IF-MANET: Interoperable framework for heterogeneous mobile ad hoc networks

Hassan, H.

UNIVERSITY OF WESTMINSTER

IF-MANET: Interoperable Framework for Heterogeneous Mobile Ad Hoc Networks

Hamid Hassan

A thesis submitted in partial fulfilment of the requirements of the University of Westminster for the degree of Doctor of Philosophy in Engineering and Computer Sciences

October 2015

Table of Contents

Table of Figures

Table of Tables

Abbreviations

Abbreviation	Meaning
AODV	Ad hoc On-Demand Distance Vector Routing
API	Application Programming Interface
ARP	Address Resolution Protocol
BFH	Bloom filter hash function
BGP	Border Gateway Protocol
CBR	Constant Bit Rate
CBRP	Cluster Based Routing Protocol
CIDR	Cluster Based Inter Domain Routing
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CSMA	Carrier Sense Multiple Access
DNS	Domain Name Server
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing Protocol
E2E	End to End
EIGRP	Enhanced Interior Gateway Routing Protocol
EMMA	Epidemic Messaging Middleware for Ad hoc networks
GW	Gateway
GWRT	Gateway Routing Table

IARP	IntrA-zone Routing Protocol
IDRM	Inter-Domain Routing for MANET
IEEE	Institute of Electrical and Electronics Engineers
IERP	Inter-zone Routing Protocol
IF-MANET	Interoperable Framework for MANET
InterMR	Inter-MANET Routing in Heterogeneous MANETs
IP	Internet Protocol
ITS	Interface Tuple Space
ISO	Organization of Standarization
LAN	Local Area Network
LIME	Linda in a Mobile Environment
MAC	Medium Access Control
MANET	Mobile Ad hoc Network
MAODV	Multicast AODV
MN	Mobile Node
MOM	Message Oriented Middleware
MTL	Message Translator
NRL	Normalized Routing Load
NS2	Network Simulator 2
OLSR	Optimised Link State Routing
OSI	Open Systems Interconnection
P2P	Peer to Peer

PDA	Personal Digital Assistant
QoS	Quality of Service
RERR	Route error message
RFC	Request for Comment
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RREP	Route reply message
RREQ	Route request message.
SDK	Software Development Kit
TCL	Tool command language
TCP	Transmission control protocol
TTL	Time To Live
UDP	User datagram protocol.
UML	Unified Modelling Language
UP	Universal Packet
URL	Uniform Resource Locator
WARF	Wireless and Autonomic Routing Framework
XML	eXtensible Markup Language
ZRP	Zone Routing Protocol

Abstract

The advances in low power micro-processors, wireless networks and embedded systems have raised the need to utilize the significant resources of mobile devices. These devices for example, smart phones, tablets, laptops, wearables, and sensors are gaining enormous processing power, storage capacity and wireless bandwidth. In addition, the advancement in wireless mobile technology has created a new communication paradigm via which a wireless network can be created without any priori infrastructure called mobile ad hoc network (MANET). While progress is being made towards improving the efficiencies of mobile devices and reliability of wireless mobile networks, the mobile technology is continuously facing the challenges of un-predictable disconnections, dynamic mobility and the heterogeneity of routing protocols. Hence, the traditional wired, wireless routing protocols are not suitable for MANET due to its unique dynamic ad hoc nature. Due to the reason, the research community has developed and is busy developing protocols for routing in MANET to cope with the challenges of MANET. However, there are no single generic ad hoc routing protocols available so far, which can address all the basic challenges of MANET as mentioned before. Thus this diverse range of ever growing routing protocols has created barriers for mobile nodes of different MANET taxonomies to intercommunicate and hence wasting a huge amount of valuable resources. To provide interaction between heterogeneous MANETs, the routing protocols require conversion of packets, meta-model and their behavioural capabilities. Here, the fundamental challenge is to understand the packet level message format, meta-model and behaviour of different routing protocols, which are significantly different for different MANET Taxonomies.

To overcome the above mentioned issues, this thesis proposes an Interoperable Framework for heterogeneous MANETs called IF-MANET. The framework hides the complexities of heterogeneous routing protocols and provides a homogeneous layer for seamless communication between these routing protocols. The framework creates a unique Ontology for MANET routing protocols and a Message Translator to semantically compare the packets and generates the missing fields using the rules defined in the Ontology. Hence, the translation between an existing as well as newly arriving routing protocols will be achieved dynamically and on-the-fly. To discover a route for the delivery of packets across heterogeneous MANET taxonomies, the IF-MANET creates a special Gateway node to provide cluster based inter-domain routing.

The IF-MANET framework can be used to develop different middleware applications. For example: Mobile grid computing that could potentially utilise huge amounts of

aggregated data collected from heterogeneous mobile devices. Disaster & crises management applications can be created to provide on-the-fly infrastructure-less emergency communication across organisations by utilising different MANET taxonomies.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Acknowledgements

This thesis would not have been possible without the help and support of many people.

First of all, I would like to express my sincere gratitude to my supervisor Dr. Philip Trwoga for his constant support and immense knowledge towards my PhD study and research work. His continuous guidance, encouragement, inspiration and enthusiasm have always motivated me for progressing in this research and writing of the thesis. I could not have imagined having a better supervisor for my PhD course.

I am also deeply thankful to my co-supervisor Professor Izzet Kale for his professional and personal support for resolving issues and motivated me throughout my studies.

I am also thankful to my friend Dr. Fahad Anwar for having valuable discussions, sharing his experiences and encouragement throughout the PhD studies.

Finally, I am grateful to my family for always encouraging me to do my best and provide me with all the help and support. Their attitude, support and patience enabled me to achieve this goal.

Dedication

All praise to Allah who helped in this endeavour without any might or power from myself

To my family members

List of Publications

- Hassan, H., Trwoga, P. and Kale, I. (2015). IF-MANET: Interoperable Framework for Mobile Ad Hoc Networks. In: Computer Networks. Springer, 54–68.
- Hassan, H., P. Trwoga, and I. Kale. (2010). Performance Comparison of Bluetooth Scatternet Formation Protocols. In ECS Research Conference. University of Westminster, London, UK

Chapter 1

Introduction

1.1 Overview

The continuous improvement in enabling mobile and wireless technologies has empowered the mobile devices like smart phones, tablets, PDA's, laptops, and sensors with enormous processing power, storage capacity and enhanced wireless communication. To utilize the very significant resources of mobile resources a wireless communication mechanism is required. The progression in wireless technology has created an infrastructure less wireless network called mobile ad hoc network (MANET) (Raghuveer 2008) and is best suited for communication in mobility driven devices. But due to the nature of MANET, the network topology changes frequently, unpredictably and this has created challenges as traditional wireless routing protocols are not suitable for mobile wireless networks.

The Research community has developed and is busy developing routing protocols like AODV(Perkins & Royer 1999), OLSR (Jacquet et al. 2001), ZRP (IETF 2002) etc. to overcome the above mentioned challenges. There is enormous number of routing protocols already proposed and new one are continuously arriving but there is not a single routing protocol that can fulfil the basic requirements of communication in heterogeneous MANETs. This diverse range of routing protocols have created a new challenge in this environment as in general the heterogeneous mobile devices cannot communicate with each other and thus are unable to facilitate the full exploitation of mobile resources.

1.2 Aims and Objectives

The aim of this research is to investigate the heterogeneous mobile ad hoc networks and proposes a comprehensive framework for heterogeneous MANET taxonomies. This framework enables the interoperability in order to provide communication between the mobile nodes of heterogeneous routing protocols. The framework confronts the above mentioned challenges, hides the complexities of heterogeneity and provides homogeneous layer for applications to seamlessly communicate with heterogeneous MANETs.

1.3 Challenges of Interoperability in Heterogeneous MANETs

Due to the rapidly changing topology and heterogeneity of resources there is a high demand to provide interaction between the heterogeneous MANETs which uses different routing protocols. To achieve this goal there are a number of key challenges which need to be addressed first. The following are the major challenges which are barrier in achieving interoperability in heterogeneous MANETs.

1.3.1 Route Discovery in Heterogeneous MANETs

Route discovery is a process initiated by a source node when it wants to start communication with a destination node which is not in its routing table or belongs to another MANET domain. If the destination node belongs to another MANET domain then the source will not be able to find the destination especially in heterogeneous MANETs where different MANET domains have different routing taxonomies (e.g. reactive, proactive) or the same taxonomy but different routing protocols (e.g. AODV, DSR of reactive taxonomy). In these types of scenarios, the route request messages sent by the source node are not received by the destination nodes or are discarded. As a result, the route between the source and the destination nodes are not constructed.

1.3.2 Routing Protocol Heterogeneity

Different routing protocols, even though they belong to same MANET taxonomy, have different packet format, data types and data processing behaviour. The communication between heterogeneous routing protocols requires conversion of packets from one protocol to another. To connect MANETs of different routing protocols, when encounter dynamically, must understand one another and exchange data. Here, the fundamental challenge is to understand the packet level message format and behaviour of routing protocols in order to map and generate the missing fields between source and target routing protocols.

1.3.3 Data Heterogeneity

Applications may use data that is represented in different ways and or have different meanings. Thus even the mobile nodes exchange data but they are still unable to understand the commands and messages. There must be a generic abstract data format which will act as a bridge between the different data formats and transforms them semantically.

1.3.4 Dynamic Runtime Configuration

When mobile devices, while moving, joins or leaves the ad hoc network, the system must be able to recognize the protocol types and dynamically configure the system at runtime.

1.3.5 Limited battery power, processing and memory

The mobile devices are small in size and have limited processing power, storage and battery life. They also rely on battery power which drains out while heavy processing. In addition, due to the mobility and lack of power, these devices are more prone to frequent disconnections. Thus the proposed system must use techniques which will take care of the resource scarcity.

1.4 Motivations & Contributions

To utilize the mobile resources effectively and provide interoperability between the different routing protocols, this thesis has proposed the interoperable framework called IF-MANET (Interoperable Framework for MANET). The IF-MANET is a reusable artefact of software architecture, design and implementation for the development of MANET middleware applications. The framework addresses the challenges of Interoperability in the heterogeneous MANETs and provides a plug-in style API's for multiple MANET routing protocols. Its abstraction layer hides the complexities of heterogeneity of MANETs and provides a seamless homogeneous API's to external applications.

Unlike other proposed frameworks which are implemented in operating system at kernel space, the IF-MANET belongs to a user space and will be implemented at application layer to provide platform and implementation independence. The framework will address the Interoperability of heterogeneous routing protocols and will provide a component based plug-in style adapters to communicate across different routing protocols. IF-MANET's abstraction layer will hide the complexities of heterogeneity and will provide a generic layer to access the mobile resources seamlessly and homogeneously. The IF-MANET is a reusable artefact of software architecture, design and implementation for the development of middleware applications for MANETs.

Middleware systems built from this framework serve as a communication infrastructure for mobile ad hoc applications.

In order to accomplish this aim, we investigate the following key aspects of Interoperability in heterogeneous MANETs. These aspects form the basis of the IF-MANET and complement each other to achieve the interoperability in heterogeneous MANETs. The following sub-sections briefly describe these key contributions (aspects) of this research:

1.4.1 Component Based Extendable Architecture

The IF-MANET Framework provides a component based architecture where each component acts as an independent service to provide a distinct set of functionalities. Based on the service oriented component model, the IF-MANET is designed to be loosely coupled so that the new components, services, and algorithms can be easily integrated without affecting the existing functionality. It also enables the integration and configuration of components at runtime and hence newly arriving routing protocols can easily be integrated within the framework at runtime.

1.4.2 Gateway Routing Table

To provide communication between MANETS of different routing protocols, the Gateway nodes (MANET cluster heads) require special storage to save the state of routing information in such a way that the outgoing packets can be associated with the incoming packets. Also the storage schema must be generic such that it can handle packet formats of different routing protocols. It is fundamental to keep track of routing information across the MANET domains in order to provide the interoperability between them.

The IF-MANET has proposed a special routing table called GWRT to store and associate the routing information of nodes communicating with external MANETs. It is designed in such a way that it is independent of any specific routing protocol packet. Each IF-MANET Gateway creates a GWRT and maintains the node routing information while communicating with external MANETs.

1.4.3 Abstract Message

Mobile nodes in different MANET domains running different routing protocols may have no or minimal knowledge about the packet formats of each other. In spite of the fact, that the protocols belonging to same MANET taxonomy e.g. reactive routing may have same behaviour but due to different packet formats, data types, field names and their values, they cannot understand each other.

To address these issues, the IF-MANET has proposed an Abstract Message which is independent of the type of a routing protocol. The IF-MANET Gateway node converts the native routing protocol to the abstract message using IF-MANET‘s MTL (Message Translator) and send it over to different MANETs. The destination Gateway node, after receiving the abstract message, uses the MTL and converts it into the local (native) MANET routing protocol. Due to protocol independence, the size of an abstract message varies because different routing protocols have different packet sizes and the abstract message contains only their mandatory fields to reduce memory processing and battery power consumption.

1.4.4 IF-MANET Gateway

The IF-MANET Gateway acts as a head node for a MANET cluster and enables communication with other MANETs. It maintains the state of the source mobile node packets, discovering and communicating with destination node in another MANET, to allow relationship between outgoing and incoming packets of same request. For example, if a node sends a route discovery and the receiving node is running different routing protocols then the receiving node discards all the RREQ messages and hence they cannot communicate with each other. To overcome this problem, the Gateway node intervenes and assumes that the destination node is in different MANET or is running different routing protocol. It then transforms the source packet into format compatible to destination packet, add entry into its routing table and forward the request to the destination node. On receiving route response (RREP), it updates the related entry in its routing table, converts the packet back to format compatible to source node and sends reply to source node.

1.4.5 MANET Ontology and Message Translator

The communication between heterogeneous routing protocols requires conversion of data packet from one protocol to another. The packet format, their data types and meaning for different routing protocols, even though they belong to the same routing taxonomy, are different. The IF-MANET framework has proposed Ontology (Euzenat et al. 2007) for MANET routing protocols and a Message Translator (MTL) to semantically compare the packets and generate missing fields from the rules saved in the Ontology. Hence, the translation between existing as well as the newly arriving routing protocols is achieved dynamically and on-the-fly.

The Ontology transforms the heterogeneous routing protocols from one system to another in three phases. In first phase, it discovers the packet in the vocabulary of Ontology and identifies its class. If the packet is not found then the Ontology learns the new packet by adding it into its vocabulary. The second phase provides the comparison between the packets of the two routing protocols with the help of semantic rules defined within the Ontology. The rules compare the packets of the two routing protocols and produces similarities and differences between them. The third phase takes the differences found in previous phase and applies rules defined in Ontology to generate the missing fields in source packet to convert it into target packet.

1.4.6 Route Discovery in Heterogeneous Routing Protocols

Route Discovery is a process initiated by a source node when it wants to start a communication with a node which is not in its routing table or belongs to another MANET domain. If the source node initiates a route discovery and the destination node belongs to another MANET domain with different routing taxonomy (e.g. reactive, proactive etc.) or the same taxonomy but different routing protocol, then they cannot understand each other and hence cannot communicate. To achieve the interoperability, the IF-MANET has proposed a route discovery mechanism by using special Gateway nodes. The Gateway nodes with the help of IF-MANET GWRT, abstract message, translator and Bordercasting (Haas & Pearlman 2002) technique, enable the communication across heterogeneous MANETs. Unlike, other proposed solutions, which modifies the behaviour of original routing protocols to trigger Gateway Nodes by sending extra data signals, this solution does not changes the behaviour of original routing protocols.

1.5 Tools and Technologies for Implementation

The IF-MANET routing protocol has been implemented in C++ (Stroustrup 2013), TclCL (TclCL 2015) using Eclipse IDE for C++ (Eclipse Foundation, 2011) and Linux/Ubuntu (Ubuntu, 2012) as an operating system. The implementation was deployed on network simulator NS-2.35 (Ns2, 2008) to run the simulations and evaluate the performance of IF-MANET in heterogeneous MANET environment. For proof of concept we have simulated two different routing protocols i.e. AODV (Perkins et al. 2003) and MAODV (Viswanath & Obraczka 2004) along with IF-MANET to evaluate the performance and connectivity ratio of heterogeneous MANETs using IF-MANET.

1.6 Thesis Structure

The rest of the thesis is structured as follows: Chapter-2 presents background knowledge necessary for an in-depth discussion regarding mobile ad hoc network (MANET), its routing protocols, inter-domain communication and interoperable frameworks. It then analyzes the existing work which has been conducted in relation to the research concept. Chapter-3 proposes the interoperable framework for the heterogeneous MANETs i.e. IF-MANET and discusses its design in depth followed by algorithms. Chapter-4 implements the proposed IF-MANET design and its algorithms. Chapter-5 concentrates on simulating the implementation of the proposed work in network simulator NS-2. It then analyzes the results produced to validate our research concept. Chapter-6 summarises the research efforts carried out in this thesis, followed by a suggestions on possible future work that can be carried out in relation to this research work

Chapter 2 Literature Review

This section discusses different technologies required for the IF-MANET. Thereafter, it analyses the related work that is close to the research approach outlined in this thesis. This chapter is organised as: section 2.1 presents the background of technologies used in this research, section 2.3 explains the work related to this research work, section 2.3.3 evaluates the related work and section 2.4 summaries the chapter.

2.1 Background

The main objective of this section is to provide background information that is required for the understanding of subsequent chapters. As such, the chapter is organised as follows:

- Sections 2.1.1 to 2.1.4 provide a justification of the wireless, ad hoc networks and routing protocols used in this study.
- Sections 2.1.5 to 0 explain the middleware frameworks in general and specific to the MANET.

2.1.1 Wireless Networks (Infrastructure Based)

Conventional wireless networks are Infrastructure based LAN (Local Area Network) which uses a fixed base station for communication between different wireless nodes (Raghuveer 2008). The wireless communication uses different frequency bands, transmission range and wireless technologies (Sarkar et al. 2007). Table 2.1 below, presents the comparison between the features of different wireless technologies.

Wireless Technologies	Data Rate (Mbps)	Range (Meters)	Relative Cost	Radio Frequency	Data Network
802.11 (Wifi)	10 - 100	50	Medium	2.4/5.0 GHZ ISM Band	TCP/IP
802.16 (WiMax)	30 - 40	100	High	2.4 GHZ ISM Band	TCP/IP
IrDA	16	< 2	Low	N/A	PPP
Bluetooth	1	10 - 30	Low	2.56 GHZ ISM Band	Via PPP

Table 2.1: Comparison of Various Wireless Technologies

2.1.2 Wireless Ad Hoc Networks (Infrastructure-less Networks)

Ad hoc networks are self-configuring networks of nodes connected through wireless links and can communicate with each other without relying on an infrastructure or a centralized administration (Sarkar et al. 2007). If the nodes are not static i.e. mobile then the ad hoc network is called Mobile Ad hoc Network (MANET). The nodes in the ad hoc network are free to move randomly and organize themselves arbitrarily thus the network's wireless topology may change rapidly and unpredictably. Due to this uncertainty the nodes do not have prior knowledge of the network topology around them. In general new or moving nodes in the MANET announce their presence and listen for broadcasts from their neighbours. With the passage of time each node gathers information about neighbouring nodes and maintains a list so that they have one or more paths to reach the destination node. These types of networks can operate in a standalone fashion, or can be connected to the larger network such as the Internet. The major challenges to design the effective routing protocols for communication in MANET are multi-hop routing, dynamic mobility, QoS, low resource devices and heterogeneity. Figure 2.1 below, shows a high level design of MANET where nodes (devices) are communicating with each without a centralized infrastructure.

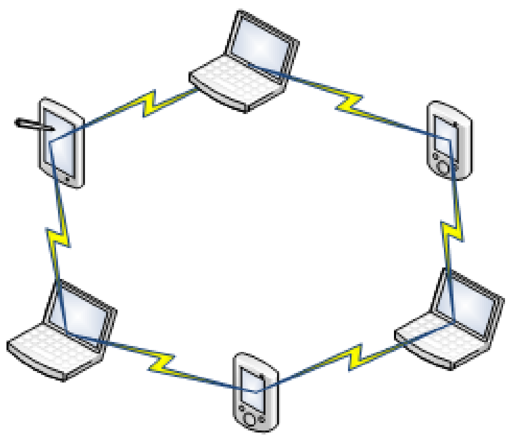


Figure 2.1: Generic Mobile Ad Hoc Network

2.1.3 Applications of Mobile Ad Hoc Networks

Unlike fixed wireless networks, nodes in the MANET are free to move and organize themselves in an arbitrary fashion. MANETs are best suited for environments where an infrastructure is unavailable or deploying an infrastructure is not cost optimal. The proposed works in (Sarkar et al. 2007; Corson & Macker 1999) have discussed different applications suitable for mobile ad hoc wireless networks and are classified into the following categories:

- Community Network (Enterprise Network): To extend the network/internet connectivity where coverage of access points is insufficient. The local nodes creates ad hoc network and connect to access points to route nodes data.
- Disaster & Recovery Environment: Where the existing infrastructure has destroyed due to catastrophic event. The nodes within the communication range connect each other to form a mobile ad hoc network to broadcast the information.
- Emergency Environment: Establishing communications for fire/safety/rescue operations or other scenarios requiring rapidly-deployable communications with survivable, efficient dynamic networking
- Vehicle Network: It enables communication where one of the tenants is vehicle e.g. Vehicle and passenger device using Bluetooth system, Vehicle to Vehicle (V2V) communication in convoy driving or accident/road block pre-emption, Vehicle to System (V2S) for vehicles communicating with Transportation Systems e.g. Toll Plaza.
- Sensor Network: To deploy low resources sensors in the field e.g. battlefield, agriculture farms to collect data.

2.1.4 MANET Routing Protocols

The main purpose of routing protocols is to discover an optimal path to reach the destination from a source node and establishes a communication link between them. The highly dynamic nature of the MANET results in frequent and unpredictable topology changes and hence creates additional complexities for routing protocols to communicate among mobile nodes (Sarkar et al. 2007). The following are the key challenges for routing in limited resource devices, heterogeneous environments and dynamic nature of MANETs:

- keep routing table small to consume minimum resources
- choose fastest route for given destination
- choose reliable route for given destination
- keep table up-to-date when nodes die, move or join
- Maintain routes and Manage nodes effectively where wireless channel is weak, unreliable and unprotected
- Quality of Services (QoS)

There are numerous routing protocols proposed for the optimal performance under various network environments and variance of these protocols, suitable for different environments, are continuously arriving. These routing protocols have been classified based on their routing and underlying architectural behaviour (Boukerche et al. 2011). These protocols are organised into following four fundamental categories and are shown in Figure 2.2 below. The detail of these routing types and their protocols are explained in the subsequent sub-sections:

- Source-initiated (Reactive or on-demand)
- Table-driven (Pro-active)
- Hybrid (Mix of Reactive & Pro-active)
- Hierarchical (Inter-Domain Cluster Based)

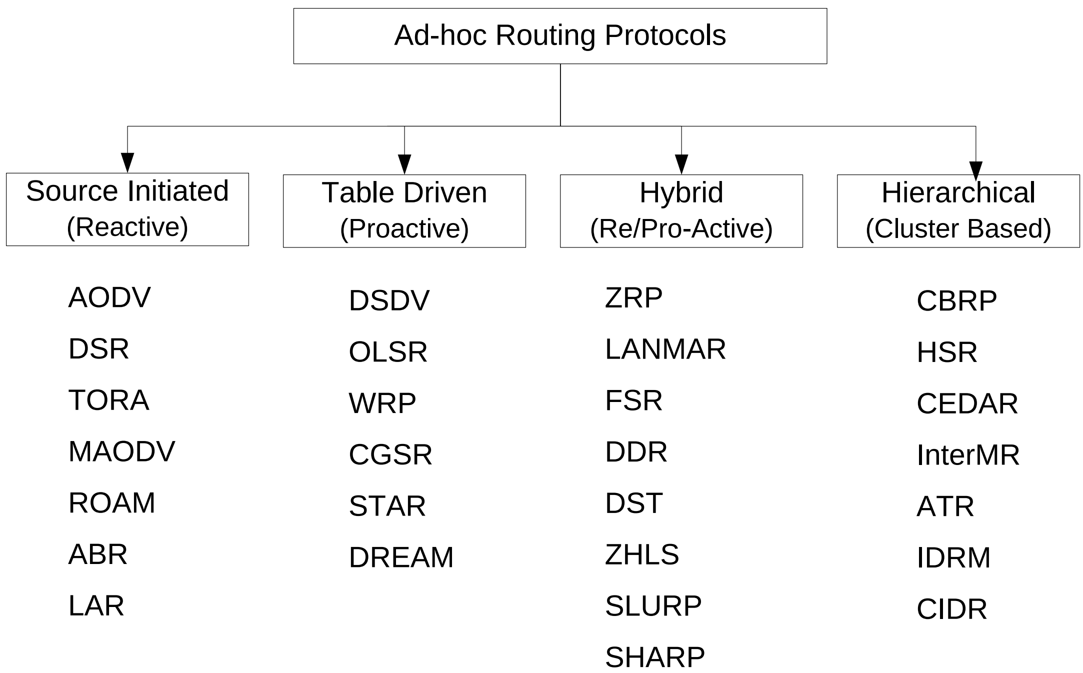


Figure 2.2: Classification of MANET Routing Protocols

2.1.4.1 Proactive Routing Protocol (Table Driven)

The proactive routing protocol maintains routing tables at each node. These routing tables contains paths to all possible reachable destinations and continuously updates them by background exchange of routing information irrespective of communication requests (Abolhasan et al. 2004). Any change in network topology triggers a propagation of updated information throughout the network to maintain a consistent network map. Due to this consistency the nodes can create instant connections to other nodes. Figure 2.3 shows a MANET in which nodes communicate with each other to route data from source node (MH1) to destination node (MH8) via intermediate nodes (MH2, MH4, MH6 and MH5).

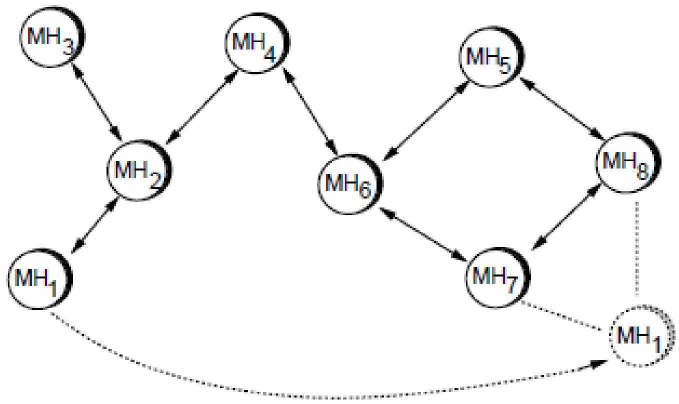


Figure 2.3: Proactive Protocol node movement in MANET (Perkins & Bhagwat 1994)

The most widely used proactive routing protocols are DSDV (Perkins & Bhagwat 1994), OLSR (Clausen & Jacquet 2003) and Babel (Chroboczek 2011). The difference between these protocols is the numbers of tables used, type of information stored and technique of beaconing and maintenance of routing tables. The brief description of how these protocols communicate is explained in the following sub-sections.

2.1.4.1.1 Destination Sequenced Distance Vector (DSDV)

The DSDV routing protocol (Perkins & Bhagwat 1994) is an improvement of Bellman–Ford algorithm (Cheng et al. 1989) to provide guaranteed loop free routes by using sequence numbers. Each entry in the routing table contains the sequence number which is generated by destination node. DSDV always uses the route with highest sequence number to identify stale routes and avoid formation of loops. If the sequence numbers are same then the route with least metric (hop count) is given precedence. The route entries which have not been updated for a "route expiry time period" are called "Stale Entries" and these entries along with routes, using these nodes as next hop, will be deleted.

To maintain the routing table consistency and to reduce network traffic two type of routing packets are periodically sent throughout the network. One is called "full dump", it carries all available routing information and are sent infrequently to avoid network load. The second type is "incremental", it contains changed information since last update and are sent more frequently for concurrency of routing tables throughout the network. However, DSDV requires regular update of routing tables which consumes battery power and causes huge network overhead even when network is idle. Therefore it is not suitable for large scale dynamic networks as most of the network bandwidth will be consumed in updating routing table information.

2.1.4.1.2 Babel

Babel (IETF 2011; Chroboczek 2011) is a loop-avoiding proactive distance vector routing protocol and operates on IPv6 and IPv4 networks. Its design is based on the concept of DSDV, AODV and Cisco's EIGRP (Albrightson et al. 1994) and is designed to work in both wired and wireless networks. Its key feature, unlike naive distance-vector routing protocols, is that it limits the frequency and duration of routing pathologies such as routing loops and black-holes during re-convergence. The use of sequence numbers, to prevent routing loops, is borrowed from DSDV. Babel does not create routing loops when every prefix is originated by at most one router but may create transient routing loop when prefix is created by multiple routers. It provides two approaches to optimise relaying performance. Firstly, it uses history sensitive route selection i.e. intelligently selects previously created route when more than one route of same link quality is available, to minimise the impact of route switching between source and destination pair which can causes route instability. Secondly, it sends a reactive update for route request when found a link failure from its neighbour. However, due to proactive nature, Babel relies on periodic routing table updates hence in large networks it generates more traffic than reactive protocols and is not good match for dense network of mobile ad hoc nodes.

2.1.4.1.3 Optimized link state routing (OLSR)

OLSR (Jacquet et al. 2001; Clausen & Jacquet 2003) is a proactive routing protocol which optimises traditional link-state algorithm. Like table driven routing it periodically exchange link-state messages to maintain the topology information. The optimisation of OLSR is that it compact the size of information sent in the messages and reduces the number of rebroadcasting nodes during each route update by using multipoint relaying (MPR) technique. MPR reduces duplicate transmission of broadcast packets to reduce the flooding of packets in the network. Each node selects a set of one-hop neighbours which are called the multipoint relays (MPR) for the node. The neighbours of the node which are not MPRs process the packets but cannot retransmit. The multipoint relay set of node N must satisfy the condition that every node in the two hop neighbours of N must have a bi-directional link with nodes in the MPR set of N and keep the MRP set to smaller in order to broadcast the minimum packets. The bi-directional links can be determined by periodically broadcasts Hello packets containing list of its one hop neighbours and their link status. From the list of nodes in hello messages, each node selects a subset of one hop neighbours, which covers all of its two hop neighbours.

2.1.4.1.4 Advantages & Disadvantages of Proactive Routing Protocols

Advantages	Disadvantages
• Quick Route Creation. • Less E2E Transmission Delay. • Reliable and Efficient in networks of less topological changes & low density.	• Extra network overheads. • Delay or drop of data packets in overloaded networks. • Maintenance of unused paths occupies a significant bandwidth especially in networks of frequent topology changes. • Requires more memory, processing power and hence more battery power.

Table 2.2: Advantages and Disadvantages of Proactive Routing Protocols

2.1.4.2 Reactive Routing Protocols (On Demand / Source Initiated)

The reactive protocols are developed to reduce the routing overheads in proactive protocols by sending routing packets only when source initiates a route discovery. The RREQ (Route Request) packets are flooded into the network by the source in search of a path to the destination. The route discovery completes when a route is found and the destination node (or an intermediate node with a fresh route to the destination) sends a RREP (Route Reply) back or all the possible outgoing paths from the source are searched. There are number of proposed reactive routing protocols, the most widely used one are DSR, AODV and are explained in the following sub-sections:

2.1.4.2.1 Ad hoc on-demand distance vector (AODV)

AODV (Perkins & Royer 1999; Perkins et al. 2003) is on-demand reactive routing protocol and is based on the DSDV and the DSR routing techniques. It borrows the concept of beaconing and sequence numbering from the DSDV and route discovery procedure from the DSR. The AODV discovers routes on as needed basis to reduce the number of broadcast messages forwarded throughout the network. AODV broadcasts RREQ (Route request) Packet to discover the current route and the destination (or an intermediate node with the current path to the destination) node forwards RREP (Route Response) Packet back to the source node. Each mobile node in AODV operates as a router and maintains a monotonically increasing sequence number to discard stale cached routes and provide loop free routing. Unlike DSR, which sends complete routing information in a packet, AODV packet carries only the destination address. Similarly in RREP, DSR packet carries the address of every node along the route whereas AODV packet contains only the destination address and the sequence number. Therefore, AODV has routing overhead compared to DSR and best suited for highly dynamic and dense networks. However, nodes experience larger delays during route construction and increases E2E (End to End) transmission delays as network density increases.

2.1.4.2.2 Dynamic source routing (DSR)

DSR (Johnson et al. 2007) is a multi-hop routing protocol based on "on-demand" algorithm. The on-demand routing algorithm consists of two phases i.e. Route Discovery and Route Maintenance. In Route Discovery Phase, when a source (initiator) node need to send a packet to the destination (target), and can't find the route to destination in its Route Cache, then it will broadcast route request packet to all nodes within its wireless range. Every node, which receives the Route Request, will add its own node id in the Route Request message and re-broadcast the packet if it is not the destination. The Route Request id remains same in the route discovery cycle initiated by initiator node. If an intermediate node has already received another Route Request with same Request id and/or its own address is already listed in route record of Route Request then it will discard the Route Request packet. If an intermediate node finds a fresh route to requested destination in its route cache then it will return a Route Reply message to the source node rather than forwarding the route request message. If the receiving node is a destination then it will first check its route cache for route back to initiator and if found then it will send a route reply packet containing the complete route from source to the destination otherwise it will reverse the sequence of hops in route record of Route Request and use this as source route on packet carrying Route Reply. Source node on receipt of Route Reply will store the routing path in its Route Cache and uses it for sending subsequent data packets to this destination.

In Route Maintenance phase both originating and forwarding node requires confirmation of packet delivery. Packet is retransmitted up to a maximum number of attempts until the confirmation of receipt is received. If no receipt confirmation is received, this node returns a Route Error message to the original sender of the packet identifying link to next hop is down. The initiator node then removes that link from its route entry and initiates a new route discovery phase if it can't find another valid route to destination in its route cache.

2.1.4.2.3 Multicast Ad hoc On-Demand Distance Vector (MAODV)

MAODV (Zhu & Kunz 2004) protocol is an extension to the AODV routing protocol. Unlike uni-casting in AODV, the MAODV protocol discovers multicast routes on-demand, through the use of broadcast mechanisms. When a node wants to join a multicast group or to send a multicast message and does not have the route in its own table, it sends a route discovery message. For each multicast group, a bi-directional tree is created. The tree contains members of two distinct classes. Member can be either a node that has joined the multicast tree or a node that is has not joined the multicast group but is forwarding the multicast messages towards other nodes in the tree. Like uni-cast route discovery in AODV, in MAODV, multicast routes are discovered on demand, based on a broadcast route Request-Reply mechanism.

2.1.4.2.4 Advantages & Disadvantages of Reactive Routing Protocols

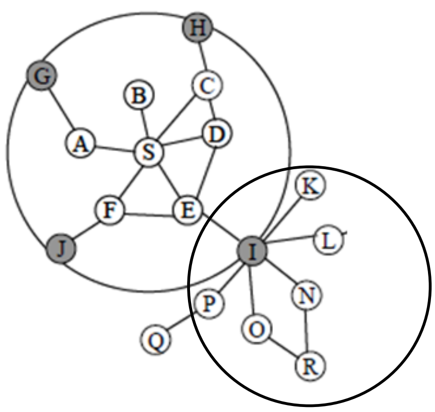
Advantages	Disadvantages
• Maintain only routes that are currently in use thereby reducing the network overheads. • It provides good performance for the dense networks with frequent disconnections.	• As routes in use are not maintained so they perform route discovery before exchange of information among nodes. It will cause delay in delivering the first packet. • The network traffic creates overhead when topology changes frequently even though only currently used routes are maintained.

• Each node doesn't maintain route information of every reachable node rather it discovers the route on request. Therefore, it reduces memory footprint and hence increases battery life.	• The network traffic creates overhead when topology changes frequently even though only currently used routes are maintained. • If a route to a destination changes during data transfer the route to destination packets will be lost. • Overhead includes the bandwidth consumed by RREQ/RREP messages.

Table 2.3: Advantages & Disadvantages of Reactive Routing Protocols

2.1.4.3 Hybrid Routing Protocols

The hybrid routing protocols combines the advantages of proactive and reactive routing schemes. The routing is established with proactive protocols at areas where mobility is low and reactive protocols for areas with high mobility. To achieve higher scalability the Hybrid routing scheme proactively maintain routes to nodes in close proximity and determine routes to distant nodes or across different zones (proximities) using on-demand reactive route discovery strategy. The optimal combination of these two strategies will yield reduced network overheads and hence increased overall performance. Figure 2.4 shows a hybrid routing scheme with a source node S and a destination node N are in different zones. It uses a proactive scheme within a zone to maintain the routes and reactive across the zones to find the route to a destination node. The following section explains the most widely used hybrid routing protocols.



Keys:

Source Node : S

Destination Node : N

Peripheral Nodes : H, I, G, J

Figure 2.4 : Routing in Hybrid Routing Scheme

2.1.4.3.1 Zone routing protocol (ZRP)

ZRP (Beijar 2002; IETF 2002) is a hybrid of proactive and reactive routing algorithms. It uses proactive mechanism for communication between nodes in close proximity (zone) whereas reactive mechanism for inter-zone communication. The size of zone radius is based on P factor which is defined as the number of hops to the perimeter of the zone. The nodes whose minimum distance to the central node is less than zone radius P are called Interior nodes and uses IntrA-zone Routing Protocol (IARP) to pro-actively maintains up-to-date routing tables. The nodes outside the zone i.e. whose distance is equal to or greater than the zone radius P are called Peripheral Nodes and uses Inter-zone Routing Protocol (IERP) for route discovery. IERP uses a reactive approach for communicating with nodes in different zones. The route discovery and packet delivery to Peripheral nodes is provided by the Bordercast Resolution Protocol (BRP). The ZRP architecture is shown in Figure 2.5 below. It illustrates that, whenever a source node wants to send a packet, it first check destination within its local zone using IARP route discovery service. If a route is found locally then the source node will send packet using proactive protocol. If the destination is outside the zone then it will send a route request packet to its peripheral nodes using BRP. If it will find a route to destination then source node will use IERP to route packets to external zones.

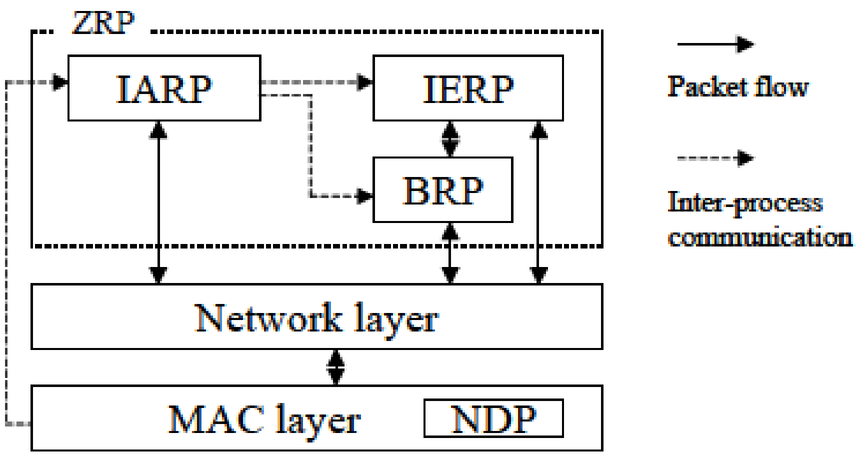


Figure 2.5: ZRP Architecture

2.1.4.3.2 Advantages & Disadvantages of Hybrid Routing Protocols

Advantages	Disadvantages
• It has a flat view of network which reduces the organizational overheads • Appropriate combination of Reactive and Proactive routing protocols will decrease E2E transmission delays and increases overall performance	• Large values of routing zones can behaves like a pure proactive protocol • Small values of Zones can behave like a pure reactive protocol

Table 2.4: Advantages & Disadvantages of Hybrid Routing Protocols

2.1.4.4 Hierarchical Routing Protocols

Hierarchical routing scheme organizes nodes in a hierarchical manner, typically through clustering techniques. It divides nodes to different groups (called domains or clusters) and assigns different functionalities to nodes inside and outside of a cluster. The clustering scheme reduces the routing table and packet sizes by including in them only part of the network information thus minimizes the control overhead and improves the efficiency of routing. There are different techniques to develop hierarchical network, the most popular is to build a group of nodes called clusters or zones. Each cluster has a leading node called cluster head which communicates with other cluster head nodes or directly with their nodes on behalf of its own cluster. Figure 2.6 below, shows the construction of hierarchical clusters from connected ad hoc nodes. There are number of proposed hierarchical routing protocols out of which the important ones are explained in the following sub-sections:

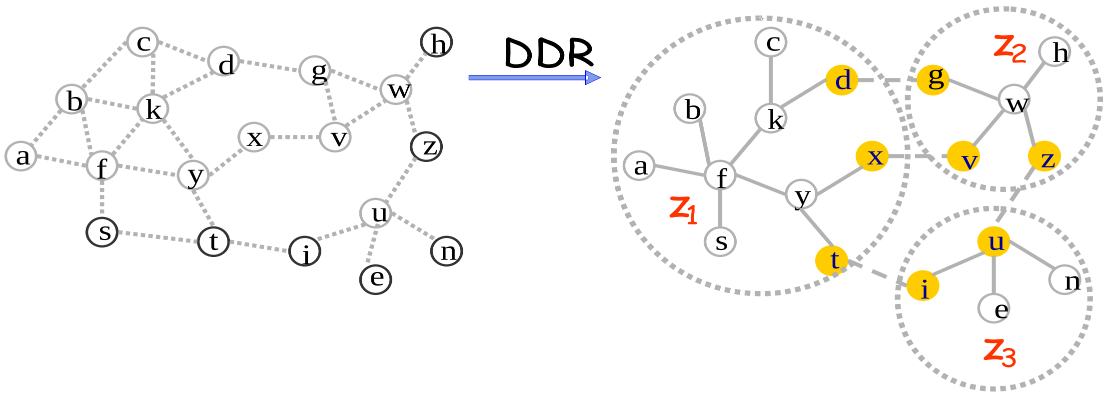


Figure 2.6: Construction and Communication of hierarchical routing protocol

2.1.4.4.1 CBRP (Cluster Based Routing Protocol)

CBRP (Khatkar & Singh 2012) is classified as hierarchical routing protocol where the nodes are grouped into clusters. Initially all nodes are in ‘undecided’ state i.e. node state before joining a cluster. A cluster algorithm will be performed when an "undecided" node wants to joins a cluster. The "undecided" initiate joining process by setting a time-out period and broadcast a Hello message. When a Cluster Head node receives the Hello message, it replies back with “Triggered Hello” message. The undecided node when receives the triggered hello message from Cluster Head, indicating the bi-directional link between them, then it will change its status to "member" state. If the time-out expires and undecided node does not contain any bi-directional link with neighbours then it re-enters the undecided state and repeats the joining process after some time interval. Otherwise it will select itself as a Cluster Head and changes the status from "undecided" to "cluster Head" in subsequent Hello messages.

Every node in CBRP maintains a Neighbour Table with their neighbour’s information i.e. link status (uni/bi-directional), neighbour id and Role (Cluster Head) and periodically broadcasts this table in Hello messages. The Cluster Head keeps information of all its cluster member nodes and cluster heads of its neighbouring clusters.

Whenever source node need to send a packet and no active route is found in its routing table then it will initiate a Route Discovery and floods Route Request Packet (RREQ) to Cluster Heads. The cluster head will discard the packet if it has already received the packet otherwise it will check if destination is in its local cluster. If destination is found locally then it will forward the packet to the destination otherwise it will flood the RREQ to its neighbouring cluster heads which in turn broadcast to their neighbouring cluster heads if not found in their local cluster. If destination is found then it will send a reply including the route information in Route Reply Packet (RREP). If the source does not receive any RREP message until time expires then it will go into exponential back-off before re-sending RREQ.

2.1.4.4.2 Cluster-based Inter-Domain Routing (CIDR)

CIDR (Zhou et al. 2009) is a cluster-based inter-domain routing protocol. It uses a clustering algorithm to discover the group of travelling companions, based on their affinity characteristics like geography, motion or task, to form a cluster in each domain. It then elects a Cluster Head (CH) for each cluster (i.e. affinity group) to act as a Domain Name Server (DNS) for its local cluster (i.e. subnets) and neighbour clusters.

The CH then advertises its connectivity, members, and domain information to its neighbours and rest of the network.

2.1.4.4.3 InterMR (Inter-MANET Routing in Heterogeneous MANETs)

InterMR (Lee et al. 2010) proposes an inter domain routing protocol to provide interoperation across heterogeneous MANETs. It extends IDRM (C. Chau et al. 2008) by introducing attribute based address scheme and doesn't require Domain Name Server (DNS). It uses the core design concepts BGP (Rekhter & Li 1995) such as Intra and Inter gateway protocol (i-BGP and e-BGP). But unlike static prefix-based address scheme in BGP, the InterMR uses attribute based address scheme that defines the address of a MANET from the attributes such as symbolic names, properties, services, in the MANET to provide transparency from split/merge of dynamic MANETS. Domain in the paper is interchangeably used with heterogeneous MANETS. For inter-domain and intra-domain changes, InterMR uses periodic update beacons to disseminate information and uses new algorithm to perform dynamic gateway election to reduce computation. Like IDRM, it uses the Bloom Filter technique (C. Chau et al. 2008) for destination resolution to increase the scalability of MANET.

2.1.4.4.4 Advantages & Disadvantages of Hierarchical Routing Protocols

Advantages	Disadvantages
• Only cluster heads exchange routing information, therefore the number of control overhead transmitted through the network is far less than the traditional flooding methods	• Overheads associated with cluster formation and maintenance. • They suffer from temporary routing loop. It is because some nodes may carry inconsistent topology information due to long propagation delay. • Cluster heads are potential bottlenecks which can increase the End-to-End delay, which in-turn decreases the overall performance of the network.

Table 2.5: Advantages & Disadvantages of Hierarchical Routing Protocols

2.1.4.5 Summary of MANET Routing Protocols

Several routing protocols for each of the routing category discussed in previous section have been analysed and compared. While different classes of protocol operate under different scenarios, they usually share the common goal to reduce control packet overhead, maximize throughput, and minimize the end-to-end delay. The main differentiating factor between the protocols is the ways of finding and/or maintaining the routes between source and destination pairs. Table 2.6 compares the features required for routing in MANET against different MANET routing taxonomies.

Features	**Proactive Routing**	**Reactive Routing**	**Hybrid Routing**	**Hierarchical Routing**
Quick Route Creation	Yes	No	Partial	Partial
Instant Connection	Yes	No	Partial	Medium to High
Reliable in less topology changes	Yes	No	Partial	Partial
Extra Overheads in frequent topology changes	Yes	Partial	Medium	Low to Medium
Route maintenance	Yes	Partial	Medium	Low
Low Bandwidth usage	No	Yes	Partial	Low
Routing Overhead	Medium to High	Only on RREQ	Partial	Medium

Control Traffic	High	Low	Medium	Medium
Periodic Messages	Yes	No	Hybrid	Medium
Routing structure	Flat	Flat	Flat	Hierarchical
Route Availability	Yes	No(On-Demand)	Hybrid	Partial (Depends on Destination Location)
Route Discovery	Periodic	On-Demand	Partial	Partial (Depends on Destination location)
Storage	High	Low	Medium (High within & Low between Zones)	Medium
Delays (Packet Delivery Delay)	Low	Medium (route is in cache) High (discovering new route)	Small for Intra-Zone High for inter-zone	Medium to Low
Scalability	Efficient for Small networks (up to 100 nodes)	Efficient for dense networks. i.e. upto few hundred nodes	Fair for network of 1000 or more nodes	Efficient for network of 1000 or more nodes
Quality of Service	Low	Medium	Medium	High

Table 2.6: Comparison MANET Routing Taxonomies

2.1.5 Middleware Frameworks

Middleware is a software layer above the operating system and below the applications to provide a higher degree of abstraction in distributed programming (Bernstein 1996) and its architecture is shown in Figure 2.7. Whereas, Framework is the development environment that is primarily characterized by API and defines how the middleware functionality can be used. The main purpose of the middleware is to hide the low-level complexities of heterogeneity, concurrency, operating systems, programming languages and networking communication to facilitate the application programming and management (Da Silva & Albini 2014). Figure 2.7 below, shows a traditional middleware architecture which hides the processing details and platform heterogeneity from application layer. It will intercept the calls from application layer, processes the request, generates a message and establish a connection with server or peer node to transfer the messages. When the results are ready, a response is sent back to client middleware, which returns to the application layer (Bruneo et al. 2007). The user only experiences a local call to the given API.

Figure 2.8 showed a relationship between middleware layer and International Organization for Standardization (ISO) Open Source Initiative (OSI) network reference model. Middleware address shortcomings of the network operating system; therefore, it implements the session and presentation layers of the ISO/OSI reference model. The middleware act as a facade layer and provides API to application developers to request parameterized services from remote components and execute them without worrying about implementation of the session and presentation layers (Tanenbaum 2002).

Example of widely used middleware's include but not limited to OMG's CORBA (OMG 2012), Microsoft's COM (MS COM 2014) , SUN's Java Remote Method Invocation (RMI) (Downing 1998), Remote procedure Calls (RPCs) (Waldo 1998), IBM's MQSeries™ (Gilman & Schreiber 1996) etc. In general, these solutions were designed for using on traditional fixed environments, as the Internet and hence these approaches are not suitable for MANETs as they present a heavy computational load and do not deal easily with the dynamic topology (Hadim 2006).

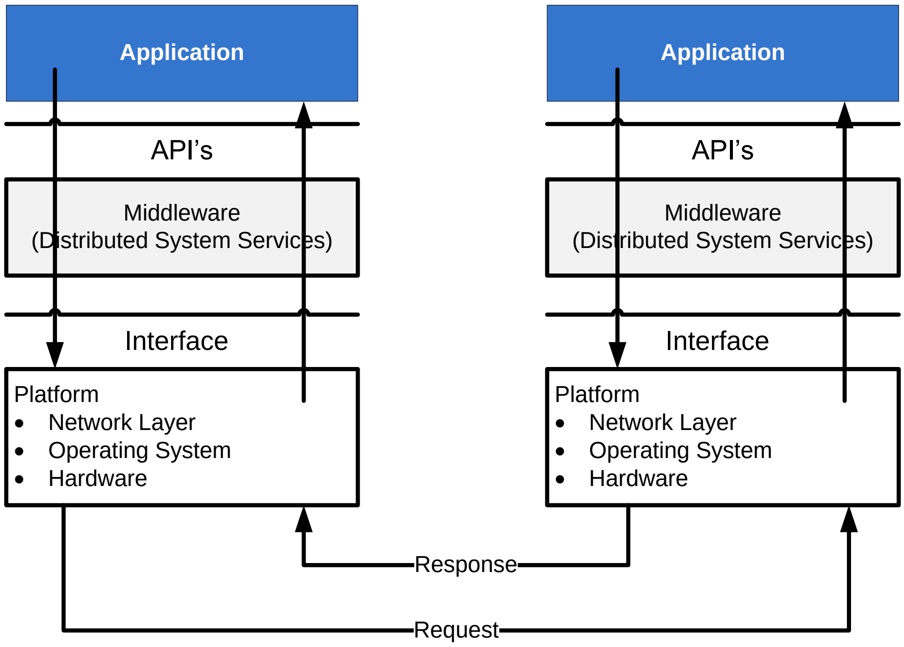


Figure 2.7: Traditional Middleware Architecture (Bernstein 1996)

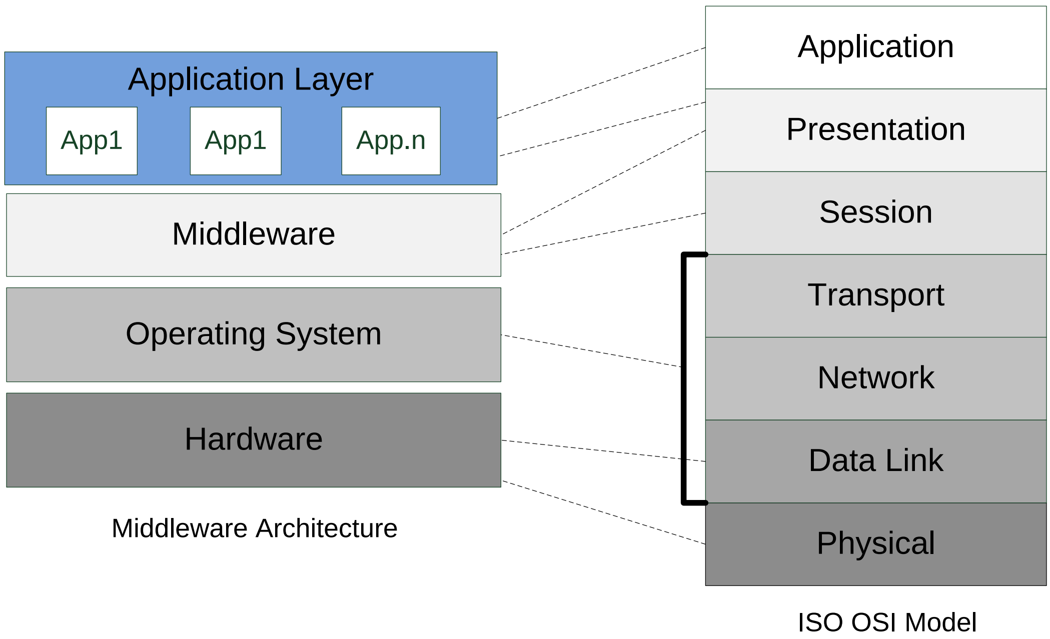


Figure 2.8: Middleware Model and its relation with OSI Model

2.1.6 Middleware Frameworks for MANET

Middleware for mobile devices must be computationally lightweight due to the scarce resources and asynchronous since low bandwidth and unpredictable disconnections are the norm in MANETs. Thus, the traditional middleware solutions for fixed networks are not applicable for MANETs. There has been some progress carried out in transforming the traditional middleware solutions for mobile environment such as IIOP (Internet Inter ORB Protocol), Mobiware, Alice, DOLMAN, Odyssey and Jini. But these solutions are still not feasible for mobile ad hoc environments as they rely on semi fixed networks and require more resource rich devices. The design of a successful middleware solution for MANETs is not a trivial task and must consider the following challenges:

- Heterogeneity
- Mobility and Dynamic Topology
- Scalability
- Limited Resources
- Quality of Service

There are number of middleware solutions proposed for MANET. They are classified, according to their design strategy and communication characteristics, into four main categories i.e. Component and Mobile agents based middleware, Event based and Message Oriented Middleware (MOM), Peer to Peer based middleware, Tuple spaces based middleware. The detail of these categories including the proposed middleware solutions under them are explained in the subsequent sub-sections:

2.1.6.1 Component and Mobile Agent Based Middleware

A component is an encapsulated unit of common functionalities and deployment that is an instantiation of a component Type. The component based middleware contains a set of common functions (components) communicating with each other or to the external systems via well defined abstract interfaces. Figure 2.9 shows a generic architecture of component based middleware communicating using required and provided interfaces.

The components provide an autonomous functionality and clear decoupling which allows operations to be decentralized and enables dynamic reconfiguration. The clear separation of key functionality and aspects of deployment on the functional side, such technology offers potential technical advantages. Among them is reduced communication cost, reduced bandwidth usage, the possibility of using remote interfaces, and support for offline computation. In addition, it allows dynamic adaptation to changing conditions which is a fundamental requirement for MANET routing in highly heterogeneous environment. This approach allows functionality to be divided into smaller and lightweight components and hence increases the scalability and yields considerable energy saving for MANET middleware's. Examples of proposed middleware's for MANET are Mobile Gaia, SELMA, Chandrakant et al.'s, ManketKit , WARF, StarLink etc.

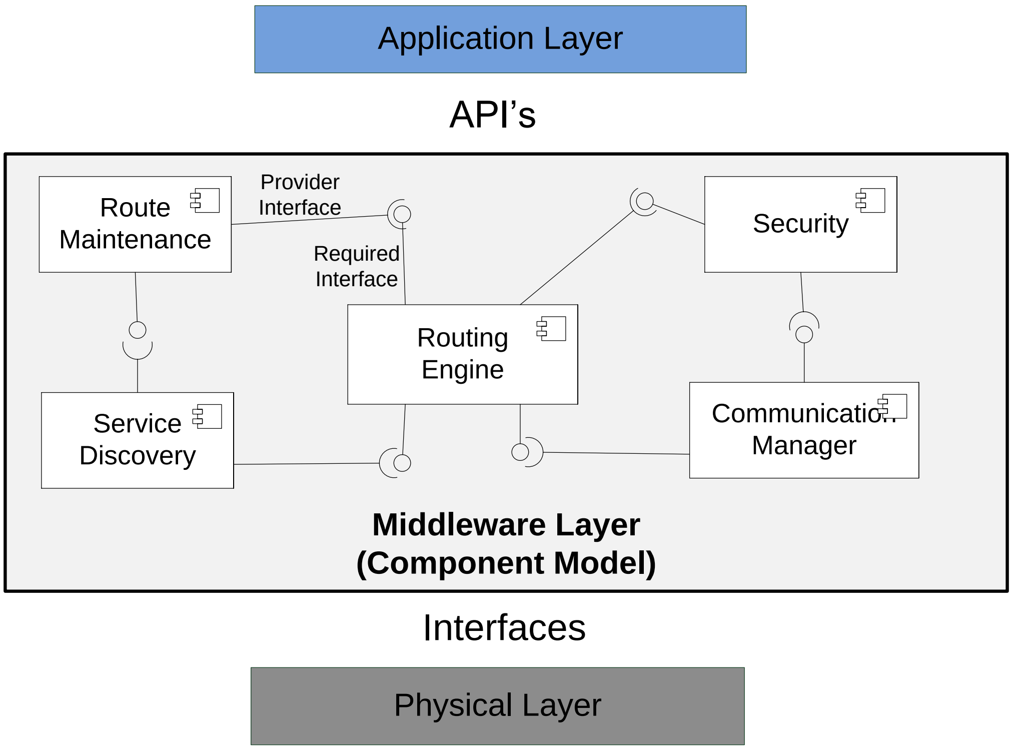


Figure 2.9: Component Based Middleware for MANET

2.1.6.2 Tuple Space Based Middleware

Tuple space is the implementation of an associative memory which provides asynchronous, anonymous and content based communication decoupling application components in time, space and flow (Gelernter 1985). Tuple contains informational data and is queried through a mechanism based on content matching. Tuple space shares information stored in a globally accessible, persistent, content-addressable data structure, typically implemented as a centralized tuple space. Tuple space based middleware's provides high degree of decoupled communication and simple interface to exchange data anonymously as tuples are addressed in an associative way by specifying their contents (Da Silva & Albini 2014). Figure 2.10 shows a logical architecture of tuples space based middleware.

Mobile ad hoc environments are characterized by low and variable bandwidth, frequent disconnections…etc. Thus a decoupled and opportunistic style of communication is required. Decoupled means that communication happens even in the presence of disconnections, and opportunistic as it exploits connectivity whenever available (Hadim 2006).

The decoupled facility allows communication to happen even at the time of disconnections and anonymous exchange of data allows sharing data in distributed environment. These features makes tuple space based middleware a best suited for MANETs where nodes disconnects frequently and communicates in a distributed manner. Example of tuple space based middleware for MANET are LIME, LIMONE, TOTA (Tuples on the Air), MESHMdl, and JADE etc

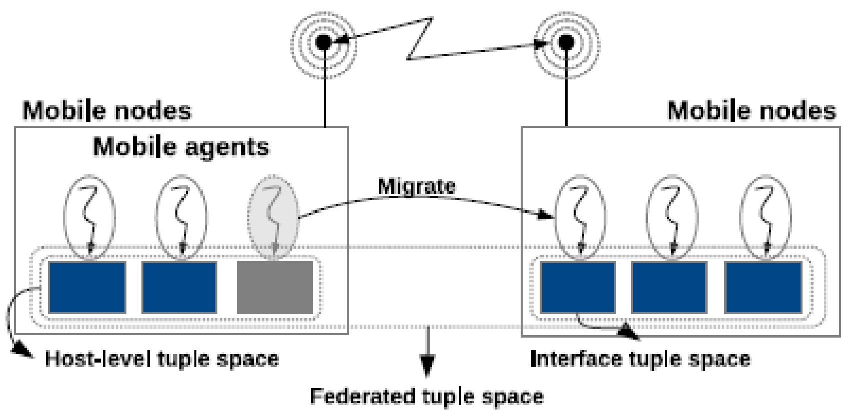


Figure 2.10: Logical Architecture of Tuple Space based Middleware (Murphy et al. 2001)

2.1.6.3 Event Based Message Oriented Middleware (MOM)

Message-oriented middleware (MOM) provides asynchronous, loosely coupled communication between sender and the receiver nodes. MOM uses event based publish-subscribe mechanism to facilitate message exchange between nodes. This approach is quite suitable for dynamic environments such as MANETS, where most applications are based on events (Denko et al. 2009). Examples of proposed MOM for MANET are: STEAM, Spontaneousware and JMS for MANET. The most widely used middleware are discussed below:

The most significant limitations of current MOM for mobile platforms are that they typically support a single, predefined messaging style (e.g., publish/subscribe). This restriction limits the scope of platforms in that they cannot easily accommodate, or easily be extended to accommodate richer or more specialized forms of interaction. One of the constraints is that some middleware components of the event services cannot be located on independent physical machines. In addition, such components may not be co-located with mobile entities and pose problems regarding availability, consistency, coverage and computational resources.

2.1.6.4 Peer to Peer (P2P) Middleware

Peer-to-Peer (P2P) (Schollmeier 2001) networks which allow direct sharing of resources (e.g., CPU, bandwidth, storage) among a large number of users in a decentralized manner (Schollmeier 2001). In P2P architecture, no single host is permanently seen as a server, but each single host is able to play both the role of server and client according to the user’s and application’s needs. This decentralized architecture makes it a suitable backbone for ad hoc environments. Hence appropriate middleware is needed to provide abstractions to the upper applicative layers and cope with high dynamics such as mobility and resource discovery (Hadim 2006).

P2P networks main focus is to provide resource sharing on top of existing reliable communication infrastructures whereas MANETs provides multi-hop wireless connectivity where no infrastructure exists or is inadequate. Many middleware solutions for MANET, using P2P approach, have been proposed due to its decentralized characteristics e.g. Proem, ExPeerience, JMobiPeer , Peer2Me and are surveyed in (Da Silva & Albini 2014)

2.1.6.5 Summary of MANET Middleware

In this section, we surveyed different middleware approaches specifically adopted for MANET. We then identified the major challenges like power awareness, limited resources, scalability etc that the design of middleware for MANETs faces. Table 2.7, compares the different MANET middleware taxonomies against the key MANET requirements. From the table, it is apparent that the component based middleware is fully compliant to most of the requirements of MANET as compared to its counterpart middleware taxonomies. In addition, due to its dynamic runtime configuration it will provide extendable platform for heterogeneous MANET components and hence is best candidate for dynamic interoperable MANET middleware framework.

Features	Event Based Message Oriented Middleware	Component & Mobile Agent Based Middleware	Tuple Space Based Middleware	Peer to Peer Based Middleware
Power Realization	P	P	P	P
Limited Resources	P	F	N	P
Scalability	P	F	P	P
Mobility	P	F	P	F
Heterogeneity	P	F	P	P
Limited Storage	F	P	N	N
Light Weight	P	F	P	P
Extendibility	N	F	P	N
Runtime Configuration	P	F	N	P
Keys: F: Fully Compliant, P: Partial Compliant, N=No (don't Compliant)				

Table 2.7: Comparison of MANET Middleware Taxonomies

2.2 Challenges of Communication in MANET

There are numerous MANET routing protocols proposed, as discussed above, and with technical advancement the new one are emerging continuously. Due to the rapidly changing environment and heterogeneity of resources there is a high demand to interoperate these protocols in order to utilize them. There are few solutions available who have addressed the limitations of heterogeneous mobile communication but none has addressed the issues when mobile nodes move across different domains each running different routing topology. Unfortunately, current mobile computing technology lacks interoperability between different infrastructures which prevent the technology from reaching a wider spectrum of applications. Following are the major challenges to achieve the interoperability against heterogeneous MANET's:

- Heterogeneity of resources
- Heterogeneity of routing protocols
- Service Discovery Heterogeneity

- Low Power, limited processing and memory
- Heterogeneous Hardware and OS Resources
- Dynamic Mobility

2.3 Related Research Work

There are many research works proposed to address the challenges of heterogeneous MANET which were introduced in chapter 1. This section summarises and gives brief analysis of their research work. Initially, interoperability of routing protocols through inter-domain systems is discussed and analyzed followed by the different middleware frameworks for solving heterogeneous MANET problems. Finally, a comparison matrix analyzing their features in accordance with comparison criteria to address the interoperability in heterogeneous MANETs will be presented.

2.3.1 Inter-domain Cluster Based Interoperability of Routing Protocols

Inter-Domain routing methodology provides scalable MANET routing and support heterogeneity of routing protocols. It divides the networks into smaller independent routing units called domains (clusters) where each cluster contains the nodes with same type of routing protocol. Each cluster then elects head node(s), also called border nodes, from each cluster and they are responsible for inter-domain communication (Ma & Chuah 2005). In this section different proposed research works addressing the heterogeneity of routing protocols, using Inter-Domain based approach, are discussed and analysed.

Border Gateway Protocol (BGP) (Rekhter & Li 1995) is a standard inter-domain routing protocol to provide communication across heterogeneous network systems in the internet world where nodes are static. However, in ad hoc network environment, the nodes are mobile and network topologies changes randomly. The BGP is designed for wired networks and doesn't support dynamic mobility of nodes and hence it cannot be used to communicate between heterogeneous ad hoc networks.

Hincapi´e, et al (Hincapié et al. 2006), **Ratish Agarwal** (Agarwal et al. 2009) have presented surveys on clustering in MANET and highlighted key issues like creating cluster-based MANET, network overhead and selection of cluster heads. They have discussed different clustering algorithms for MANET like Lowest-ID heuristic and Highest degree heuristic. Both the researches have discussed the techniques and issues in the formation of clustering and electing head nodes in MANETs but they have not addressed the issues of routing packets within and across the clusters.

Plutarch (Crowcroft et al. 2003) translates address spaces and transport protocols among domains to support interoperation of heterogeneous Internet networks. It divides the world into contexts, each comprising some set of hosts, routers, switches, network links and provides communication across these contexts by interstitial functions, which map between the set of functionalities encapsulated by contexts. **TurfNet** (Schmid et al. 2005) supports inter-domain networking without requiring global network addressing or a common network protocol. However, these protocols provide interoperability of heterogeneous wired and wireless networks but do not address the issues of infrastructure less mobile ad hoc networks.

Zone Routing Protocol (ZRP) (IETF 2002), (Beijar 2002) creates routing zone which define a range (in hops). The nodes within local zones use a proactive protocol whereas they use a combination of a reactive routing and border-cast protocol for inter-zone routing. Cluster Based Routing Protocol (**CBRP**) (Jiang et al. 1999) uses hierarchical routing approach to group nodes into clusters and uses cluster-heads to interact with nodes within and outside the clusters. Landmark routing **(LANMAR)**(Pei, Gerla & Hong 2000) creates subnet of groups of nodes and uses Fisheye Routing (FSR) (Pei, Gerla & Chen 2000) within the local scope. Clusters are formed based on node mobility characteristics and cluster heads within each group become the Landmarks for inter-domain routing (Boukerche et al. 2011). These protocols reduces the network overheads and increase packet transmission time as compared to reactive and proactive routing protocols but these protocols doesn't address the interoperability of nodes with heterogeneous routing protocols.

SHARP (Ramasubramanian et al. 2003) is a hybrid routing protocol and uses both proactive and reactive routing protocols to adapt different traffic patterns and improve performance. The basic idea of SHARP is to create proactive routing zones around the nodes with lots of data traffic, and use reactive routing in other areas. Although the hybrid routing protocols enable communication between proactive and reactive routing protocols, they require nodes to be controlled by the same administrative policies and do not support autonomous operations by multiple MANETs. Thus they do not provide a systematic solution to interoperability among multiple MANETs with different routing protocols.

Ad hoc Traversal Routing (ATR) (Fujiwara et al. 2012) is an inter-domain routing protocol and provides the interoperability between different ad hoc networks (domains) using gateways called ATR. ATR connects two different networks to each other by converting control messages from one network to another network and adding the node address of different networks into the routing table of routing protocols. For inter communication between networks, firstly intra (internal network) nodes communicate with their local ATR node which then communicate with ATR nodes of other networks to find the destination node. If the local protocol is reactive, ATR uses the packet conversion mechanism to communicate with local routing protocols whereas it uses the address sharing mechanism for communication, if the local protocol is pro-active. If source and destination are of different types and belongs to different networks then ATR gateway nodes will transform the route request messages according to the destination routing protocol and forward to neighbour ATR node. In case of networks with proactive routing protocols, ATR nodes of different networks will share the node address information with each other. Figure 2.11, below shows the heterogeneous networks where nodes of different routing protocols can only communicate through an ATR head node.

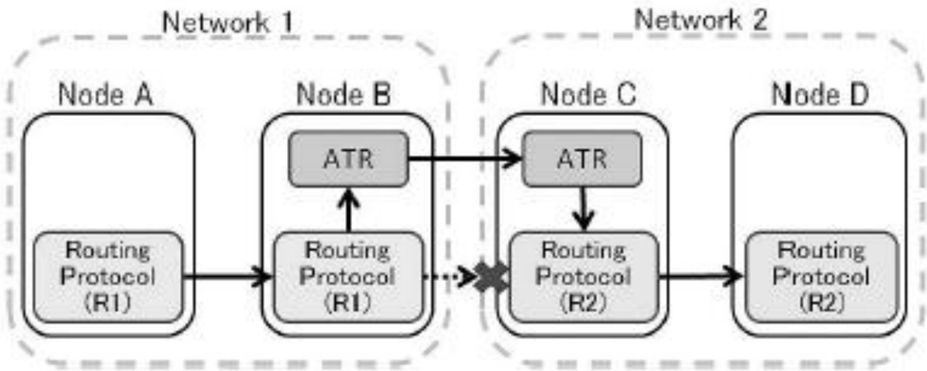


Figure 2.11: ATR - Communication between heterogeneous MANETs

ATR nodes share the node address information in all networks with each other whereas many nodes do not become the destination node that provides any services in the network. As a result, the high overhead is incurred to exchange the node address information among ATR nodes. Also, it doesn’t address the ATR (Gateway) node selection criteria and has ignored the node heterogeneity and focuses only on routing among heterogeneous networks each of which are comprised of homogeneous nodes

Inter-Domain Routing for MANET’s (IDRM) (C.-K. Chau et al. 2008) is a networking protocol which provides an interoperation among MANETS. It uses core design principles of BGP to enable inter-domain routing in MANETs but unlike BGP, it addresses the MANET challenges of dynamic network connectivity and environment specific specialised routing protocols. It uses special nodes called Gateways for inter-domain communications by bridging the technical seam that exists between different MANETs.

For routing of packets, it uses semi-proactive path vector routing mechanism i.e. at the inter-domain level the routing of packets is proactive whereas at the intra-domain level the routing can be reactive or hybrid. If a node wants to send a packet to the destination node in another domain then it will forward the packets to intra-domain gateways. In case of reactive domain, the source node will initiate a route request and a gateway node with a valid route to destination will respond whereas for proactive domain, the source node will select an intra-domain gateway from its local list of gateways. If the selected gateway is directly connected to the destination domain then it will forward the packet directly otherwise it will forward the packets to a gateway which is connected to the destination domain. For incoming packets, the gateway performs a protocol translation and initiates a route discovery process if the domain is reactive otherwise it will determine the destination from its local routing table. However, it doesn’t address the challenges of communication between different topologies and how the routing protocols of different types discover and communicate with each other.

Cluster-based Inter-Domain Routing (CIDR) (Zhou et al. 2009) is a cluster-based inter-domain routing protocol. It uses clustering algorithm to discover the group of travelling companions, based on their affinity (common) characteristics like geography, motion or task, to form a cluster in each domain. It then elects a Cluster Head (CH) for each cluster (i.e. affinity group) to act as a Domain Name Server (DNS) for its local cluster (i.e. subnets) and neighbour clusters. The CH then advertises its connectivity, members, and domain information to its neighbours and rest of the network.

The routing in CIDR is two level operations i.e. if source node wants to send packets to destination node in another cluster then these packets are routed via cluster-head advertised routes and packets to local destination are routed using the local routing algorithm. For membership management, the CH nodes broadcast the control packets containing the domain membership information in the form of membership digest. It uses Bloom Filter technique to map a member list to a bit vector to reduce the membership verification operation and hence decreases the size of the advertised control packet as compared to conventional control packet which contains a plain member list. To detect domain split, CH nodes send periodic beacons to other CHs. If a CH can't hear beacons from other CH within with in timeout threshold then CIDR considers the domain as partitioned and trigger a new CH election process to elect a new CH within the isolated nodes. Once a new cluster is formed, a unique new AS-ID using pseudo random functions, will be generated for the new-born cluster.

However, it doesn't explain the communication mechanism between heterogeneous routing protocols i.e. how and what transforms the control, the data packets to/from source and destination nodes of different routing protocols and topologies

InterMR (Inter-MANET Routing in Heterogeneous MANETs) (Lee et al. 2010) proposes an inter domain routing protocol to provide interoperation across heterogeneous MANETs. It extends IDRM (C.-K. Chau et al. 2008) by introducing attribute based address scheme and doesn't require Domain Name Server (DNS). It uses the core design concepts of BGP (Rekhter & Li 1995) such as Intra and Inter gateway protocol (i-BGP and e-BGP). But unlike static prefix-based address scheme in BGP, the InterMR uses attribute based address scheme that creates the MANET address from attributes such as symbolic names, packet properties and services of MANET. This attribute scheme provides transparency from split/merge of dynamic MANETS.

The InterMR assumes that the nodes within same MANET directly communicate with each other without InterMR whereas communication across MANET must go through gateways. Hence the gateway is responsible of translating from one routing protocols to another (Rekhter & Li 1995). For routing packets, as shown in Figure 2.12, a node in MANET 'A' can communicate with another node in a neighbouring MANET 'B' by first sending a packet to a gateway in MANET A (i.e. A1), which then passes the packet to a neighbour gateway in MANET B (i.e. B1) and then finally delivers the packet to the destination.

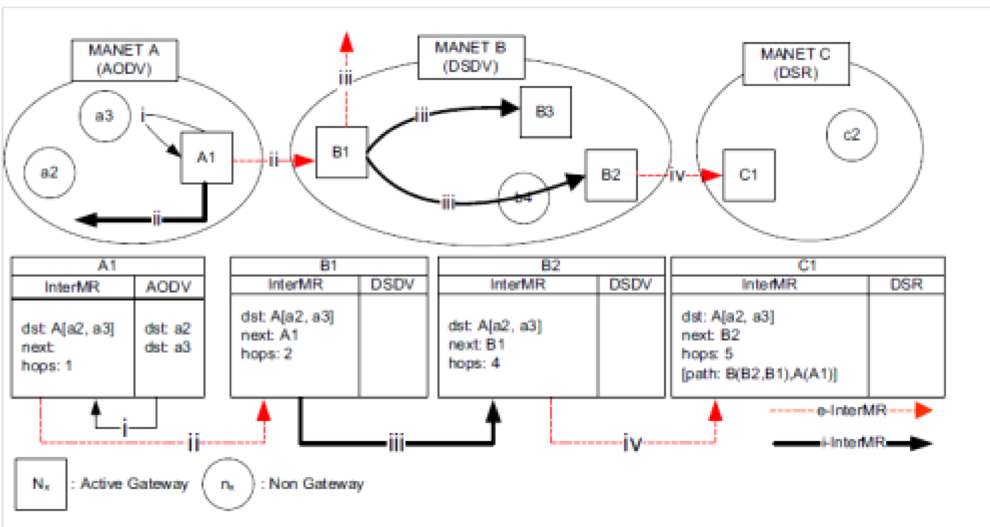


Figure 2.12: InterMR - Propagation of routing information

For inter-domain and intra-domain changes, the InterMR uses periodic update beacons to disseminate information and uses new algorithm to perform dynamic gateway election to reduce computation (Lee et al. 2010). Like IDRM, it uses Bloom Filter for destination resolution to increase the scalability of MANET.

2.3.2 Middleware Frameworks for MANET

Middleware is a software layer that resides between the operating system and applications. It hides the complexities of scalability, load balancing and heterogeneity from the applications (Da Silva & Albini 2014). There has been number of proposed works to create middleware frameworks to address the challenges of routing in MANET. It is assumed that any proposed middleware solution must support the basic requirements of MANET i.e. scalability, dynamic topology and limited resources. This section, apart from the basic MANET middleware requirements, discusses and evaluates the different proposed middle framework approaches and find out their suitability based on following key MANET criteria:

- Heterogeneous of MANET Routing Protocols
- Heterogeneous service discovery protocols
- Lightweight Middleware
- Modularity, Extendibility and Runtime Configuration

In traditional Infrastructure based networks, there are number successfully used middleware framework technologies to address the challenges of heterogeneity, resource discovery and distributed communication. For Example: CORBA, Microsoft COM, Java/RMI and JXTA. However, these middleware frameworks are heavy weight in terms of processing power, memory, network bandwidth and cannot be used where the environments are mobile and network topology changes randomly and frequently (Hadim 2006).

To address the limitations of mobility, heterogeneity, scarce resources and interoperability across networks of different topologies, MANET research groups have proposed several middleware frameworks and are discussed below:

PICA (Calafate & Manzoni 2003) provides multi-platform functionality for threading, packet queue management, socket-event notifications to waiting threads, and network device listing, as well as minimising platform-related differences in socket APIs, and kernel. It provides MANET specific APIs which can be used to developed components in users space e.g. routing protocols. However, these systems are restricted to providing programming abstractions for operating system-level services only and they ignore generic routing protocol commonalities that could be reused across implementations. **ReMMoC** (Grace et al. 2003) allow clients to be developed transparently from the heterogeneous middleware's that may encounter in the future. While suitable for systems that know they will need to interoperate with heterogeneous protocol, this approach cannot solve the problem of two legacy platforms required to interoperate with one another.

Unik-olsrd (Da Silva & Albini 2014) is an implementation of OLSR that supports a plug-in framework. Though these works are proposed for MANET but they have addressed only the basic problems of limited processing and routing in ad hoc networks. However these frameworks do not offer the runtime configuration, interoperability of routing protocols which are the main challenges of hetero MANETs.

Different **Tuple Space** (Gelernter 1985) based representations have been proposed in the past to represent middleware frameworks for MANET. The most widely used technique in this regard is the **LIME** (Linda in a Mobile Environment) (Murphy et al. 2006) middleware for MANET. It extends **Linda** (Gelernter 1985), which provides tuple space data structure for fixed distributed systems. LIME permanently associates each tuple space with a mobile node and hence reduces end-2-end delay of message exchange when a connection is established. Each mobile node has a tuple space called Interface Tuple Space (ITS) and contains the tuples of other available nodes. It uses engagement, disengagement operation upon arrival or departure, respectively, of new mobile node to process its contents received through it's ITS. It uses mobile agents to carry ITS to connected nodes. The tuple spaces of the connected nodes merge to form a federated tuple space. The middleware returns a tuple from any ITS of federated tuple space when mobile agent of a node queries it's ITS. **LIMONE** (Fok et al. 2004) presents a tuple space based middleware for MANET and is an enhancement of LIME middleware. The model is based on individual agents having full control on the distributed transaction it participates with. This is done by making each host maintain an acquaintance list that provides a global view of the operating context and is customizable using admission policies depending on the network dynamics and the application requirements. The main features of LIMONE are context management, explicit data access, reactive programming, neighbour discovery and code mobility or agent migration. This approach copes better with scalability, limited hardware resources and security issues. **MESHMdl** (Herrmann et al. 2007) middleware uses mobile agents for logical mobility and tuple spaces for decoupling applications components in order to address the dynamic mobility and frequent disconnections of mobile ad hoc networks. Due to tuple space the decoupled nodes can be at different locations in order to communicate. The MESHMdl middleware consists of 4 core layers: 1- Generic connection layer provides APIs to discover and connect to neighbour nodes, 2- Event space provides decoupling for ad hoc communication, 3- Agent runtime is responsible for executing and maintaining mobile agents, 4-Interaction manager to provides communication with neighbour nodes.

There are number of **Peer 2 Peer** based middleware frameworks proposed for MANET. The most widely used middleware under this category is the **ExPeerience** (Bisignano et al. 2003) which is based on **JXTA** (Gong 2002) approach and adds various modules needed for MANET. ExPeerience enhances the services offered by JXTA and adds new features like: management of the intermittent connections and multiple interfaces, efficient resource discovery mechanisms and code mobility in order to meet the requirement of MANET. The code mobility allows the middleware to dynamically adapt to situations at run-time. JXME (Carlo & Emiliano De 2007), implements a framework named JXBT (JXME over Bluetooth) (Blundo & Cristofaro 2007), which allows the JXME infrastructure to use Bluetooth as the communication medium. JXBT uses the basic features of JXME i.e. interoperability of binding peer-to-peer system to single infrastructure, platform and programming language independence and ubiquity. JXME main focus is on peer-to-peer ad hoc communication and requires proxy for inter-communication. **JMobiPeer** (Bisignano et al. 2005) is a framework to develop P2P applications for MANET. Its main goal are to provide interoperability with JXTA protocols, overcoming of JXME (proxy) architectural constraints and the provide communication in ad hoc mobile environment. JMobiPeer extends JXTA to overcome JXME proxies limits and constraints to work in MANET. It provides a modular layered architecture with Virtual Messenger Service providing transport and service protocols to manage node communication in the network. Endpoint Service abstracts the physical address of the peer into logical address to provide communication among mobile nodes in an ad hoc manner. In contrast to JXTA, which uses adaptive source-based routing, Endpoint Routing Service of JMobipeer provides multi-hop communication to peers which are not directly communication range. The Service layer manages the advertisements of peer identities and provides higher level services such as pipes management or resources discovery for interaction with peer nodes. **Peer2Me** (Wang et al. 2007) presents a framework to hide the network communication technology and allow developers to create mobile peer-to-peer applications without knowing the complexities of MANETs. It uses J2ME with Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) to develop middleware applications. The Framework design is based on layered architecture to gain modularity and transparency. Its message layer allows exchange of data between peers including Java Objects. The Management layer supports maintenance and communication between nodes in mobile ad hoc networks (MANETs). The Peer2Me runs on Bluetooth device, which uses Master-Slave protocol for communication between peers and is not suitable for MANETs due to its dependency on Master device. To overcome this issue, Peer2Me uses dynamic establishment of Master-Slave connection when two nodes want to communicate. This avoids the dependency from Master Device and hence allows communication in MANETs. **OnehopMANET** (Mojamed & Kolberg 2014) proposes a one hop MANET as a structured P2P over MANET the uses cross-layering with a proactive underlay. Unlike traditional MANETs, One hop MANET uses a P2P overlay that is capable of achieving lookups in a single hop.

Event-Based and Context Oriented communication model provides loosely coupled communication relationship between entities compared to the traditional client & server approach. Several middleware frameworks utilizing the event-based communication model have been proposed, the key one are discussed here. **STEAM** (Scalable Timed Events and Mobility) (Meier & Cahill n.d.) is an event-based middleware that has been designed for wireless area networks utilizing the mobile ad hoc network model. It addresses specific constraints of event-model related to MANETs i.e. middleware components of the event services cannot be located on independent physical machines. The STEAM event service implements an implicit event model that allows consuming entities to subscribe to particular event types rather than at another entity or a mediator, without having to rely on system-wide services to locate entities or mediators, or on intermediate middleware components through which entities interact. STEAM provides event filtering by combining three different types of event filters to address the dynamic aspect of the network topology i.e. Subject filters, Proximity filters and Content filters. STEAM uses the proximity (geographical and functional) group communication model to allow mobile application components to discover each other and therefore communicate**. EMMA** (Epidemic Messaging Middleware for Ad hoc networks) (Musolesi et al. 2006) is based on Java Message Service (JMS) for mobile ad hoc networks and utilizing an event-based model. It modifies the message passing used in JMS and adding an epidemic routing mechanism that facilitate delivery of messages in a MANET environment. As in JMS, EMMA applications use the point to point or publish-subscribe communications pattern. In point to point, applications use queues for asynchronous message exchange between the producer and possible consumers. The optimal location of the queues is determined by a negotiation process that is application dependant, which makes the middleware context aware (Hadim 2006). To allow the hosts that are not within range to receive messages, the asynchronous epidemic routing protocol is used. Each host maintains a buffer of messages created and messages received and messages are dropped if the buffer overflows. As a result the reliability of this protocol increases but it does not guarantee that all messages are delivered.

Component Based (Costa et al. 2007) approach in creating middleware frameworks is very famous because it provides modularity, distributed processing, extendibility, lightweight interfaces and dynamic loading and unloading of resources. **SELMA** (Self-organized Marketplace-based Middleware for Mobile Ad hoc Networks) (Görgen et al. 2004) presents a component based middleware platform for distributed applications in mobile multi-hop ad hoc networks. The middleware uses mobile agents to communicate through 'marketplace' pattern where mobile applications forward data or agents to specific geographical locations called marketplaces. Due to location awareness, the probability of finding resources is high and this pattern performs well in large scale ad hoc networks. The middleware applications send data and agents back and forth between interested devices and marketplaces by using variations of geographic routing and agent transport protocol. SELMA middleware architecture is divided in three main component layers i.e. communication abstraction, agent platform, application and service agents. The communication layer provides mobile devices positioning, wireless communication, and device discovery to achieve communication hardware and positioning system independence.

The agent platform layer provides transport mechanism using agent map computation techniques for agents to communicate between specific marketplaces. The application and service agent's layer provides location based services to applications created in user space. SELMA middleware fits well in an ad hoc scenario by being self-configurable and power aware. It supports hop-to-hop communication and multi-hop communication as well.

Mobile Gaia (Chetan et al. 2005) is based on component based framework approach and decomposes application services into smaller components that can run on a cluster of different heterogonous devices. This saves memory and power as the middleware allows only the required component to be loaded and unloaded to a device based on its role. For routing it uses event based publish-subscribe model. The main focus of the proposed is on power and memory saving in heterogeneous mobile ad hoc networks.

MANETKit (Ramdhany et al. 2009b) proposes a component based framework that supports the development, dynamic (re) configuration of multiple MANET protocols. It allows protocols to be composed, decomposed and hybridised for dynamic reconfiguration which can safely be executed at run-time. The main features of ManetKit are: i) reduce MANET implementation effort, ii) enhance the portability of protocol implementations, iii) facilitate the exploration of protocol optimisation/hybridisation efforts, iv) seamlessly integrate MANET routing in a wider middleware framework and v) dynamic reconfiguration in MANET protocols. ManetKit uses OpenCom (Ramdhany et al. 2009b) as a software component to handle low-abstraction-level systems, e.g. routing systems for mobile ad hoc networks and allows application developers to develop routing protocols in multiple languages to run in user space.

Figure 2.13 shows the ManetKit architectural design that uses OpenCOM as a run-time deployable software components and CFs (component frameworks) to decompose and configure protocol functionality. CFs are sub-components and identify the common functionalities across ad hoc routing protocols and can be implemented in different languages. Manetkit framework comprises of two key sub-CFs: 1) the System CF which encapsulates common system-related functions and 2) the ManetProtocol CF which encapsulates protocol-related functions. The System CF performs event handling and provide the generic operating system interface. It parses generalised packet format into data structures containing protocol messages and act as a generic surrogate to target OS-specific APIs. The ManetProtocol CF accommodates MANET protocol diversity and enables the coverage of diverse ad hoc routing protocol taxonomy.

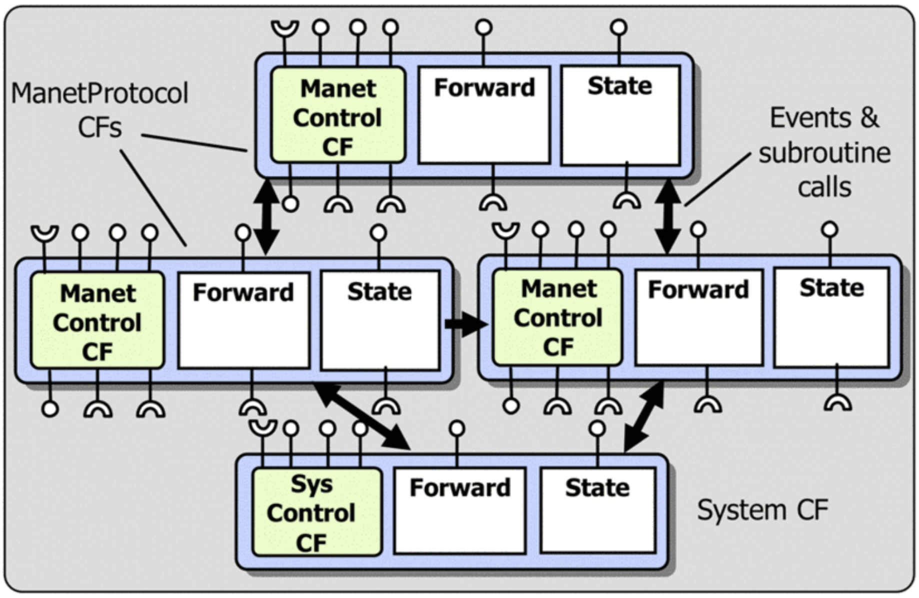


Figure 2.13: The Manetkit framework architecture (Ramdhany et al. 2009b)

I-Jeng Wang et al. (Wang & Jones 2004) proposed a component based framework for MANET routing protocols. The framework groups common set of routing protocol functionalities into different system components. The granularity of (sub) components depends upon the complexity of routing protocols algorithms and is used to build different routing protocols. The high level components of the proposed framework representing the set of common functionality which includes: Route Information Representation, Route Determination/Selection, Packet Forwarding, Neighbour Discovery and Maintenance, Route Information Initialization, Dynamic Route Management, Failure Response, and Route Discovery. The constructed protocol from these components as a whole addresses a unique routing functionality and performance requirement under different environment. The main focus of this work is to characterise existing routing protocols, map them with different components base on their complexities and construct new hybrid routing protocol from these components. However, it doesn't explain how the components interconnect with each other and it would communicate with devices running different type of routing protocols.

WARF (Wireless and Autonomic Routing Framework) (Kukliński 2011) has proposed a component based middleware framework for WMN (Wireless Mesh Networks). The middleware enables component based implementation of different routing protocols and their mutations in IPv6 networks. It is an extension to the component-based framework for analyzing and designing routing protocols for MANET and supports the features i.e. cross-layer operations, multiple radio interfaces, real-time resource monitoring, dynamic resource allocation and multipath adaptive forwarding (Wang & Jones 2004).

As shown in Figure 2.14, the WARF architecture is decomposed into four independent components: Resource Maintenance, Route Maintenance, Data Forwarding, and Policy Control. These components are based on common features of different routing protocols and provide protocol flexibility and simplicity of incremental improvements of protocols. Data forwarding component provides multi route data forwarding and data transfer across different routing protocol devices. Route Maintenance support: Route Discovery, Route(s) Selection, Route Quality Monitoring, Route Representation and Route Fault Detection. The Resource Maintenance component is responsible for the physical layer configuration, i.e., for channel management and monitoring and is composed of two sub-components i.e. Resource State Information (RSI): monitors and disseminates resource information and Resource Control (RC): allocation of resource (channels) in 802.11 WMNs. Policy Control allows nodes configuration and setting control parameters of WARF components. However, WARF requires updating the existing routing protocols in align with WARF Component Model and depends on the IPv6 environment. WARF doesn't explain the communication between heterogeneous MANET routing protocols.

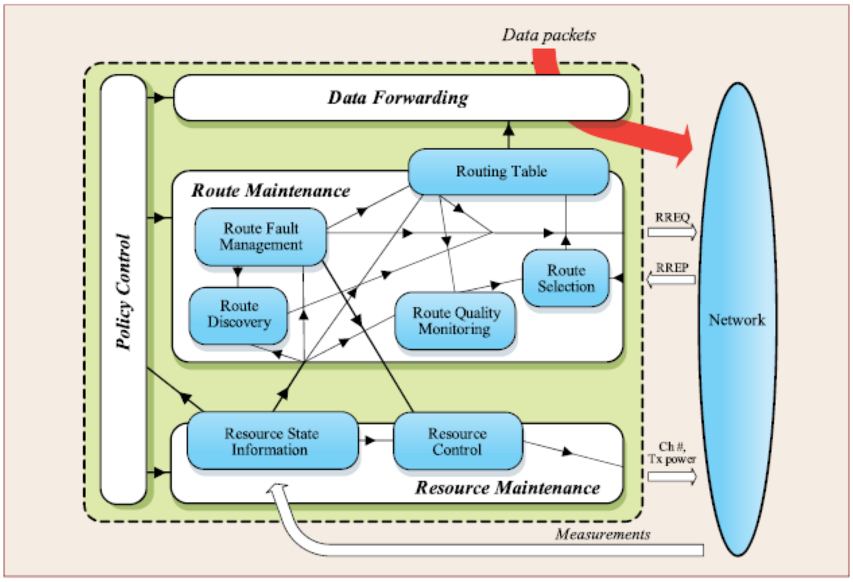


Figure 2.14: The WARF Architecture (Kukliński 2011)

2.3.3 Evaluation of Related Work

We will evaluate each of the proposed research work by concentrating on how well they meet the comparison criteria's listed in 2.3.3.1 to fully address the interoperability in heterogeneous MANETs. The basic requirements of MANET i.e. mobility, limited resources, route request, scalability and power awareness have been excluded from the comparison criteria due to the fact that these mandatory to provide the routing in MANET. Table 2.8 below, presents a comparison matrix to highlight the key features of each propose work and level of compatability with comparison criteria's.

2.3.3.1 Comparison Criteria's

The following criteria's are used to compare the proposed MANET solutions discussed in section 2.3 and are analyzed in Table 2.8:

- Interoperability of Heterogeneous Routing Protocol
- Interoperability Heterogeneous Service Discovery
- Interoperability Data & Control Packet Heterogeneity
- Group Support to reduce E2E transmission delay and network overheads
- Runtime (Re)Configuration
- Autonomous: Each set of functionalities must be independent and modular to easily adjust new routing protocols
- Modular and Lightweight
- Do not Change Existing Protocol (MP): The solution must not modify the existing routing protocols in order to fully provide interoperability with heterogeneous routing protocols.

From Table 2.8 below, it is evident that the solutions presented in the first section (Inter-Domain Routing Approach) have focused on minimizing network traffic and were mainly concerned with providing communication between hybrid routing protocols. They used the cluster based approach to group the same type of routing protocols into separate clusters and elect cluster head to provide communication among them. However these approaches do not provide autonomous routing, framework or API's for extension and mainly interoperability of heterogeneous routing protocols.

The research proposals in second section i.e. Middleware Frameworks Approach for MANET of Table 2.8 have presented middleware frameworks approach to hide the complexities of MANET and provide a seamless platform (middleware) to application developers. However they don't fully provide the mechanism to enable the communicaiton across heterogeneous routing protocols.

Hence, from Table 2.8 we can analyze that that there is no single solution except IF-MANET which has addressed all the challenges of heterogeneity in MANET.

The works proposed in (Fujiwara et al. 2012), (Lee et al. 2010), (Ramdhany et al. 2009a), (Kukliński 2011), (Davoudpour et al. 2014) are similar to our concept of interoperable MANET framework (IF-MANET). However, these approaches are specific to cluster based routing for hybrid protocols, Ontology based framework CANthings (Davoudpour et al. 2014) for standardizing data from heterogeneous objects and middleware platforms for application developers to create new routing protocols. Whereas the IF-MANET uses cluster based inter-domain approach to support the routing in heterogeneous MANET and component based extendable middleware approach to provide seamless homogenous platform to application developers. The details of IF-MANET design approach will be discussed in Chapter 3.

Research Work	Key Features	LR	RP	RT	SD	DH	PH	GS	RC	SC	AM	MP
Inter-Domain Routing Approach for MANET												
ZRP	Zone Based, Hybrid Routing	P	N	N	P	N	N	Y	N	P	N	N
CBRP	Cluster Based, Hierarchical Routing	Y	N	N	P	N	N	Y	N	P	N	N
SHARP	Hybrid Routing, Support Proactive & Reactive routing	P	Y	P	P	N	N	P	N	N	N	N
ATR	Inter Domain Routing, Interoperability across ad hoc networks, Inter and Intra communication using ATR Cluster Heads.	Y	Y	P	N	P	N	Y	P	Y	N	N
IDRM	Inter-Domain Routing, semi-proactive routing for inter-domain communication, Protocol Translation	Y	P	N	P	P	N	Y	N	P	N	N
CIDR	Cluster based, group travel companions into cluster, Independency of underlying routing protocols, periodic beacons to detect domain split	Y	P	N	P	P	N	Y	N	P	N	N
InterMR	Inter Domain Routing, Attribute based	Y	P	P	N	P	N	Y	Y	Y	N	N

	addressing, Inter and Intra Gateway Protocol, Periodic Beacon, Dynamic Gateway election,											
Middleware Frameworks Approach for MANET												
LIME	Data Sharing Middleware, Tuple Spaces System, Extends Linda, Interface Tuple (ITS)	Y	P	N	N	P	Y	N	N	P	N	N
LIMONE	Tuple Space Middleware, Extends LIME, Enhances the scalability	Y	P	N	N	P	Y	N	N	P	N	N
MESHMdl	Object Oriented Tuple Spaces, Mobile Agents, Mobility Aware, Decoupling coordination primitives	Y	P	N	N	P	Y	N	N	Y	N	N
ExPeerience	JXTA, P2P Framework, Code Mobility, Resource Discovery Mechanisms, management services	Y	P	N	N	P	Y	P	Y	Y	Y	N
JMobiPeer	P2P Middleware, Enhances JXTA and ExPeerience, limited resource devices	Y	N	N	P	P	Y	N	Y	Y	P	N
Peer2Me	P2P Middleware, Bluetooth technology	Y	N	N	P	N	Y	P	Y	P	N	N

STEAM	Event Based Middleware, Context aware, Proximity Group Communication, Publish-Subscribe Mechanism, Event Filters	P	P	N	N	N	N	P	N	P	p	N
EMMA	Message Oriented Event Based Middleware, Context aware, JMS, Point to Point, Communication, Pub-Sub Communication, Epidemic Routing	P	P	N	N	N	P	N	N	P	P	N
SELMA	Marketplace pattern, Component Based, Mobile Agents, Home zones, Neighbour discovery, geographical region	P	P	N	N	N	P	P	N	Y	Y	N
Mobile Gaia	Component based Middleware, Active Spaces, Clusters, Publish-Subscribe, “WYNIWYG”, Coordination	F	P	N	N	N	P	N	N	Y	F	N
MANETKit	Component based Framework, Runtime (re) Configuration; Pub-Sub Event based communication, OpenCom for communication, Pluggable functionality to reduce routing protocol implementation efforts.	F	P	N	P	P	F	N	F	F	F	N

Wang et al.	Component based, Dynamic Route Management	F	N	N	N	P	N	N	P	P	F	N
WARF	Component based Middleware, Extending Wang et al., IPv6, multiple radio interfaces, resource maintenance & monitoring	F	P	N	N	P	P	N	P	F	F	N
IF-MANET	Modular Component Based Middleware, Inter-Domain Routing for heterogeneous MANETs, Runtime Dynamic (Re)Configuration, Event Driven Communication, Semantic Match Making, Packet Transformation	F	F	F	F	F	F	F	F	F	F	F

Keys: **LR**: Limited Resources, **RP**: Routing Protocol Interoperability, **RT**: Routing Topologies Interoperability, **SC**: Scalability, **SD**: Service Discovery, **DH**: Data Heterogeneity, **GS**: Group Support, **PH**: Platform Heterogeneity, **RC**: Runtime Configuration, **AM:** Autonomous, **P**: Partial, **F**: Full, **N**: Not Supported

Table 2.8: Comparison matrix of Interoperable MANET routing protocol Approaches

2.4 Summary

The focus of this chapter was on examining the previous research contributions to support routing in mobile ad hoc networks. After providing the general background on different ad hoc routing protocols and middleware frameworks, the chapter has then focused on examining the previous research contributions on interoperability of heterogeneous MANETs and conferred how the work presented previously differs from the one proposed in this thesis. Table 2.8, compares these existing proposed research works against the MANET evaluation criterias in section: 2.3.3.1, which shows that none of these works, except the IF-MANET if addresses, have proposed a complete solution for interoperability of routing protocols in heterogeneous MANETs. The IF-MANET provides a framework which creates a homogeneous layer to hide the complexities of the MANET taxonomies in order to provide interoperability between them (Hamid et al. 2015).

The next chapter explains the design approach of the proposed IF-MANET framework.

Chapter 3

IF-MANET Design Approach

3.1 Introduction

This chapter explains the novel architecture of the proposed IF-MANET Framework. Thereafter, the design and algorithms of the IF-MANET interoperable routing protocol elucidates the Ontology based packet translation to support a route discovery in heterogeneous MANETs.

The IF-MANET Framework provides a component based architecture where each component acts as a service to provide a distinct set of functionalities. Due to the service oriented component model, the IF-MANET is designed to be loosely coupled so that new components, services and algorithms can be easily integrated without affecting the existing functionality. It also enables the integration of new components at runtime and hence new routing protocols can easily be integrated within the framework. For communication between different MANETs, the framework uses inter-domain cluster based approach with the help of IF-MANET Gateways. The Gateway uses Ontology based MTL (Message Translator) to semantically translate the packets from one routing protocol to other and hence enables interoperability between heterogeneous routing protocols. The IF-MANET framework provides the following key features to achieve the interoperability:

- Provides communication between heterogeneous routing protocols
- Provides route discovery across heterogeneous routing protocols
- A special routing table to maintain the protocol type along with other details of reachable nodes to pre-empt data transformation and hence reduces network overheads
- An Inter-domain cluster based routing for internal (Intra domain) and external (inter domain) communication.
- Packet translation from source to destination routing protocols and vice versa.
- Provides communication between different MANET taxonomies.
- Runtime packet conversion to accommodate new arriving protocols.

3.2 Assumptions

In this thesis, the following assumptions are made:

- MANET is defined as a logical grouping of mobile nodes, where all nodes in the same MANET employ the same routing protocol.
- Intra-MANET is defined as a node or collection of nodes within the same MANET e.g. mobile nodes interacting in intra-domain means they belong to same MANET and are interacting with nodes within that MANET.
- Inter-MANET is defined as nodes interacting with external MANET(s).
- Domain, MANET, Cluster defines the same context and are used interchangeable.
- Direct communication between nodes of different MANETs is not allowed except through IF-MANET Gateways.
- The system uses wireless interface 802.11 to communicate with mobile wireless devices

3.3 Contributions

The following are novel contributions of the IF-MANET Framework to achieve the interoperability across heterogeneous MANETs.

- Light Weight Component Based Service Oriented Framework
- Route Discovery in Heterogeneous MANET Taxonomies:
    - Reactive Route Discovery
    - Proactive Route Discovery
    - Heterogeneous Route Discovery
- MANET Ontology and Message Translator
- Special Gateway Routing Table, Abstract Message and Universal Packet
- Initialization and Maintenance Phase

3.4 Context Diagram of IF-MANET Framework

Figure 3.1 shows a context diagram of the IF-MANET framework and its interfaces with external systems. For Example, a node running the Protocol-A sends a route request to discover the destination node running the Protocol-B. When the IF-MANET System (Gateway Node) receives a route request it invokes the Gateway Engine to translate the Protocol-A packet, using the Message Translator into an abstract message. It then conducts a route discovery to find out the destination node and its protocol taxonomy.

The Gateway node in the destination MANET, when receives the packet, converts the abstract message into a packet compatible to the Protocol-B and then sends it over to destination node running Protocol-B. The destination node processes the incoming message and acknowledges with route reply back to the Gateway node. The destination's Gateway node follows the same process, converts the protocol-B into Protocol-A and forward the route reply to Protocol-A. This process of conversion between different types of packets, allows the IF-MANET to enable interoperability across heterogeneous routing protocols.

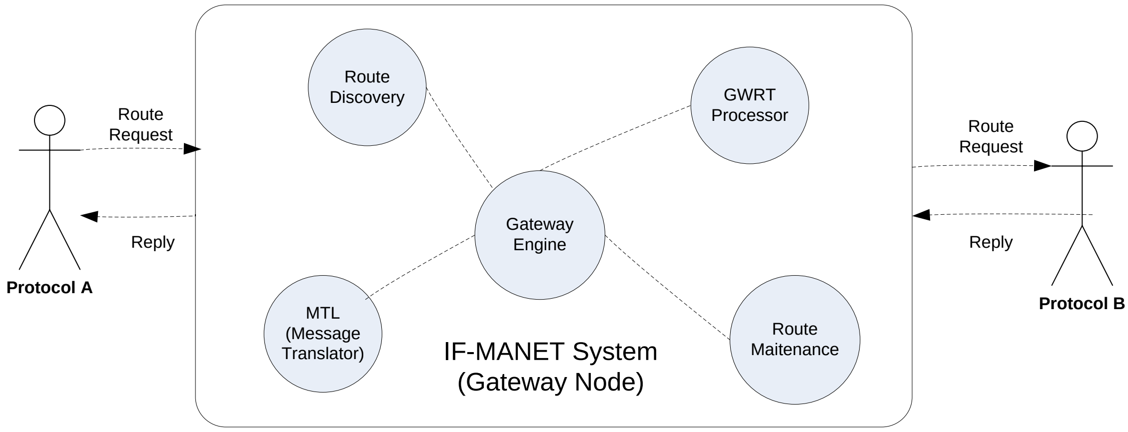


Figure 3.1: Context Diagram of IF-MANET Framework

3.5 High Level Architecture of the IF-MANET Framework

Figure 3.2 presents a high level architecture of the IF-MANET Framework. The architecture groups the related features of the IF-MANET into the self contained modular components. The external applications interact with the system using the API provided by Communication API's layer. The system communicates with the physical layer through the "Wireless Stack" layer. This layer provides different components each implementing different type of wireless protocol stacks e.g. IEEE 802.11, Bluetooth etc. The Message Translator loads the relevant adapter to convert a packet from one protocol to another. The routing engine is a core of the IF-MANET framework and provides the following key capabilities:

- Route Discovery
- Route Maintenance
- Message Translator
- Packet Processing
- Resource Management

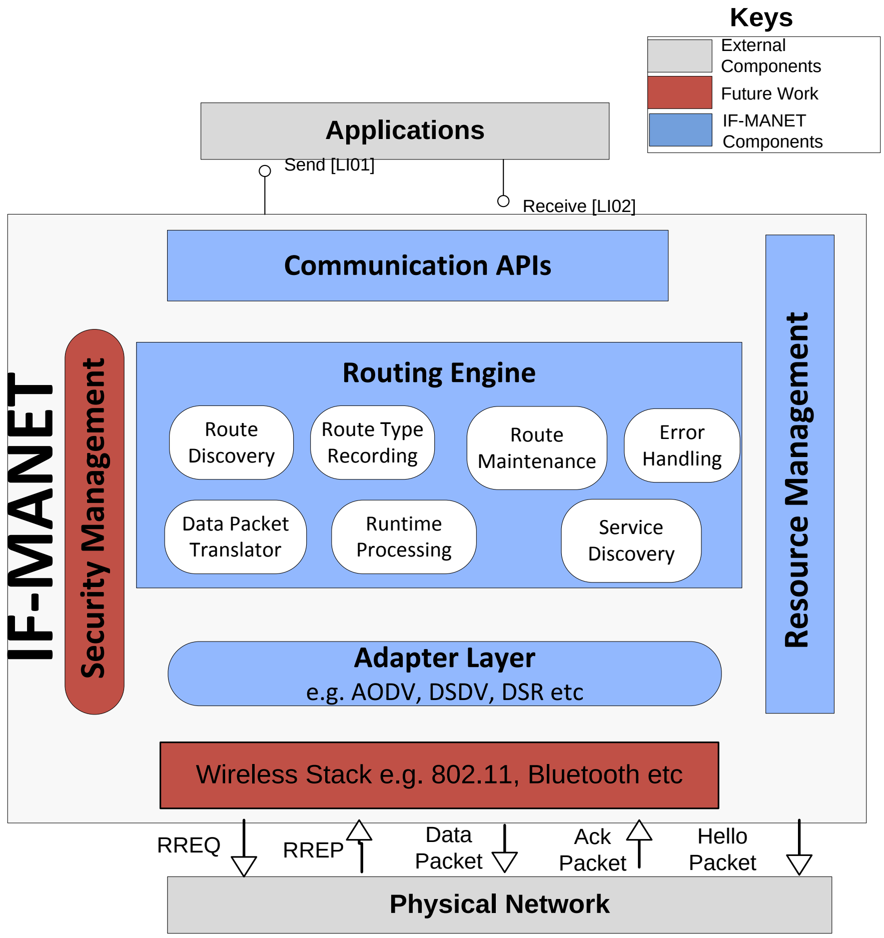


Figure 3.2: High level Architecture of IF-MANET Framework

3.6 Logical Architecture of IF-MANET Framework

The IF-MANET provides a component-based model where each component is designed as a lightweight service and is composed of a distinct set of capabilities. Due to the service oriented model, these components are loosely coupled so that the new components, services and algorithms can be easily be integrated without affecting the existing functionality of the system. It also enables the integration of new components at runtime and hence the new routing protocols can easily be integrated within the framework. Figure 3.3 shows the logical architecture of IF-MANET Framework for Interoperable Routing Protocol. The components of the routing protocol, providing distinct set of functionalities, are explained in the following sub-sections:

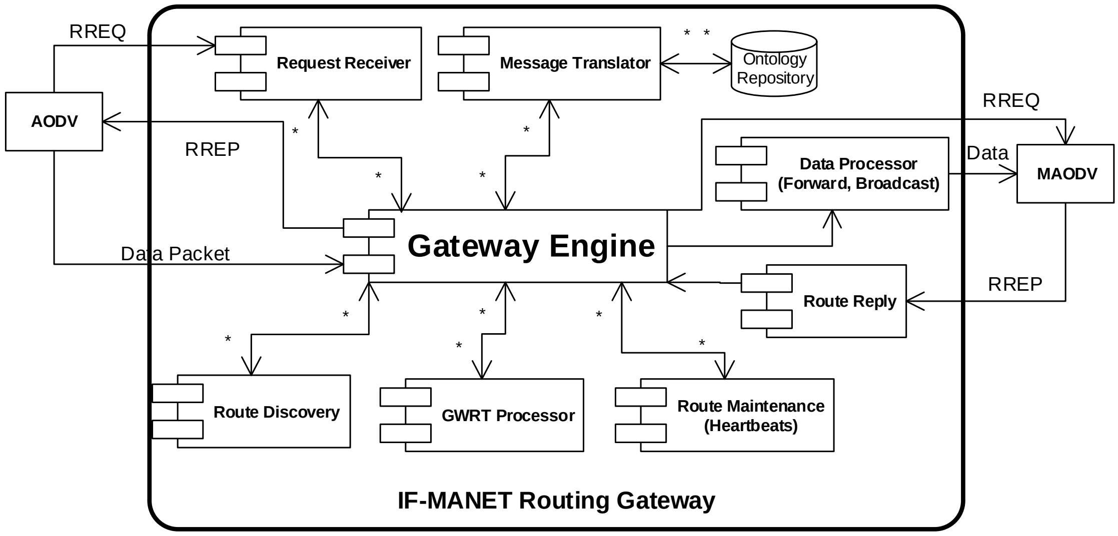


Figure 3.3: Logical Architecture of the IF-MANET Framework

3.6.1 Gateway Engine

This is the core component of the IF-MANET framework and enables communication with different components of the framework for processing the incoming/outgoing requests and data packets. For Example, if a node sends a RREQ and the destination node, running different routing protocol, receives the request, it will not be able to understand the packet format and hence discards all the RREQ packets. To overcome this problem and achieve the interoperability without changing the functionality of existing routing protocols, the Gateway Engine maintains the RREQ counter. When the counter reaches RREQ Threshold (i.e. completes the network wide search), the Gateway assumes that no node has replied because the packet is for external MANET. The Gateway node first searches its special routing table (GWRT) for the type of destination or next hop node. If found then it will load the relevant routing protocol adapter and transform the packet similar to destination routing type, otherwise it will broadcast the RREQ message at different channels. On receiving RREP, the Gateway node will update the GWRT, converts the packet and forwards RREP to source node. For data transfer, the Routing Engine keeps track of source and destination protocols in the GWRT and transforms the data from source to destination type and vice versa.

3.6.2 Route Discovery Service

This component is responsible of discovering a routing path from a source to a destination node. If a node needs to send a packet and if it is a Reactive MANET, then the RDS will first discover the route by broadcasting a RREQ message. For Proactive MANET, the routes are maintained by periodically sending RREQ messages, and RREQ is broadcasted only when route to a destination is stale in the routing table.

3.6.3 Route Reply Service

This component implements the Route Reply algorithms. The node replies back with RREP message if it is a destination node or an intermediate node with a fresh route to the destination.

3.6.4 Message Translator

It will de/en-capsulate the incoming/outgoing message and evaluates the type of interface whether it is a normal node or a Gateway node. It will also check whether the source and destination nodes are of the same type or not in order to transform the routing protocol from a source to a destination routing protocol. It will communicate with the IF-MANET's MANET Ontology to semantically translate the source routing protocol and generates the missing fields of source protocol to model the target routing protocol.

3.6.5 Route Maintenance

Due to dynamic nature of MANET the network topology changes continuously and hence the nodes join and leave different MANET domains. To maintain the cluster heads routing information, the IF-MANET maintains the Gateway Nodes information by broadcasting periodic beacons. Failure to receive a beacon indicates that a Gateway is lost or out of range and hence new Gateway Head will be elected.

3.7 IF-MANET Interoperable Routing Protocol

To support a communication in dynamic MANET environments, the IF-MANET has proposed a novel routing protocol to achieve the interoperability and hence the communication between heterogeneous MANETs. It allows mobile nodes of different MANET taxonomies to interact using Gateway nodes running the IF-MANET routing protocol. The novel contributions of the IF-MANET are explained in the following sub-sections:

- Universal Packet
- Abstract Message
- Gateway Routing Table
- Initialisation Phase

- MANET Ontology and Message Translator
- Route Discovery

3.7.1 Universal Packet

A unique lightweight universal packet has been created for the IF-MANET Interoperable Routing Protocol to exchange the information between its Gateways. The IF-MANET periodically sends Hello beacons (Heartbeats) of the Universal Packet to collect and maintain the information between the Gateways. This feature allows Gateway nodes to Bordercast (Haas & Pearlman 2002) i.e. directly send packets instead of flooding the network with broadcast packets. The Universal Packet contains the fields illustrated in Table 3.1 below:

<table>
<tr><th colspan="32">Universal Packet</th></tr>
<tr><td>0</td><td colspan="3"></td><td>4</td><td colspan="3"></td><td>8</td><td colspan="3"></td><td>12</td><td colspan="3"></td><td>16</td><td colspan="3"></td><td>20</td><td colspan="3"></td><td>24</td><td colspan="2"></td><td colspan="2">28</td><td></td><td colspan="2">31</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="8">Request id</td><td colspan="8">Message Type</td><td colspan="8">MANET Taxonomy</td><td colspan="8">Reserved</td></tr>
<tr><td colspan="32">Source Gateway Node id</td></tr>
<tr><td colspan="32">MANET id</td></tr>
<tr><td colspan="32">Time-to- Live (HHMMSSMSC)</td></tr>
<tr><td colspan="32">Time-Stamp (HHMMSSMSC)</td></tr>
<tr><td colspan="32">Sequence Number</td></tr>
</table>

Table 3.1: IF-MANET Universal Packet

The description with sample value of each field is described below:

- **Request id**: An Id to uniquely identify the request message.
- **Message Type**: It defines the type of message sent and has following types:
    - 01: Hello
    - 02: Route Request
    - 03: Route Response
    - 04: Route Error
- **MANET Taxonomy**: Identify the type of routing protocol used by a source MANET
- **Source Gateway Node id**: Node id of the Gateway who has sent the Universal packet
- **MANET id**: Unique id of a source MANET. It uses distinct MANET attributes to create a unique hash value in an ad hoc network.
- **Time-to-Live**: The expiry time of this route
- **Time-Stamp**: Time at which the Universal Packet was broadcasted

- **Sequence No**: To check the freshness of information received and to avoid loops by discarding the packets with sequence number equals or less than the previously received.

3.7.2 IF-MANET Gateway Routing Table (GWRT)

The IF-MANET has created a special routing table called GWRT to store and associate the routing information, along with the MANET taxonomy, of nodes communicating with external MANETs. The main challenge in communication across different MANET domains is that the nodes with different routing protocols cannot communicate directly with nodes in other domains. Even though, in a cluster based approach, where cluster head nodes communicates on behalf of internal cluster nodes, they require special storage to save the route information to keep track of outgoing packets against incoming packets. It is fundamental to keep track of routing information across the MANET domains in order to provide the interoperability between them.

The IF-MANET GWRT is designed in such a way that it is independent of any specific routing protocol packet format. Each IF-MANET Gateway creates a GWRT and maintains a node routing information while communicating with external MANETs. Table 3.2 below, illustrates the format of the GWRT.

Gateway Routing Table (GWRT)																															
0				4				8				12				16				20				24				28			31
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
GWRT_id				Record_id				Record Type				MANET_id				Node_Type					TTL			Reserved							
0	0	0	1	0	0	0	1	R	R	E	Q	0	0	0	1	N	O	D	E	0	1	2	0								
Field_Name																															
Field_Value																															
Time Stamp (HHMMSSMSC)																															

Table 3.2: IF-MANET Gateway Routing Table (GWRT)

3.7.2.1 Description of IF-MANET GWRT Fields

The fields of GWRT are explained in Table 3.3 below:

Field Name	**Description**
Header Fields	
GWRT_id	Unique identifier for every GWRT row
Record_id	Unique id for a record where a record consists of multiple rows
Record_Type	Type of record saved e.g. Hello, RREQ, RREP, RRER etc
MANET_id	Unique MANET id of Next Hop or Destination MANET. The id represents the taxonomy of destination MANET domain e.g. Reactive, Proactive etc
Node_Type	Type of node communicating with IF-MANE Gateway e.g. N for node and G for Gateway
TTL (Time To Live)	How long the record is valid for in seconds
Reserved	Reserved for future use
Body Fields	
Field_Name	Name of field e.g. Destination Address, Sequence No etc
Field_Value	Value of field name e.g. 192.168.0.1, 01234 etc
Time Stamp	Date Time when record was saved

Table 3.3 Description of GWRT fields

3.7.2.2 Example of GWRT

Table 3.4 shows a record of GWRT entity when a local MANET node, running the AODV protocol, sends a message of type RREQ to a gateway node. Let us assume that, the originator node “Node-A1” wants to send a message to the destination node “Node-B3” and is broadcasting a Route Request (RREQ) message.

Field Name	Value
Header Fields & Values	
GWRT_id	0001
Record_id	0011
Record_Type	RREQ
MANET_id	0123456
Node_Type	N
Time_To_Live	120
Body Fields & Values	
Protocol Type	AODV
Source Address	Node-A1
Destination Address	Node-B3
Sequence No	0012
RREQ id	0002
Next Hop	GW-A3
Hops	1
Timestamp	2014-05-05 10:00:00

Table 3.4: Sample record of GWRT

3.7.3 Abstract Message

Mobile nodes in different MANET domains running different routing protocols have no or minimal knowledge about cross-domain routing messages. In addition, different routing protocols have different packet formats and data fields. In spite of the fact that different routing protocols will have the same purpose and meaning of the data but due to different format, field names and value types, they cannot understand each other.

To address these issues, the IF-MANET has proposed an Abstract Message which is independent of the type of routing protocols. The size of an abstract message depends on the packet size of the routing protocol in communication. It will only store the information mandatory for the packet translation in order to reduce the memory footprint and hence processing power and battery consumption. The IF-MANET node converts the domain specific routing protocol to the abstract message using IF-MANET's MTL and sends over to different MANETs. The destination node then converts the Abstract Message, by using MTL, to its local domain specific routing protocol. The format of the Abstract message is given in Table 3.5 below.

<table>
<tr><th colspan="32">Abstract Message</th></tr>
<tr><td colspan="4">0</td><td colspan="4">4</td><td colspan="4">8</td><td colspan="4">12</td><td colspan="4">16</td><td colspan="4">20</td><td colspan="4">24</td><td colspan="3">28</td><td>31</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="8">Record id</td><td colspan="4">Message Type</td><td colspan="4">MANET id</td><td colspan="4">MANET Taxonomy</td><td colspan="4">Gateway id</td><td colspan="8">Reserved</td></tr>
<tr><td colspan="32">Field id</td></tr>
<tr><td colspan="32">Field_Name</td></tr>
<tr><td colspan="32">Field_Value</td></tr>
<tr><td colspan="32">Onto_Field id</td></tr>
<tr><td colspan="32">Time To Live (HHMMSSMSC)</td></tr>
<tr><td colspan="32">Time Stamp (HHMMSSMSC)</td></tr>
</table>

Table 3.5: IF-MANET's Abstract Message

3.7.3.1 Description of Abstract Message Fields

The description of abstract message fields is given below:

Field Name	Description
Record id	Unique id of the message and is associated with the cached message in IF-MANET Routing Table
Message Type	Specifies the type of message e.g route request, reply, data, error
MANET Taxonomy	Type of MANET e.g. Proactive, Reactive etc
MANET id	Unique id of source MANET i.e. the MANET sending the message
Gateway id	Unique Address of the IF-MANET Gateway node
Field_id	Uniquely identify the record in the system
Field_Name	Name of protocol feature e.g. protocol name, source Address etc
Field_Value	Value(s) of the Field Name e.g. AODV, 192.168.0.1
TimeToLive	Expiry time of the message e.g.120 sec
TimeStamp	Date and Time when record is created e.g. 10:10:00
Onto-Field_Id	Mapping id of the record to the Ontology vocabulary id.

Table 3.6: Description of Abstract Message Fields

3.7.3.2 Example of Abstract Message

A sample record of the Abstract Message created by a source MANET Gateway running the AODV routing protocol is shown in Table 3.7. The originator node, for example, “Node-A1” wants to send a packet to destination node “Node-B3” and is broadcasting a route request (RREQ) message. The GW node translates the source native message into Abstract message using MTL leveraging the MANET Ontology.

Field Name	Value
Header Fields and Values	
Record id	00001
Message Type	Route Request (RREQ)
MANET Taxonomy	Reactive Routing
MANET id	0123456
Gateway id	1234561
TimeToLive	120
Timestamp	10:10:00
Operational Fields and Values	
Originator Node Address	Node-A1
Destination Node Address	Node-B3
Sequence No	0012
RREQ id	00012
Field_id	0005
Hop Count	02

Table 3.7: Sample Record of Abstract Message

3.7.4 Message Translator (MTL)

The communication between heterogeneous routing protocols requires conversion from one protocol to another. To connect MANETs of different routing protocols, when encounter dynamically, must understand one another and exchange data. Here, the fundamental challenge is to understand the packet level message format and behaviour of the routing protocols in order to generate the compatibility. The packet format, its data types and meaning of different routing protocols, even though they belong to the same routing taxonomy, is different. Table 3.8, compares the packet format of different routing protocols under different routing taxonomies.

MANET Routing Protocol Packet Format		
Reactive Routing Taxonomy		Proactive Routing Taxonomy
AODV	DSR	OLSR
RREQ id	Identification	N/A
N/A	Payload Length	Packet Length
Sequence No	N/A	Packet Sequence Number
Destination Address	Target Addresses (Add-1, Add-2...)	N/A
Originator Address	Source Address	Initiator Address

Table 3.8: Comparison of Routing Protocols Packet Format

By comparing the packets of reactive routing protocols i.e. AODV, DSR, the common denominator between them is source address, destination address and unique request identification. In addition, due to a reactive nature, both protocols send and receive RREQ, RREP messages. In spite of the commonalities, the field names and behaviour of the data are different between these protocol packets. For Example, In AODV each node saves the path of a next hop (node) whereas, in DSR every node stores information of all nodes along the route path in the Target Addresses field. By comparing the packet format and their behaviour, it is evident that their fields cannot be mapped directly.

On the other hand, by comparing the packet of reactive routing protocol (e.g. AODV) with that of proactive routing packet (e.g. OLSR), it is clear that both the protocols have significantly different behaviour of communication and hence the packets formats. Unlike AODV the OLSR (proactive protocol) maintains a routing table with information of all its neighbour nodes by periodically exchanging the route information. Due to the difference in behaviour and packet formats, AODV and OLSR protocols cannot be directly mapped and hence cannot communicate with each other.

To map the packets and behaviour of different routing protocols, one solution is to create an adapter for every routing protocol which will provide a static mapping between different packet types at compile time. The downside of this technique is to create an adapter for every combination of protocols e.g.

AODV → DSR, DSR → AODV

AODV →OLSR, OLSR → AODV

Also, it requires creating new adapters forever for new arriving routing protocols. The other challenge is dynamic nature of MANET where mobile nodes spontaneously encounter one another on-the-fly and hence requires conversion of messages at runtime rather than at compile time.

To provide a dynamic transformation between different routing protocols there is a need to automatically learn the context and behaviour of routing protocols at run time and then generate a dynamic bridge between them. The IF-MANET has proposed a unique MTL and MANET Ontology to provide a semantic based mapping between packets of different types at runtime. The run time conversion allows translation of newly arrived routing protocols on-the-fly.

3.7.4.1 Architecture of Message Translator

Figure 3.4 shows the architecture of IF-MANET Message Translator. The source and destination MANETs uses different taxonomies and have their own Ontologies. The architecture of the MTL is divided into following main sub-components:

- **Message Interoperability Component**: It is dedicated to the interpretation of protocol packets to/from MANET systems. They receive the messages from different routing protocols, parses the messages, applies domain ontology to map packet structure and transforms the message into the IF-MANET Abstract Message
- **Behavioural Interoperability Component**: It intervene the interacting protocols running by the different MANETs and translates the behaviour to fill the missing fields from one protocol to other using semantic matching.
- **Domain Ontology**: It maintains the MANTE domain ontology, learn/discover new ontology's and provide matching/mapping different MANET protocols.

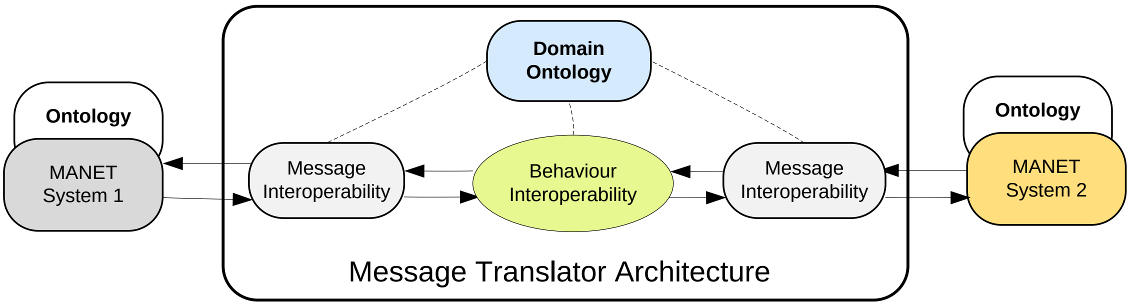


Figure 3.4: Architecture of the IF-MANET Message Translator

In order to interpret incoming packets, MTL read those incoming packets and extract their field labels at run time. These field labels are then cached in the IF-MANET GW Routing Table (GWRT). MTL then loads the IF-MANET ontology, the field names from GWRT at run time, applies semantic matching between them and creates a new abstract message based on these values. Following equations shows the flow of packet conversion from source to destination protocol and return back.

Native Protocol → Source Gateway → MTL → Abstract Msg → Destination Gateway

Destination Gateway Received → Abstract Msg → MTL → Convert to Native Protocol

The proposed IF-MANET Ontology and how the MTL semantically map the heterogeneous routing protocols is explained in the following sub-sections.

3.7.4.2 Proposed IF-MANET Ontology

MANETs have different taxonomies and each of them can only understand the packet formats it has defined for its routing protocol. Hence, if heterogeneous MANETs intend to interact with each other, they must be able to translate the fields of incoming packets to the target MANET system. To address this challenge and provide interoperability, IF-MANET has defined a MANET Ontology to create a vocabulary of different routing protocols and rules to semantically map their packet fields. The Ontology provides the flexibility to classify the new incoming routing protocol packets under the relevant routing taxonomy updates the vocabulary and enable the packet to interoperate with the existing MANET systems.

Figure 3.5 below, illustrates the proposed MANET Ontology which plays the key role in achieving the interoperability across heterogeneous MANET taxonomies. The Ontology uses the comparison of routing protocols, provided in the Section 2.1.4, to identify the similarities and differences between protocols and classify them into the common functional groups. It then uses this classification to semantically match the similar features and uses behavioural reasoning to map the dissimilar features. The packets of different routing protocols are classified into the following functional groups:

- **MANET Taxonomies**: It classifies the behaviour of different routing protocols e.g. AODV uses Reactive Approach for communication whereas OLSR uses Proactive approach.
- **Operations**: Different protocols perform different operations for route discovery and interaction with mobile nodes e.g. AODV uses RREQ, RREP to discover route where proactive protocols like OLSR uses route advertisement to maintain route to neighbour nodes.
- **Routing Protocols**: The routing protocols which mobile nodes uses for multi-hop communication in Mobile Ad hoc networks. Each protocol has different packet format and operation behaviour for communication. Each protocol belongs to different operation groups which in turn belong to MANET taxonomy.

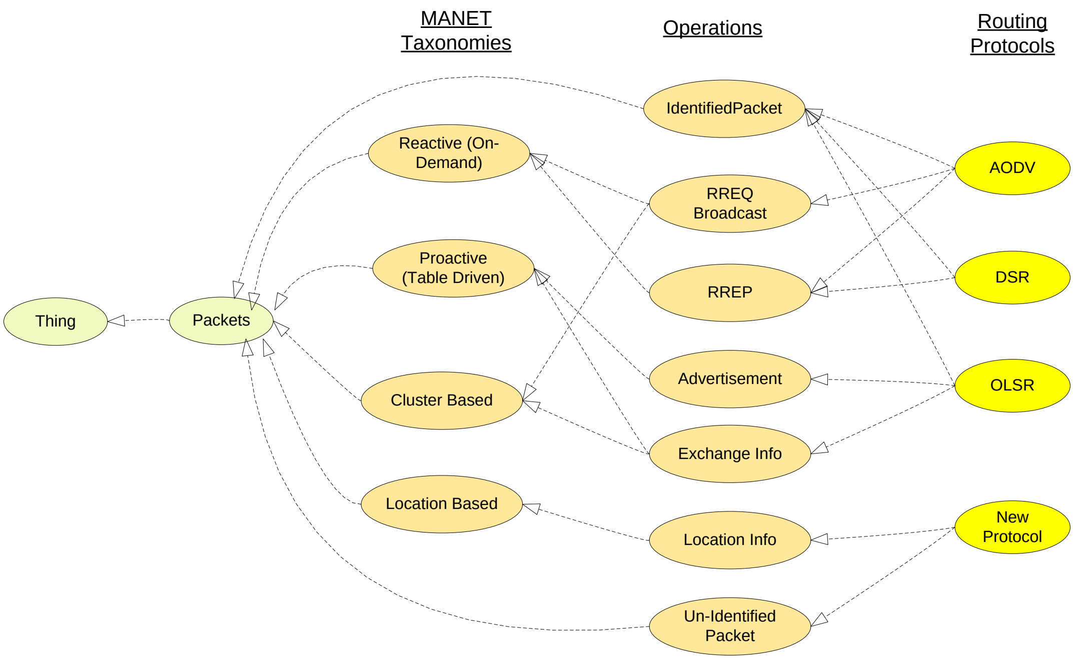


Figure 3.5: IF-MANET Ontology for MANET Routing Protocols

The Ontology transforms the heterogeneous MANET protocols from one system to another by using the following three phases:

3.7.4.2.1 Discovery Phase

This phase defines the MANET domain within the Ontology. The Ontology gives semantic meaning to different routing taxonomies applicable to MANETs along with definition of known packet formats classified under relevant routing strategy. This classification of packet plays a key role in comparing packets belonging to different routing taxonomies. If the incoming routing protocol (packet) does not belong to any routing taxonomy defined within the Ontology then the system learns new packet by classifying its concept and stores in the vocabulary of Ontology.

3.7.4.2.2 Comparison Phase

This phase provides comparison between two routing protocols with the help of semantic rules defined within the Ontology. The rules compare the packets of two routing protocols and produces similarities and differences between them. These similarities, differences are significant in determining the possibility of mapping from one routing protocol to another.

3.7.4.2.3 Modelling Phase

This phase provides mapping technique to convert the packet of source protocol to that of destination. The differences in two packet formats, found in Matching Phase, are analysed in this phase to determine how to generate the missing fields in destination packet. It uses the rules defined in the Ontology and generates the missing fields to bridge the gap between two different routing protocols.

Figure 3.6, shows the context diagram depicting the above mentioned Ontology phases to convert a routing protocol A into routing protocol B.

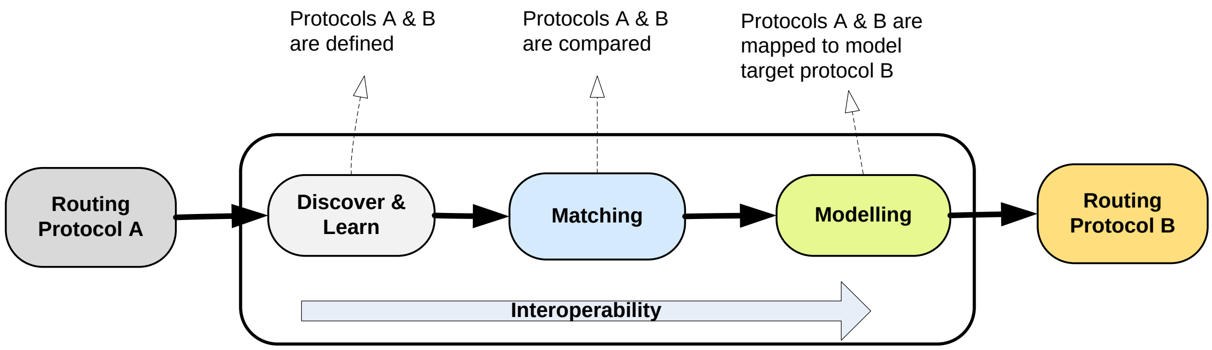


Figure 3.6: MTL Ontology Interoperability Phases

Figure 3.7 shows a sequential flow of the MTL process using Ontology to convert a packet from one routing protocol to another. The IF-MANET node on receipt of a source packet P1 queries the Ontology repository and finds out, whether the repository contains the source packet P1 fields or not. If the fields are not found then the packet is classified as Un-identified packet and is sent over to Discovery Phase to define the packet format, its classification and learn the new packet by adding its fields and rules into the repository of Ontology. If the packet is identified, then the “Match” component compares the fields of source and destination and find out the similarities and differences between them. These similarities and differences are then passed over to “Modelling” Component to map the fields. For different fields the modelling component loads the rules from Ontology and generates the missing fields. It will then map the fields and create a target packet P2.

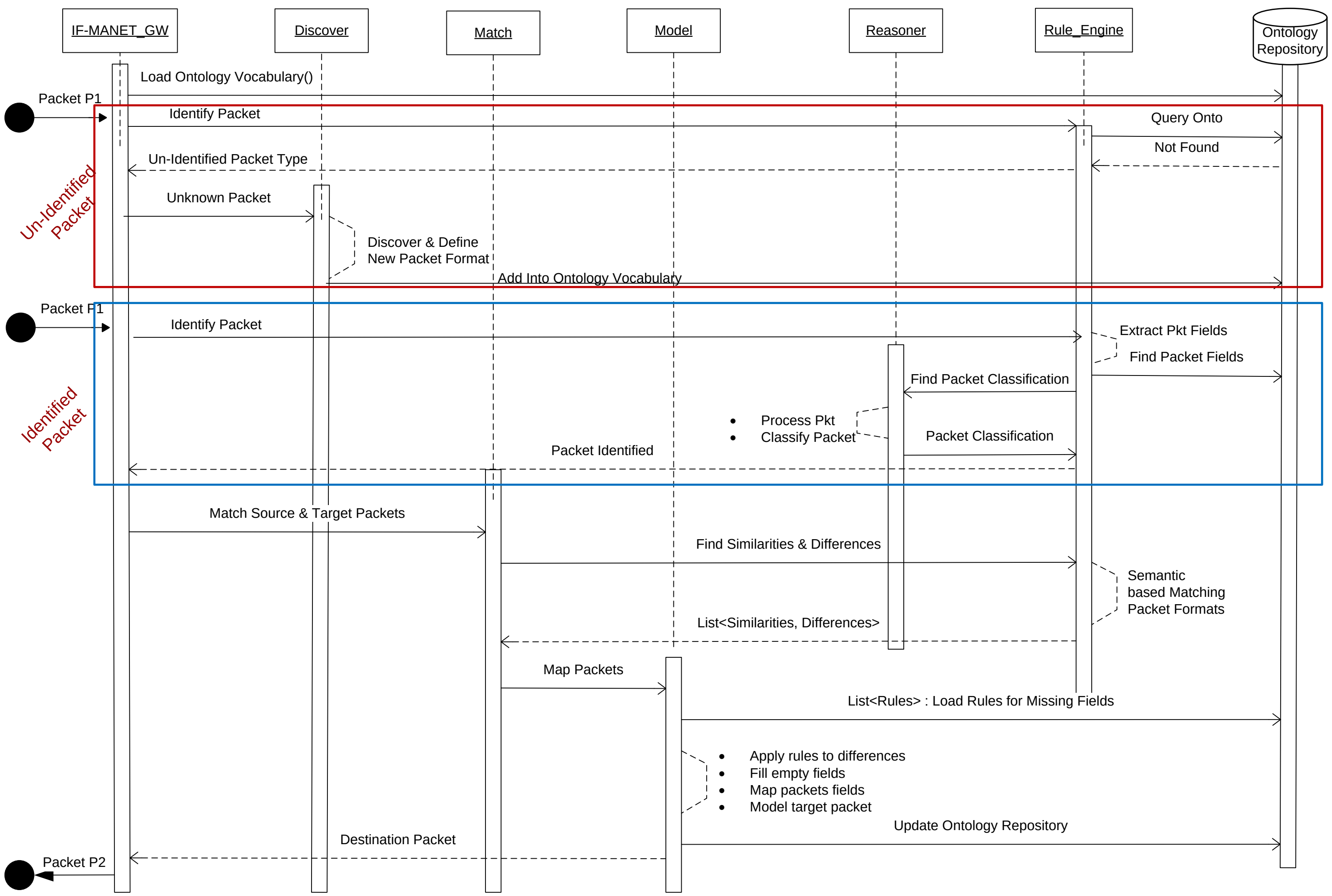


Figure 3.7: Sequential Diagram showing MTL Packet Conversion Process

3.7.4.3 Algorithm of IF-MANET Ontology Based Packet Translation

Algorithm 3.1 shows the pseudo code describing how the Ontology works in MANET. During semantic matching, the Ontology is used for understanding the meaning of data. When ontology is not available, the discovery component is invoked to define the ontology for the new protocol and learn by adding the packet into the existing Ontology repository. Once the Ontology is discovered, matching and reasoning components are used to analyse the semantics of data. The matching component compares the fields of different packets and find out similarities and differences between them. The modelling component uses reasoning and rules from the Ontology to generate the missing fields found in differences and map the packets to create a target packet.

```
Proc: Startup
        Global Ontology = LoadOntologyVocabulary() // Load existing Ontology

Proc: ReceivePacket(Packet Pkt) // Assume P1 is source and P2 is target Packet

        List<Fields> fields = Extract_Fields(Pkt) // Extract fields of P1 packet

        If ( Not Ontology.Contains(fields) ) Then
                Pkt_Class = Un-Identified        // New packet not in Ontology vocabulary
                Proc: Discover (Pkt) // Call Discover function to learn new packet
        Else
                Pkt_Class = Identified   // Packet fields found in Ontology vocabulary
        End-If

        List<Fields, Fields> matched_fields = Proc: Match (source_packet, Pkt_Class,
        target_packet)
        // Match source and target packet fields to find out similarities and differences

        Packet target_packet = Proc: Model (matched_fields) // Model (create) target packet by
        mapping matched (similar, different) fields

        Return target_packet
```

```
Proc: Discover (Packet pkt) // Define and learn packets

        format = DefinePacketFormat(Pkt) // Define packet format and rules
        List<Fields> fields = Extract_Fields(Pkt)
        Ontology.Add(fields, format) // Learn new packet by adding into Ontology

// Match the fields of source and target packets to find out the similarities and differences
Proc: Match (Packet P1, Packet P2) // Assume P1 is source and P2 is target Packet

        List<Fields> P1_Fields = Extract_Fields (P1)
        List<Fields> P2_Fields = Extract_Fields (P2)

        List<Fields> similarFields = P1.hasFields(P2_Fields).SelectDistinct()
        List<Fields> differenceFields = P2.dontHaveFields(P1_Fields).SelectDistinct()
        List<Fields, Fields> match_fields = (similarFields, differenceFields)

        Return match_fields

// From matched fields: apply rules to create missing fields of target packet
// Map source and target fields to create the target packet
Proc: Model (List<Fields, Fields> matched_fields, string sourcePktType, string targetPktType)

        Rules rules = Ontology.LoadRules(matched_fields.MissingFields(),targetPktType)
        List<Fields> missing_fields = Reasoner.GenerateMissingFields(rules)
        Packet target_packet = Map.SourceToTargetFields(matched_fields, missing_fields)

        Return target_packet

End // End of Algorithm to translate packets
```

Algorithm 3.1: MTL Using Ontology for Packet Translation

3.7.4.4 Example of MTL using Ontology for Packet Translation

This example is based on the Ontology explained in Section: 3.7.4.2 and addresses the interoperability problem of different message formats. The main challenge in exchanging the messages is difference in packet formats and the way they are formulated. Here, IF-MANET uses Ontologies to match the message formats of different routing protocols and semantically map their differences to provide interoperability between them. For this, IF-MANET MTL defines a MANET Ontology containing vocabulary of the various routing protocols and defines their rules.

To explain the system, the example uses two different reactive routing protocols i.e. AODV and DSR. The MTL uses Ontology to interpret the packets and convert them from one protocol to another. Table 3.9 shows a comparison between the packet fields and data types of AODV and DSR.

AODV		DSR	
Route Request Packet Fields			
Field Name	**Data Type**	**Field Name**	**Data Type**
Type	String	Option Type	String
Hop Count	Int	N/A	
RREQ id	Int	Identification	Int
Destination Address	String	Destination Address	String
Destination Sequence No	Int	N/A	
Originator Address	String	Source Address	String
Originator Sequence Number	Int	N/A	
N/A		Opt Data Len	Int
N/A		Hop Limit (TTL)	Int
N/A		Target Addresses Address[1]....Address[n]	Struct:List

Table 3.9 : Comparison between AODV and DSR Packets

To establish a path from node A to node B (A → B), node A broadcasts a RREQ (Route Request) message. When a Gateway node receives a request, it translates the AODV RREQ into IF-MANET abstract message and sends over to MANET running DSR routing protocol. The Gateway node in DSR MANET converts the abstract message into DSR Route Request. The process of interoperability will be explained in following sub-section with the help of Ontology phases described above.

Phase-1: This phase defines the MANET Ontology and creates vocabulary of different routing protocols along with their packet formats. It then verifies the incoming packet and then classifies it accordingly. For instance, AODV protocol broadcasts a RREQ packet and the packet fields are found in Ontology. Then the MTL classifies it under an Identified Packet and RREQ Packet as shown in Figure 3.5 above. The requirements for RREQ packet are Source Address, destination number and an identifier. These fields form a part of the RREQ packet format and hence the protocol is classified as AODV RREQ Packet. Similarly it will find the DSR packet in the Ontology and classifies as DSR RREQ Packet. If the fields of incoming packet are not found in the Ontology then it will be classified as Un-Identified packet and its packet fields along with their rules are added in vocabulary of the Ontology.

Phase-2: This phase dynamically compares the packets of both the routing protocols to find out the similarities and differences between them. From Table 3.9, it is clear that there is no direct mapping between the packet formats of AODV and DSR and hence requires rule-based reasoning to enable matching. The Algorithm as shown in Algorithm 3.1, finds out the differences between AODV and DSR RREQ Packets i.e. missing fields from AODV RREQ packet to function as DSR RREQ Packet. The following differences have been found between these two protocols:

- Sequence numbers of both the protocols have different field names and sizes. AODV uses a sequence number for the freshness of route whereas DSR relies on concatenated node address to destination.
- AODV uses a hop count to represent the number of hops from destination whereas DSR uses concerted ids to calculate the hop number
- DSR requires Target Addresses field to store all the addresses along the route to destination whereas AODV stores only next hop (destination) address.

Phase-3: This phase enables the translation of one protocol to another. It takes the differences found in previous step, applies requirements for missing fields to generate them and provide mapping to convert source protocol into destination protocol.

For Example, MTL applies following rules to determine the missing fields in AODV packet i.e. Target addresses, hop number and id to convert into DSR protocol.

- MTL stores AODV sequence number against a unique request number in GWRT and associates it with DSR route request.
- MTL generates the hop number for DSR, stores in GWRT and associate it with the AODV sequence number. E.g. the Gateway node (GW) communicates with DSR node then the GW node acts as the source node and adds hop number equals one in the DSR packet.
- Initially the GW node communicates with DSR node, so it will add its address into DSR target addresses. On receipt of RREP (reply) from DSR node, containing

multiple addresses into its target addresses, the GW node stores them against unique id in GWRT for relating them in subsequent RREQs.

3.7.5 Initialization Phase

In mobile ad hoc networks, route discovery is a complex mechanism and to achieve it in an effective and efficient manner each Gateway node should have information about the Intra and Inter Gateway nodes. The purpose of this phase is that each IF-MANET Gateway maintains the reachability information of all the active and passive Gateways within and outside the MANET Domain. The Initialization Phase utilizes the IF-MANET Universal Packet and GWRT to discover and maintain the IF-MANET Gateways.

3.7.5.1 Explanation of Initialization Process

When IF-MANET Gateway node powers on, it broadcasts the Universal Packet at a configurable periodic time. At first, it queries the MANET-id from other Gateways and if does not exist then it creates a unique id for its MANET domain. It then creates a unique Gateway id and calculates a Gateway Rank (Weight) using the IF-MANET formula, described in Algorithm 3.2 and stores in its global variable for future use. The Gateway node then creates a Universal Packet and set its Message Type = 01 (Hello Message), MANET id, GW id, GW Rank, Request id, Sequence no and TTL value as shown in Table 3.10. The TTL is an expiry time of the packet and Request-id along with MANET-id uniquely identifies the packet. The purpose of the MANET-id is to distinguish between the internal and external MANETs i.e. if MANET-id of sender and receiver are same then they belong to the same MANET otherwise of different MANETs.

When Gateway node exchanges the Universal Packet with a Gateway node in a same MANET (i.e. MANET ids are same), the receiving node compares its Gateway Rank with the one received. If it is less than the received GW Rank then the receiving GW will change its status to passive GW otherwise an active Master GW. It will then initiate the IF-MANET Master Gateway Election Process to find out the Master Gateway within the MANET and inform all reachable GW nodes in other MANETs. The initialization phase will complete when all the active and Master Gateway nodes are identified in all MANETs. The receiving Gateway nodes will create a new record or update an existing one, in their routing tables from the Universal Packet as shown in Table 3.11.

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	GW Weight	Seq. No
01234	01	AODV	19201234	1920123401	262	0001

Table 3.10: Example of Universal Packet at Initialization Phase

Record id	Request-id	Message Type	Src. MANET id	Src. GW id	GW Ranking	Status	Seq. No
01	01234	01	19201234-0001	1920123401	00123	P	0001

Table 3.11: Example of Gateway Routing Table (GWRT) at initialization Phase

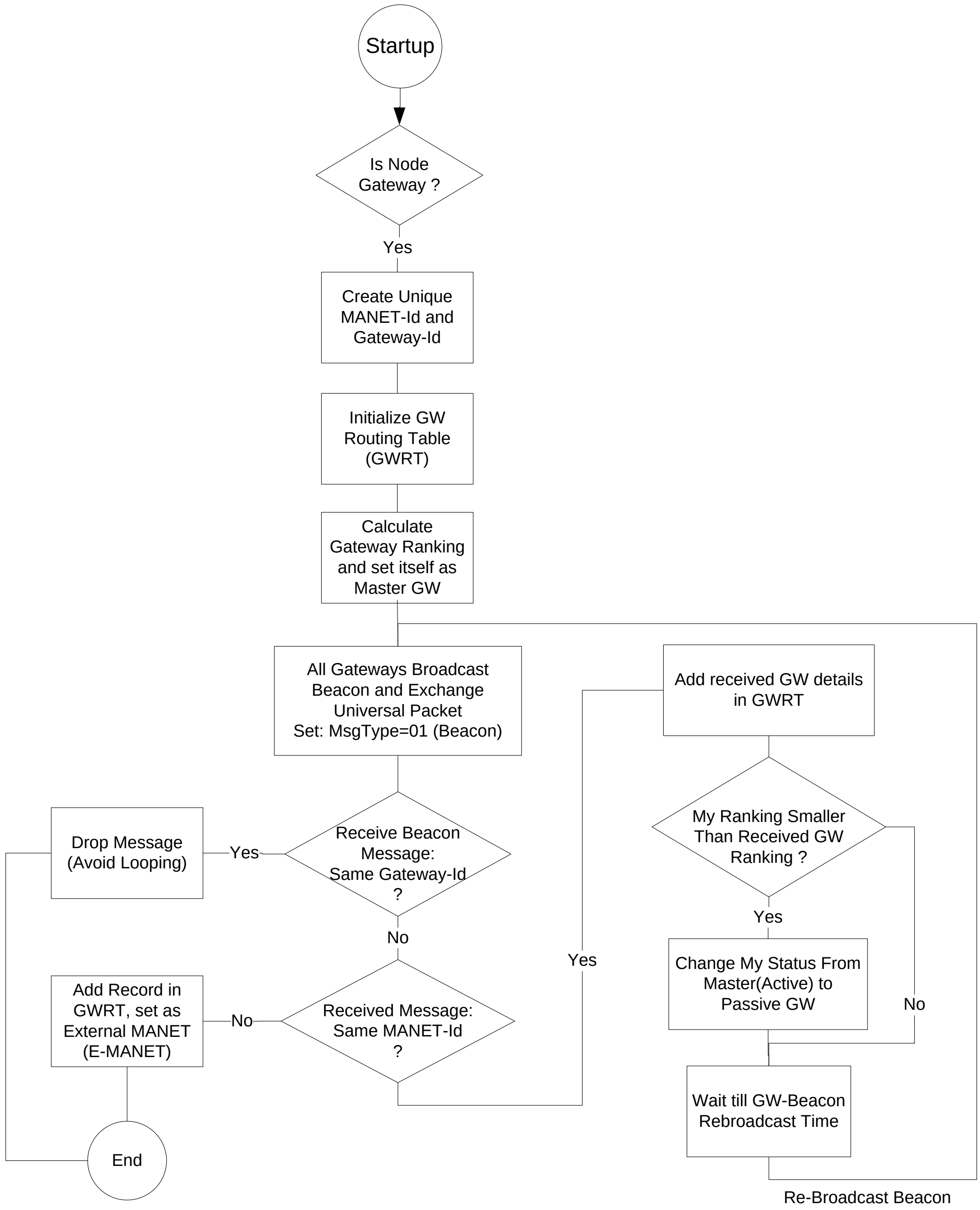


Figure 3.8: Workflow of Initialisation Phase

Proc: Initialize and Send Beacon Message

```
Generate a new MANET id and Gateway id  //using IF-MANET mechanism
Calculate Gateway Weight   // using IF-MANET Ranking Formula
Create IF-MANET Universal Packet
Set: Message Type= 01 (Hello) and populate Universal Packet
Create record in IF-MANET GWRT
Set GWRT Status = M // Master Gateway
Broadcast IF-MANET Universal Packet
```

Proc: Receive Beacon Message

```
Check SeqNo, MANET_id, GW_id
If(GW_id = this.GW_id)
    Drop Packet // to avoid circular looping
Else If (Record No And Seq. No already received) // Found in GWRT
    Drop Packet // Avoid duplicate messages
Else
    Update GW Routing Table // Create new record
End If
If (My. GW Ranking <= Received GW Ranking)
    Update GW status = P // Passive
Else
    Update GWRT: Set GW Status = M // Master
End If
```

Proc: Calculate Gateway Ranking

GW Rank = Node Transmission Range (m) + Power + Connectivity Metric
e.g. Rank = 250 + 10 + 3 = 263

// where Connectivity Metric = No of Connected External MANETs (e-MANET)

Algorithm 3.2: Gateway Node Initialisation Process

3.7.6 Route Discovery

Route Discovery provides a core functionality of the IF-MANET Routing Protocol. It is initiated by a source node to start a communication with a node that is not in its routing table or belongs to another MANET domain.

To start the process, the source node initiates a route discovery in a local domain by sending a route request message as defined by the routing protocol in use. If the receiving node is a destination or an intermediate node with a fresh route to the destination then it will uni-cast a route reply to the source node as defined by the specification of the local routing protocol. If the destination belongs to another MANET domain then the source will not be able to find the destination especially in heterogeneous MANETs where different MANET domains have different routing taxonomies (e.g. reactive, proactive) or the same taxonomy but different routing protocols (e.g. AODV, DSR of reactive taxonomy). In these types of MANETs, if the source and destination nodes are in different networks then the route request messages sent by the source node are not received by the destination nodes. As a result, the route between the source and destination node is not constructed.

To achieve the interoperability between heterogeneous MANETs, The IF-MANET provides a novel route discovery. It uses the special Gateway node, GWRT, abstract message, message translator and Bordercasting technique to provide an effective and efficient communication between different MANETs.

Unlike other proposed solutions, discussed in section 2.3, which modifies the behaviour of original routing protocols to trigger gateway nodes by sending extra data signals, our solution does not changes the behaviour of original routing protocols. It is because of the fact that, it is not practical to change other user's application before communicating with them and or change all existing as well as newly arriving routing protocols. The IF-MANET Gateways uses a unique mechanism to address this challenge by introducing a Route Discovery Retry_Threshold_Counter. The IF-MANET Gateway maintains and increments a threshold counter for every route discovery request made by a source node. The threshold counter will be different for different source node request_id e.g. if there are 3 different source nodes requesting a route discovery, the IF-MANET gateway will maintain three separate threshold counters for whole lifespan of the route request id. Thus, if a source node does not receives a route reply and keep on sending route discovery messages, the Gateway node will assume, after its Threshold Counter equals to the Route Request Retry Threshold, that the destination node is outside the MANET Domain. The Active Gateway Node will take charge, converts the route discovery packet into an abstract message using IF-MANET Message Translator and then forwards the abstract message to all external IF-MANET Gateways using Bordercasting technique. The route to the Gateway Nodes is determined from the Gateway Routing Table (GWRT) which was built in the Initialization phase. The Bordercasting technique reduces the path determination complexity as well as network overheads. The Gateway Node, of receiving MANET, converts the route request abstract message to local routing protocol and finds the route to the destination. If it finds the route within its local routing domain then it generates a Route Reply Packet and sends to the source Gateway Node otherwise it will generate an abstract message of route discovery packet and Bordercast to its neighbouring MANET domains.

In heterogeneous environments, the source and destination MANETs can be running different routing taxonomies. This thesis has categorised these routing taxonomies in the following three route discovery techniques:

- Route Discovery in Reactive Routing MANETs
- Route Discovery for Proactive Routing MANETs
- Route Discovery in Heterogeneous Routing MANETs

3.7.6.1 Route Discovery in Reactive Routing MANETs

If source and destination nodes exist in different MANETs and both MANETs are running reactive routing taxonomy but different routing protocols, the source node still cannot communicate with the destination node. Firstly, because the nodes within the MANET cannot interact with nodes in external MANETs and secondly, different MANETs they are running different reactive routing protocols. To address this challenge, IF-MANET uses a reactive routing approach to enable communication between heterogeneous Reactive Routing MANETs. The following sub-section explains the behaviour of IF-MANET Reactive Route Discovery.

3.7.6.1.1 IF-MANET Reactive Route Discovery by Example

Figure 3.9 shows a context diagram whereas Figure 3.10, Figure 3.11 explains the logical flow of the IF-MANET reactive route discovery process between heterogeneous MANETs. Let us assume that the all nodes in MANET-A are running Reactive Routing Protocol e.g. AODV whereas MAODV in MANET-B. As per assumptions in Section: 3.2, nodes of different MANETs cannot communicate directly and due to the reason nodes in MANET-A cannot communicate with the nodes in MANET-B without using the IF-MANET Gateways.

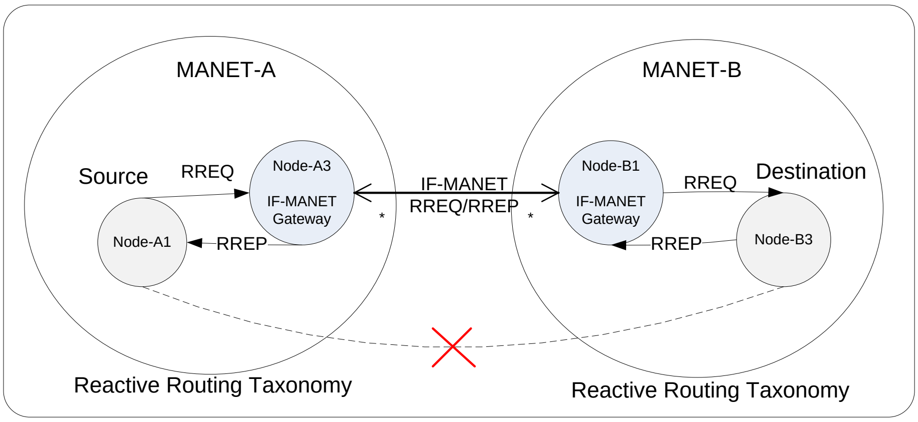


Figure 3.9: Context Diagram of Route Discovery in Reactive MANETs

Let us assume that the nodes in this example are using the addresses as shown in Table 3.12 below:

Node Name	Address (id)
Node-A1	00
Node-A3	01
Node-B1	02
Node-B3	03

Table 3.12: Node addressees for Route Discovery in Reactive MANETs

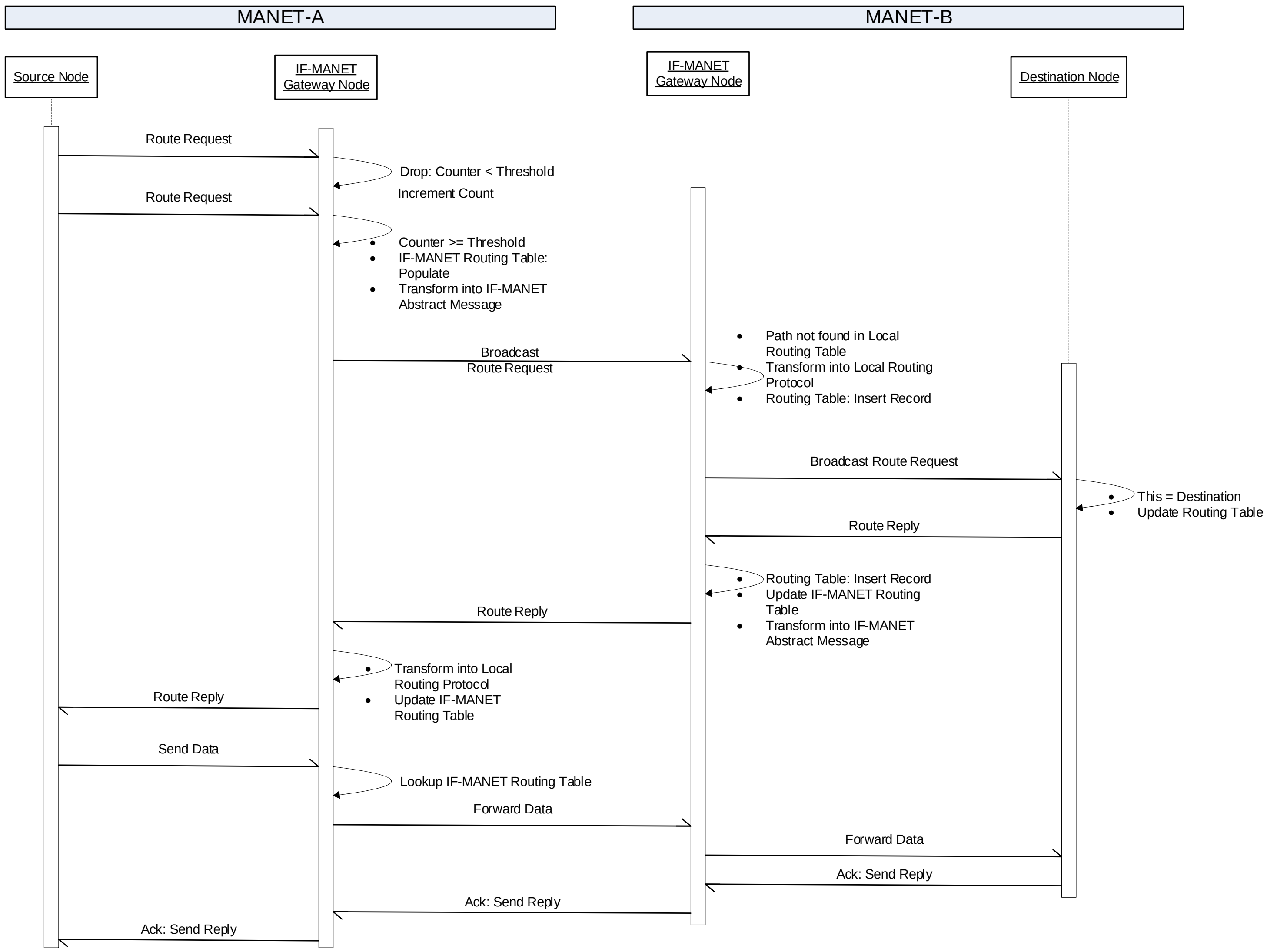


Figure 3.10: Sequence Diagram Showing Route Discovery in Reactive MANETs

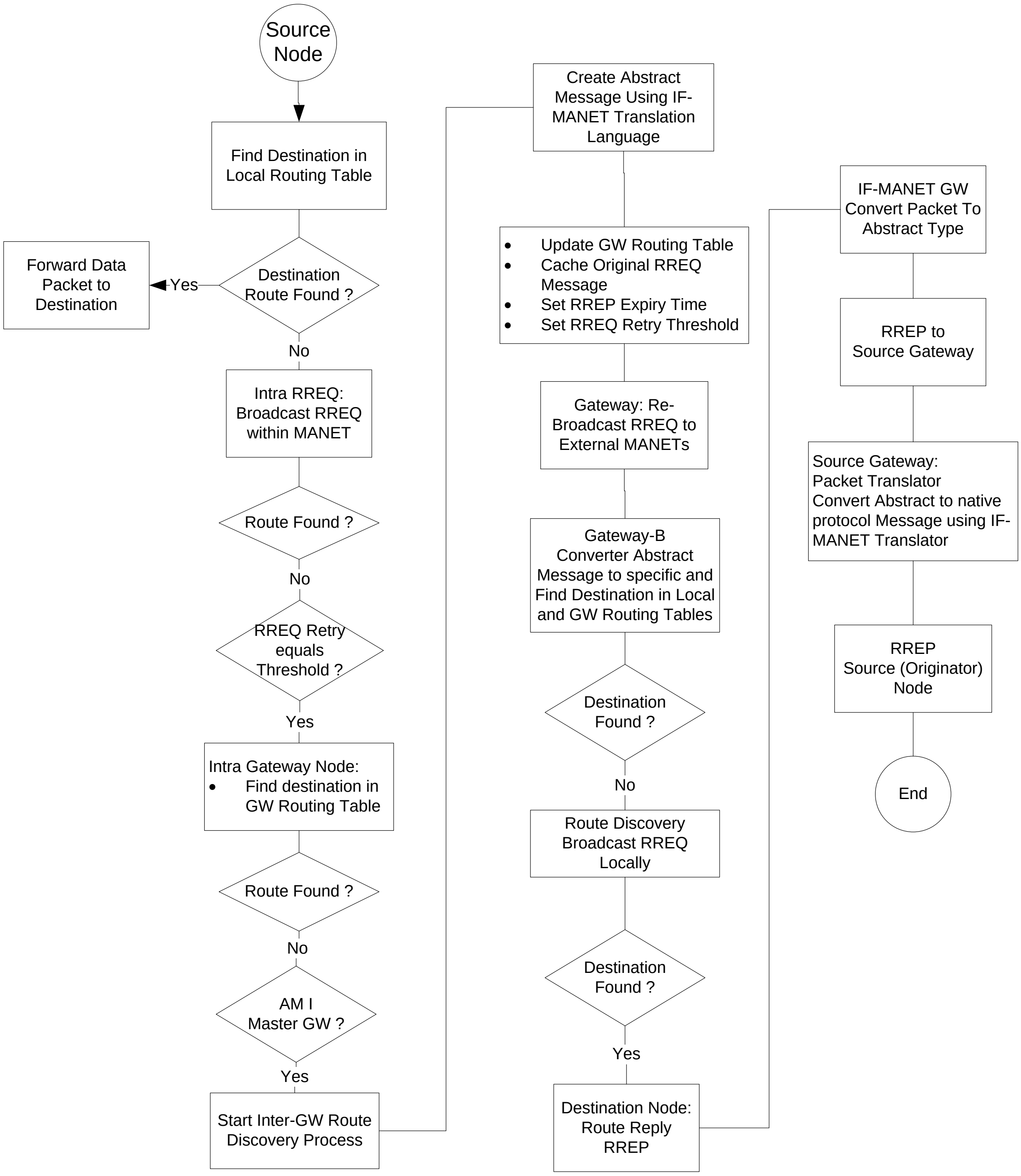


Figure 3.11: Logical Flow of Route Discovery in Reactive MANETs

For Example, if a source “Node-A1” wants to send a message to a destination “Node-B3” and cannot find a route in its local routing table then it will initiate a route discovery by broadcasting a RREQ message. The source node periodically sends RREQ messages until it receives a Route Reply (RREP) message or RREQ Retry Threshold value reaches a configurable network wide search limit. The Source Node-A1 will not receive a RREP message because the destination node belongs to a different MANET and therefore its route discovery request counter reaches RREQ Retry Threshold. The IF-MANET Gateways of the source MANET, increments their GW RREQ Retry Counter, for every RREQ received from the Source Node. When the RREQ Retry Counter of the Gateway node reaches a RREQ Retry Threshold (network wide search) the IF-MANET Gateways will assume that the destination node belongs to an external MANET. The active master IF-MANET Gateway will then take the lead, save sender’s node information in GW routing table as shown in Table 3.13. The Gateway node will add the following two records into the GWRT:

- Route from the Gateway Node to the Source Node. Here the originator is a node that has initiated the RREQ, Gateway Node will become the Source Node, Destination Node from this records point of view is the node to whom Gateway will reply back i.e. originator node (Node-A1), Next Hop is the node to whom Gateway will communicate to reach the destination node (Node-A1). Here, the next hop is the Gateway node itself as there is no intermediately node between Gateway and source node. Hop is the number of nodes between the source and destination and its value is one as the destination requires only 1 hop to reach to the destination.
- Route from the Gateway node to a destination node. Here the Originator and Source nodes are same but route to the destination is unknown. The Gateway node is discovering the route to destination on behalf of the originator node. Here, the destination route details are empty because the IF-MANET has not yet discovered the route. The Hops has been assigned a dummy value of 255 i.e. an unreached destination.

Request-id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
01234	0 (A1)	1(A3)	19201234-0001	0 (A1)	19201234-0001	0 (A1)	1	60	0001
01234	0 (A1)	1 (A3)	19201234-0001	3 (B3)			255	60	0001

Table 3.13: Source Gateway Routing Table (GWRT) after RREQ sent

The Gateway node then converts the route request packet into an abstract message using IF-MANET Message Translator as shown in Table 3.14. The Message Type “02” represents the Route Request message, MANET Taxonomy is the senders MANET type i.e. AODV, Source Address is the address of the sender node i.e. Gateway Node (Node-A3) in this example and destination address is the target node to be reached i.e. Node-B3 here. Here we can see that the source address is a RREQ Sender Gateway address not the originator, it is because the Gateway Nodes will keep track of the forward and backward routes into its GWRT to reduce network overheads and size of the abstract message.

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	Orig. Address	Source Address	Dest. Address	Seq. No	TTL
01234	02 (RREQ)	Reactive (AODV)	19201234-0001	1920123401	0 (A1)	1 (A3)	3 (B3)	0001	001030

Table 3.14: RREQ Abstract Message Sent by Source GW Node

The Gateway then uses Bordercasting technique to directly send the abstract message to all the external MANET gateway nodes instead of broadcasting to all nodes. The Bordercasting will significantly reduce the network overhead and increases the overall performance of the IF-MANET protocol.

Let us assume that, the IF-MANET Gateway (Node-B1) receives the route request (abstract message) from the source Gateway (Node-A3). It will convert the abstract message into the local MANET routing protocol and create/update its GWRT as shown in Table 3.15. Then it will search the destination node in its local routing table as well as in GWRT (for external MANETs). If it will find the destination or an intermediate node with a fresh route to the destination in any of its routing table then it will update its GWRT as shown in Table 3.16. If the destination (Node-B3) is one hop then it will update the relevant record (2nd record) with node id “3” (Node-B3) as a next hop and “1” as a hop count. If the Gateway is not a direct neighbour of the destination node then it will add the address of an intermediate node as its Next Hop node.

Request-id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
01234	0 (A1)	2 (B1)	19201234-0002	1 (A3)	19201234-0001	1 (A3)	1	60	0001
01234	0 (A1)	2 (B1)	19201234-0002	3 (B3)	19201234-0002		255	60	0001

Table 3.15: Receiving Gateway Routing Table (GWRT) after RREQ received

Request-id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
01234	0 (A1)	2 (B1)	19201234-0002	1 (A3)	19201234-0001	1 (A3)	1	60	0001
01234	0 (A1)	2 (B1)	19201234-0002	3 (B3)	19201234-0002	3 (B3)	1	60	0001

Table 3.16: Receiving Gateway Routing Table (GWRT) before RREP Sent

If the destination is found then the Gateway will create a route reply abstract message using IF-MANET message translator as shown in Table 3.17. After that, it will uni-cast a RREP to the source Gateway (Node-A3). If the destination is not found in GWRT of MANET-B, then the Gateway node will broadcast the route request to the local MANET. If the Gateway node doesn't receive a route reply (RREP) until the RREQ Retry Threshold has reached then it will Bordercast RREQ to all the Gateway Nodes. If it will receive a RREP from a destination node or an intermediate node with a fresh route then the Gateway Node will update its routing table, creates RREP abstract message and uni-cast reply back to the source Gateway node i.e. "Node-A3.

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	Source Address	Dest. Address	Seq. No	TTL
01234	03 (RREP)	Reactive (MAODV)	19201234-0002	1920123402	2 (B1)	3 (B3)	0001	001030

Table 3.17: RREP Abstract Message Sent by Destination GW Node

The source Gateway node (Node-A3), on receipt of RREP abstract message will convert it into the local routing protocol, update its GWRT as shown in Table 3.18, creates a RREP packet and send it directly (forward) to the source node (originator of RREQ) i.e. Node-A1.

Request-id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
01234	0 (A1)	1 (A3)	19201234-0001	0 (A1)	19201234-0001	0 (A1)	1	60	0001
01234	0 (A1)	1 (A3)	19201234-0001	3 (B3)	19201234-0002	2 (B1)	1	60	0001

Table 3.18: Source Gateway Routing Table (GWRT) after RREP Received

The source node “Node-A1” will update its local routing table and uses this route to forward the data packet to destination Node-B3. The Gateway node on receipt of data packet will find the type of next hop or destination node from its GWRT, transform the data packet into an abstract message and forward directly to external Gateway which will in-turn forward data packet to destination node. Hence, the process will seamlessly provide interoperability between heterogeneous reactive routing protocols.

3.7.6.1.2 Algorithm of Route Discovery in Proactive MANETs

Algorithm 3.3 below, presents the pseudo code of IF-MANET algorithm for route discovery in Reactive Routing MANETs.

```
Proc: GW Receive Route Request
    IF-MANET Gateway received route request
    If (Sender Not Gateway Node) Then
        If (Destination Address Found in Native Routing Table) Then
            Route Reply by using Native Routing Protocol
        Else If (Destination Found in IF-MANET GWRT) Then
            Route Reply destination path from GWRT
        End
        If (Sender and Gateway have different MANET Type) Then
            Drop Packet // Nodes of different MANETs can't communicate
        End
        If (This Gateway RREQ Retry Counter >= Retry Threshold) Then
            Invoke Proc: IF-MANET GW-Bordercasting ()
            Initialize RREQ Retry Counter
        Else
            Increment RREQ Retry Counter and Release Packet
        End
    ELSE
            Invoke: Gateway Routing Proc (param: Gateway_id, MANET_id)
```

```
        End
// Gateway Operation to Broadcast directly to GWs
Proc: IF-MANET GW Bordercasting
        If (Master/Active Gateway) Then
                // Confirmation that I am Master and have processed the message
                Broadcast to Gateways within MANET
        Else (Wait Till GW_Active_Confirmation_Expiry_Time)
                If(Wait time expired) Then
                        Proc: GW Process RREQ Message
                //other Master/Active Gateway has processed the message
                Else If (Confirmation received) Then
                        Drop Packet
        End
Proc: GW Process RREQ Message
        If (Destination Not Found in GWRT) Then
                Create Abstract Message using IF-MANET Message Translator
                Broadcast Abstract Message to all MANETs
                Update this node Gateway Routing Table (GWRT)
                Cache original route request message
                SET RREP expiry time, Increment GW Counter (GW_Counter)
                Wait for Response
        Else
                RREP destination route path from GWRT
        End

Proc: Gateway Receive External MANET Message
        If (MANET ID Not Equals This.MANET.ID) Then // check to avoid looping
                Convert Abstract Message to Local Routing Protocol
```

Search in Local and GWRT of the Gateway Node

If(Destination Found) Then

Send RREP

Else

Create Entry in GWRT and cache original message received

If(Route to destination found) Then

Call Proc: GW RREP Process ()

End

Broadcast RREQ locally and wait for response

End

Proc: RREP Process

Route to destination found

Local Gateway: Update local routing table and GWRT

Convert RREP message to Abstract Message

Find External Source Path from GWRT

Send (Unicast) RREP abstract message to source MANET GW

Proc: Source MANET Receives RREP

Convert Abstract Message to local/native routing protocol

Update Local native routing table

Update GWRT

Load cached RREQ Message and update with route/path found details

Forward (Unicast) to Intra source node of RREQ

Source Node add entry into Routing Table

Node directly send data packet to destination using the route discovered

Algorithm 3.3: Route Discovery in Reactive MANETs

3.7.6.2 Route Discovery for Proactive Routing MANETs

In proactive routing scheme, each node maintains a route record of every node in the network to send data without initiating a route discovery. But if a node wants to send a packet to a node in other network, they cannot communicate with each other. Firstly, because the nodes do not have route record of external network nodes and secondly different MANETs might be running different proactive routing protocols e.g. DSR, OLSR etc. To address this challenge, IF-MANET uses the Proactive Route Discovery Approach to enable communication between heterogeneous Proactive Routing MANETs. The unique algorithm of the proposed solution is described in Algorithm 3.4 and the logical flow is explained in Figure 3.13. The following sections explain the process of IF-MANET proactive route discovery with the help of example.

For route discovery in proactive routing MANETs, the IF-MANET Gateways periodically broadcasts a Gateway_Advertise_Packet, after a configurable gateway time period i.e. GW_Advertise_Period. The Gateway nodes within the transmission range receives the advertisement and if any of them do not have a route to the sender Gateway then they will create a route entry in their GWRT for the advertising Gateway. The Gateways also exchange their local routing table information and save them against their MANET-id and Gateway-id. The Gateways who already have a route to other gateways will update their route entries. The process continues until all Gateway nodes are synchronised and have fresh route of other Gateways.

To handle duplicated broadcasted message and avoid circular looping, a Gateway when receives a RREQ message, checks whether it has already received a message with same MANET-id, RREQ-id and Source Address. If it has already received the same advertisement then the Gateway node discards the newly received RREQ message. Unlike, traditional Proactive Routing Protocol who exchanges their complete routing tables, the IF-MANET Gateways stores only external MANET Gateway id and their local node ids against their MANET-id and Gateway-id in GWRT. Thus, if a node wants to send a data to other network and cannot find its route in its local routing table, then there are maximum chances that the destination node will be in GWRT. This technique will not only avoid overloading the network traffic but also decreases end to end packet delivery time and hence increases overall performance of the system.

3.7.6.2.1 IF-MANET Proactive Route Discovery Process by Example

Figure 3.12 shows a high level context diagram of mobile nodes communicating across different Proactive MANETs using IF-MANET Gateways whereas Figure 3.13 explains the logical flow of the IF-MANET proactive route discovery process.

This section explains how the route discovery process takes place when nodes in different MANETs need to communicate with each other. From Figure 3.12, we assume that all nodes in 'MANET-A' are running DSDV whereas OLSR in MANET-B. Also, nodes of different MANETs cannot communicate directly due to differing types of routing protocols i.e. nodes in MANET-A cannot communicate with the nodes in MANET-B except by using IF-MANET Gateways.

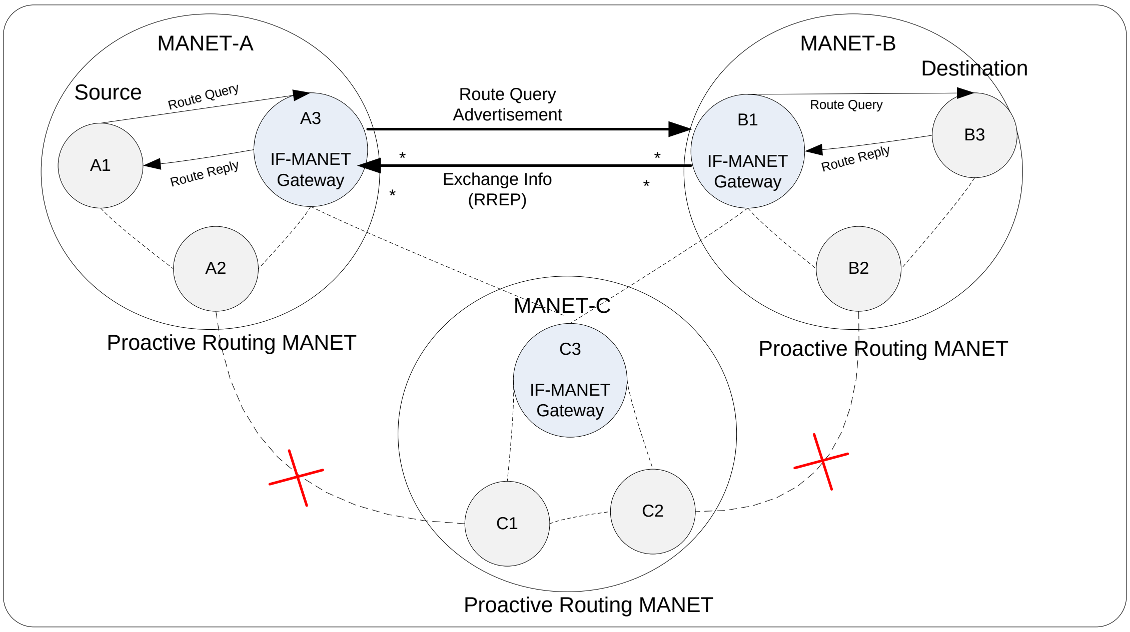


Figure 3.12: Context Diagram of the route discovery in proactive MANETs

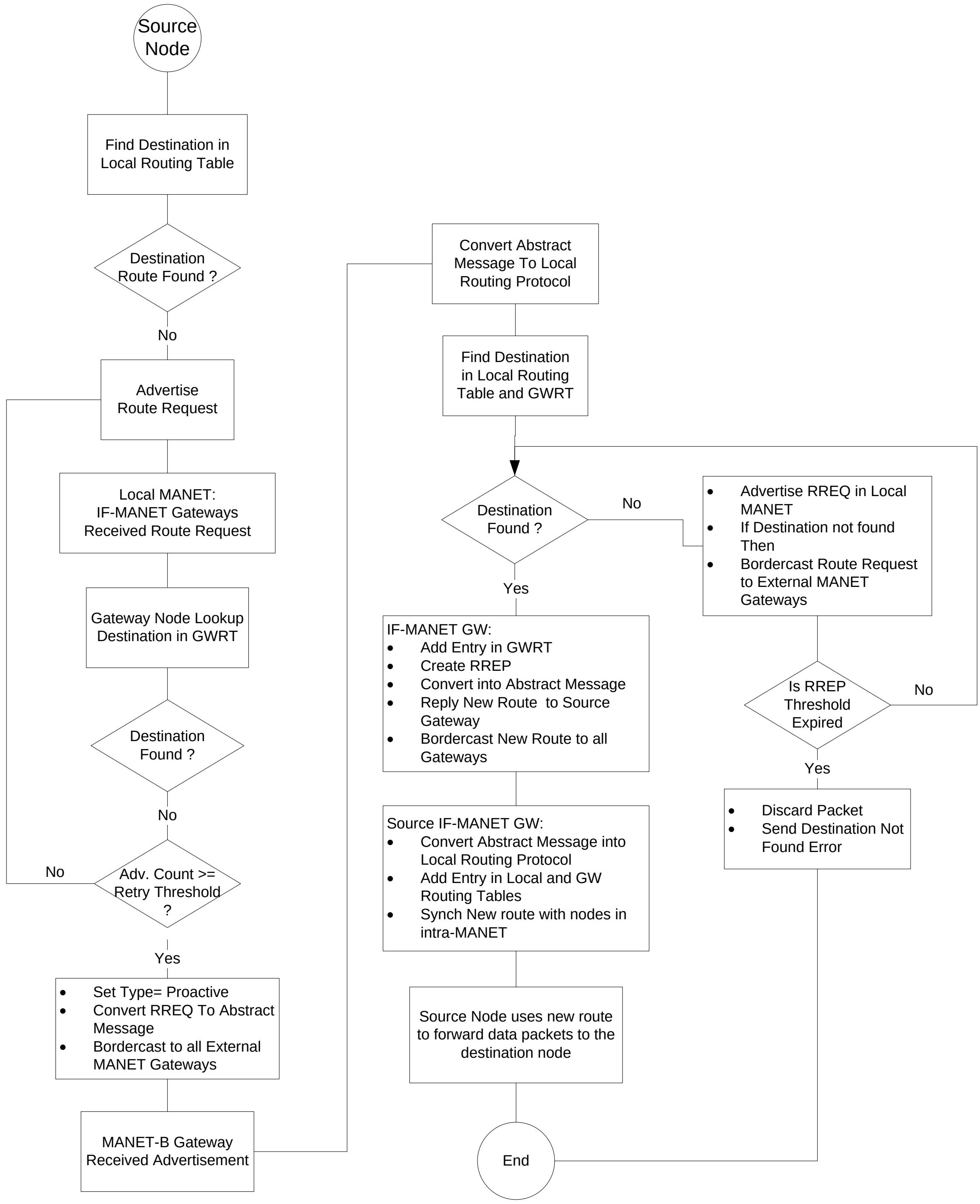


Figure 3.13: Workflow of Proactive Route Discovery Process

For illustrative purpose the ids, as shown in Table 3.19, have been assigned to Nodes, Gateways and MANETs presented in context diagram Figure 3.12.

Node Name	Address (id)
Node A1	000
Node A2	001
Node A3	002
Node B1	003
Node B2	004
Node B3	005
Node C1	006
Node C2	007
Node C3	008
Gateway A3	00011
Gateway B1	00021
Gateway C3	00031
MANET-A	0001
MANET-B	0002
MANET-C	0003

Table 3.19: Sample addresses of nodes for Proactive Route Discovery

First of all the Gateway Nodes generate a unique address for themselves and for their MANETs using a bloom filter hash technique (BFH) for destination resolution. From Figure 3.12, if a node A1 wants to send a message to a node B2, first of all it checks its routing table to find a route to the destination. If a route is not found then it will advertise for route discovery in the local MANET according to the proactive protocol in use. The Gateway Node A2, on receipt of route discovery request, lookup it's GWRT. If the route is found then it will update its local routing table and reply by exchanging the route information with the source node A1 which will then update its local routing table and broadcast the change in local MANET. If the route is not found then the Gateway waits until its Route_Advertisement_Counter equals RREQ_Retry_Threshold. It then assumes that the destination node belongs to an external MANET and create the entries for source and destination nodes in its GWRT with a unique new Request id (e.g. 0005), as shown in Table 3.20. The newly created GWRT entries are explained below:

- **Record-1**: Route from a Gateway Node A2 to a Source Node A1. Here, the Gateway Node A2 becomes the Source Node and the Originator (Source) node A1 becomes the Destination Node. The Next Hop A1 is an intermediate node with which the Gateway will communicate to reach the destination node. The Hop is the number of nodes between the source and destination nodes.

- **Record-2**: Route from a Gateway Node A3 to a destination Node B3. Here, the Gateway node A3 becomes the source of route discovery and the destination node becomes node B3 (destination of the originator node A1). Also the destination details i.e. MANET-id, node type and next hop node are empty. It is because the IF-MANET has not yet found the route to the destination node. The Hops has been assigned a dummy value of 255, reflecting an unreached destination.

Request id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. Node Type	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
0001	1(A2)	2(A3)	0001	3 (B1)	G	0002	3 (B1)	1		
0002	1(A2)	2(A3)	0001	4 (B2)	N	0002	3 (B1)	2		
0003	1(A2)	2(A3)	0001	8 (C3)	G	0003	8 (C3)	1		
0004	1(A2)	2(A3)	0001	6 (C2)	N	0003	8 (C3)	2		
0005	1(A2)	2(A3)	0001	7 (C2)	N	0003	8 (C3)	2		
0006	0 (A1)	2(A3)	0001	0 (A1)	N	0001	0 (A1)	1	60	0011
0006	0 (A1)	2 (A3)	0001	5 (B3)				255	60	0011

Table 3.20: Proactive Route Discovery - Source GWRT after RREQ:

The Gateway node then converts the route discovery packet into an abstract route request message as shown in Table 3.21. It sets Message Type to 02 (where 02 = Route Request), MANET Taxonomy to the DSDV, Source node to sender of RREQ i.e. Gateway Node A3 and Destination node to target node i.e. B3. The Gateway then Bordercast abstract message to external MANET Gateways.

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	Orig. Address	Source Address	Dest. Address	Seq. No	TTL
01234	02	Proactive (DSDV)	0001	00011	0 (A1)	2 (A3)	5 (B3)	0011	60

Table 3.21: Proactive Route Discovery - RREQ Abstract Message Created by Source Gateway

On receipt of route request message, all Gateway nodes lookup their local routing table (GWRT) for a route to the destination. If the route is not found then Gateway nodes will create an entry in their GWRT, update abstract message and re-bordercast the route request to Gateway nodes in their neighbour domains. The process continues until the destination is reached or a fresh route to the destination is found in an intermediate node. Here, we assume that the IF-MANET Gateway (B1) receives the route request abstract message, converts it into a local routing protocol and creates an entry in its GWRT as shown in Table 3.22. Due to proactive taxonomy, the GWRT periodically maintains the route information of all External MANET nodes. The Gateway then searches the destination in its local routing table (for local MANET) and GWRT (for external MANETs). If the route is found then it will update the newly created record in GWRT with details: Destination Node Type='N' (Node), Destination MANET id=0002, Next Hop= 3 (B1) and Hops=1.

Request id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. Node Type	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
0001	1(A2)	1(A2)	0001	3 (B1)	G	0002	3 (B1)	1		
0002	1(A2)	1(A2)	0001	4 (B2)	N	0002	3 (B1)	2		
0003	1(A2)	1(A2)	0001	8 (C3)	G	0003	8 (C3)	1		
0004	1(A2)	1(A2)	0001	6 (C2)	N	0003	8 (C3)	2		
0005	1(A2)	1(A2)	0001	7 (C2)	N	0003	8 (C3)	2		
0006	0 (A1)	2(A3)	0001	0 (A1)	N	0001	0 (A1)	1	60	0011
0006	0 (A1)	2 (A3)	0001	5 (B3)	N	0002	3 (B1)	1	60	0011

Table 3.22: Proactive Route Discovery - Receiving Gateway GWRT after RREQ received

After finding a route to the destination, the Gateway creates a route reply message as shown in Table 3.23, set its Message Type to 03 (RREP) and uni-cast message to source Gateway (A3).

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	Source Address	Dest. Address	Seq. No	TTL
0006	03	OLSR	0002	00021	3 (B1)	3 (B3)	0011	60

Table 3.23: Proactive Route Discovery - RREP Abstract Message Sent by Destination Gateway

The source Gateway node A3, on receipt of RREP abstract message, converts it into a local routing protocol and update its GWRT. Then the Gateway uni-cast's the RREP directly to the originator node A1 of the RREQ. Due to the proactive routing taxonomy, the Gateway Node synchronizes the new route with all the external gateways.

Request-id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. Node Type	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
.... Same as in Table 3.20										
0006	0 (A1)	2 (A3)	0001	5 (B3)	N	0002	3 (B1)	1	60	0011

Table 3.24: Source Gateway Routing Table (GWRT) after RREP Received

The source node “A1” will update its local routing table and uses this route to forward the data packet to destination Node-B3. The Gateway node on receipt of data packet will find the type of next hop or destination node from its GWRT, transform the data packet into an abstract message and forward directly to external Gateway which will in-turn forward the data packet to destination node. Hence, the process will seamlessly provide interoperability between heterogeneous proactive routing protocols. Figure 3.13 illustrates the workflow of route discovery phase in Reactive Routing MANETS. The Algorithm 3.4 presents the pseudo code of IF-MANET algorithm for route discovery in Proactive Routing MANETS.

3.7.6.2.2 Algorithm of Route Discovery in Proactive MANETs

Algorithm 3.4 below, shows the pseudo code of the IF-MANET Proactive Route Discovery.

```
Proc: Startup

        Source Node Advertise Route in MANET

Proc: Local GW Received Route Query:

Gateway Node received Route Query

If (Source Protocol Type = Proactive) Then

        If (Destination Not in GWRT) Then

                If (GW Advertisement Counter >= RREQ Retry Threshold) Then

                        Create Route Request Packet from Route Query

                        Set: Req.id, MANET-id, GW-id, SeqNo, Reply_Time_Expiry,
                                Proc_Type, Source_Add, Destintatio_Add

                        Convert Route Request Packet into Abstract Message

                        Bordercast Route Request Advertisement to External Gateways

        End

End

Proc: External GW Received Route Request:

If (MANET_id Equals this.MANET_id) Then

        Drop Packet

If (MANET_id Not Equals This.MANET_id) Then

        Convert Abstract Message into Local Routing Protocol

        Lookup Destination in Local and GW Routing Table

If (Route to Destination Found) Then

        Call: Send Route Reply Proc

Else
        Route Reply: Destination Not Found
```

End

Proc: Send Route Reply

GW Node Create entry in GW Routing Table

Create Route Reply Message

Convert Route Reply into IF-MANET Abstract Message

Unicast Message to Source Gateway

Bordercast New Route to all MANET Gateways

Proc: Receive Route Reply

Convert Abstract Message to Local Routing Protocol

Gateway Updates local and GW routing table

Propagate new entry in the local MANET

Source (Originator) node updates its local routing table

Source Node forward data packet using the new route found

End // Proactive Route Discovery

Algorithm 3.4: Route Discovery in Proactive MANETs

3.7.6.3 Route Discovery for Heterogeneous MANET Taxonomies

The Hybrid Route Discovery combines the individual route discovery taxonomies (e.g. reactive and proactive) and extends their behaviour to provide interoperability between them. In Hybrid Route Discovery, mobile nodes within MANET uses local routing protocol e.g. Reactive or Proactive etc. Whereas Gateway nodes use local native protocol within MANETs and hybrid approach across different MANETs to discover route and maintain the Gateway Routing Table (GWRT). Unlike proactive and reactive approaches, that store all nodes from all MANETs and only the Gateway address of all MANETs respectively, GWRT maintains all Gateway Nodes and address of only those external native nodes (Non-Gateway) that were discovered during route discovery.

The gateway periodically broadcasts an IF-MANET Universal Message with Message type = 01 (Beacon) after configurable ADVERTISEMENT_INTERVAL. All gateway nodes residing in the gateway's transmission range receive the Universal Message.

Upon receipt of the message, the Gateway nodes that do not have a route to the gateway create a route entry for it in their routing tables. The Gateway Nodes, unlike proactive route discovery, do not broadcast their local routing tables i.e. all internal nodes, to external MANET Gateways. It will significantly reduce the network as well Gateway resource utilization overheads. For route discovery across MANETs, it will use reactive approach and will discover the destination route on demand. Unlike reactive protocol, it will store the route information, found during route discovery, in GWRT for future use. This strategy will reduce end to end (E2E) delivery of packets and increases the overall performance of the system.

Mobile nodes that already have a route to the gateway update their route entry for the sender gateway. The mobile nodes want to communicate with mobile node in other MANET then they will interact only through Gateway Nodes.

3.7.6.3.1 Heterogeneous Route Discovery Process by Example

This section explains with example that how the route discovery process takes place when nodes in heterogeneous MANETs want to communicate with each other.

Figure 3.14 shows a high level context diagram of communication between nodes of different MANETs using IF-MANET hybrid route discovery. The logical flow, explaining the activities of different components and their interaction is explained in Figure 3.15. The pseudo-code of IF-MANET heterogeneous route discovery protocol is presented in Algorithm 3.5.

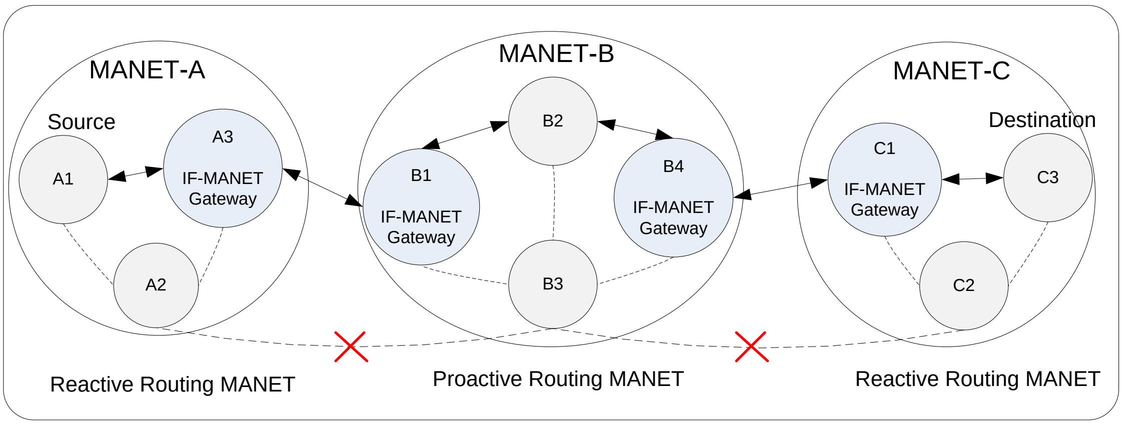


Figure 3.14: Route Discovery in Heterogeneous MANETs

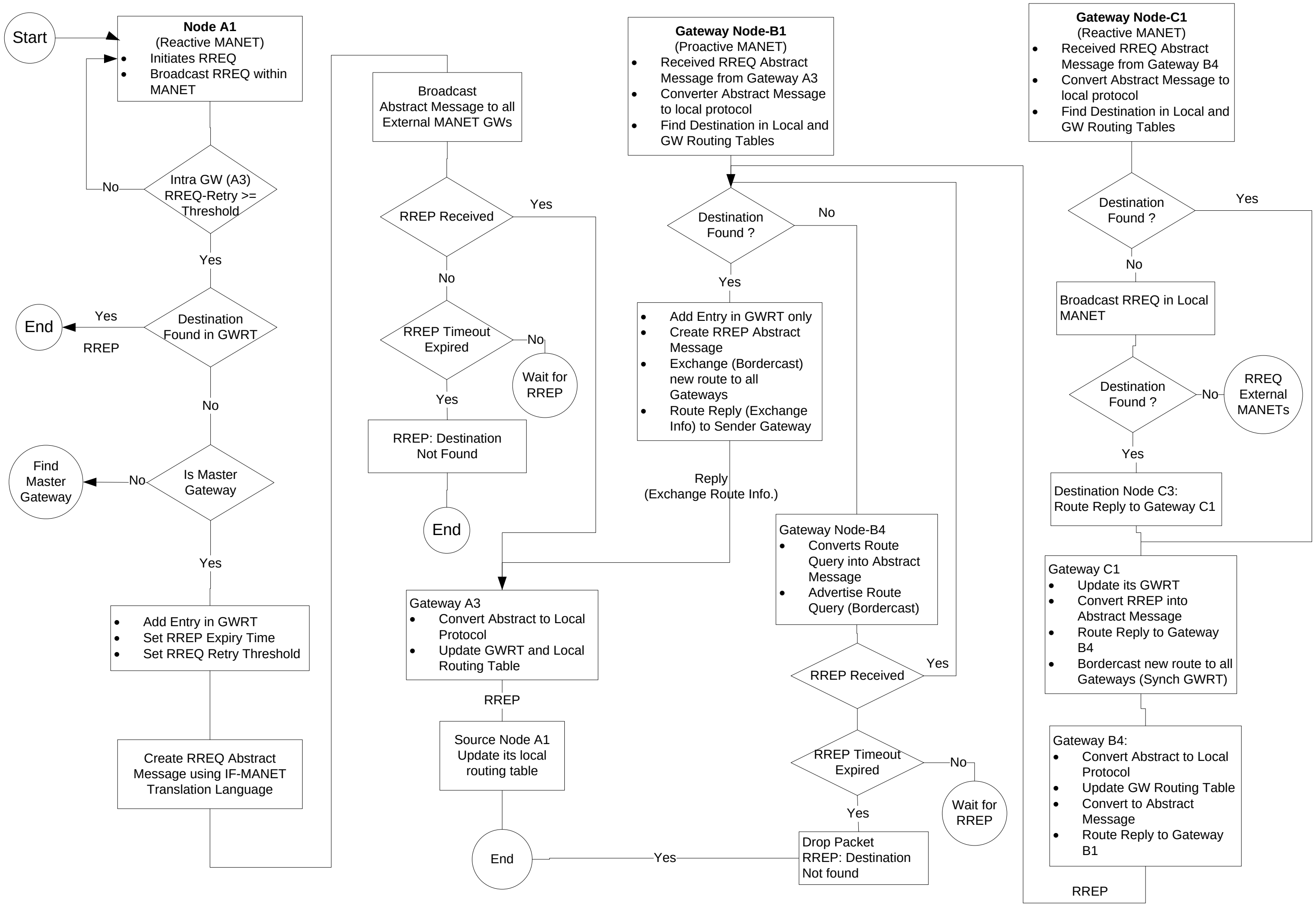


Figure 3.15: Route Discovery in Heterogeneous MANET Routing Protocols

Figure 3.14 assumes that MANET-A is running Reactive Routing Protocol i.e. AODV, MANET-B is running Proactive Routing Protocol i.e. OLSR and MANET-C is running a Reactive Routing Protocol i.e. MAODV. For illustration purpose, the id's of Nodes and MANETs used in this example are given in Table 3.25 below.

Name	Address (id)
Node A1	000
Node A2	001
Node A3	002
Node B1	100
Node B2	101
Node B3	102
Node B4	103
Node C1	200
Node C2	201
Node C3	202
Gateway A3	00201
Gateway B1	10001
Gateway B4	10303
Gateway C3	20201
MANET-A	0001
MANET-B	0002
MANET-C	0003

Table 3.25: Sample addresses of nodes in Heterogeneous Route Discovery

The Node A1, from reactive MANET-A, wants to send a message to destination node C3. If A1 cannot find a route to the destination is in its local routing table then it will follow the Reactive Route Discovery Process, explained in Section: 3.7.6.1, discovers the route in Local MANET. If the destination is not found, then the Master Gateway Node (A3) will create new entries in its GWRT as shown in Table 3.26 against Request ids: 0004 and 0005. It will then create an Abstract Route Discovery Message, as shown in Table 3.27 and Bordercast to all the external gateways.

Request id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. Node Type	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
0001	00201 (A3)	00201 (A3)	0001	10001 (B1)	G	0002	10001 (B1)	1	60	001
0002	00201 (A3)	00201 (A3)	0001	10303 (B4)	G	0002	10001 (B1)	2	60	002
0003	00201 (A3)	00201 (A3)	0001	20201 (C1)	G	0003	10001 (B1)	3	60	003
0004	0 (A1)	00201 (A3)	0001	0 (A1)	N	0001	0 (A1)	1	60	004
0005	0 (A1)	00201 (A3)	0001	202 (C3)				255	60	005

Table 3.26: Heterogeneous Route Discovery - Reactive Source GWRT

Request id	Message Type	MANET Taxonomy	MANET id	Gateway id	Orig. Address	Source Address	Dest. Address	Seq. No
01234	02	Reactive (AODV)	0001	00201	0 (A1)	00201 (A3)	202 (C3)	0011

Table 3.27: Heterogeneous Route Discovery - Abstract Message

Let us consider that, the Gateway Node B1 of the proactive MANET-B receives an Abstract RREQ message. It will then follow the same process as described in Proactive Route Discovery to process the received RREQ message. Unlike the Proactive approach to create entries of all MANET mobile nodes in the GWRT, the Hybrid Protocol will only add addresses of the Gateway Nodes and only those mobile nodes that were part of route discovery. The Gateway node B1 cannot find the destination node in its local MANET (B) because the destination belongs to external MANET i.e. MANET-C. The Gateway-B1 creates an entry in GWRT, an abstract route discovery message and broadcast the route discovery. The GWRT of Gateway-B1, after new entries, is shown in Table 3.28. The abstract RREQ message values are shown in Table 3.29 below.

Request -id	Orig. Address	Src Address	Src. MANET id	Dst. Address	Dst. Node Type	Dst. MANET id	Next Hop	Hops	TTL	Seq. No
0001	00201 (B1)	00201 (B1)	0002	00201 (A3)	G	0001	00201 (A3)	1	60	001
0002	00201 (B1)	00201 (B1)	0002	101 (B2)	N	0002	101 (B2)	1	60	001
0002	00201 (B1)	00201 (B1)	0002	10303 (B4)	G	0002	101 (B2)	2	60	001
0003	00201 (B1)	00201 (B1)	0002	20201 (C1)	G	0003	0303 (B4)	3	60	001
0006	0 (A1)	00201 (B1)	0002	00201 (A3)	G	0001	00201 (A3)	2	60	0011
0006	0 (A1)	00201 (B1)	0002	202 (C3)	G	0003	10303 (B4)	3	60	0011

Table 3.28: Heterogeneous Route Discovery - GWRT of Gateway after RREQ

Request id	Message Type	MANET Type	MANET id	Gateway id	Orig. Address	Source Address	Dest. Address	Seq. No
0006	02	OLSR	0002	00201	0 (A1)	00201 (B1)	202 (C3)	0011

Table 3.29: Hetero Route Discovery - Gateway sent Abstract RREQ Message

The network-B is running proactive protocol whereas the network-A, network-C are running reactive routing protocols and the source node from the network-B wants to send packet to destination node in network-C. In this case, when Gateway node of the network-B cannot find the destination in its routing table and therefore it will advertise the route discovery to all intra-MANET nodes. The Gateway Nodes of Network-B will receive the advertisement message, checks the advertisement counter and the Master Gateway Node will convert the Route request message to the IF-MANET's abstract message and broadcast the route discovery to all the MANETs. The Gateway nodes of the network-C when receives the request with source routing type equals Proactive, it will convert the abstract message to local routing protocol, finds the route to destination and send back the routing information back to the source Gateway (B1). The Gateway B1 then converts the abstract the message to its local routing protocol using MTL, add entry in its local and GWRT routing tables, update the abstract message with new routing information and follow the proactive mechanism to send reply back to the source node via Gateway-A3.

3.7.6.3.2 Algorithm of Route Discovery in Proactive MANETs

Algorithm 3.5 below, shows the pseudo code of the IF-MANET route discovery in heterogeneous MANET taxonomies.

```
Proc: Startup
If (Destination Not Found in Local Routing Table) Then
        Source Node Initiate Route Discovery (RREQ)
Proc:  Intra-GW Receive Route Request
        IF-MANET Gateway receives route request (RREQ)
        If (Sender Not Gateway Node) Then // Non-Gateway Node
                If (Sender.MANET-ID Not Equals Gateway.MANET-ID) Then
                     Drop Packet // Nodes can't communicate with External MANETs
                If (Destination In Gateway's Local Native Routing Table) Then
                        Route Reply using Native Routing Protocol
                        Stop Processing
                If (Destination Found in GWRT) Then
                        Route Reply with Destination Route from GWRT
                        Stop Processing
                If (Gateway RREQ Retry Counter >= RREQ Retry Threshold) Then
                        Create Entry against RREQ in GWRT
                        Invoke Proc: Master GW Process Message ()
                        Initialize RREQ Retry Counter
                If (Gateway RREQ Retry Counter < RREQ Retry Threshold) Then
                        Increment RREQ Retry Counter
                        Release Packet
                End
        ELSE
                Call Proc: Active_GW_Process_Message( Gateway_id, MANET_id)
```

End

Proc: Master GW Process Message

// Find Active Gateway to Send RREQ to avoid network flooding

If (This.Gateway = Master Gateway) Then

Proc: GW Process RREQ Message

Send Confirmation to Intra-Gateways // *I am Master and processing the message*

Else (Wait for GW_Active_Confirmation_Expiry_Time)

If (Wait time expired) Then // No Master GW Confirmation received

Proc: GW Process RREQ Message

Else If (Confirmation received) Then

//Other Master/Active Gateway has already processed the message

Drop Packet

End

Proc: GW Process RREQ Message

Create Route Request Abstract Message using IF-MANET ITL

Update Gateway Routing Table (GWRT)

Cache original route request message

Increment GW Advertisement Counter (GW_Counter)

IF-MANET Bordercast Abstract Message to all External MANET GWs

Set RREP Expiry Time

Wait and Listen for RREP

// Gateway Operation to send Query directly to GWs

Proc: IF-MANET GW-Bordercasting

Search GWRT

Find All Gateways where MANET-id **Not Equals** My.MANET_id

Directly Send Route Request Abstract Message to All External MANETs

// Avoid flooding network by broadcasting to all nodes in all MANETs

Proc: Gateway Receive RREQ Message from External MANET

// Using IF-MANET MessageTranslator

Convert Abstract Message to Local Routing Protocol

If (RREQ.MANET- ID == This.MANET-ID) Then

Drop Packet //Must be different MANET to avoid looping

Stop Processing

If (Sender Node Type != GW && RREQ.MANET- ID != This.MANET-ID) Then

Drop Packet //Only process GW requests

Stop Processing

If (RREQ SeqNo < Last Seq No && RREQ. Request-id == This.Request-id) Then

// Seq. No of Req-id must be latest to avoid processing old and indirect requests from multiple channels

Drop Packet

Return

// Gateway of other MANET received the RREQ

Search Destination in Routing Table and GWRT

If (Destination Found In Local Routing Table) Then

Create Route Entry in GWRT

Set Source=Gateway-Address and Destination=Target Node Address

Propagate (Advertise) New Route to all Gateways

Destination Node: Add route in its Routing Table

Invoke Proc: Route Reply Process

If (Destination NOT Found In Routing Table) Then

If (This.MANET Type == Proactive) Then *// Destination not in local MANET*

GW Create Route Query Message

Convert Into Abstract Message

Create Entry in GWRT

Exchange GWRT (new entry) with Local MANET Gateways

Advertise (Bordercast) Route Query to External MANETs

Wait for RREP *// Invoke: Receive Route Reply Proc*

If (This.MANET Type == Reactive) Then

//First: On-demand Route Discovery in Local MANET

Create Route Request Message

Create Entry in GWRT

Broadcast RREQ in Local MANET

If (Destination Not Found) Then *// No RREP received within Expiry Time*

Convert RREQ Into Abstract Message

Bordercast RREQ to all External MANETs

Wait for RREP *// Invoke: Receive Route Reply Proc*

Else If (Destination Found) Then *//Send RREP received*

Update GWRT

Invoke Proc: Route Reply Process

Proc: Route Reply Process

Create Route Reply Packet from GWRT

Convert RREP into If-MANET Abstract Message

Unicast (Directly Send) to Source Gateway Node *// Exchange new route information*

Proc: Receive Route Reply (RREP)

Convert Abstract Message to local native routing protocol

Update Local native routing table

Update GWRT

Forward (Unicast) to RREP to Source node

Source Node add entry into Routing Table

Source Node directly sends data packets to destination using the route path found

End *// Route Discovery for Hybrid MANET Taxonomy*

Algorithm 3.5: Route Discovery in Heterogeneous MANET Routing Protocols

3.8 Summary

To overcome the heterogeneity of MANET routing protocols, a novel Interoperable Framework “IF-MANET” has been proposed in this chapter. The IF-MANET provides interoperability between heterogeneous routing protocols and enables them to seamlessly communicate with each other. The Framework uses component based architecture, where each component acts as an independent service to provide a distinct set of functionalities. Due to the service oriented component model, newly arriving protocols can easily be integrated into the system without affecting the existing functionality.

For communication between dynamic heterogeneous MANET taxonomies, IF-MANET has created a new Interoperable Routing Protocol. The protocol uses a cluster based inter-domain routing approach and has created a unique Gateway that enables mobile nodes of one MANET system to communicate with the nodes of another MANET System. The Intra-MANET (Internal MANET) nodes communicate with their local Gateways for communicating with external MANETs. These Intra Gateways keep the state of local source mobile nodes into its special GWRT (Gateway Routing Table) for association, translate their packets into destination packets, and communicate with Gateways of destination MANET. The IF-MANET routing Framework consists of the following components to achieve the interoperability in heterogeneous MANETs.

- **Universal Packet**: It provides a packet format, specialised for IF-MANET, to exchange information between Gateways. The IF-MANET periodically broadcast beacons (Heartbeats) of Universal Packet to collect and maintain the information between the Gateways.

- **Gateway Routing Table (GWT):** It stores and maintains the routing information of nodes, communicating with external MANETs, in a protocol independent format. The information allows nodes to associate the packets received in the route reply.
- **Gateway Engine:** It receives requests/replies from nodes, communicates with other components to translate and process incoming/outgoing packets, and send requests/replies to intra or inter MANET nodes.
- **Initialisation Phase & Route Maintenance**: At start-up of IF-MANET node, it will create a unique MANET-ID and Gateway-ID, calculate the ranking of the Gateway node to elect it as Active or Passive Gateway and periodically exchange gateway information with other MANET gateways.
- **Message Translator (MTL):** MTL defines MANET Ontology to handle the semantic differences that arise between different routing protocols, in order to enable them to interoperate. It uses three phases to perform interoperability, which are: Discovery Phase to identify the packets, Comparing Phase to find out the differences and missing fields between the packets, Conversion/Mapping Phase to apply rules to semantically generate the missing fields.
- **Route Discovery in Heterogeneous MANETs**: The route discovery enables a source node to find a route and send packets to a destination node. The IF-MANET has proposed a route discovery mechanism to find a route between node, running different routing protocols, in heterogeneous MANET taxonomies. The IF-MANET uses Gateway nodes to translate the source packet into the target packet using MTL and provides an interoperable bridge between MANETs running heterogeneous routing protocols.

The next chapter explains the implementation of the design presented in this chapter with the help of UML diagrams and code snippets.

Chapter 4

Implementation of IF-MANET

4.1 Introduction

This chapter explains the implementation of the proposed IF-MANET interoperable routing protocol presented in Chapter 3. The IF-MANET implementation is written in C++ language (Stroustrup 2013) and the scripts for configuring and creating the mobile nodes are in TclCL language (TclCL, 2014). The Eclipse Editor (Eclipse Foundation, 2011) is used as an IDE for developing, debugging and executing the application. The operating system Linux (Ubuntu 12) (Ubuntu, 2012) has been used for development and running the simulations.

The IF-MANET code is deployed on the Network Simulator NS-2 (Ns2 2008) to run real life scenarios in a simulated environment. The NS-2 is a discrete event driven simulation tool written in two programming languages i.e. C++ and Tcl/OTCL .The Tool Command Language (TCL) is a scripting language which is responsible for interfacing with the precompiled C++ objects to create mobile nodes and simulation scenarios. The NS-2 provides enriched library for simulating the communication between routing protocols in mobile ad hoc network environments. The IF-MANET gateway is derived from a standard mobile node of NS-2 to provide ad hoc communication functionality.

Figure 4.1 presents a class model of the IF-MANET architecture. Different classes and data structures implements different functionalities of the IF-MANET protocol. The IF-MANET node derives the standard mobile ad hoc communication functionality of an existing NS-2 routing protocols i.e. AODV, in order to avoid re-implementing the NS-2 communication layer. Though AODV was used in this study, our approach can be applied to any MANET routing protocol to implement the IF-MANET framework in order to achieve the routing of packets in heterogeneous MANETS. The subsequent sub-sections of this chapter present and discuss the implementation of the following IF-MANET proposed algorithms:

1. Route Discovery
2. Initialisation & Maintenance Phase
3. Message Translator (MTL)

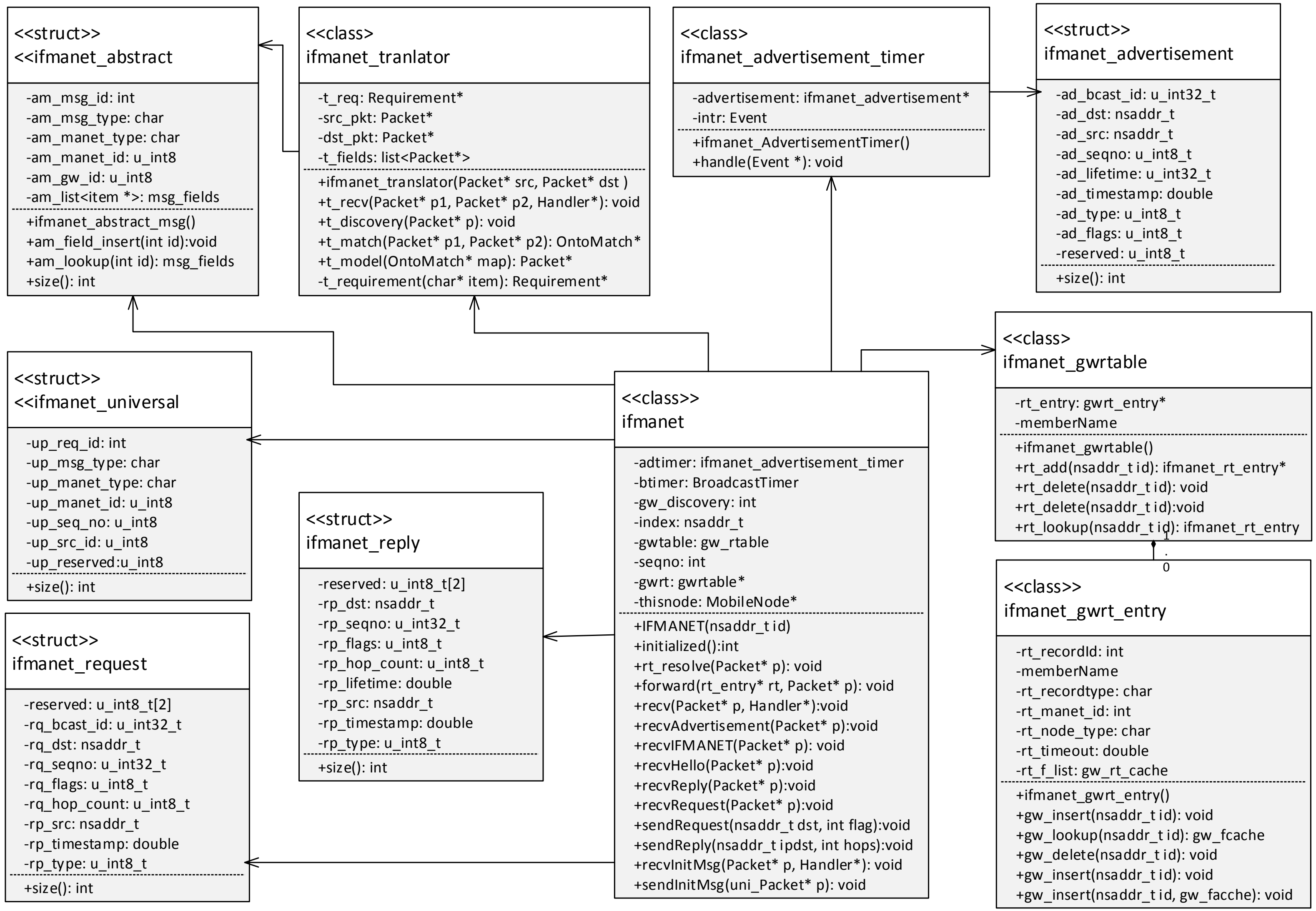


Figure 4.1: Class Model of IF-MANET Interoperable Routing Protocol

4.2 Implementation of Route Discovery Algorithms

This section explains the implementation of route discovery algorithms proposed in section [3.6.7]. The main functionality of the algorithms is implemented in ifmanet.cc class and the functions are declared in ifmanet.h. The following sub-sections explain the classes and their methods used for implementing the route discovery algorithms.

4.2.1 Ifmanet.cc and ifmanet.h

The ifmanet.h is a header file and declares all the methods, configurable variables and data structures used in ifmanet.cc. The ifmanet.cc is a main class which provides the functionality to send, receive and process the packets. The ifmanet.cc invokes rt_resolve() to find a route to the destination when a mobile node wants to send a data packet to the destination. If the mobile node does not have any valid route to the destination it broadcasts a RREQ message using sendRequest() method. The RREQ message is eventually received by the destination or an intermediate node and is processed in recvRequest() method. The destination node or an intermediate node with fresh route to the destination sends a RREP message back to the originator of the RREQ using the method sendReply(). The originator of the RREQ receives the RREP message in a method recvReply() and starts sending data packets to the destination. The functionality of ifmanet.cc, ifmament.h is explained in the following sub-sections.

4.2.1.1 rt_resolve ()

This function is invoked under two scenarios i.e. firstly, when a source node wants to send a data packet and secondly when an intermediate mobile node receives a packet for forwarding towards the destination node. These two scenarios are explained below:

Scenario-1: If the current node is a gateway node, its network wide search counter equals RREQ_Retry_Threshold value and the source node has not received a route response (RREP). Then in that case, the IF_MANET Gateway node assumes that the received message is destined for an external MANET. The Gateway node then converts the packet into abstract message, sends (broadcasts) to other gateways and add record into its GWRT. Whereas, if the current node is a mobile node then it processes the message as described in the native routing protocol specification.

Scenario-2: If the function is invoked by an intermediate node, who has received the packet from the source node then the request will be processed differently depending on the type of a packet i.e. whether it is sent by mobile node or a gateway node. If mobile node is the source of the packet then the intermediate node processes the packet and forwards it to the next hop node. If the source node is a gateway node then the intermediate node invokes the Gateway functionality to translate and forward the message to the destination node. The Gateway node also updates its Gateway routing table (GWRT) with the details received in a packet. Figure 4.2 below, explains the implementation flow of the rt_resolve () method.

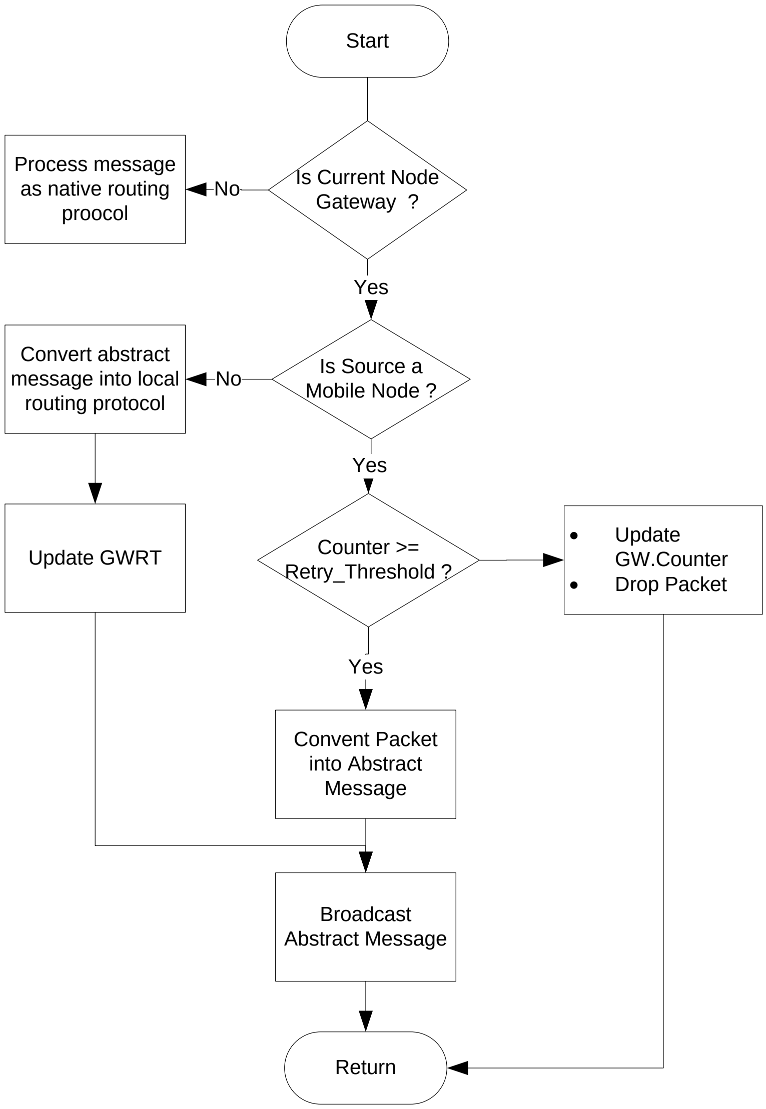


Figure 4.2: Route Resolve method of IF-MANET

4.2.1.2 sendRequest (nsaddr dest)

The mobile node invokes sendRequest() method to find a route to the destination node by sending a route discovery request. The mobile node continuously broadcasts a RREQ message, according to the expanding ring search mechanism, until it receives a RREP. The IF-MANET Gateway gets triggered when a RREQ reaches a maximum value of RREQ_Retry_Threshold without receiving any corresponding RREP and assumes that the destination node belongs to an external MANET. The Gateway node then starts the Route Discovery to find out a route to the destination in external MANETs.

Firstly, the Gateway node creates a new entry or updates an existing in its GWRT. If the route to the destination node is not found in GWRT then it creates a new route request packet, translate the source packet into abstract message, and populate the newly created packet and Bordercast to Gateway nodes of external MANETs. The Gateway node sets an expiry time of RREP and increments the GW_Counter. If no response is received before the expiry of RREP time then the Gateway drops the packet and sends No_Route_Found back to source node.

4.2.1.3 receiveRequest (Packet *p)

Figure 4.3 shows the workflow of receiveRequest () functionality. This function is invoked for every network node when a mobile node receives any type of route request packet. If the receiving mobile node is the originator of the packet or it has already received the packet with same request number then the node immediately drops the packet to avoid circular looping.

The received message gets processed differently based on type of source node i.e. mobile or gateway node. In case of mobile node, the code executes without any modification whereas for Gateway node, it first checks the type and source of the packet. If the received packet is of type PT_IFMANET i.e sent by IF-MANET Gateway, then the Gateway node first converts the received abstract packet into the native local routing protocol and calls SendReply() method to RREP to the originator of the RREQ message. If the sender is a non-Gateway node (mobile node) and belongs to different MANET then the current node drops the packet because of the assumption that the external mobile nodes cannot directly communicate with Gateways. However, if the sender node belongs to the same MANET then the Gateway node converts the packet into abstract message, update its GWRT and calls SendRequest() method to Bordercast the packet to external MANETs for route discovery.

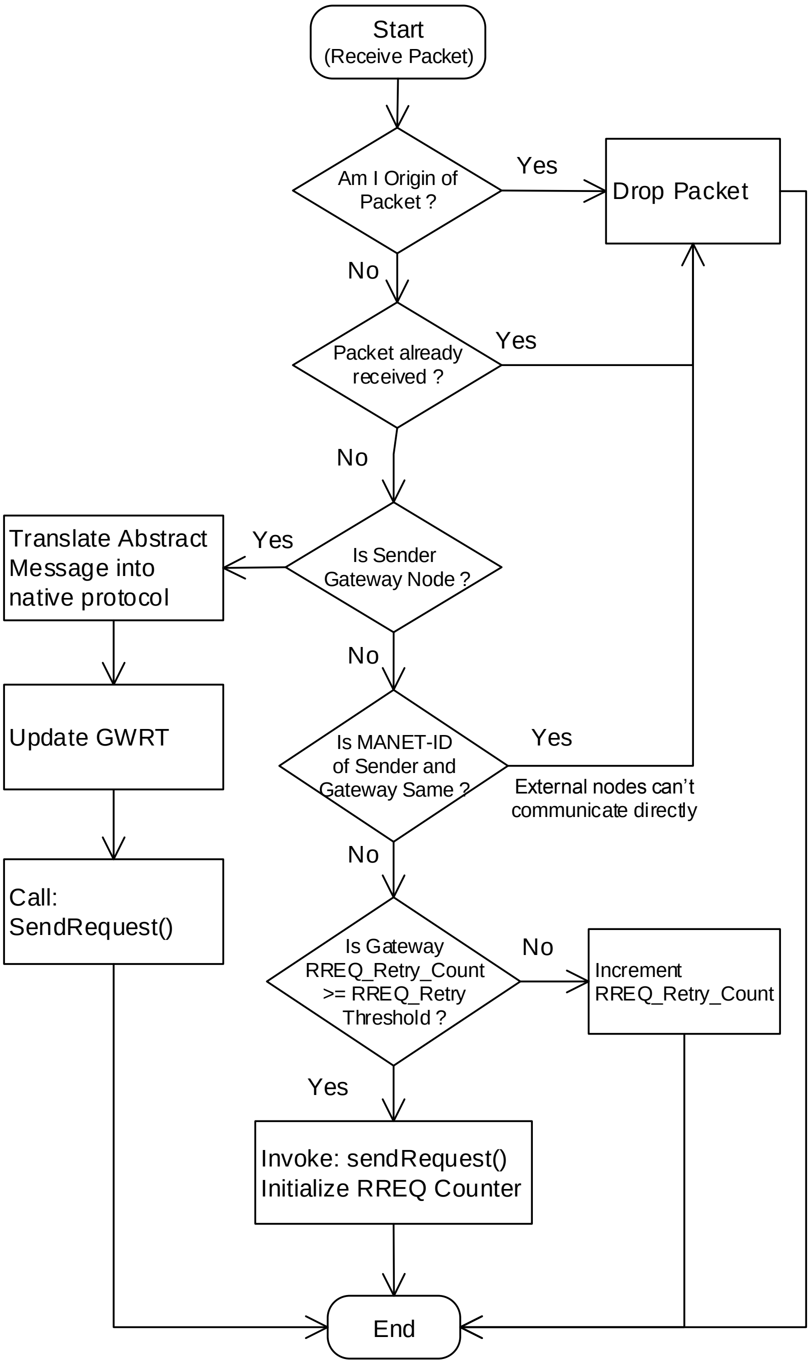


Figure 4.3: Gateway Node Receive Request Method

4.2.1.4 . sendReply(nsaddr ipdst, int hop_count, nsaddr rpdst, int rq_seq)

The node triggers this function if it has received a RREQ messages and either it is a destination node or an intermediate node with a fresh route to the destination. The Gateway node processes the sendReply() method differently depending upon the type of destination node. If the RREP is for local mobile node then the Gateway node converts the packet from abstract message to local routing protocol and then uni-cast it to the originator of the RREQ. Whereas, the external Gateway node wants to send RREP to the source Gateway node, then firstly it converts the packet into an abstract message and then unicast it to the source Gateway node.

4.2.1.5 recvReply(Packet *p)

Mobile node invokes recvReply() method when it receives a RREP message. The message is processed differently depending on the type of RREP packet. If the RREP is received by a mobile node then the code runs without any change. However, if the message is received by a Gateway node from a mobile node then the Gateway node update its GWRT corresponding to RREP's RREQ_id, creates an abstract message, a new RREP packet and unicast the newly created packet to the source node. Whereas, if the RREP is received by Gateway node from external Gateway node with different MANTE-id, then it will convert the abstract message into local routing protocol, update its GWRT and forward (unicast) the RREP message to the source node.

4.2.1.6 sendAdvertisement(int ttl)

This method implements the route discovery in proactive MANET as explained in Section: [3.6.7.2]. For a route discovery, the IF-MANET gateways periodically broadcast the Gateway Advertisement packet after a configurable gateway time period i.e. GW_Advertiesement_Period. The method creates a packet and populates with ifmant_advertisement data structure. It then assigns the Ifmanet_advertisement: Type = IFMANET_Advertisement, source=index (current node address), broadcast_id = broadcast_id++, Manet_id = this.MANET_id and Gateway_id = this.Gateway_id. The method then assigns the packet to scheduler (advertisement_timer) to broadcast periodically at pre-configured time. The logic of send advertisement is presented in Algorithm [3.4].

4.2.1.7 ReceiveAdvertisement(Packet *p)

The ReceiveAdvertisement method receives the advertised messages sent for the exchange of Gateway route discovery information in a proactive route discovery mechanism as explained in Section: [3.6.7.2] and in algorithm [3.4]. This method first checks the source of the packet and if it is same as of the current node then it drops the packet. If the Gateway node doesn't have a fresh route to the received Gateway node in its GWRT then it will create one and add it into the GWR. If the Gateway node cannot find the destination in GWRT and GW_Advertisement counter is greater than or equals to RREQ_Retry_Threshold then it will create the route request packet and assigns the Request id, MANET_id, GW_id, new sequence no, source address, destination address and then broadcast the packet to external Gateways. The external Gateways lookup their GWRT for the destination, if the record is not found it then creates a new entry and search on behalf of source GW node. If it finds a destination in its routing table then it will create a route reply packet and unicast back to the source gateway node. This way the proactive MANETs maintains the routing path of all the external Gateway nodes.

4.2.2 ifmanet_packet.h

This header file defines the structure of different packets used in this routing protocol. These packet structures are explained below:

4.2.2.1 struct ifmanet_request

The sendRequest() method creates ifmanet_request packet, populates with data and send's this packet in a route discovery request (RREQ). For Example, to send a route discovery, the IFMANET allocates the following values to ifmanet_request packet:

```
rq->rq_type            = IFMANETYPE_RREQ;  // type of packet
rq->rq_bcast_id        = bid++; // request id
rq->rq_dst             = dst;  // destination node address
rq->rq_dst_seqno       = (rt ? rt->rt_seqno : 0); // sequence number
rq->rq_src             = index; //  address of source node
rq->rq_src_seqno        = seqno++; // sequence number of packet
rq->rq_timestamp       = CURRENT_TIME; // current date time as timestamp
```

Where "rq" is an instance of ifmanet_request (i.e. rq*= IFMANET_REQUEST(p)), IFMANETYPE_RREQ is a property whose value is defined as 0x02 (#define

IFMANETYPE_RREQ 0x02), index is the address of the current node and dst is the address of destination node.

4.2.2.2 struct ifmanet_reply

The SendReply() method creates the ifmanet_reply packet to send a route response (RREP) back to the source node. For Example, the Gateway node populates the ifmannet_reply packet with following values:

```
*rp = ifmanet_reply(p); // an instance of ifmanet_reply packet

 rp->rp_type              = IFMANET_RREP; // Type of packet

 // Data Transformation
 if(rt->rt_dst_ptype == PT_AODV)
         rp->rp_type      = AODVTYPE_RREP;
 if(rt->rt_dst_ptype == PT_MAODV)
         rp->rp_type      = MAODVTYPE_RREP;

rp->rp_dst                = rpdst; // address of destination node
rp->rp_dst_seqno          = rpseq; // sequence number of route response
rp->rp_src                = index; // address of source node
```

Where “rp” is an instance of ifmanet_reply packet (i.e. *rp = ifmanet_reply(p);), IFMANET_RREP is a type of packet whom default value is set to 0x04 ((#define IFMANETYPE_RREP 0x04).

4.2.2.3 ifmanet_abstract

This is the IF-MANET’s proposed abstract packet defined in Section 3.6.4 (Abstract Message) and is declared in ifmanet_packet.h file. The Gateway node creates and populates this packet from the abstract message to communicate with heterogeneous routing protocols. The structure and details of this packet is explained in Section 3.6.4.

4.2.2.4 ifmanet_universal

The IF-MANET has proposed this packet to exchange the information between heterogeneous IF-MANET Gateway nodes. The IF-MANET creates this packet to periodically send Hello beacons (Heartbeats) to collect information between Gateways in order to pre-empt the destination type and address for bordercasting. Table 3.1 illustrates the ifmanet_universal packet format. The structure and detail of ifmanet_universal packet fields are described in Section 3.6.2 (Universal Packet).

4.2.3 ifmanet_gwrtable and ifmanet_gwrt_entry

The class's ifmanet_gwrtable and ifmanet_gwrt_entry are used to implement the functionality and packet format of the IF-MANET proposed Gateway Routing Table (GWRT) as presented in Section 3.6.3 and Table 3.2 respectively.

As shown in Figure 4.1, the ifmanet_gwrtable and ifmanet_gwrt_entry has one-to-many relationship with each other and are associated with ifmanet.cc. The ifmanet_gwrt_entry defines the data structure of the GWRT, maintains a list of gwrt_entries and provides access methods to add new record, delete/update existing record and lookup a record from the list. The ifmanet_gwrtable is composed of many ifmanet_gwrt_entry records. The ifmanet_gwrtable maintains a routing table which stores gwrt_entry record against a unique node record id. The class also provides the functionality embedded into the class to add new route_entry record, delete/update existing and lookup against a unique record id. The class model of these classes is shown in Figure 4.1.

4.2.4 ifmanet_advertisement_timer and ifmanet_advertisement

The ifmanet_advertisement_timer class and ifmanet_advertisement data structure are created to implement the proposed proactive route discovery mechanisms. The ifmanet_advertisement_timer extends the base handler class to use event model for scheduling the packets to be sent and send/broadcast them when time period elapses. The packet sent is defined as a new IF_MANET packet with type =IFMANET_Advertisement.

The ifmanet_advertisement is a data structure of the packet to be advertised for proactive route discovery. It defines he packet type = IFMANET_Advertisement, hop_count= hops (number of nodes) to reach the next node, destination address, destination sequence number, source address and broadcast (request) id.

The class model of the IF_MANET advertisement and its associated data structure is shown in Figure 4.1

4.3 Initialization & Maintenance Phase

Each IF-MANET node broadcasts a Hello message, when it starts-up and at regular configurable time interval, for exchanging the Gateway route information with other Gateway nodes. The Gateway nodes uses the IF-MANET proposed Universal Packet to discover and maintain the list of IF-MANET Gateways. At start-up the Gateway node set the following fields of Universal Packet:

- Message Type=01 (Hello Message)
- Assign MANET_id
- Create new Gateway_id for the Gateway node
- Calculate and assign the Gateway Rank to GW_Status
- Create and assign the Request id, Sequence No and TTL

The details of the initialization phase are explained in Section: [3.6.6] and the algorithm explaining the logic is presented in algorithm [3.2]. The implementation flow of the initialization phase is shown in Figure 4.4 and the functionality (classes, methods) are explained below:

Ifmanet::initialized(): This method executes when the gateway node starts-up. It then creates the Universal Packet, Gateway id and registers the HelloTimer event handler for sending hello events.

Ifmanet::rt_resolve(): This function is invoked when a new packet is received by the node. This function checks the packet type and if it is “Hello” then it invokes the recvInitMsg() method.

HelloTimer::handler(Event*): This function registers the Hello Event to be triggered periodically at configurable time. This function is invoked only if “IFMANET_LINK_LAYER_DETECTION” switch is NOT defined in IFMANET.h

sendInitMsg(Packet *p): The HelloTimer() function invokes this method to broadcast the Hello message. This method populate the Universal packet, set its Message_Type=01 (Hello Message), assigns the Gateway_id, MANET_id, new Request_id, Sequence No and TTL (Time to live). It then creates an entry in GWRT with same information and broadcasts packet to IF-MANET Gateways.

recvInitMsg(Packet *p): This function is invoked by rt_resolve() function when a new message of type Hello (01) is received. This function then verifies the packet received and applies the validation checks as defined in Algorithm [3.2] and explained in Figure 4.4. If the data is valid and there is no route entry, in GWRT, with the same request_id and time then this function creates a new ifmanet_gwrt_entry. It then populates the GWRT entry from the universal packet received and store the record in GWRT i.e. ifmanet_gwrtable.

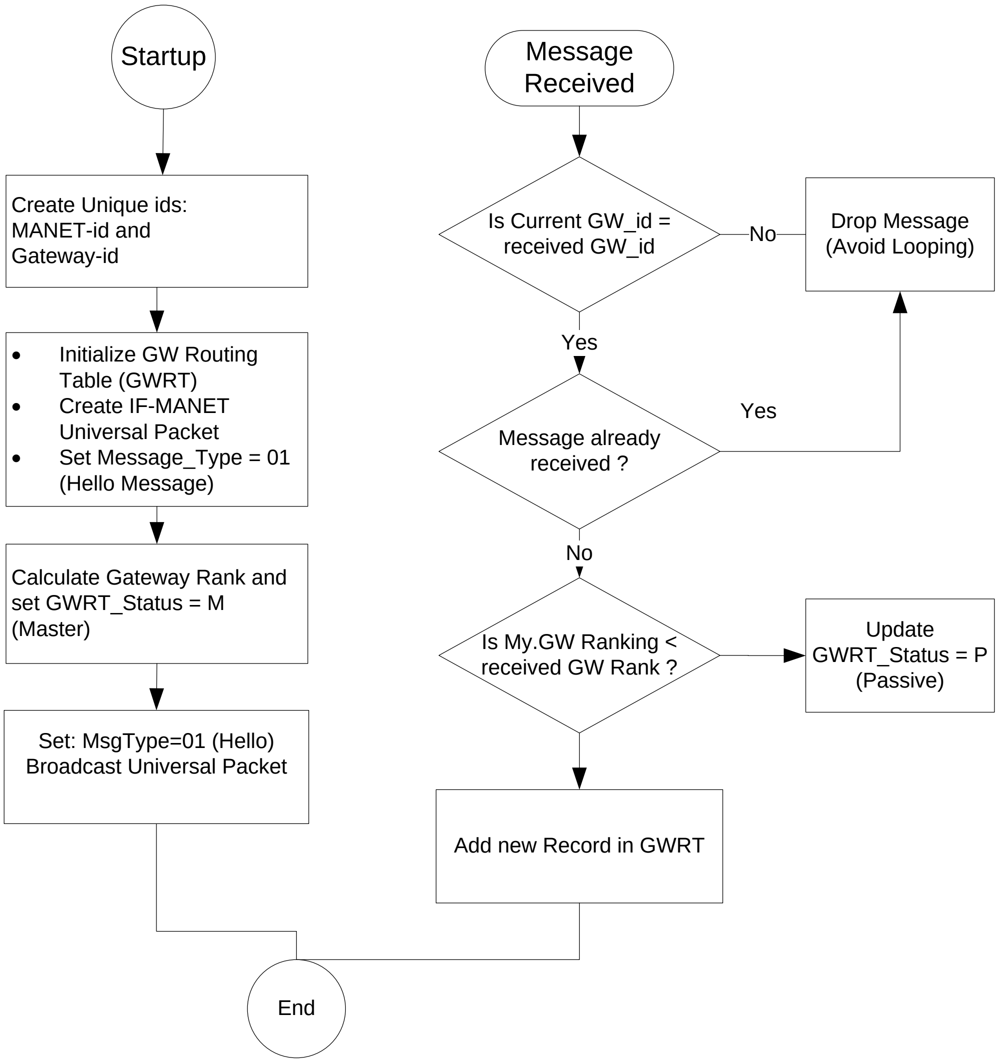


Figure 4.4: IF-MANET Initialization phase send and receive beacons

4.4 Message Translator

The message translator dynamically converts the packet of source routing protocol into the target routing protocol using the MANET Ontology. The IF-MANET node at start-up loads the existing vocabulary of MANET Ontology into the system. When the Message Translator receives a packet from a source node to convert it into a target packet, it first extracts the fields of the packet and find the similarities and differences between them. It then uses semantic mapping technique to generate the missing fields of source packet to create a target packet. The detail of the proposed IF-MANET Message Translator is explained in Section [3.6.5]. Figure 4.5 presents the implementation flow of the Message Translator Algorithm [Algorithm 3.1]. As shown in Figure 4.1, the message translator uses the ifmanet_translator.cc and ifmanet.cc classes to implement the packet translation. The ifmanet.cc invokes the ifmanet_translator.cc for converting the packets from source to target. The functionality of these classes to transform the heterogeneous packets using three phases of Ontological transformation is given below:

4.4.1 Ifmanet::Initialized()

This method loads the vocabulary of MANET Ontology into the global variable at initialization phase of the node.

var onto_vocabulary = LoadOntologyVocabulary() // Load existing Ontology

4.4.2 Ifmanet_translator:: t_recv(Packet* src, Packet* dst)

The ifmanet.cc class invokes t_recv() method and passes the source and the target packets as parameters for translation. This method then extracts the fields of the source packet, calls the discovery, match and model methods to convert it into the target packet. After conversion, it returns the new target packet back to the calling ifmanet.cc method. The sample code implementing the above logic shown below:

list<Fields> fields = Extract_Fields(src); // Extract fields of src packet*

t_discovery(Packet p, fields); // call to discover or learn new packet*

list<Fields, Fields> matched_fields = t_match(src, dst); // call match method to find similarities and differences between the Source and the target packets

Packet target_packet = t_model(matched_fields); // map the fields and generate the missing one to create a target packet*

return target_packet; // return packet back to calling method

4.4.3 Ifmanet_translator:: t_discovery(Packet* p, list<Fields>* f)

This method finds the classification of the source routing protocol. It queries the onto_vocabulary against the list of fields extracted from the source packet. If it finds the packet fields in the onto_vocabulary then the routing protocol is classified as "Identified" and if the field (s) doesn't exist in the vocabulary then the source routing protocol is classified as "un-identified". If the routing protocol is "un-identified", then this method learns the new packet by adding its fields and associated rules into the onto_vocabulary. The code snippet showing the above logic is given below:

If(! onto_vocabulary.contains(f) then {

pkt_format = definePacketFormat(p); // define packet format and rules

onto_vocabulary.add(fields, pkt_format); // add new packet into vocabulary }

4.4.4 Ifmanet_translator:: t_match (Packet P1, Packet P2)

This method compares the fields of source and target packets and produces a list of similarities and differences between them. The following is a code snippet showing queries to find the similarities and differences between source and target packets:

list<Fields> P1_Fields = Extract_Fields (P1);

list<Fields> P2_Fields = Extract_Fields (P2);

list<Fields> similarFields = P1.hasFields(P2_Fields).SelectDistinct();

list<Fields> differentFields = P2.dontHaveFields(P1_Fields).SelectDistinct();

OntoMatchFields matched_Fields = LoadMatchedFields(similarFields, differentFields)*

4.4.5 Ifmanet_translator::t_model(OntoMatchFields* matched_fields)

This method uses semantic based mapping to convert the source packet into the target packet. It uses the results of matched fields, applies rules defined in the MANET

Ontology and generates the missing fields in source packet to construct the target packet. The code snippet is shown below:

Rules rules = Onto_Vocabulary.LoadRules(matched_fields.MissingFields, targetPkt);*

Packet target_packet = MapSourceToTargetFields(matched_fields);*

return target_packet;

4.5 Summary

This chapter has presented the implementation of IF-MANET Interoperable routing protocol on NS-2 simulator. The chapter has introduced the NS-2 software simulator in brief and presented how to create and initialize mobile ad hoc nodes using TCL scripts. The proposed IF-MANET protocol has been written in C++ using Eclipse editor and was deployed on network simulator NS-2. The IF-MANET protocol drives the standard communication functionality (the layer below the application layer) of NS-2 mobile ad hoc protocol i.e. AODV to communicate with other mobile ad hoc nodes.

Furthermore, the implementation of the Universal Packet, GWRT, Initialization & Maintenance phase, Message Translator and Route Discovery algorithms were explained in detail and a relational class model diagram of IF-MANET architecture was presented. Thereafter, the methods and data structures implementing the functionality were explained with the help of code snippets and UML diagrams.

The next chapter simulates the implementation of the proposed IF-MANET routing protocol in NS-2 simulator and investigates the impact of evaluation criteria's on IF-MANET algorithms.

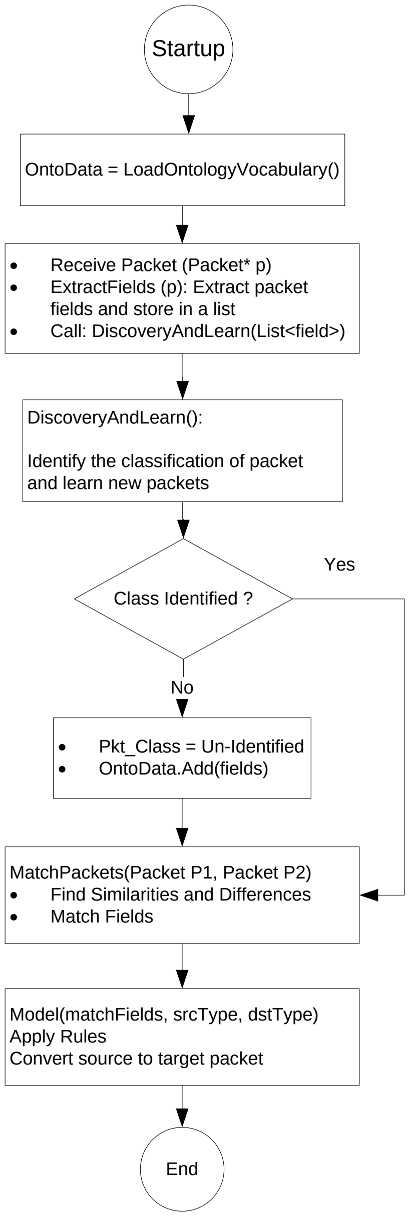


Figure 4.5: Message Translation Implementation flow

Chapter 5

Simulations and Results

5.1 Introduction

This chapter describes the implementation of the IF-MANET interoperable routing protocol, proposed in Chapter Three, evaluates its performance and compares the results against the existing available routing protocols. The thesis uses Network Simulator NS-2 to simulate the implementation of algorithms presented in Chapter Four and analyzes the results using MANET performance criteria's explained in Section 5.5.

It is assumed here that, the heterogeneous mobile nodes (MN) belonging to different MANET domains cannot communicate directly except through IF-MANET Gateway nodes. The Gateway node, on receipt of route discovery message from local MANET nodes, converts the message into an abstract message using the IF-MANET Message Translator and then broadcast that abstract message to Gateway nodes of external MANETs. The receiving Gateway converts the abstract message into a local native routing protocol, find the route to destination and forward the route reply message to the source Gateway.

In the following sub-sections, the route discovery mechanism, message translation and initialization (Bootstrap) process of the IF-MANET routing protocol will be investigated and evaluated against the performance metrics described in Section 5.5.

This chapter also defines the parameters for simulation and investigates the performance of the IF-MANET routing protocol. It randomly deploys the mobile ad hoc nodes and Gateways to form a multi-hop MANET and uses the Random Way Point Mobility model (Resta et al. 2003) to move them randomly in a simulation field. The scenarios for distributing nodes randomly and moving with different speeds are created with the Setdest utility of NS-2. The NS-2 is setup with IEEE 802.11 physical interface using 11 Mbps channel capacity and 250 meters transmission range. The hardware for the simulations consists of Intel Core i7 Quad Processor with 1.4 GHZ each, 2 GB RAM and Linux Ubuntu Operating System. The complete list of baseline parameters for simulations in this chapter is provided in Section 5.3.

For proof of concept and evaluating the connectivity of IF-MANET in heterogeneous MANETs, this thesis uses two different type of routing protocols i.e. AODV and MAODV along with IF-MANET routing protocol. Figure 5.1 shows the deployment scheme of different routing protocols in different MANET Clusters. The AODV nodes are deployed in Cluster-A, MAODV in Cluster-B whereas IF-MANET nodes are deployed in such a way that they can communicate with both internal and external MANET nodes. The placement and transmission range of AODV, MAODV nodes are configured in such a way that they cannot communicate with external cluster nodes except via IF-MANET Gateway nodes.

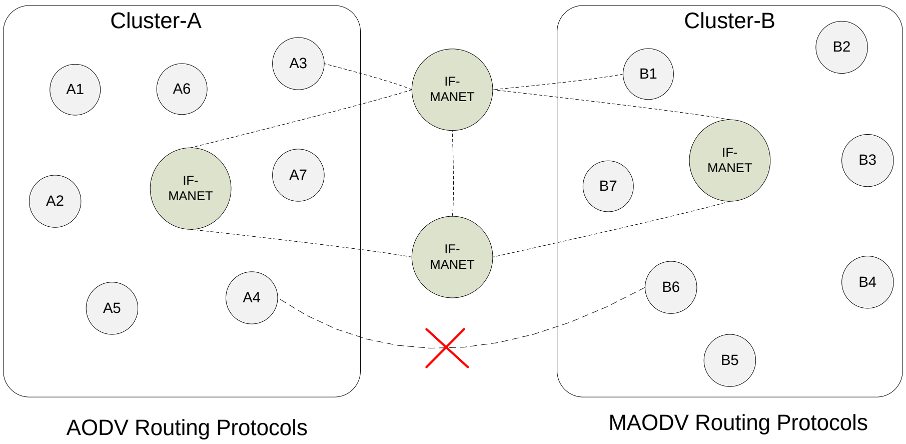


Figure 5.1: IF-MANET Simulation Field Map

After explaining the simulation environment, the chapter defines the methodology to analyze and evaluate the performance of the IF-MANET routing protocol. This chapter uses different simulation models: 1) Mobility Model with different node speeds, 2) Topology Model with different mobile nodes and gateway densities, to investigate the efficiency of the routing protocol. It then uses the MANET performance evaluation criteria's i.e. Connectivity (Packet delivery ratio), Average End-to-End delivery and Normalized routing load (Overheads) against different simulation models to evaluate the impact of varying node speeds and densities on the performance of the IF-MANET. It then analyzes the results of the performance metrics and compares them with two baseline evaluation expectations presented in section 5.6 i.e. 1) Low performance (no connectivity), 2) High performance (using single routing protocol) to evaluate the overall performance especially the connectivity of IF-MANET interoperable routing protocol. The organization of rest of this chapter is shown in Figure 5.2 below:

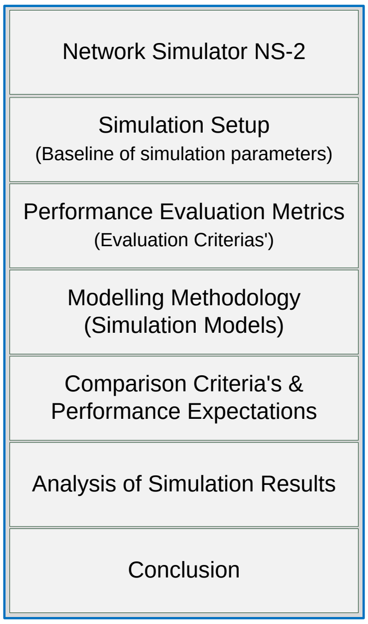


Figure 5.2: Organization of Chapter: Simulations

5.2 Network Simulator (NS)

NS 2.35 is selected as a network simulator mainly because of the range of features it provides and above all it has an open source code that can be modified and extended. NS-2 is an object-oriented, discrete event simulator for networking and routing protocols. It provides extensive support for simulating background traffic e.g. TCP/IP, CBR/UDP, uni/multicast routing over wired and wireless networks and implementation of a wide range of mobile ad hoc network routing protocols.

NS-2 is developed in C++ with an OTcl interpreter for command and configuration interface. The complex protocols are implemented in C++ for fast processing whereas OTcl, quick to change, creates commands and configuration for simulation. The main advantage of this dual programming approach is that it enables to quickly create large scenarios without updating, compiling and executing the C++ code. For IF-MANET, this thesis has used NS-2's base functionality for MANET communication (i.e. from physical to network layer) and implemented the algorithms of interoperable routing at application layer.

5.2.1 Justification of using the NS-2 Simulator

Figure-1 shows the ratio of most widely used network simulations for evaluating the wireless network research works. The NS-2 with 38% is the most popular and highly used simulator among the widely used network simulating tools. The MATLAB is second highly used tool but is based on the analytical method of evaluation. The other simulator tools (8%) includes self-developed or custom made simulation tools, TOSSIM, Montecarlo, J-SIM etc

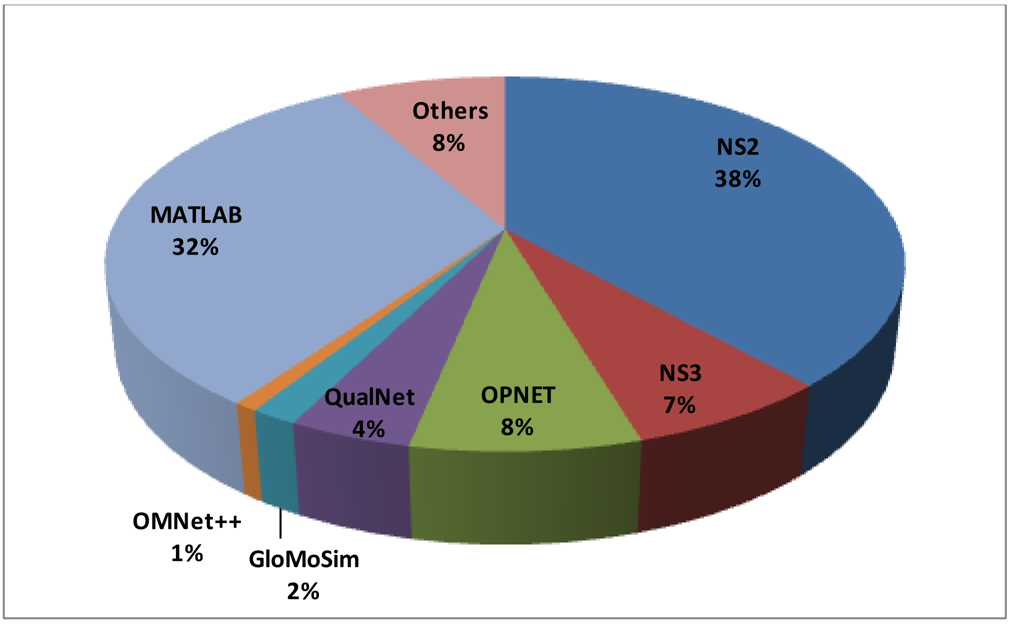


Figure 5.3: Simulators Used in Selected IEEE Journals and Conferences papers published

Table-1 below compares the strengths and weaknesses of six most widely used network simulating tools. The selection criteria of simulator, for the evaluation of the IF-MANET, are based on the following features:

1. Open Source License Type
2. Availability of MANET Protocols Implementation
3. Running Heterogeneous protocols in single environment (plane)
4. Customize/Change existing protocol functionality
5. Documentation and Community Support

Though commercial tools are features enriched and provides free academic version but the free versions are limited in functionality and doesn't include support. While implementing the IF-MANET in the commercial tool "OPNET", we faced the issue of running heterogeneous protocols in a simulation plane, customising the functionality of existing MANET routing protocols and above all there was no support available from OPNET in the academic version. In addition, the source code was not available to understand the behaviour and extend the MANET functionality to produce interoperability between heterogeneous protocols. It was then decided to use Open Source tool as they provide the source code which will make it easy to understand the logic of protocols and hence customize them.

We then evaluated the NS-3 simulator as it was upgraded version of most widely used tool ns-2 and claims to object oriented, extendable and efficient with small memory and processing foot prints. However, we found that it is not backward compatible to NS-2 , therefore huge implementation of ns-2 based MANET protocols implementations are not valid in ns-3 and the new implementations are not available or are in Beta phase. In addition, there is not a big user community of NS-3 and therefore it requires active maintainers to respond user questions like in OPNET and QualNet.

By comparing the simulation tools presented in Table-1, it is evident that the ns-2 meets all the five selection criteria’s mentioned above. Also from Figure-1, the ns-2 is the most widely used tool for simulating the networks and therefore has huge user community to get support quickly. There is large number of scripts available for processing the output trace files of ns-2 and huge library of existing and continuously arriving implementations of MANET routing protocols. Therefore, based on the above facts, this thesis has selected the NS-2 tool for simulating the IF-MANET along with other MANET routing protocols to provide interoperability between them.

#	Features	Simulation Tools					
		NS2	**NS3**	**OPNET**	**GloMoSim**	**QualNet**	**OMNet++**
1.	License	Open Source	Open Source	Commercial (Limited functionality free for Academics)	Open Source	Commercial	Commercial (Free for Academics)
2.	Interface	C++/OTCL	C++/Python	C++	C	C	C++/NED
3.	Mobility Support	Yes	Yes	Yes	Yes	Yes	No
4.	Scalability	Medium	Large	Large	Large	Large	Large
5.	Documentation	Excellent	Good	Excellent	Poor	Good	Good
6.	User/Community Support	Excellent	Fair	Poor	Poor	Fair	Fair
7.	GUI	Limited	Limited	Good	Limited	Good	Good
8.	MANET Protocols Implementation	High	Low	Medium	Low	Low	Low

9.	Heterogeneous protocols in single environment (plane)	Yes	Yes	No	No	No	No
10.	Customize/Change existing protocol functionality	Yes	Yes	No	Yes (Difficult)	No	No
11.	Ongoing Development and Support	Yes	Yes	Yes	No	Yes	Yes

Table 5.1: Comparison of Network Simulators

5.3 Simulation Setup

This sub-section defines the simulation parameters and their default values for simulating the routing protocols in this chapter. The initialization parameters that are common to all the simulation scenarios are presented in Table 5.2 whereas the parameters specific to simulations scenarios are presented in their respective sections. The TCL script is used to create and initialize these variables, define topology and create mobile nodes.

For simulations in this thesis, the nodes are randomly deployed in a 500 x 500 m area as shown in Figure 5.1. The transmission range, which is a communication distance between two mobile nodes, is set to 250m. It was found during the simulations that the two nodes don't communicate with each other even if the distance between them is 251m. The mobile node uses a DropTail queue, which is used to buffer the packets between sender and receiver. The DropTail queue is used because it is of type waiting queue and it drops the new entered packets if the queue overflows, similar to buffer of general realistic network. For optimal queue length, we performed few simulations, with varying the size of the Queue length, and compared the impact of size on the performance provided. We found that if the queue size becomes smaller, the system drops packets quickly and hence increases the packet loss ratio. Whereas if the queue size is very large then the system buffer the packets between source and destination for long time and hence increase's the End-to-End Delay of packet delivery. In addition, the higher the size of the queue, the higher is the memory consumption. The higher memory consumption leads to high processing power and battery consumption. After comparing the performance, the queue length for the simulations was set to 30 packets because at this size the system performance was optimal i.e. fewer packets loss and low memory utilization. The physical interface for wireless communication used in this simulation is IEEE 802.11 and the Traffic type used is (Constant Bit Rate) CBR. The simulation time for the execution of each simulated scenario is 200 sec whereas each scenario is simulated for three times with the same configuration and their mean is taken for optimal values.

Table 5.2 below, shows the list of parameters with default values for all scenarios whereas the values in blue changes according the simulation scenario.

Parameter	Value
Channel Type	Channel/WirelessChannel
Propagation Type	Propagation/TwoRayGround
Physical Type	Phy/WirelessPhy
Mac Type	Mac/802_11
Queue Type	Queue/DropTail/PriQueue
Queue Length	30
Traffic Model	Application/Traffic/CBR (Constant Bit Rate)
Traffic Agent	Agent/UDP
Packet Size	512 bytes
Packet rate	5 packets/s
Start Simulation	5 sec
Simulation Time	200 sec
Number of mobile nodes	10, 25, 50, 100
Number of Gateways	5%,1 0%, 20% of mobile nodes
Number of Sources	5
Topology Size	500x500m, 1000x1000m
Mobility Model	Random Way Point
Transmission Range	100m, 250m
Routing Protocols	AODV, MAODV, IF-MANET
Antena	Antenna/OmniAntenna
Phy/WirelessPhy Frequency	2.472e9
Phy/WirelessPhy RXThresh	2.62861e-09
Phy/WirelessPhy Bandwidth	11; # Mbps

Table 5.2: Initialization Parameters for all Simulations

5.4 Simulation Models

This thesis uses different models to simulate the routing protocols and uses different values to investigate and evaluate the performance of IF-MANET routing protocol. The following are the MANET models used in this thesis in order to evaluate the IF-MANET's performance:

5.4.1 Mobility Model

It provides the movement pattern through which mobile nodes can move within the defined topological plane. This thesis uses "random waypoint" model to randomly move the nodes within the simulation area. In random waypoint, the nodes moves towards a randomly selected destination with a random movement speed after a random start time. Once the node reaches the destination, it pauses for a defined wait time and then chooses another random destination within the simulation plane. This random movement pattern is repeated for the complete duration of the simulation.

The pause time is set to 20 sec and the movement speed varies between 1m/s and 20m/s. The random movement patterns are generated by using NS-2's movement generating utility "Setdest". To reflect the realistic pattern, this thesis has modified the "setdest" utility to define the lower and upper bound of simulation plane so that the nodes can move only within their own MANET cluster. For Example, in this case the simulation area of 500m x 500m is divided into two clusters (0,0 to 200,200 and 300,300 to 500,500) where each cluster is running different type of routing protocol. But the Setdest utility doesn't allow to be defined the lower bound of axis and takes a default starting value i.e. 0, 0. To address this issue and define different cluster plane for separate type of routing protocols, we have modified the utility to accept and process the lower bound of axis while generating the mobility movement pattern.

5.4.2 Topology Model (Impact of Node and Gateway Density)

The number of mobile nodes within a unit area is called 'node density' and plays an important role in evaluating the connectivity of ad hoc networks. This thesis investigates the impact of mobile node density by using different number of nodes within MANET clusters. The simulation uses different number of mobile nodes i.e. 25, 50, 75 and 100 to evaluate the impact of node density on the performance of the IF-MANET.

The other factor which directly influences the performance of IF-MANET routing protocol is the density of Gateway nodes. The thesis uses different number of Gateway nodes (percentage of mobile nodes) i.e. 10%, 20%, 30% in addition to mobile nodes to investigate the impact of Gateway nodes in relation to connectivity in heterogeneous routing protocols.

For Example, in a plane of 500 × 500 metres the probability of finding multi-hop route to destination increases with increasing the number of nodes. But it is not true if node density increases a certain threshold as it creates more network overheads, interferences and hence decreases the connectivity. To find an optimal value of node density, simulations were carried out with different number of nodes. The connectivity ratio, i.e. the number of successful established connections, is then investigated to find out the best suitable value of node density.

5.4.3 Traffic Model

Traffic model defines the type for background traffic in order to stress the ad hoc network and examine the behaviour of protocols in communication. The background traffic is set between source and destination nodes where the nodes remains same within a simulation scenario but changes their position randomly according to random way point model.

This thesis uses constant bit rate (CBR) over UDP as a traffic source. The source nodes generates packets after 0.5 sec i.e. each node generates 2 packets per second. The payload of each packet is 512 bytes thus the amount traffic generated by each source node is 2x512x8 bit/s = 8kbit/s. The background traffic starts with the start of simulation run i.e. at 5 seconds and end with the simulation stop time i.e.200 seconds.

To generate the background traffic between source and destination, the thesis has used traffic generator utility '**cbrgenl**' by CMU. The input parameters to the utility are: number of total nodes, maximum connections, packet rate and stop time. The thesis has modified the **cbrgen** output pattern to explicitly define the source and destination nodes which belongs to different routing protocols. It is because the **cbrgen**, by default, randomly selects the source and destination nodes which might be from same MANET. Whereas to simulate the realistic scenarios of interoperability between heterogeneous routing protocols, the source and destination nodes must belongs to different MANETs.

5.5 Performance Evaluation Metrics

This section identifies the different quantitative criteria's to evaluate and compare the performance of IF-MANET routing protocol approach. These are the key metrics to evaluate the MANET routing protocols and are extracted from RFC2501 (Corson & Macker 1999).

5.5.1 Connectivity (Packet Delivery Ratio (PDR))

It is the ratio of data packets received by the destinations to those generated by the sources. The PDR evaluates the connectivity of nodes in heterogeneous MANET which is the main objective of the proposed IF-MANET interoperable routing protocol. In multi-hop MANET each intermediate node generates a packet on behalf of the source node and forwards along the path to destination. Hence there will be many routing control packets for a single route request message. This thesis counts each unique route request as a single transaction (from source to destination) to find out the real number of successful connections. It uses the following formula to calculate the PDR:

PDR = (Total Packets Received / Total Packets Sent) x 100

(Where Total = Sum of all packets)

5.5.2 Normalized Routing Load (NRL) (Overhead):

The total number of routing packets transmitted per data packet delivered at the destination. It provides the performance evaluation in terms of delivering the total number of packet to establish a connection and calculate the protocol overheads in performing these actions. Here, each transmission of a routing packet over multi-hop MANET is counted as one transmission. NRL is calculated using the following formula:

NRL = Routing Packets Transmitted / Received Packets

5.5.3 Average End-to-End Delay

It is the average time data packets take to be transmitted across a MANET from source to destination. It includes all possible delays caused by buffering during the route discovery latency, queuing at the interface queue, retransmission delay at the MAC, propagation and transfer time. The formula to calculate this delay is given below:

Delay = Total Receive Time – Total Sent Time
(Where Total = Sum of Average times)

5.6 Comparison Criteria for Performance Evaluation

The purpose is to create the benchmark for performance metrics defined in Section 5.5 and compare the results of IF-MANET protocol against the benchmarked values. This thesis has simulated the AODV and MAODV routing protocols under the simulation setup defined in Section 5.3 and created the following two baseline cases in order to compare their performance with IF-MANET routing protocol.

5.6.1 Case-1: Lower End Performance:

There will be N number of MANETs and each MANET contains S number of mobile nodes. Different MANETs contains different type of routing protocols whereas nodes within same MANET are of same type. Also, there will be no IF-MANET Gateways and therefore there will be no (zero) connectivity across heterogeneous MANETs.

5.6.2 Case-2: High End Performance:

There will be only on MANET and each node uses the same routing protocol. This scenario will achieve the highest performance as similar nodes can communicate directly and route multi-hop packets via shortest path.

5.7 Performance Expectations

The main aim of this research is to achieve the high connectivity between heterogeneous routing protocols at the cost of moderate overheads. For comparison, there is no direct performance relationship between case-1 and case-2 as the first case uses heterogeneous MANETs but without connectivity whereas the 2nd case provides maximum connectivity but no interoperability. Based on the above analyzes, our expectation from IF-MANET stands in-between these two cases i.e. 50% connectivity at the cost of 60-70% overheads as compared to AODV. It is because of the fact that IF-MANET uses Gateway nodes for interoperability. The Gateway nodes increases a hop count in multi-hop MANET, provides packet translation to/from native routing protocols and broadcast route discovery request to external MANETs.

5.8 Simulation Results

This section simulates the performance evaluation criteria's explained in Section 5.5 using different simulation modelling parameters from Table 5.3. This section presents three simulation scenarios where each scenario executes one of the three simulating models i.e. Node Density, Gateway Density and Mobility Model (nodes speed). The three performance evaluation criteria's i.e. PDR, NRL and E2E Delay are then analyzed and calculated for each of the simulation scenario i.e. simulation model. Further, each scenario executes simulations for two type of use cases: 1) a MANET with N nodes where each node is running only AODV protocol, 2) two MANET clusters with N nodes where one MANET contains AODV and the other contains MAODV protocols. The IF-MANET Gateway nodes are distributed randomly between the two MANETs. The simulation field graph showing multiple MANET clusters with IF-MANET Gateways as shown in Figure 5.1. The results of the simulations are then evaluated using the criteria described in section 5.6 and compare against the performance expected in section 5.7.

Each simulation is executed for three times using the same simulation model and their average values are taken for optimal results. The output data of simulated scenarios is analyzed and processed to calculate the performance metrics i.e. PDR, NRL and E2E Delays.

The graphs in this chapter are using 3 to 4 data points to evaluate the behaviour of an ad hoc networks and an impact of the IF-MANET protocol in providing the connectivity. The rationale behind using few points is that the data had a sequential pattern after the 3rd data point and therefore the graphs after the 3rd point was a smooth continuous graph with no "sudden jumps" or breaks.

The following Table 5.3 provides a matrix of different values used in different simulation scenarios and describes the legends used in graphs. For Example, "IF-MANET: G=10%" means that the simulation includes AODV, MAODV and IF-MANET protocols with Gateway nodes equals 10% of the total mobile nodes. "IF-MANET: N=25" means a scenario that includes 25 nodes along with other fix and variable simulation parameters.

Legend	**Protocol(s)**	**Mobile Nodes**	**Gateways % of Nodes**	**Speed m/s**	**Plane (m x m)**	**Range (m)**
AODV	AODV	25, 50, 75, 100	0	5,10,15	500x500	250
AODV, MAODV	AODV, MAODV	25, 50, 75, 100	0	5,10,15	500x500	250
IF-MANET: G=10%	AODV, MAODV, IF-MANET	25, 50, 75, 100	2, 6, 8, 10 (10% of nodes)	5,10,15	500x500	250
IF-MANET: G=20%	AODV, MAODV, IF-MANET	25, 50, 75, 100	4, 12, 16, 20 (20% of nodes)	5,10,15	500x500	250
IF-MANET: N=25	AODV, MAODV, IF-MANET	25	2, 4, 6 (10/20/30%)	5,10,15	500x500	250
IF-MANET: N=50	AODV, MAODV, IF-MANET	50	5, 10, 15 (10/20/30%)	5,10,15	500x500	250
IF-MANET: N=75	AODV, MAODV, IF-MANET	75	8, 15, 23 (10/20/30%)	5,10,15	500x500	250
IF-MANET: N=100	AODV, MAODV, IF-MANET	100	10, 20, 30 (10/20/30%)	5,10,15	500x500	250
Keys: G = Gateways and N = Nodes						

Table 5.3: Parameters for different Simulation Scenarios

5.8.1 Scenario-1: Impact of Node Density

This section evaluates the impact of different number of mobile nodes on the performance of the IF-MANET interoperable routing protocol. To evaluate the performance and efficiency of IF-MANET algorithms presented in Chapter 3, this section uses the three simulation evaluation criteria's mentioned in Section 5.5. This section uses the following values for the simulation parameters defined in Table 5.2:

- Field Plane: 500x500 m²
- Transmission Range=250m
- Mobile Nodes =25,50,75,100
- Gateway Nodes=10%,20% of mobile node
- Mobility Model= Random Walk Mobility Model with nodes moving at 5m/s
- Simulation Time = 200 sec

The following sub-sections analysis the behaviour of proposed routing protocol under different node densities

5.8.1.1 Criteria-1: Connectivity (Packet Delivery Ratio (PDR))

Figure 5.4 shows the packet delivery ratio at different node densities i.e. 25, 50, 75 and 100 nodes. It then evaluates the connectivity between heterogeneous routing protocols (AODV, MAODV) using IF-MANET Gateways and compares them with that of single protocol i.e. "AODV" and multiple protocol (AODV, MAODV) without IF-MANET. In this simulation, the nodes are distributed in two domains (MANETs) of equal sizes and number of nodes where each MANET contains the nodes using same protocol as shown in simulation field map Figure 5.1.

As the Figure 5.4 shows, the AODV, as expected, has obtained the connectivity greater than 90% especially when the node density was higher whereas the connectivity is zero (0%) between heterogeneous routing protocols (i.e. AODV, MAODV) while communicating directly. The zero connectivity is due to the fact that both the protocols do not understand each other’s packets and hence drops the messages.

From Figure 5.4 it is evident that the connectivity has been achieved between heterogeneous routing protocols using IF-MANET Gateways, which is the main objective of this research. The connectivity ratio achieved is comparatively less at low node density irrespective of gateway nodes density as compared to high node density. It implies that; at fewer nodes, the MANET has to create and maintain long multi-hop routing links to reach the destination nodes via IF-MANET Gateways and hence decreased connectivity. From the graph, it is clear that the percentage of connectivity increased rather quickly in dense networks. For example, at node density of 50 - 75 nodes, the percentage of connectivity achieved, using IF-MANET Gateways, was greater than 70% which is 10% less than that of “AODV” whereas 80% more than that of AODV, MAODV connectivity. The decrease in connectivity as compared to AODV protocol is due to the fact that the IF-MANET performs dual functionality of native routing protocol as well as Gateway node. In addition, the Gateway has to perform extra steps of finding and communicating with Gateways of other MANETs, translate the packets and maintain the GWRTs. Also, it can be seen that there is a slight decrease in connectivity when node density reaches at 100 nodes along with increase in gateway nodes (because number of gateway nodes are directly proportional to mobile node density). It is because of high noise and overheads caused due to collisions of broadcast messages in dense networks.

By comparing the connected ratio of IF-MANET protocol with that of the expected results in section 5.5, the IF-MANET has achieved the connectivity up-to 80% which is far better than the benchmarked value of 50%. Hence, it is proved from the simulations that under optimal number of gateways and mobile nodes the IF-MANET can outperform the connectivity in heterogeneous MANETS.

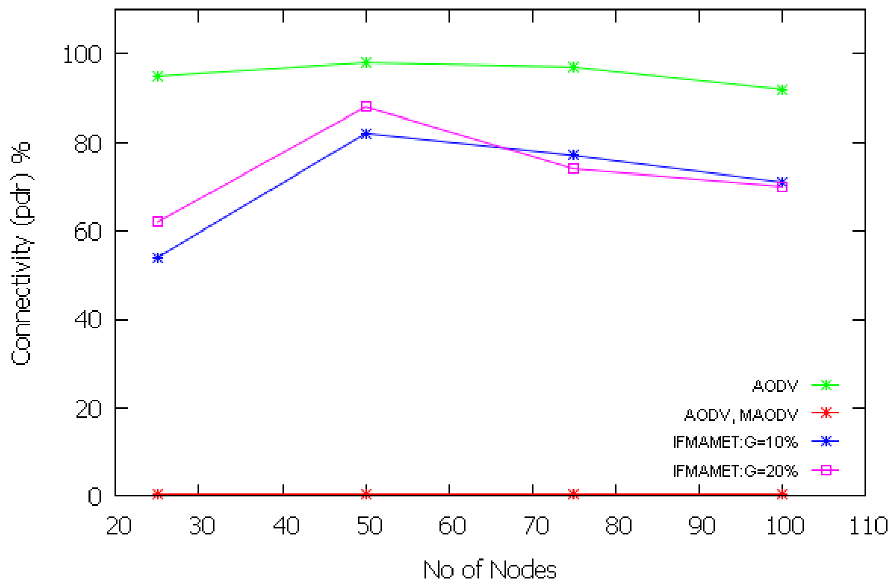


Figure 5.4: Connectivity vs Node densities

5.8.1.2 Criteria-2: Average End to End Delay

Figure 5.5 shows the average end-to-end delay between different node densities i.e. 25, 50, 75 and 100 nodes and compares the transmission delay of IF-MANET packets with that of AODV and AODV-MAODV simulation scenarios. For IF-MANET scenario, the simulation uses varying gateway nodes i.e. 10%, 20% of mobile nodes. From Figure 5.5 it can be seen that, the end-to-end delay for AODV protocol is less for low node density and increased with increase in node density. It is because, higher number of nodes creates more shorter routing paths at each node which leads to higher processing times to process packets at each node and hence increases the delay time.

As seen from Figure 5.5, the transmission delay for IF-MANET is low (under 0.4 sec) up-to the node density of 75 nodes (i.e. 0.4sec is less than 50% of total end-to-end delay time of 1 sec) as compared to the high node density i.e.100 nodes. As compared to AODV, the average delay in IF-MANET is greater than 10% at 25 nodes and 30%

between 50 to 75 nodes. The high transmission delay of IF-MANET is because of the fact that the IF-MANET Gateways requires: 1) extra number of hops in multi-hop, multi-cluster MANETs, 2) high number of broadcast messages to discover neighbouring MANET Gateways and 3) Gateway nodes requires high processing in translating packets from one protocol to another and maintaining the GWRTs, as explained in route discovery algorithms in section [3.6.7]. Due to these reasons, the IF-MANET has shown increased end-to-end delay especially at high density of mobile and gateway nodes.

The IF-MANET has shown optimal performance, by taking minimum time to transmit the data packets to the destinations at up-to node density of 75 nodes, irrespective of number of Gateway nodes. It implies that by increasing the node density the network connectivity increases but decreases packet delivery performance. It is encouraging that, despite increased packet transmission delay, the performance of connecting heterogeneous routing protocols with the IF- MANET is much higher than our expectation in Section 4.3.

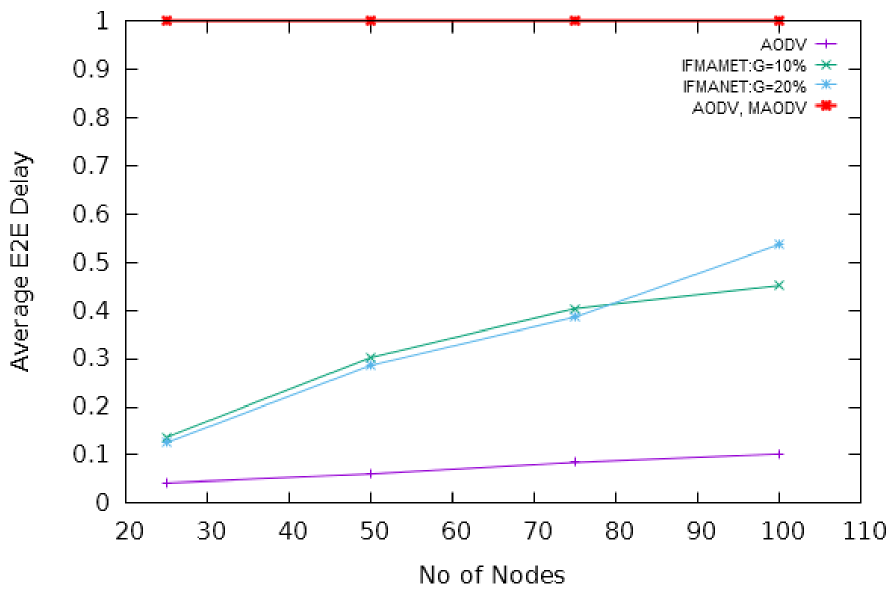


Figure 5.5: Avg. End-to-End Delay (Sec) vs Node Density

5.8.1.3 Criteria-3: Normalized Route Load (NRL)

Figure 5.6 shows the IF-MANET routing overhead with mobile node densities between 25 and 100 nodes and Gateway nodes. As shown in Figure 5.6, the IF-MANET overhead is comparatively greater than AODV but much less than AODV-MAODV. This is an expected result as the IF-MANET approach periodically broadcast the gateway information to maintain the inter-Gateway information and abstract messages for route discovery in heterogeneous MANETs.

It is clear from Figure 5.6 that the IF-MANET generates less routing overheads at low node density whereas the routing overheads increases with increase in mobile and gateway node densities. This behaviour of IF-MANET is due to the fact that by increasing the mobile nodes increases the Gateway nodes (Gateway nodes are proportional to MN) in the network. Although increased number of Gateways increases packet delivery ratio but it cost more routing overhead. As explained in route discovery algorithms in Section [3.6.7] the reason of increased routing overheads in IF-MANET are due to following reasons:

- Firstly, the Gateway listens to internal route requests and doesn't activate its functionality until the RREQ counter reaches the RREQ_Threshold. The behaviour causes internal nodes to keep on re-broadcasting the route requests and hence creates extra control overheads
- Secondly, the Gateways broadcast and re-broadcast inter-MANET Route Discovery (RREQ) and Route Responses (RREP) messages on behalf of local nodes which causes very much overhead

By comparing the results with our expectation of achieving 50% NRL, in section 5.7, the IF-MANET has reduced the NRL up-to 20% between mobile nodes 25 to 50 irrespective of Gateway nodes. It implies that the IF-MANET has achieved high connectivity across heterogeneous protocols by keeping the overheads low.

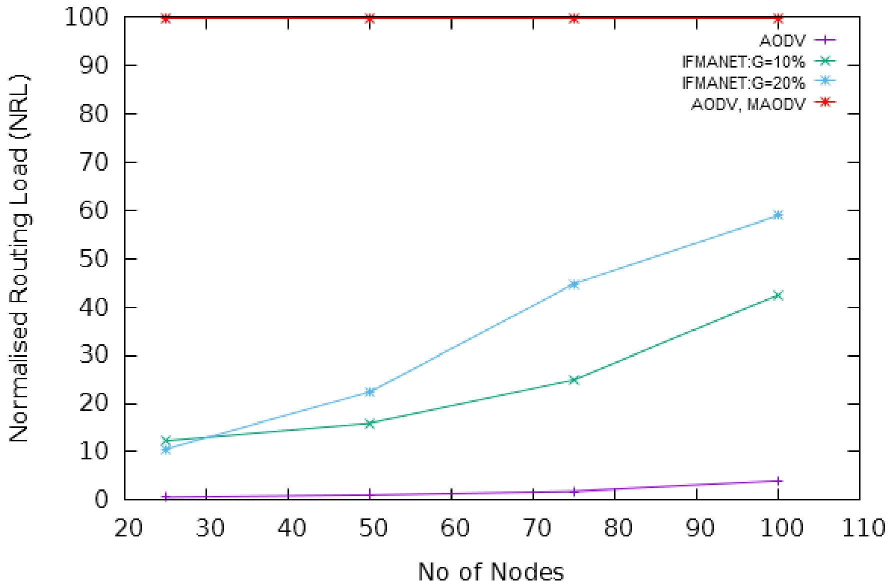


Figure 5.6: Normalised Routing Load vs Node Density

5.8.2 Scenario-2: Impact of Gateway Density

This section evaluates the impact of varying number of Gateway nodes on the performance of IF-MANET interoperable routing protocol. It then analysis the efficiency of algorithms presented in Chapter 3, by using the three simulation evaluation criteria's mentioned in section 5.5. As the AODV based MANET has no gateway nodes, therefore the evaluation of each criteria compares the nodes of different densities at different number of Gateway nodes.

For simulation this section uses the following parameters:

- Field Plane: 500x500m^2
- Transmission Range=250m
- Mobile Node Densities=25,50,75,100 nodes
- Gateway Densities=10%,20%,30% of mobile node
- Mobility Model= Random Walk Mobility
- Simulation Time = 200 sec

5.8.2.1 Criteria-1: Connectivity (Packet Delivery Ratio (PDR))

Figure 5.7 presents the impact of having different number of gateways (represented as percentage of nodes) in the MANET to that of packet delivery ratio (Connectivity). The figure plot the connectivity results with different node densities (25, 50, 75, 100 nodes) at different IF-MANET gateway nodes.

Figure 5.7, shows the percentage of connectivity increases with increase in number of gateway nodes especially in dense networks between 50 and 75 nodes. In particular, for networks with 50 to 75 nodes, the connectivity becomes more than 80% with 20-30% Gateway nodes. Whereas, the connectivity at 25 nodes was comparatively low irrespective of the Gateway density.

As described in route discovery algorithms in Chapter 3, that the inter-MANET nodes cannot communicate with each other except through IF-MANET Gateway nodes. The Gateway nodes are responsible of translating the native protocol packets into interoperable packets using IF-MANET's Message Translator and broadcasting to inter-MANET Gateways for route discovery. It justifies the low connectivity at low gateway node density as there are few, resource limited, mobile gateways to process the high traffic between external MANETs. For Example, at node density of 25 nodes, there is only one Gateway (10% of 25 nodes equal's 2.5 gateway node) per MANET responsible of translating the messages and communicating with external MANETs.

On average the connectivity achieved by IF-MANET in heterogeneous routing protocols is greater than 65% irrespective of the node density and attained more than 70% between 50-70 nodes and 30% gateway density. By comparing the results with the expected results of 50% connectivity in section 5.7, the IF-MANET has performed very well and has achieved connectivity more than 80%.

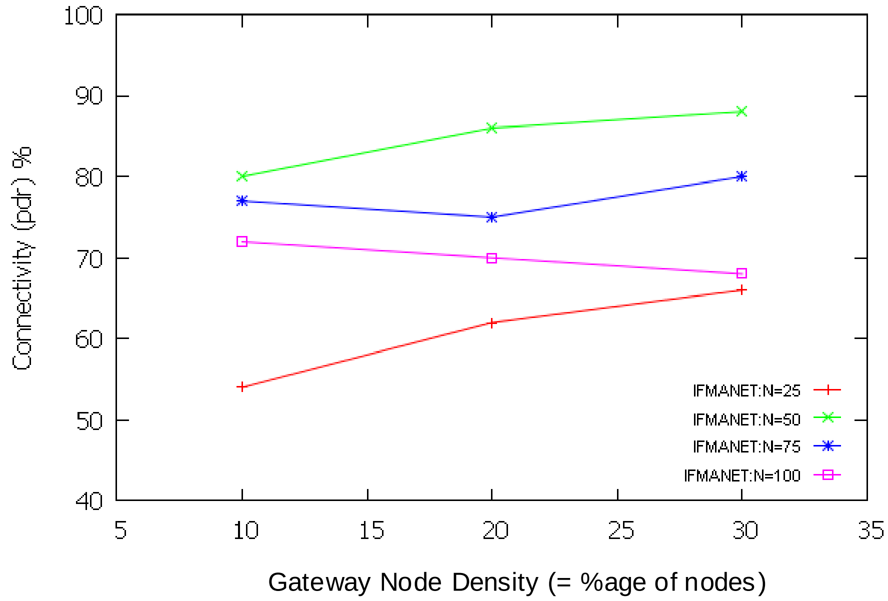


Figure 5.7: Connectivity vs Gateway node densities

5.8.2.2 Criteria-2: Average End to End Delay

Figure 5.8 presents the impact of having different number of gateways and node densities to that of Average End to End Delay (in Seconds). The simulation uses mobile node densities from 25 to 100 nodes with IF-MANET gateways between 10 to 30% of mobile nodes.

Figure 5.8 shows that the end-to-end delay is low at low mobile and gateway node densities and increases with increase in node densities. By comparing the delays caused by different node densities, it can be observed that the transmission delay becomes significant by adding more nodes where there is less or no impact of adding more Gateway nodes. In particular, the transmission delay is below 0.2 sec at 25 nodes irrespective of Gateway nodes whereas it goes above 0.5 sec at 100 nodes, which is approximately 30% increase. It is because of the fact that IF-MANET requires extra hops and high number of broadcast messages to discover destination in external MANETs. This behaviour is explained in algorithms proposed in Chapter 3.

In general, by comparing the above results with the expected results in section 5.7, the IF-MANET has achieved the high connectivity by keeping the end-to-end delay lower than the expected delay of 50% (0.5 sec) even at high node and gateway densities.

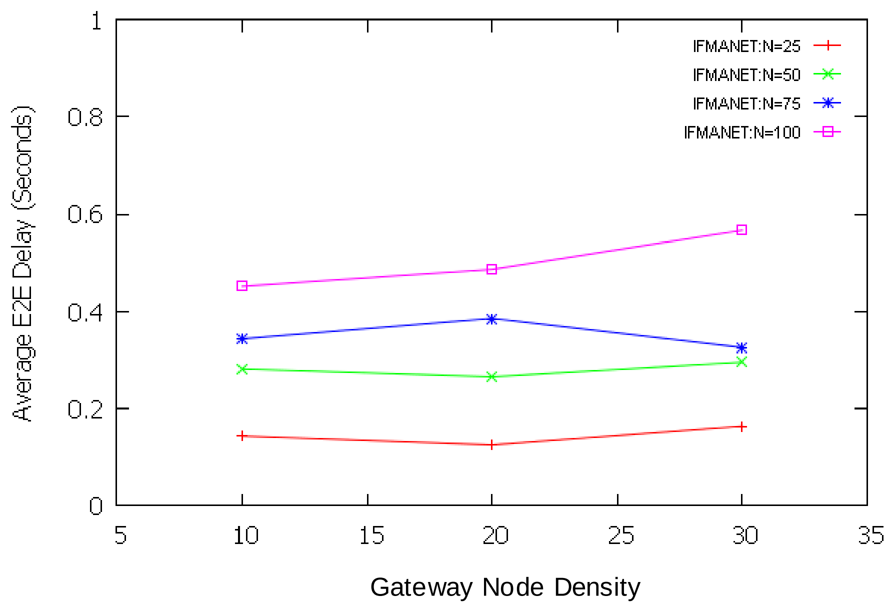


Figure 5.8: Average End-to-End Delay vs Gateway Node Densities

5.8.2.3 Criteria-3: Normal Route Load (Protocol Overhead)

Figure 5.9 evaluates the impact of gateway node densities to that of NRL (routing overheads). It is clear from the graph that, when node density increases it increases the routing packet overhead. Whereas the impact of Gateway node density as compared to the mobile node density is less i.e. it creates less NRL. For example, the maximum difference between NRL is 45 due to node densities between 25 and 100 nodes whereas it is 20 due to Gateway densities between 10% and 30% at node density of 100 nodes. Therefore, we can say that the IF-MANET Gateway nodes are creating fewer overheads than the native nodes itself. In particular, the node density of 100 mobile nodes with 30% of Gateway nodes has created a transmission overhead 60 which is a maximum compared to other densities. It is because of the fact that the size of the routing packet to neighbouring MANETs becomes bigger because of more possible routing path entries in the domain.

By comparing the results in Figure 5.9, it is evident that by increasing the Gateways nodes increases the packet delivery ratio (connectivity) but it costs more transmission overheads. The reason is that each MANET Gateway processes the routing discovery on behalf of local routing protocol and needs to broadcast such information to its neighbouring MANET Gateways.

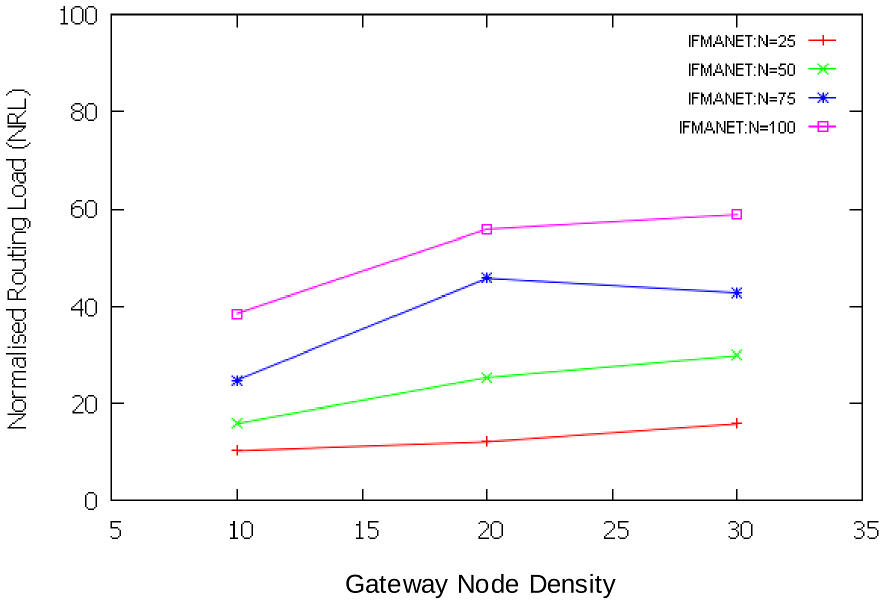


Figure 5.9: Normalized Routing Load vs Gateway Node Density

5.8.3 Scenario-3: Impact of Nodes Mobility (Mobility Model)

This section evaluates the impact of different node speeds on the performance of IF-MANET interoperable routing protocol and compares the performance with the single routing protocol “AODV” and multiple protocols AODV, MAODV. After that, it analysis the efficiency of algorithms, presented in section [3.5], by using different node speeds and find out its effect on connectivity and packet transmission overheads in heterogeneous MANETs.

Figure 5.10 shows the relationship between different node velocities (1, 2 5, 10 and 20 m/s) with different node densities (25, 50, 75 and 100 nodes) to that of network connectivity. It is obvious from the Figure 5.10 that the packet delivery ratio (connectivity) increases with increase in node densities at low to moderate node speed. Whereas the connectivity drastically decreases with increase in speed, irrespective of the node density. In particular, the IF-MANET achieved the higher connectivity i.e. greater than 80% at node densities between 50-100 nodes and speed 10m/s. The connectivity becomes worse when node density increases to 100 nodes and speed increases to 20m/s. It is because of the fact that when nodes move faster, the radio links between the nodes changes frequently which causes frequent change in connectivity and thus reduces the packet delivery ratio.

According to the route discovery algorithms in Sections 3.6.7 the IF-MANET Gateway listens to route requests of local mobile nodes, translates their route request packets and broadcast a route discovery to external Gateways. When node speed increases, it frequently breaks the connection with existing nodes and creates a new one with other nodes. This uncertain behaviour of frequent disconnections asserts extra burden on the Gateway Nodes to translate the messages of new nodes, broadcast/re-broadcast the messages to external MANETs, and maintain the GWRT with new entries. Due to these reasons the network connectivity decreases at high node velocity and in dense networks.

In addition to the connectivity, the increase of node speed increases the routing overhead as well. It is because the faster node movement produces more routing update packets. The same trend remains in denser network, but the routing control overhead dramatically increases when the node number increases. The reason is that there are much more intra-domain and inter-domain control packets needed in a denser network.

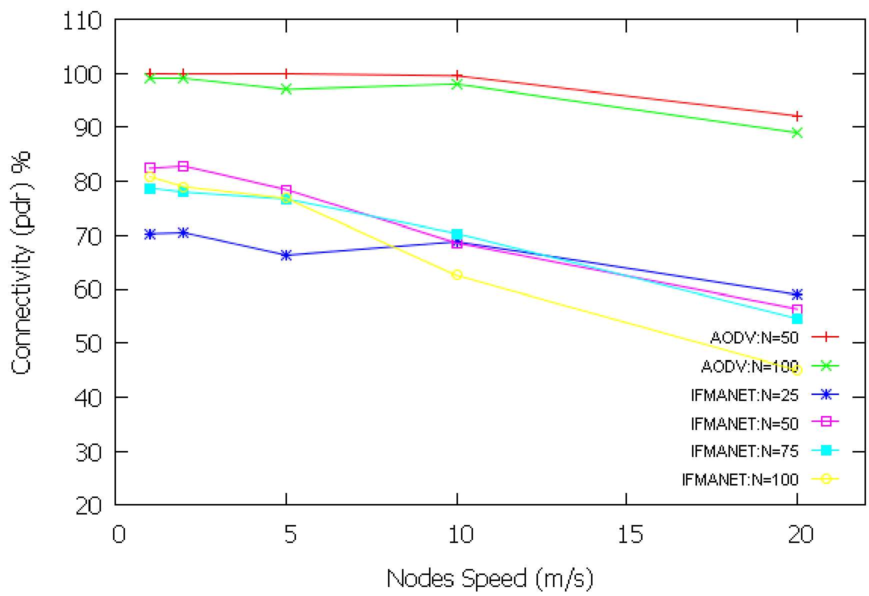


Figure 5.10: Impact of Varying Nodes Speed

5.9 Summary

This chapter has evaluated the performance of IF-MANET routing protocol, proposed in Chapter 3, within the heterogeneous ad hoc network environment consisting of AODV and MAODV routing protocols. It uses the network simulator NS-2 to implement and simulate the algorithms presented in Chapter 4. The simulation environment was setup with IEEE 802.11 interface, 11 Mbps channel capacity, 250 m transmission range between nodes, Random way point mobility model to randomly move nodes in different directions in a plane of 500x500 meters and CBR/UDP to generate the background traffic. The Table 5.2 defines the complete baseline parameters for all the scenarios simulated in this chapter. The chapter uses the following simulation methodologies:

- Topology Model: Impact of Node and Gateway Densities
- Mobility Model: Impact of different nodes speed

Each of the above models was used to evaluate and analyze the results of simulations against the following three key performance evaluation metrics:

- Connectivity (PDR)
- End to End (E2E) Delay
- Normalized Routing Load (NRL)

The results of performance metrics were compared with two baseline evaluation expectations presented in section 4.2 i.e. 1) Low performance (no connectivity), 2) High performance (using single routing protocol).

This chapter firstly investigated the IF-MANET routing protocol with the density of nodes, Gateways and evaluated their impact on connectivity (PDR), E2E Delay and NRL. The results showed that with the increase in mobile nodes and gateway densities the connectivity between heterogeneous routing protocols was increased. It was also observed that the dense networks created more transmission delays and routing overheads. This was an expected behaviour as the Gateway nodes in dense network have to translate a higher number of routing protocols, maintain larger GWRT and broadcast/re-broadcast large number of route discovery requests.

Further the routing protocols' performance was investigated based on the varying nodes mobility speeds and evaluated its impact on Connectivity (PDR) and Overheads. From the results, it is found that the connectivity was increased with increase in node density at low to moderate node speed but drastically reduced at high speed (20m/s) irrespective of the node densities.

The results of simulations has confirmed that the heterogeneous routing protocols were communicating with each other through IF-MANET routing protocol and by comparing the results with the baseline expected expectations it was clear that IF-MANET has performed well by providing maximum connectivity at low NRL and E2E Delay.

Chapter 6

Conclusions and Suggested Future Works

All the research efforts which have been carried out for this thesis have been presented in the previous chapters. The purpose of this chapter is to summarise the work presented and the goals achieved in this thesis. It will then discuss some of the potential new directions which can be explored as future research work.

6.1 Conclusions

This thesis has investigated the challenges of communication between mobile devices in heterogeneous MANETs and has proposed a framework to overcome these challenges and provide interoperability between them.

The thesis began with an introduction to the wireless ad hoc networks, its routing protocols and middleware frameworks and contributed an in-depth knowledge towards the interoperability of routing protocols. The thesis then analyzed the features and route discovery mechanisms of heterogeneous MANET routing protocols and classified them in different operational groups as shown in Figure 2.2. Then these heterogeneous routing protocols, of differing taxonomies, were compared based on their capabilities and are shown in Table 2.6. Thereafter, different middleware frameworks proposed for mobile ad hoc networks were surveyed and the major challenges were identified in designing the framework for MANET. Based on these surveys, different MANET frameworks were compared against the key requirements and the comparison chart is provided in Table 2.7Table 2.7. From comparison in Table 2.8 it is evident that the component based approach for middleware framework is fully compliant to most of the MANET requirements and due to its dynamic runtime configuration facility, component model is best suitable for interoperable MANET framework which requires dynamic translation of protocols at runtime.

After providing the background, the thesis has identified the key issues and challenges in providing interoperability between mobile nodes of heterogeneous MANETs. It then focused on examining the previous proposed research works that have addressed the challenges of interoperability and has conferred how the related work proposed previously differs from the one provided in this thesis.

Based on the analysis, a comprehensive comparison between different proposed interoperable solutions is presented in Table 2.8Table 2.8. It is evident from the comparison Table 2.8, that there is no single solution, but the IF-MANET proposed is an attempt to address all the challenges of interoperability in heterogeneous MANETs.

After analyzing the existing proposed solutions, this thesis has proposed a novel IF-MANET Framework to addresses the challenges of heterogeneity in MANETs to enable interoperability between them. The abstraction layer of IF-MANET hides the complexities of heterogeneity and provides a seamless homogeneous layer with well defined interfaces to develop external applications. The IF-MANET uses a cluster based approach for routing between different MANET domains and logically divides the network into Intra-MANET (Internal MANET) and Inter-MANET (External MANET). It has created a unique Gateway Node which enables the mobile nodes of one MANET system to communicate with the other MANET system. The IF-MANET allows Intra-MANET nodes to communicate with their local Gateway Nodes, which keep their state in special GWRT (Gateway Routing Table), translate their packets using MTL into the Abstract Message and communicate with Gateways of destination MANET. To achieve these objectives, the IF-MANET Framework has presented the following key contributions, along with algorithms.

To achieve the above mentioned objectives of interoperability, the key contributions of IF-MANET are divided into three main areas i.e. 1- Framework Architecture, 2- Route Discovery in Heterogeneous MANET Taxonomies and 3- MANET Ontology & Message Translator (MTL) whereas the sub-contributions of the IF-MANET, supporting the route discovery, consists of an Abstract Message, Gateway Engine and Initialization & Maintenance Phase. These contributions complement each other to achieve the interoperability in heterogeneous MANETs and are explained as below.

First of all the Framework Architecture of the IF-MANET has been described. The framework is created in such a way that it hides the complexities of heterogeneous MANET taxonomies from application layer. There are number of MANET routing protocols proposed so far to provide interoperability but their approach is to create new protocols and hence they are not compatible to existing as well as new arriving protocols. However, there is a lack of research on defining the framework for providing interoperability. After investigation, we have found that the framework approach is best suitable to provide not only the interoperability but also the extendibility for future protocols. Therefore, we have developed a framework for heterogeneous MANET routing protocols based on the concept of a service oriented component model, where each component acts as an independent service to provide a distinct set of functionalities. Due to the service oriented component model the new arriving protocols can more easily be integrated into the system without affecting the existing functionality. These components are classified in respect to the operational requirements and provide extendibility to load/unload the required components. Due to this model, the IF-MANET enables the interaction of components at runtime and reduces the processing and battery usage by loading/unloading the light weight components on as per request basis. The IF-MANET Framework hides the heterogeneity complexities and provides a seamless layer with well defined interfaces to application layer in order to develop interoperable middleware applications.

The thesis then has focused on defining the unique Abstract Message design. Due to the heterogeneity of routing protocols the mobile nodes cannot understand the packet syntax, semantics and data, and hence requires a message type which every protocol can understand. The IF-MANET has created an Abstract Message in such a way that it is not only protocol independent but also extendable to handle the different packet types of existing as well future arriving routing protocols. The structure of the Abstract Message is shown in Table 3.5. To reduce the memory footprint and hence the processing power and battery life, the structure of abstract message is defined in a way that it uses minimum mandatory fields whereas the optional fields can be expanded as per requirements of specific routing protocols.

The thesis then discussed the development of the MANET Ontology for semantic annotation of MANET routing protocols. The Ontology defines a set of predicates to describe various relationships between events and entities. It then has created a set of classes for routing protocols along with their properties and relationships. To semantically identify and map the data packets of different routing taxonomies, the capabilities of the MANET Ontology are classified into three functional groups i.e. 1)

MANET Taxonomies, 2) Operations and 3) Protocols. This grouping provides flexibility to classify the packet formats of new arriving routing protocols.

To transform the messages semantically and syntactically from source routing taxonomy to destination and vice versa, the thesis has created a Message Translator (MTL). The MTL exploits the MANET Ontology to find the semantic differences that arise between different routing protocols. It then uses the three phase process i.e. 1) Discovery & Learn Phase, 2) Matching Phase and 3) Modelling Phase, to identify the source and destination routing protocols, make comparison and find the missing data fields between their data packets and then model the destination routing protocol by applying rules to semantically generate the missing fields of source protocol. We then presented examples to explain how simple and complex routing packets of different taxonomies can be transformed through the IF-MANET MTL.

The route discovery, through which a source node finds a route to a destination node, is a fundamental requirement of the MANET to allow communication between source and destination nodes. However, if the source and the destination nodes belong to different MANET taxonomies then they will not be able to discover each other and hence cannot communicate. To address the challenges of heterogeneity, this thesis has presented an interoperable route discovery mechanism to find a route between nodes in heterogeneous MANET taxonomies. It uses the special Gateway Nodes, GWRT, abstract message, message translator and Bordercasting technique to provide an effective and efficient communication between heterogeneous MANETs. Unlike other proposed solutions, discussed in section 2.3, which modifies the behaviour of original routing protocols to invoke gateway nodes, our solution does not changes the behaviour of original routing protocols. For that, the IF-MANET Gateway nodes uses a unique mechanism by introducing a Route Discovery thresh hold counter i.e Retry_Threshold_Counter to store number of route discovery requests against every unique request id. When the value of Retry_Threshold_Counter reaches the configured retry threshold value then the Gateway node assumes that the destination node belongs to different MANET taxonomy or in different cluster and invokes itself to conduct extended route discovery on behalf of the source node. The Gateway node then converts the route discovery packet into the Abstract Message using IF-MANET Message Translator and Bordercast to all external IF-MANET Gateways. The IF-MANET uses Bordercasting technique instead of broadcasting to significantly reduce the path determination complexity as well as network overheads.

The IF-MANET provides the following three types of route discovery mechanisms that are best suitable for the type of MANET taxonomy in use:

- Route Discovery in Heterogeneous Reactive MANET Taxonomies
- Route Discovery in Heterogeneous Pro-active MANET Taxonomies
- Route Discovery in Hybrid Heterogeneous MANET Taxonomies

In Reactive route discovery, the Gateway Nodes find the route on demand and maintain the state of existing discovered routes. In Pro-active route discovery, the Gateway nodes maintain the information of internal as well as external domain nodes. In Hybrid Route Discovery, mobile nodes within MANET uses local routing protocol e.g. Reactive or Proactive routing etc, whereas Gateway nodes use local native protocol within MANETs and hybrid approach across external MANETs to discover. Unlike proactive and reactive approaches, that stores all nodes from all MANETs and only the Gateway address of all MANETs respectively, Gateway Nodes in Hybrid route discovery stores addresses of only those external native nodes (Non-Gateway) that were discovered during route discovery. This technique not only reduces the network overheads, end to end delays, processing power but also increases the ratio of nodes connectivity.

To evaluate the performance and mainly the Packet Delivery Ratio of IF-MANET, a series of experiments were conducted using different routing protocols i.e. AODV, MAODV and IF-MANET under different simulation models i.e. Mobility Model, Topology Model (nodes densities) and Traffic & Movement Model. It has then analyzed the results against three key performance criteria's i.e. 1) Connectivity (PDR), 2) Normalized Routing Load (NRL), and 3) End to End (E2E) Delay. The results were compared with two baseline benchmarked expectations i.e. 1) Low performance (no connectivity), and 2) High performance (single routing protocol). The results showed that with the increase in mobile nodes and gateway densities the connectivity between heterogeneous routing protocols was increased. It was also observed that the dense networks created more transmission delays and routing overheads. This was an expected behaviour as the Gateway nodes in dense network have to translate a higher number of routing protocols, maintain larger GWRT and broadcast/re-broadcast large number of route discovery requests. It was also observed that the connectivity was high at low to moderate nodes speed but drastically reduced at high speed (10 – 20 m/s) irrespective of the node densities. The results of simulations have proved that the heterogeneous routing protocols were communicating with each other through the IF-MANET and by comparing the results with the baseline benchmarks it is clear that the IF-MANET has outperformed our expectations by providing high connectivity at low NRL and E2E Delay.

.

6.2 Suggested Future Works

The following points offer some suggestions that could be interesting to investigate and implement as an extension to the research work proposed in this thesis.

- Although the proposed IF-MANET framework uses the widely used wireless interface IEEE 802.11 for communication, but it can only interact with devices using the same wireless interface. The proposed IF-MANET framework can be extended by providing interoperability across different wireless interfaces i.e. IEEE 802.11, Bluetooth, Zigbee etc. For Example, a mobile device using Wi-Fi 802.11 can communicate with its neighbour node using Bluetooth wireless technology to create and maintain a connected ad hoc network otherwise they cannot communicate and consume the significant resources of mobile devices. This feature has significant importance as it enhances the scope of our objective to utilise the heterogeneous resources of mobile devices. To provide the feature, it requires investigation on these interfaces and how to interact with them especially Bluetooth which does not have built-in facility of ad hoc networks rather it uses a mechanism of creating Scatternet from Piconents to act like ad hoc network. In addition, the interaction with other interfaces requires extending the IF-MANET routing protocol to provide interoperability with these interfaces.
- The thesis has proposed the interoperability based on two MANET taxonomies i.e. reactive and proactive routing and created Ontology for MANET routing protocols AODV and MAODV along with IF-MANET. In Future, an extension can be made to provide interoperability between more MANET taxonomies and create Ontology vocabulary and rules for various routing protocols to extend the scope of interoperability.
- Security is a major issue in public networks and ad hoc networks are vulnerable to security threats. Currently the IF-MANET Gateway nodes don’t apply any security checks but in future the IF-MANET can be extended to have a security component

integrated. The security component can use public/private key mechanism to verify and validate the requests before processing. This area requires more investigation to find out the feasible approach which is best suited for limited resources mobiles devices.

- Wireless sensor networks (WSNs) are considered as a variant of Ad Hoc Networks. Whereas the challenges of WSN include the design of routing protocols and network security for energy constrained tiny devices. To extend the ad hoc routing protocol to WSNs, security level and energy level need to be considered into the routing algorithms.

6.2.1 Potential Exploitation of the IF-MANET Implementation

The IF-MANET algorithms are implemented in C++ for the NS-2 simulator and runs in the application layer. It uses the MANET protocol stack (from Network layer) of the NS-2 to communicate with other mobile ad hoc nodes. In Future, the prototyped implementation of the IF-MANET can be extended to have its own complete MANET protocol stack to be deployed independently or the existing implementation can be modified for widely used mobile operating systems i.e. iOS and Android. The iOS had a 19.7% share of the Smartphone mobile operating system units shipped in the fourth quarter of 2014, behind Android with 76.6% (Wu 2015). However, iOS and Android are using different operating systems, architecture frameworks, development languages and have different security levels for applications to interact with the lower layer of the protocol stack. For Example, Android is built on Linux, written in C/C++ and the applications will be written in C++/Java, whereas the iOS was created by Apple Inc. iOS is based on Apple's desktop operating system OSX that is in turn is based on a variant of Unix BSD. Applications for iOS are written in Objective-C or SWIFT languages and are sandboxed from each other and additionally can only communicate with underlying operating system by well-defined and restricted APIs. Also, the security system of these operating systems doesn't allow any application to directly access the network and physical layer of the mobile device. Therefore, a research is required to investigate the security system of different mobile operating systems and find out how they can be addressed to allow interaction with the lower layers of their protocol stack. The other important capability of these mobile operating systems is to find out whether they provide the implementation of MANET protocol stack or not. If not, then the IF-

MANET should be extended to have its own complete implementation compatible to the target mobile operating systems.

In addition, the IF-MANET implementation can be extended to develop an Add-on Plug-in for network simulators like NS-2 and OPNET. However, it requires investigating the architectures, MANET protocol stacks and API's provided by these network simulators to implement IF-MANET as a plug-in adapter for them.

The IF-MANET implementation can be exploited to develop a middleware for applications, like mobile grid, emergency and disaster management and fire/safety & rescue system etc, to provide communication in a heterogeneous infrastructure-less mobile wireless networks. The middleware applications can be developed for commercial mobile companies by seeking collaboration and special agreement to host the IF-MANET on their Operating Systems. Also the network simulator’s market can be exploited by making IF-MANET implementation available to most commonly used simulators like NS-2 and OPNET. It will allow community of developers across the globe to analyze the IF-MANET's code, routing protocol and contribute for continuous improvement. The IF-MANET routing protocol will be published in the routing protocol standards-setting body i.e. Internet Engineering Task Force (IETF) as a Request for Comments (RFC). It will not only secure the rights of IF-MANET routing protocol but also allows different companies to contact us if they want to use the protocol.

6.2.2 Planned Papers for Publication

The following research papers are planned to be published in Journals and Conferences:

- Route Discovery in Heterogeneous MANET Taxonomies
- MANET Ontology and Message Translator for Heterogeneous Data Transformation
- IF-MANET Interoperable Routing Protocol for Heterogeneous MANETs
- Survey: Comparison of Middleware Frameworks for Heterogeneous MANETs

References

[1] Abolhasan, M., Wysocki, T. & Dutkiewicz, E., 2004. A review of routing protocols for mobile ad hoc networks. *Ad hoc networks*, 2(1), pp.1–22.

[2] Agarwal, R., Motwani, D. & others, 2009. Survey of clustering algorithms for MANET. *arXiv preprint arXiv:0912.2303*.

[3] Albrightson, B., Garcia-Luna-Aceves, J. & Boyle, J., 1994. {EIGRP} -- A fast routing protocol based on distance vectors. *Networld/Interop*, pp.1–13.

[4] Chandrashekar, K., Morera, R., McAuley, A. and Baras, J., 2005, October. Domain based hierarchical routing for large heterogeneous MANETs. In *MILCOM* (Vol. 2, p. 1284).

[5] The Network Simulator - NS-2, 2014. Available at: http://www.isi.edu/nsnam/ns/ [Accessed on January 1, 2014].

[6] TclCL, 2014. Available at: http://otcl-tclcl.sourceforge.net/tclcl. [Accessed on January 1, 2014].

[7] Ubuntu, 2012. Available at: http://www.ubuntu.com/. [Accessed on January 1, 2014].

[8] Beijar, N., 2002. Zone Routing Protocol (ZRP). *Networking Laboratory, Helsinki University of Technology, Finland*, 9, pp.1–12.

[9] Bernstein, P.A., 1996. Middleware: A Model for Distributed System Services. *Communications of the ACM*, 39(2), pp.86–98.

[10] Bisignano, M. et al., 2003. Expeerience: A Jxta Middleware for Mobile Ad-Hoc Networks. *Peer-to-Peer Computing*, pp.214–215.

[11] Bisignano, M., Modica, G. Di & Tomarchio, O., 2005. JMobiPeer: a middleware for mobile peer-to-peer computing in MANETs. *25th IEEE International Conference on Distributed Computing Systems Workshops*.

[12] Blundo, C. & Cristofaro, E. De, 2007. A Bluetooth-Based JXME Infrastructure. In *On the Move to Meaningful Internet Systems 2007*. Springer, pp. 667–682.

[13] Boukerche, A. et al., 2011. Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13), pp.3032–3080.

[14] Bruneo, D., Puliafito, A. & Scarpa, M., 2007. Mobile Middleware: Definition and Motivations. *The Handbook of Mobile Middleware*, pp.145–167.

[15] Calafate, C.M.T. & Manzoni, P., 2003. PICA: A multi-platform programming interface for protocol development. *Parallel, Distributed and Network-Based Processing*, pp.243–249.

[16] Carlo, B. & Emiliano De, C., 2007. A Bluetooth-based JXME infrastructure. *Proceedings of the 2007 OTM Confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I*.

[17] C Chau, C.K., Crowcroft, J., Lee, K.W. and Wong, S.H., 2008. IDRM: Inter-Domain Routing Protocol for Mobile Ad Hoc Networks. University of Cambridge Technical Report UCAM-CL-TR-708.

[18] Cheng, C. et al., 1989. A loop-free extended Bellman-Ford routing protocol without bouncing effect. In *ACM SIGCOMM Computer Communication Review*. pp. 224–236.

[19] Chetan, S. et al., 2005. Mobile Gaia: a middleware for ad-hoc pervasive computing. *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*, pp.223–228.

[20] Chau, C.-K. et al., 2008. IDRM: Inter-domain routing for mobile ad hoc networks. *Proceedings of the 3rd international workshop on Mobility in the evolving internet architecture*, pp.61–66.

[21] Chroboczek, J., 2011. The Babel Routing Protocol. Available at: http://tools.ietf.org/html/rfc6126. [Accessed on Mar 1, 2014]

[22] Clausen, T. & Jacquet, P., 2003. RFC 3626 - Optimized Link State Routing Protocol (OLSR) P. Jacquet, ed. *Ietf Rfc3626*, p.75.

[23] Corson, S. & Macker, J., 1999. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501. Available at: https://tools.ietf.org/html/rfc2501. [Accessed on Feb 01, 2014].

[24] Costa, P. et al., 2007. Reconfigurable component-based middleware for networked embedded systems. *International Journal of Wireless Information Networks*, 14(2), pp.149–162.

[25] Crowcroft, J. et al., 2003. Plutarch: an argument for network pluralism. *ACM SIGCOMM Computer Communication Review*, 33(4), pp.258–266.

[26] Davoudpour, M. et al., 2014. A Formal Ontology Alignment for CANthings (Context Aware Network for the Connected Things). In *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on*. pp. 175–180.

[27] Denko, M.K., Shakshuki, E. & Malik, H., 2009. Enhanced cross-layer based middleware for mobile ad hoc networks. *Journal of Network and Computer Applications*, 32(2), pp.490–499.

[28] Downing, T.B., 1998. *{Java RMI}: {Remote Method Invocation}*, IDG Books Worldwide, Inc.

[29] Eclipse Foundation, 2011. Eclipse CDT. Available at: http://www.eclipse.org/cdt/ [Accessed on Feb 20, 2014].

[30] Euzenat, J., Shvaiko, P. & others, 2007. *Ontology matching*, Springer.

[31] Fok, C.-L., Roman, G.-C. & Hackmann, G., 2004. A lightweight coordination middleware for mobile computing. In *Coordination Models and Languages*. pp. 135–151.

[32] Fujiwara, S., Ohta, T. & Kakuda, Y., 2012. An inter-domain routing for heterogeneous mobile ad hoc networks using packet conversion and address sharing. *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pp.349–355.

[33] Gelernter, D., 1985. Generative communication in Linda. *ACM Transactions on Programming Languages and Systems*, 7(1), pp.80–112.

[34] Gilman, L. & Schreiber, R., 1996. *Distributed computing with IBM MQSeries*, John Wiley & Sons, Inc.

[35] Gong, L., 2002. *JXTA for J2ME extending the reach of wireless with JXTA technology*,

[36] Görgen, D. et al., 2004. SELMA: A middleware platform for self-organizing distributed applications in mobile multihop ad-hoc networks. *Western Simulation MultiConference WMC '04*, pp.1–16.

[37] Grace, P., Blair, G.S. & Samuel, S., 2003. ReMMoC: A Reflective Middleware to Support Mobile Client Interoperability. *Conference on Cooperative Information Systems*, pp.1170–1187.

[38] Haas, Z. & Pearlman, M., 2002. The bordercast resolution protocol (BRP) for ad hoc networks. *IETF, MANET Internet Draft*, pp.13801–14853. Available at: http://tools.ietf.org/id/draft-ietf-manet-zone-brp-02.txt.

[39] H Hadim, S., Al-Jaroodi, J. and Mohamed, N., 2006. Trends in middleware for mobile ad hoc networks. Journal of Communications, 1(4), pp.11-21.

[40] Herrmann, K., Mühl, G. & Jaeger, M. a., 2007. MESHMdl event spaces - A coordination middleware for self-organizing applications in ad hoc networks. *Pervasive and Mobile Computing*, 3, pp.467–487.

[41] Hincapié, R.C., Correa, B.A. & Ospina, L., 2006. Survey on clustering Techniques for mobile ad hoc networks. *IEEE, Los Alamitos*.

[42] IETF, 2011. The Babel Routing Protocol. Available at: https://tools.ietf.org/html/rfc6126. [Accessed on Feb 01, 2014].

[43] IETF, 2002. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. *Draft-Ietf-Manet-Zone-Zrp-04*, 04(1), p.11.

[44] Jacquet, P. et al., 2001. Optimized Link State Routing Protocol for Ad Hoc Networks. *IEEE International Multi Topic Conference*, pp.62–68.

[45] Jiang, M., Li, J. & Tay, Y.C., 1999. *Cluster Based Routing Protocol (CBRP).*

Draft-ietf-manet-cbrp-spec-01, Available at: https://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01.

[46] Johnson, D. et al., 2007. *The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4*,

[47] Khatkar, A. & Singh, Y., 2012. Performance Evaluation of Hybrid Routing Protocols in Mobile Ad Hoc Networks. *2012 Second International Conference on Advanced Computing & Communication Technologies*, pp.542–545..

[48] Kukliński, S., 2011. WARF: Component Based Platform for Wireless Mesh Networks. *The Smart Computing Review*, 1(2), pp.125–138.

[49] Lee, S.-H. et al., 2010. Intermr: Inter-manet routing in heterogeneous manets. *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pp.372–381.

[50] Ma, W. & Chuah, M.C., 2005. Comparisons of inter-domain routing schemes for heterogeneous ad hoc networks. *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005*, pp.368–373.

[51] Meier, R. & Cahill, V., STEAM: Event-Based Middleware for Wireless Ad Hoc Networks. *Distributed Event-Based Systems*.

[52] Mojamed, M. Al & Kolberg, M., 2014. OnehopMANET: One-hop Structured P2P over Mobile ad hoc Networks. In *Next Generation Mobile Apps, Services and Technologies IEEE(NGMAST), 2014 Eighth International Conference on*. pp. 159–163.

[53] MS COM, 2014. COM. Available at: https://www.microsoft.com/com. [Accessed on June 1, 2014]

[54] Murphy, A.L., Picco, G. Pietro & Roman, G., 2001. LIME: A Middleware for Physical and Logical Mobility. *International Conference on Distributed Computing Systems*, pp.524–533.

[55] Murphy, A.L., Picco, G. Pietro & Roman, G.-C., 2006. Lime. *ACM Transactions on Software Engineering and Methodology*, 15(3), pp.279–328.

[56] Musolesi, M., Mascolo, C. & Hailes, S., 2006. EMMA: Epidemic Messaging Middleware for Ad hoc networks. *Personal and Ubiquitous Computing*, 10(1), pp.28–36.

[57] Ns2, 2008. Network Simulator. http://www.isi.edu/nsnam/ns/, Mar. Available at: http://www.isi.edu/nsnam/ns/. [Accessed on June, 2014]

[58] OMG, 2012. CORBA. Available at: http://www.omg.org/spec/CORBA/. [Accessed on June, 2014]

[59] Pei, G., Gerla, M. & Chen, T.-W., 2000. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*. pp. 70–74.

[60] Pei, G., Gerla, M. & Hong, X., 2000. LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility. In *Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. pp. 11–18.

[61] Perkins, C., Belding-Royer, E. & Das, S., 2003. RFC 3561-ad hoc on-demand distance vector (AODV) routing. *Internet RFCs*, pp.1–37. Available at: http://www.rfc-editor.org/info/rfc3561.

[62] Perkins, C.E. & Bhagwat, P., 1994. *Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*,

[63] Perkins, C.E. & Royer, E.M., 1999. Ad-hoc On-Demand Distance Vector Routing. *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, p.90.

[64] Raghuveer, M., 2008. *Interoperability of ad-hoc routing protocols*. PhD Thesis. Wichita State University..

[65] Ramasubramanian, V., Haas, Z.J. & Sirer, E.G., 2003. SHARP: A hybrid adaptive routing protocol for mobile ad hoc networks. *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp.303–314.

[66] Ramdhany, R. et al., 2009b. MANETKit: Supporting the Dynamic Deployment and Reconfiguration of Ad-Hoc Routing Protocols. *International Conference on Distributed Systems Platforms and Open Distributed Processing/Open Distributed Processing*, pp.1–20.

[67] Rekhter, Y. & Li, T., 1995. A border gateway protocol 4 (BGP-4). Available at: http://tools.ietf.org/html/rfc1771.

[68] Resta, G., Bettstetter, C. & Santi, P., 2003. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3), pp.257–269.

[69] Ramdhany, R. et al., 2009a. MANETKit : Supporting the Dynamic Deployment and Reconfiguration of Ad-Hoc Routing Protocols. *System*, pp.1–22

[70] Sarkar, S.K., Basavaraju, T.G. & Puttamadappa, C., 2007. *Ad hoc mobile wireless networks: principles, protocols and applications*, CRC Press.

[71] Schmid, S. et al., 2005. TurfNet: An architecture for dynamically composable networks. *Autonomic Communication*, pp.94–114.

[72] Schollmeier, R., 2001. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*. pp. 101–102.

[73] Da Silva, E. & Albini, L.C.P., 2014. Middleware proposals for mobile ad hoc networks. *Journal of Network and Computer Applications*, 43, pp.103–120.

[74] Stroustrup, B., 2013. *The C++ programming language*, Pearson Education.

[75] Tanenbaum, A.S., 2002. Computer Networks (4th Edition). *Prentice Hall*, p.912.

[76] Viswanath, K. & Obraczka, K., 2004. Interoperability of Multicast Routing Protocols in Wireless Ad-Hoc Networks. , (1), pp.1–20.

[77] Waldo, J., 1998. Remote procedure calls and Java Remote Method Invocation. *IEEE Concurrency*, 6(3), pp.5–7.

[78] Wang, A.I., Bjørnsgård, T. & Saxlund, K., 2007. Peer2Me - Rapid application framework for mobile peer-to-peer applications. *Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems, CTS*, pp.379–388.

[79] Wang, I. & Jones, S.D., 2004. Component-Based Analysis and Design of Routing Protocols for Mobile Ad Hoc Networks 1. , pp.1–3.

[80] Wu, Y.-H., 2015. The Impact of Consumers' Purchase Intention Influenced by Review Types. Available at: http://www.idc.com/getdoc.jsp?containerId=prUS25450615. [Retrieved on Aug 2015]

[81] Zhou, B., Cao, Z. & Gerla, M., 2009. Cluster-based inter-domain routing (CIDR) protocol for MANETs. *Wireless On-Demand Network Systems and Services, 2009. WONS 2009. Sixth International Conference on*, pp.19–26.

[82] Zhu, Y. & Kunz, T., 2004. MAODV Implementation for NS-2.26. *Systems and Computing Engineering, Carleton University, Technical Report SCE-04-01*, (January).

[83] Hassan, H., Trwoga, P. and Kale, I., 2015. IF-MANET: Interoperable Framework for Mobile Ad Hoc Networks. In *Computer Networks* (pp. 54-68). Springer International Publishing.