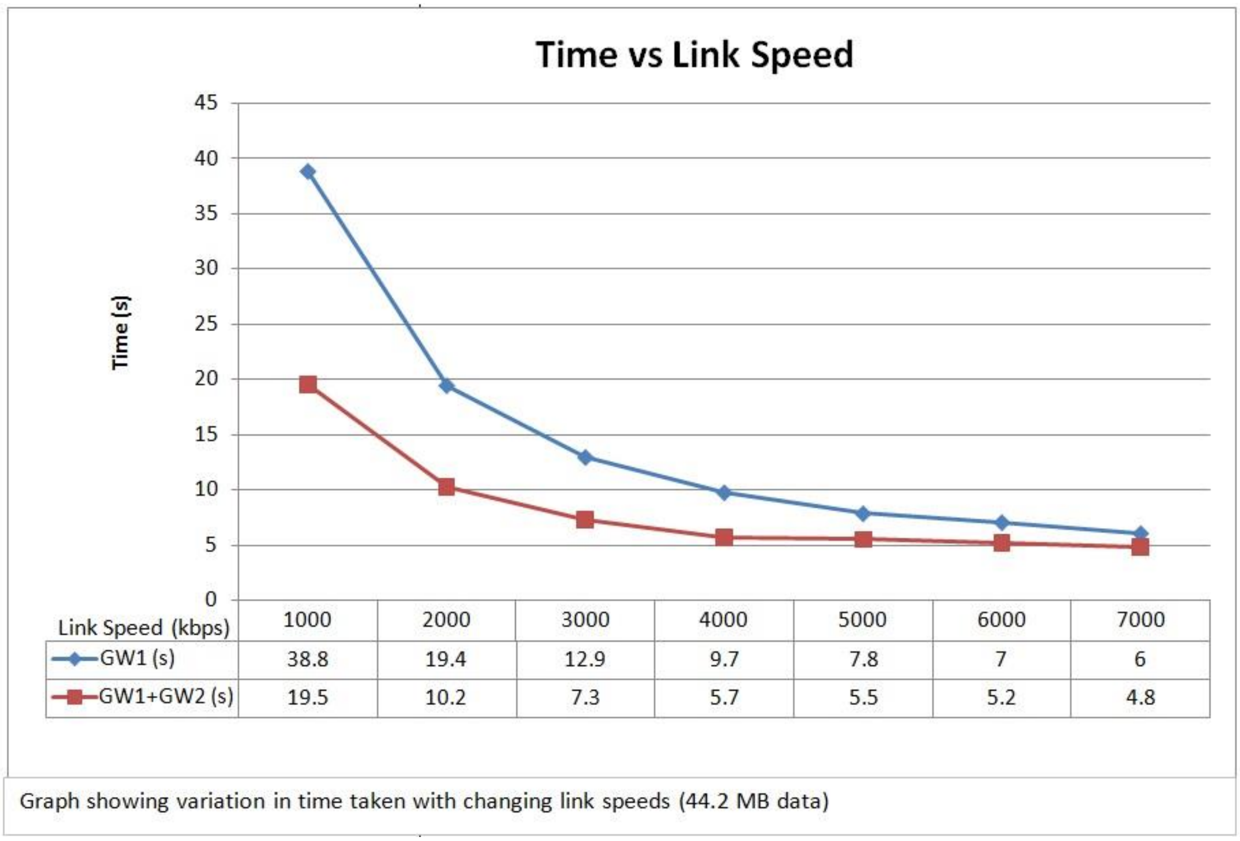
	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
GW1	169.2	92.4	61.96	46.73	37.91	32.48	27.14	24.51	24.53	23.98	24.49	24.06	23.84	24.53
GW2	183.3	92.62	62.99	47.69	37.88	32.71	27.31	25.62	30.63	28.72	27.3	26.96	27	26.51
GW1 + GW2	91.83	47.53	31.24	23.72	19.16	21.77	17.59	17.91	16.12	16.01	19.15	16.68	17.11	17.72

Fig 6.2 Experiment 1 graph (Time taken vs Link speed)

The graph below was created with experimental readings and it clearly shows gains in the network when download times are considered. It is evident from the graph that the gains are linear until the underlying link becomes a bottleneck.

Experiment 2:

Setup a mesh network as shown in the Figure: x. In this experiment, the server was outside the mesh network, so that server does not participate in spectrum sharing. The 802.11n (HT40+ mode) wireless networking standard was used in order to gain higher speeds. Downloads of 44.2 MB data were performed using one gate and then two gates combined.



Link Speed (kbps)	1000	2000	3000	4000	5000	6000	7000
GW1 (s)	38.8	19.4	12.9	9.7	7.8	7	6
GW1+GW2 (s)	19.5	10.2	7.3	5.7	5.5	5.2	4.8

Fig 6.3 Experiment 2 graph (Time taken vs Link speed)

The above graph was plotted from the experimental readings. As observed in experiment 1, there are linear gains until underlying hardware link capacity becomes the bottleneck. Due to the use of higher channel capacity(802.11n) and keeping the server outside the network the overall results were better.

Experiment 3:

An experiment was setup similar to experiment 1. In this case instead of one client, two clients and three clients were downloading 44.2MB of data simultaneously.

Time taken (sec) to download 44.2 MB of data by:-

2 Clients (C1, C2) with Gates GW1, GW2 using GW1 and GW1+GW2

Throttled (600kBps)			Unthrottled	
GW1		GW1 + GW2	GW1	GW1 + GW2
C1	132.1	68.1	62.9	55.4
C2	133	65.6	63.9	66.8

Table 6.1: Experiment 3 readings for 2 clients

3 Clients (C1, C2, C3) with Gates GW1, GW2 using GW1 and GW1+GW2

Throttled (600kBps)			Unthrottled	
GW1		GW1 + GW2	GW1	GW1 + GW2
C1	186	104	134	89.5
C2	184	103	125	81.3
C3	188	109	139	85.4

Table 6.2: Experiment 3 readings for 3 clients

Linear gains were not observed but the time taken by two gates together was less than using only one gate at a time. This shows that as the network scales up the gains reduce.

The gains depend on the spatial position of the clients and gateways.

Chapter 7

Scheduling

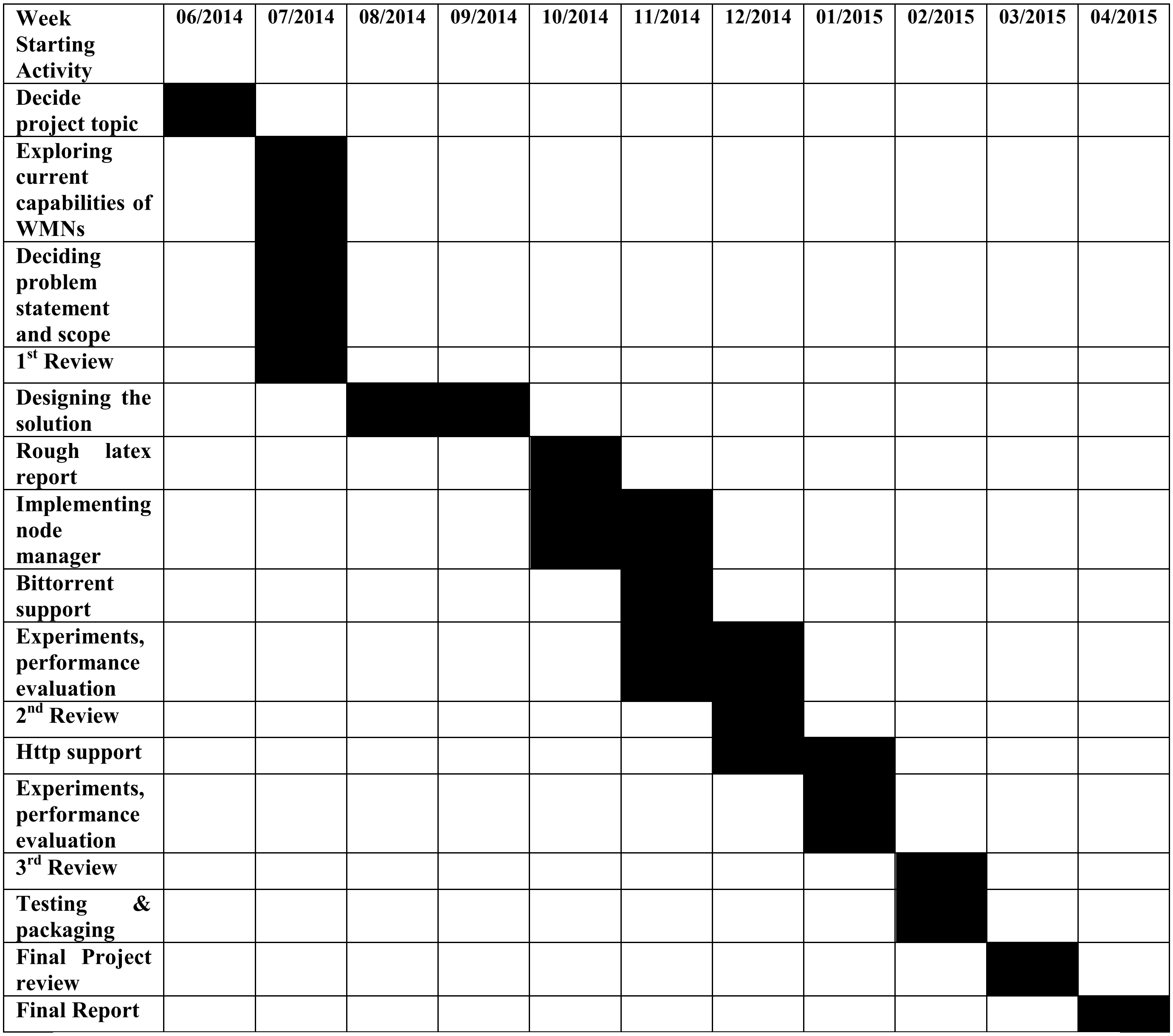


Fig 7.1:Gantt chart

Chapter 8

Future Scope and Conclusion

8.1 Future Scope

- **Patching the open80211s code in the kernel** – Currently, our solution is in userspace. If the gateway selection functionality is a part of the kernel code, the packets can be let out to the external network faster as it won't suffer kernel-userspace and userspace-kernel cycle. The basic idea has been proposed to o11s and valuable feedback was received from the community http://lists.open80211s.org/pipermail/devel/2015-February/003567.html.

- **openWRT** – openWRT is an open source router firmware. If our approach becomes a part of the firmware, the client process can be offloaded to the router. This will also help in congestion control. Hence a router-router communication will manage requests efficiently by balancing loads.

- **Community mesh** – Communities having multiple internet connections can contribute to the pool of aggregated bandwidth and reap the benefits of a high speed connection.

8.2 Conclusion

We are able to use multiple internet access gateways inside a wireless mesh network. The approach is unconventional as it moves away from the usual single associations with gateways. Various measures that need to be taken into account for a multipoint-to-multipoint association model to work mainly include gate discovery, gate selection and routing. Also, fault tolerance and fluidity for maintaining the dynamic self-healing and self-organizing nature of the mesh has been taken into account. The gains are linear. The effect will be more pronounced when the number of flows is high, especially in the case of P2P file sharing networks. Also this approach works transparently with Network Address Translation. The basis for this project has been a network based on open80211s, although the ideas may be extended to other implementations such as B.A.T.M.A.N. QoS metrics is an open ended discussion and highly dependent on the type of traffic involved and therefore may require further investigation

References

[1] Ian F. Akyildiz Georgia Institute of Technology, Xudong Wang, KiyonInc, “A Survey on Wireless Mesh Networks”

[2] Krishna Ramachandran, Irfan Sheriff, Elizabeth Belding-Royer, Kevin Almeroth, “A Multi-Radio 802.11 Mesh Network Architecture”, University of California at Santa Barbara

[3] Stefano Maurina, Roberto Riggio, Tinku Rasheed, FabrizioGranelli “On Tree-Based Routing in Multi-Gateway Association based Wireless Mesh Networks”, 20th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'09)

[4] Hiertz, G.R.Denteneer, D.Max, S.Taori, R.Cardona, J.Berlemann, L.Walke, “IEEE802.11s: The WLAN Mesh Standard”, Wireless Communications, IEEE February 2010,Vol. 17, pp. 104-111.

[5] Robert Morris, Eddie Kohler, John Jannotti, and M. FransKaashoek, “The Click modular router”, 17th ACM Symposium on Operating Systems Principles

[6] open80211s [Online] Available: www.github.com/open80211s.

[7] Click Modular Router [Online] Available: www.read.cs.ucla.edu/**click**/**click**

[8] Jonas Karlsson, Per Hurtig, Giovanni Di Stasi, Anna Brunstrom and Andreas Kassler “The Interaction Between TCP Reordering Mechanisms and Multi-path Forwarding in Wireless Mesh Networks”, 2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)

[9] Jerome Henry, “80211s Mesh Networking”, CWMP

[10] Madhusudan Singh, Song-Gon Lee and HoonJae Lee, “Non-root based Hybrid Wireless Mesh Protocol for Wireless Mesh Networks” International Journal of Smart Home Vol. 7, No. 2, March, 2013