ABSTRACT

BOPPANA,ANEESHA CHOWDHARY. A Scalable Simplified Multicast Forwarding for Mobile Ad-Hoc Networks. (Under the direction of Dr. Mihail Sichitiu).

Multicasting is increasingly important for mobile ad hoc networks (MANETs). A MANET is comprised of mobile nodes, potentially without any infrastructure. Many MANET applications need multicasting, as it provides a simple and reliable communication mechanism by using the inherent broadcasting property of wireless transmissions, and can significantly improve the bandwidth efficiency. Multicasting reduces transmission overhead and power consumption. In the past couple of years, several multicast routing protocols have been proposed for both wired networks and MANETs.

In this thesis, a Simple but Scalable Multicast Forwarding (SSMF) is developed. The proposed SSMF protocol is an extension of Simplified Multicast Forwarding (SMF) for MANETs, and is independent of any underlying unicast protocol. The performance of the protocol is evaluated in NS-2 and compared with SMF and Multicast Ad hoc On-demand Distance Vector (MAODV).

A Scalable Simplified Multicast Forwarding for Mobile Ad-Hoc Networks

by
Aneesha Boppana

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Computer Networking

Raleigh, North Carolina

2011

APPROVED BY:

Dr. Mihail Sichitiu
Committee Chair

Dr. Yannis Viniotis

Dr. Michael Devetsikiotis

DEDICATION

To my dad (Ramakrishna), my mom (Satyasri) and my sister (Prathyusha) …

BIOGRAPHY

Aneesha Boppana is a graduate student at the Department of Electrical and Computer Engineering, North Carolina State University. She was born and brought up in Hyderabad, India. In 2009, she graduated with a Bachelor of Technology in Information and Communication Technology (ICT) from DA-IICT University, Gujarat, before joining North Carolina State University for a Master of Science in Computer Networking in the same year. She is currently working towards the completion of her degree.

ACKNOWLEDGMENTS

I would like to thank Dr. Mihail Sichitiu, my advisor and committee chairman, for his support, encouragement, and invaluable guidance throughout the course of this work. I would also like to thank my colleagues in WALAN Lab for their advice. I would like to thank my roommates and friends for being my family away from home. Last but not the least, I express my deep gratitude to my family for molding me into the person I am today with their constant love, support and guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ARP - Address Resolution Protocol

AODV - Ad hoc On-demand Distance Vector

AMRIS - Ad hoc Multicast Routing protocol utilizing Increasing id-numberS

AMRoute - Ad hoc Multicast Routing Protocol

CDS - Connected Dominating Set

CBT - Core – Based Trees

CAMP - Core Assisted Mesh Protocol

DSDV - Destination Sequence Distance Vector

DSR - Dynamic Source Routing

DVMRP - Distance Vector Multicast Routing Protocol

DDM - Differential Destination Multicast

DCF - Distributed Coordination Function

E-CDS - Essential Connected Dominating Set

FGMP - Forwarding Group Multicast Protocol

GW - Gateway

HNA - Host and Network Association.

IETF - Internet Engineering Task Force

IGMP - Internet Group Multicast Protocol

IP - Internet Protocol

LGT - Location Guided Tree

LAN - Local Area Network

MANET - Mobile Ad Hoc Network

MAODV - Multicast Ad hoc On-demand Distance Vector

MPR -Multi Point Relay

MOLSR - Multicast extension for the Optimized Link State

MCEDAR - Multicast Core Extraction Distributed Ad hoc Routing

MAC - Media Access Control
NRL - Naval Research Laboratory
NS-MPR - Non-Source-based Multi Point Relay
NS-2 - Network Simulator
OLSR - Optimized Link State Routing
ODMRP - On Demand Multicast Routing Protocol
PIM - Protocol Independent Multicast
PIM-DM - Protocol Independent Multicast – Dense Mode
PIM-SM - Protocol Independent Multicast – Sparse Mode
PIM-SSM - Protocol Independent Multicast – Source Specific Multicast
PIM-BIDIR - Protocol Independent Multicast – Bidirectional
RP - Rendezvous Point
RPB - Reverse path Broadcasting
RPM - Reverse Path Multicasting
RPF - Reverse Path Forwarding
RWP- Random Way Point
SPT - Shortest Path Tress
SMF - Simplified Multicast Forwarding
SSMF – Scalable but Simplified Multicast Forwarding
S-MPR - Source-based Multi Point Relay
TRPB - Truncated Reverse Path Broadcasting
TC - Topology Control
TTL - Time to Live
UDP - User Datagram Protocol

CHAPTER 1

Introduction

Given the increasing demand for flexibility as well as technological advances in mobile communication devices such as wireless LANs, laptop computers and smart mobile phones, wireless communications are becoming more and more common. There are several advanced efforts to enable wireless communication over mobile networks. Multicasting is one such effort that strives to provide support for wireless communication in mobile networks.

1.1 Multicasting in Mobile Ad Hoc Networks (MANETs)

In the Internet, multicasting means transmission of packets to a group of zero or more hosts identified by a single destination address [1]. The idea of multicasting is intended in scenarios, where, all members in the host group need to receive the same packets from one or more sources. Membership in the multicast group can change dynamically.

A MANET comprises self-organized wireless mobile nodes that share a common wireless channel that can work without the support of fixed infrastructure or centralized administration. Two nodes can communicate either by single-hop transmission, if they are within each other's transmission ranges, or by multihop transmissions through intermediate nodes that will serve as relays. Multi-hopping is usually required due to limited transmission power. Each node participates in the network as both host and a router.

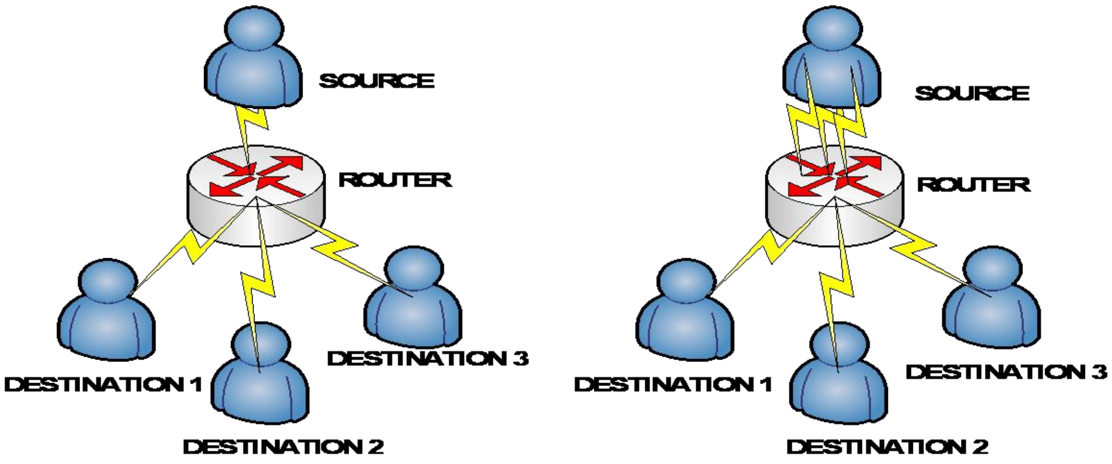
1.1.1 Advantages

Multicasting reduces the communication costs for applications that send the same data to multiple recipients. Instead of sending data through multiple unicasts, multicasting minimizes the link bandwidth consumption and delivery delay [4]. Figure 1.1 shows a topology of one source and three destinations when using both multicast and unicast and depicts this advantage.

1.1.2 Challenges

MANETs have several characteristics not present in the wired networks: rapid deployment, robustness, flexibility, inherent mobility support, highly dynamic network topology (device mobility, changing properties of the wireless channel (e.g., fading and multipath propagation), and partitioning and merging of ad hoc networks are possible), limited battery power, limited capacity, and asymmetric/unidirectional links [5,6].

The above characteristics of MANETs create challenges for multicasting [2, 3, 6-7]. The key problem of multicasting in MANETs is to enable efficient delivery of packets from a sender to multiple receivers, when the nodes are mobile. A highly dynamic topology is the biggest challenge for the robustness of a multicast protocol. In comparison, for wired networks, MANETs have a lower channel capacity, which is the result of noise and interference inherent with the wireless transmissions. As a result, there is always a tradeoff between reliability and control overhead. This tradeoff in turn affects the performance of the protocol.



(a) **(b)**

Figure 1.1: Example of a network topology with one sender and three receivers when using (a) multicast, and (b) unicast

1.2 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 discusses the related work in the area of existing multicast protocols. Chapter 3 covers our proposal of SSMF. In Chapter 4, we present the performance results of SSMF. Chapter 5 provides the conclusion and future work.

CHAPTER 2

Related Work

This chapter is devoted to presenting and classifying existing multicast routing protocols for wired networks as well as MANETs. A brief overview of a few existing multicast protocols that are relevant to this thesis is also provided.

2.1 Existing Multicast Protocols

In wired networks, two popular categories of multicast schemes exist. They are:

i. shortest path multicast trees [8], belonging to the category of source-based tree approach, and
ii. core-based multicast trees, belonging to the category of shared-based tree approach [9].

Either source-based tree or core-based tree approaches are chosen based on density of networks. Wired network multicast protocols use Internet Group Management Protocol (IGMP) for group management.

For MANETs, multicast protocols should be able to keep track of host mobility for computing multicast trees. The computation of multicast trees might however result in a substantial overhead when compared with wired network multicast protocols.

2.1.1 Wired to Wireless

There is a need for separate multicast protocols for MANETs [10]. The wired multicast routing protocols are designed assuming static hosts with stable links and this assumption is not true for the mobile environment.

Due to the importance of multicasting in MANETs and their particular challenges, many multicasting protocols have been proposed specifically for MANETs. A taxonomy of multicast routing protocols has been published in Omari et al. [11], classifying multicast routing protocols into tree-based, mesh-based, hybrid-based and flooding protocols and evaluates the performance and capacity of multicast routing protocols for MANETs.

A brief overview of all the existing multicast protocols is presented in the subsequent sections; detailed descriptions can be found in [11].

2.2 Taxonomy of Multicast Protocols in Wired Network

In the Internet, the multicast routing protocol is responsible for multicast packet delivery. In the following sections several multicast forwarding algorithms for wired networks are presented.

2.2.1 Flooding based IP Multicast

The simplest technique for delivering multicast packets to all routers in an internetwork is the flooding algorithm. In this algorithm, when a router receives a broadcasted packet for the first time, it will forward the packet on all interfaces except the one on which it received the packet. If the router has seen the packet before, it will simply discard it. This way all routers in the network will receive at least one copy of the packet.

A flooding algorithm is very simple to implement, since a router does not have to maintain a routing table and only needs to keep track of the most recently seen packets. However, flooding does not scale well for large networks, since, it generates a large number of duplicate packets and wastes a lot of network bandwidth.

2.2.2 Spanning Tree based IP Multicast

A better algorithm than flooding is the Spanning Tree algorithm [59]. In this algorithm, a subset of the network becomes a spanning tree. The tree is constructed such that there is only one active path between any two routers. This tree reaches to all nodes in the network. Whenever a router receives a multicast packet, it forwards the packet on all the links which belong to the spanning tree except the one on which the packet has arrived, thus, preventing loops and ensuring that all the multicast nodes receive the data. However, a spanning tree solution can funnel traffic on a small number of links, and may not provide the most efficient path between the source and group members.

2.2.3 Source - based IP Multicast

In the source-based multicast algorithm, a multicast tree is built using a source as the root of the tree. There are several source-based multicast algorithms [13]:

i. Reverse Path Broadcasting (RPB),
ii. Truncated Reverse Path Broadcasting (TRPB), and
iii. Reverse Path Multicasting (RPM).

All of these algorithms are variants of Reverse Path Forwarding algorithm (RPF) [12]. Both shared trees described in section 2.2.4, and shortest path trees (SPTs) or source based trees, are built with the help of RPF.

RPF eliminates loops in the flooding process by not forwarding packets on the interfaces on which they arrived and forwarding them on all other interfaces. RPF forwards only those packets that are eligible for forwarding by performing a reverse-path check. If the interface on which the packet arrived is on the shortest path back to source, it is forwarded (hence the term reverse-path) otherwise, it is discarded.

The RPB [13] algorithm is an enhancement to the flooding variant RPF. RPB is modification of spanning tree, where, instead of creating a tree for the entire network, a tree is created for each (source, group) pair. Hence multiple sources in a group will result in multiple spanning trees for a group. RPB does not consider group membership; hence it forwards packets even if there are no members in the domain.

TRPB [13] addresses the limitations of RPB by using IGMP, which allows a multicast router to determine if there is any member for a group in its domain. If not, it prunes itself from the multicast delivery tree through prune messages.

RPM [13] is an enhancement of both RPB and TRPB. In RPM, the spanning tree connects not only routers and sub networks with group members like in TRPB, but also routers and sub networks along the shortest path to sub networks with group members. The RPM tree can be pruned such that the multicast packets are only forwarded along links which lead to members of the destination group. There are several limitations of RPM. The first limitation is that multicast packets must be periodically forwarded to every router in the network. The second drawback is that each router is required to maintain state information for all groups and each source, resulting in scalability issues of RPM.

The best examples of Source based IP Multicast protocol are Protocol Independent Multicast - Dense Mode (PIM-DM) [14] and the Distance Vector Multicast Routing Protocol (DVMRP) [15]. PIM and DVMRP are similar with minor differences. DVMRP constructs source-rooted multicast delivery trees using variants of the Reverse Path Broadcasting (RPB) algorithm. DVMRP also uses RIP-like exchange messages to build its unicast routing table unlike PIM-DM, which is independent of any particular underlying unicast protocol. PIM-DM uses RPM for constructing trees and also delivers traffic to all nodes until it receives prune messages whereas, DVMRP delivers traffic only to its tree nodes. However, the fact that DVMRP periodically floods the network makes it disadvantageous to networks with few nodes and limited bandwidth.

2.2.4 Core - based IP Multicast

The Core-Based Trees (CBT) algorithm has been created to overcome RPM limitations. CBT basically divides the network into different multicast groups and creates a single shared multicast delivery tree for each group. Every group has a single router as a core for forwarding packets. All packets are forwarded through the core. To join a multicast group, a user sends an explicit JOIN toward the core of the group. The JOIN message is propagated upstream until it reaches the core. CBT efficiently handles the problem of scalability by building only one multicast tree for each group. However, CBT generates heavy traffic near the core routers since all sources and all receivers within a multicast group use the same paths leading to higher delays between source and group members. Moreover, the paths used may not be the shortest. Hence, shared trees are often not desirable if the number of hosts in multicast group increases. The best example of CBT is Protocol Independent Multicast - Sparse Mode (PIM-SM) [16].

2.3 Overview of Wired Multicast Protocols

A few wired multicast protocols are presented in this section. Since, PIM-SM is the most widely deployed multicast protocol; PIM is taken as example in this section. PIM introduced several collections of algorithms like PIM-DM [14], PIM-SM [16], PIM-SSM [29] and PIM-BIDIR [30].

PIM was mainly designed to overcome the limitations of dense mode protocols such as DVMRP, which was not scalable for large and sparse multicast network groups. Core based approaches were introduced at that time to support sparse mode, but these approaches have their own limitations. The core-based approaches sometimes result in a bottleneck at cores in applications, and placement of core(s) is critical. Hence, PIM was designed to overcome the problem of scaling in dense mode as well as core based algorithms.

2.3.1 PIM - DM (Dense Mode)

PIM-DM implements the same flood-prune approach as DVMRP. Periodic flooding is carried out in the dense-mode networks by the multicast sources with multicast traffic and then prune messages inactivate routers without clients. The prune messages are generated by the routers that do not have any interested multicast receiver for that multicast group under its subnet.

2.3.2 PIM - SM (Sparse Mode)

PIM-SM is more complex compared to PIM-DM and requires establishment of special routers called Rendezvous Points (RPs). These RPs are those points, where interested multicast receiver(s) send upstream JOIN messages to, and where sender(s) forward multicast group content. PIM-SM is more efficient than PIM-DM as it never uses flooding to distribute multicast content. Initially, receivers in a PIM-SM need to join the shared tree rooted at RP by sending explicit JOIN messages. Senders announce their existence to all RPs, and receivers need to know the existence of at least one RP to find any multicast session. Receivers can explicitly prune themselves from a tree. An RP can be common for multicast groups or just a subset of them. However, the sources need to send multicast data to a particular RP for a particular group. Every multicast group has a single RP. A key difference between PIM-SM and PIM-DM is that, in PIM-SM, routers need to explicitly announce that they need to receive multicast messages for multicast groups, while the dense-mode protocols assume that all routers need to receive multicast messages unless they explicitly send prune messages.

2.3.3 PIM - SSM (Source Specific Multicast)

PIM-SM also supports source based trees. Source based trees enable the use of SSM and allow a host to specify sources from whom they wish to receive data as well as the group they wish to join. Unlike the shared tree protocols which use (*, G) to identify the multicast data stream, SSM uses (S, G) to differentiate between multicast data streams. PIM-SSM builds SPTs rooted at the source, which allows them to bypass the RP in a shared tree and go directly to a source-based distribution tree. Thus, PIM-SSM is a subset of PIM-SM. During congestion, if receivers know the address of their multicast source for a particular group, they can explicitly send a JOIN request to the source. Sending JOIN requests will ensure optimal routing. However, there is always a tradeoff between amount of state information and optimal routes between shared trees and that of source-based trees.

2.3.4 PIM - BIDIR (Bidirectional)

Shared trees can be categorized into two types: unidirectional and bidirectional. PIM-BIDIR uses bidirectional trees. In PIM-BIDIR, multicast data is forwarded in both directions:

i. up the tree toward the RP and then RP transmits down the tree toward receivers, and

ii. directly down the tree toward receiver by passing RP.

PIM-BIDIR creates a two-way forwarding tree. Because the data travels in both directions, the amount of state information kept is minimum. In PIM-BIDIR, there are no source based trees.

In conclusion, PIM is a collection of multicast protocols, where, each variant is designed specifically to work best in certain network scenarios.

2.4 Taxonomy of Multicast Protocols in MANETs

Recently, many multicast routing protocols have been proposed specifically for MANETs. These include multicast ad-hoc on-demand vector (MAODV) [17], core assisted mesh protocol (CAMP) [18], location guided tree (LGT) [19], on-demand multicast routing protocol (ODMRP) [20], forwarding group multicast protocol (FGMP) [21], ad-hoc multicast routing (AMRoute) [22], multicast core extraction distributed ad-hoc routing (MCEDAR) [23] and differential destination multicast (DDM) [24]. Most of these multicast routing protocols are primarily based on distance-vector, stateless or link-state routing with additional functionality incorporated to assist the routing operations. The goals of all these protocols include minimizing control overhead, minimizing processing overhead, maximizing multi-hop routing capability, maintaining dynamic topology and preventing loops in the networks.

However, many multicast routing protocols do not perform well in MANETs, because, in a highly dynamic environment, network topology changes frequently and unpredictably. Moreover, bandwidth and power are limited. This section presents the life cycle of a MANET multicast protocol, and their algorithms that are largely dependent on characteristics, closely related to the stages of the life cycle.

2.4.1 Multicast Session Life Cycle

A general multicast session undergoes different stages to complete the steps of a life cycle as shown in Figure 2.1. The most important stages and their sub stages involve:

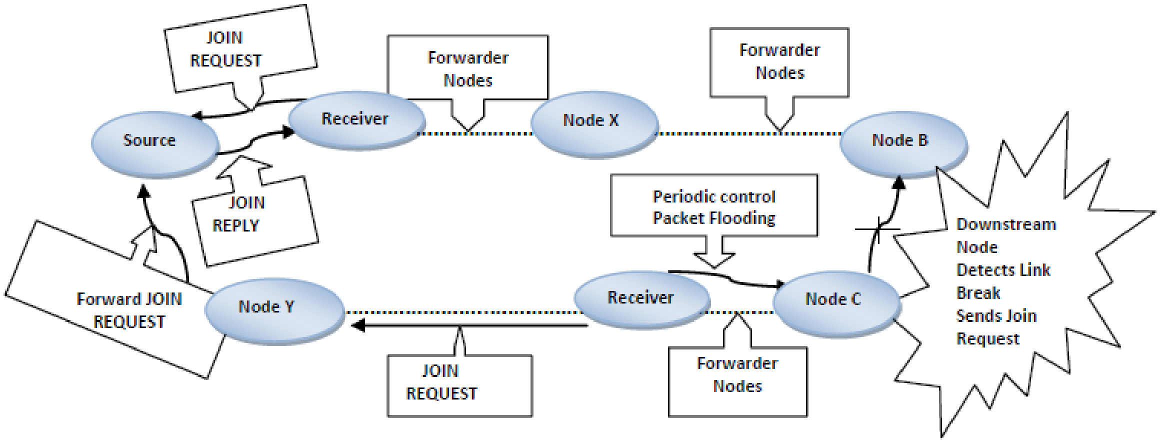


Figure 2.1: Multicast session life cycle

1) Initialization of multicast session

i. Registration

ii. De-Registration

Both Registration and De-Registration can be receiver or source initiated.

2) Multicast Information dissemination of topology

1. Flooding
2. Tree-based (Source or Shared)
3. Mesh-based

3) Multicast Topology maintenance

1. Reactive
2. Proactive

In all the lifecycle stages joining, leaving, rejoining and session maintenance affect the performance of a multicast protocol. The routing scheme used is either reactive or a proactive. A source or a receiver can sends JOIN requests to initiate a multicast group. These requests are propagated until the respective source or a receiver is found, which in turn sends JOIN replies back. The path to the source(s) and/or receiver(s) is established from the JOIN replies and requests received, as they will hold all the addresses of intermediary nodes. These intermediary nodes mark themselves as forwarder nodes.

A session can be ended by explicit leave messages or by implicit periodic updates i.e. not replying to any JOIN requests. Subsequent sections present the classification of MANET algorithms based on above stages of multicast session life cycle.

2.4.2 Initialization of Multicast Session

There are three approaches on initializing a multicast session:

i. *The Source-Initiated approach* - The source constructs a multicast mesh or tree by flooding the network with JOIN Request messages. Any receiver node wishing to join a multicast group replies with a Join Reply message.

ii. *The Receiver-Initiated approach* - any receiver node wishing to join a multicast group floods the network with a JOIN Request message searching for a route or a core (RP) to a multicast group. The membership management of a multicast group is usually assigned to a core (rendezvous) node. All sources of the same multicast group share a single multicast connection through a core.

iii. *The Hybrid approach* is neither receiver-initiated nor source-initiated approach. Both sources and receivers can send JOIN-requests in this approach.

2.4.3 Multicast Information Dissemination

Multicast routing protocols for MANETs can be classified based on how forwarding paths are constructed. Existing multicast routing approaches for MANETs can be divided into tree-based multicast protocols, mesh-based multicast protocols and hybrid multicast protocols:

i. *Tree-based* structures have high data forwarding efficiency. These approaches are also simple, but they lack robustness. Tree-based algorithms are not very efficient in a high mobility environment, as every time a node moves in the tree, a new tree has to be constructed, resulting in considerable overhead as well as data loss during the periods of tree formation. Tree-based multicast routing protocols can be further divided into source-based or core-based as in IP multicasting in Section 2.2.

ii. *Mesh-based* structures are a set of interconnected nodes. Unlike, tree-based approaches, these protocols perform better in high mobility situation as they provide redundant paths from source to destinations during multicasting. Route discovery and mesh building are simultaneously accomplished by using broadcasting to discover routes or by using cores for mesh building. However, mesh-based approaches sacrifice multicast efficiency and simplicity in comparison to the tree-based approaches.

iii. *Hybrid-based* multicast routing protocols combine the best features of tree and mesh-based approaches. Hence, hybrid protocols address both efficiency and robustness.

2.4.4 Multicast Topology Maintenance

Another classification is based on how routing information is acquired and maintained by the mobile nodes. Multicast routing protocols can be divided into proactive routing and reactive routing:

i. In a *proactive* multicast routing protocol, also called table-driven multicast routing protocol, each node periodically exchanges neighbor information with every other node in the network. The periodic exchange helps the node to maintain as well as update routing tables. However periodic updates lead to a relatively high overhead on the network. On the other hand, routes will always be available when required.

ii. *Reactive* multicast routing protocols are also called on-demand protocols. In a reactive routing scheme, routes are set up on-demand. Hence, only when a node wants to communicate with another node, the routing protocol discovers a route. Nodes may encounter long delays for establishing routes before they can forward data.

2.5 Overview of MANET Multicast Protocols

Some MANET broadcasting schemes like MPR flooding [26], Simplified Multicast Forwarding (SMF) [28], MANET unicast protocols called Optimized Link State Routing (OLSR) [32], Ad Hoc On-Demand Distance Vector (AODV) [37] and Multicast Ad Hoc On-demand Distance vector (MAODV) [17] a MANET multicast protocol are briefly described in this section. All these protocols use some of the algorithms previously discussed.

2.5.1 Multi Point Relay (MPR)-based Flooding

The concept of MPRs was developed to reduce the number of duplicate transmissions of pure flooding, while forwarding a broadcast message. In MPR-flooding [26], only subsets of neighbor nodes retransmit messages, unlike the pure flooding, where all the neighbors forward the messages. A neighbor node, which forwards a message, is referred to as relay node of the peer. MPR nodes are chosen based on messages exchanged between one hop neighbors. The information required to calculate the multipoint relays is the set of one-hop neighbors and the two-hop neighbors, i.e. the neighbors of the one hop neighbors. Most protocols use some form of periodic *keep alive* messages or commonly known as HELLO messages to obtain information about one-hop neighbors. In a mobile environment, these HELLO messages are exchanged by each node to refresh current information of their one-hop neighbors. Every node by sending HELLO messages can send its own one-hop neighbor information, so that, the two-hop neighbor set can be computed. Thus, with these HELLO messages, each node can independently calculate its one-hop and two-hop neighbor set. The multipoint algorithm is designed such that every node will select relay nodes, such that, it can reach its entire two-hop neighbors. Figure 2.2 compares normal and MPR – flooding.

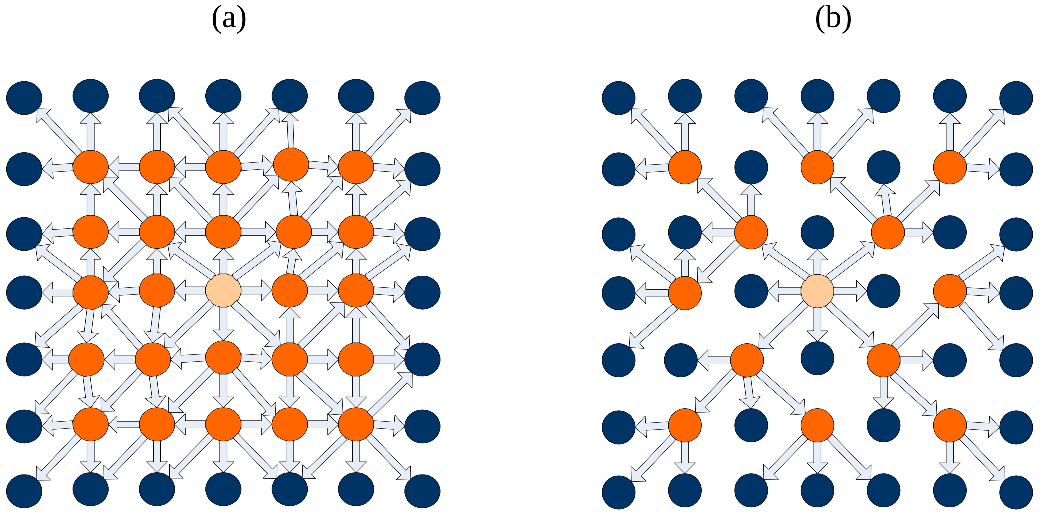


Figure 2.2: Normal (a) and MPR Flooding (b)

The MPR algorithm is designed to provide a near optimal MPR set and is very simple to implement. The problem of selecting optimal MPR set is NP-complete. The two algorithms used in MPR-flooding are:

i. MPR selection for node u

```
- Select as MPR all the neighbors of node u that are the only neighbors of a 2-hop neighbor of node u;
- While an uncovered 2-hop neighbor from u still remains:
    select as MPR a neighbor of u that is neighbor to the largest number of uncovered 2-hop nodes.
```

ii. MPR flooding

```
- Each node u that receives a broadcast message will forward it only if
    The node u is an MPR of the previous hop of the message and has never received the message before.
```

Thus, even though the classical flooding (CF) scheme is more robust and reliable, it consumes a lot of bandwidth. Multi-point relaying gives equally good results [67] with much lower overhead traffic.

2.5.2 Simplified Multicast Forwarding (SMF)

Flooding is the simplest form of broadcasting data in MANETs. However, due to broadcast storm problem in flooding, many mechanisms that minimize the packet forwarders were introduced. SMF also specifies mechanisms for applying reduced relay sets to achieve more efficient multicast data distribution within a mesh topology versus simple flooding. Flooding optimizations include Connection Dominating Set (CDS) (in graph theory, a *dominating set* (DS) for a graph is a set of vertices whose neighbors, along with themselves, constitute all the vertices in the graph, a connected DS (CDS) is a DS forming a connected graph), Multi

Point Relay (MPR) [26] set etc. SMF is one such simple scheme that tries to minimize the problems in CF scheme. SMF basically is comprised of three parts:

(i) a sequence id generator and marker to be used when and if necessary,

(ii) a duplicate detection module, and

(iii) a basic multicast packet forwarding module.

All these three modules help in providing a working prototype compatible with existing and emerging IP network protocol frameworks. The sequence generator is responsible for marking each packet with a monotonically increasing unique identification number when existing IP kernel methods are not sufficient or are not predictable. The duplicate detection mechanism is used to remove and detect duplicate packets from both entering the interface forwarding process and from being delivered to upper layer applications. The multicast forwarding module is flexible in its design and presently supports different flooding design optimizations. The current experimental mechanisms are: CF, source-specific multi-point relay (S-MPR) flooding, and non-source multi-point relay (NS-MPR), Essential Connecting Dominating Set (E-CDS) and Multipoint Relay Connected Dominating Set (MPR-CDS).

The S-MPR flooding mechanism is based on the MPR technique described in Section 2.5.1. The current algorithm selects MPRs which are one-hop away, to build a reduced relay set to reach all of its two-hop neighbors. S-MPR allows only locally elected MPRs to retransmit packets that are received from upstream nodes. Symmetric two-hop neighbor knowledge can be collected via single HELLO exchanges. Source-specific MPRs compose a connected dominating set, and using S-MPR significantly reduces redundant retransmission of packets

[31], especially in dense network neighborhoods. However, there is an implementation disadvantage of S-MPR as it requires previous hop identification to perform a proper forwarding match, thus, adds some additional state and complexity to the design.

A flooding technique that does not require previous hop information during the forwarding decision process and overcomes S-MPR drawbacks is called NS-MPR. The NS-MPR mechanism combines all source-specific elected MPRs into a common relay node set. In this case, only knowledge that a node is an MPR for at least one neighbor is used and previous hop information is not required during the active forwarding process. However, NS-MPR does not scale well as compared with the S-MPR approach. That is, there is no significant decrease in a combined resultant relay set when compared to a source-specific relay set. Research is still carried out to investigate optimization algorithms, to form common relay set not requiring previous hop knowledge.

The third flooding scheme called Essential Connected Dominating Set (E-CDS) is based on the E-CDS algorithm described in a proposal for MANET extensions to OSPF using CDS flooding [63]. The E-CDS algorithm forms a single CDS mesh for the entire network similar to NS-MPR and allows nodes to use 2-hop neighborhood topology information to dynamically perform relay self election to form a CDS. Nodes elect themselves as relays using neighborhood router priority information. Priority values need not be unique and can be a combination of values such as power level, number of one-hop neighbors and address values. For nodes to correctly assign themselves as relays, priority values need to be learned within a two-hop neighborhood. E-CDS nodes select themselves as relays if and only if:

i. the node's router priority is greater than all its two-hop neighbors, or
ii. there does not exist a path from the highest priority neighbor to all other one and two hop neighbors using only nodes with greater priorities as relays.

With E-CDS, any SMF node that has selected itself as a relay performs duplicate detection (DPD). E-CDS, unlike SMPR, does not guarantee minimal hop paths for end to end connections. Because E-CDS uses a shared CDS, there may be higher traffic concentration within the network forwarding paths compared to source based approaches.

The final algorithm presented is the MPR-based Connected Dominating Set (MPR-CDS) algorithm [64]. The number of forwarding nodes in MPR-CDS is reduced to a more efficient subset of MPRs than the simple NS-MPR described previously. MPR-CDS requires that nodes know a unique ordering identifier for each node within their two-hop neighborhood. After neighborhood discovery, a node using MPR-CDS will forward all unique packets if and only if:

i. the node's identifier is higher than all its one-hop neighbors, or
ii. node has been selected as an MPR by the node that has the highest identifier in its one hop neighborhood.

Like E-CDS, MPR-CDS approach results in a common relay set, and does not guarantee minimal hop paths. MPR-CDS also has no requirement for previous hop knowledge similar to other shared CDS algorithms. MPR-CDS has similar scaling properties to both E-CDS and S-MPR [65].

All the SMF forwarding schemes present robustness to changes in topology caused by network mobility and increasing traffic loads. However, there is still ongoing research on the interoperation of SMF with multicast MANET border routers and other existing exterior multicast protocols.

2.5.3 Optimized Link State Routing (OLSR)

OLSR is a link state algorithm modified for mobile networks. In OLSR, only MPRs forward link state information. Furthermore, only partial link state information is exchanged between MPRs. The link state information is used to calculate OLSR routing tables.

We will discuss three important stages involved in maintaining OLSR routing tables.

i. **Link sensing:**

MPR link state information is exchanged between the mobile nodes through the exchange of HELLO packets. The HELLO packet message format is shown in the Figure 2.3 [32]. HELLO packets are periodically transmitted over the interfaces to detect connectivity with the neighbors. A link is assigned a status like ‘*symmetric* ‘or '*asymmetric*' based on whether a pair of HELLO packets are heard or not heard from both the directions on the links respectively. This way a node maintains a link set, which contains the information of links to its one-hop neighbors.

ii) Neighbor detection and MPR selection:

Based on the link set on a node obtained from the exchange of HELLO packets, a neighbor set is created. A node is called a neighbor of another node if and only if there exists at least a link between them. Nodes also maintain a two-hop neighbor set, that is, a set of nodes which have symmetric link to symmetric neighbors. The MPR set is computed based on two-hop neighbor set. A node will select the MPRs such that any strict two-hop neighbor is covered by at least one MPR node.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |     Htime     |  Willingness  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                             .  .  .                           :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved    |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                                       :
```

Figure 2.3: OLSR Hello packet format [32].

The MPR list is recalculated every time there is a change in the link state information which results in a different one-hop and two-hop neighbor set.

iii) Topology control message diffusion:

A node announces its link-set by flooding *Topology Control (TC)* messages. TC messages are flooded through MPR flooding. TC messages use an *Advertised Neighbor Sequence Number* to ensure the “freshness” of the announced link-set. TC messages are sent at regular intervals, and are also triggered by link-set changes and MPR selection set changes. The TC message format is shown in Figure 2.4 [32].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ANSN             |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2.4: TC message format [32]

2.5.4 Ad hoc On-demand Distance Vector (AODV)

The information in this section was obtained from the Ad Hoc On Demand Distance Vector Protocol (AODV) RFC [37]. AODV is a reactive protocol, that is, the routes are created and maintained only when they are needed.

Route Request (RREQ) is flooded by the source host to find the path to destination host. The RREQ message includes the destination sequence number, which not only prevent loops, but prevents old information to be replied to the request. The source host finds the destination host's sequence number from its routing tables which stores information about the next hop to the destination and a sequence number.

On receiving RREQ messages, the intermediary nodes update their routing tables. The destination host or any intermediate node (if it has the path to the destination) can reply using Route Reply (RREP) message. Each host also has its own sequence number, which must be incremented in two different cases:

i. before source host sends RREQ message, and
ii. when the host sends a RREP message responding to the RREQ message.

AODV uses a third type of message called Route Error (RERR). When a node detects any breaks in the active routes it sends RERR messages toward the source. Link breaks can be detected via periodic HELLO messages. The host originating RERR messages should increment the RERR message sequence number before broadcasting it locally, to prevent replies for old RERR messages.

AODV reduces the overhead of maintaining routes at the cost of increased latency in finding new routes. The AODV protocol will perform better in networks with static traffic and relatively small number of source and destination pairs, unlike OLSR which is more efficient at a high density and random traffic.

2.5.5 Multicast Ad hoc On-demand Distance Vector (MAODV)

MAODV is a multicast extension of AODV. In MAODV, all members of a multicast group belong to a tree (which includes non-member nodes required for the connection of the tree), and the root of the tree is the group leader. Multicast data packets are propagated using the tree.

The core of the MAODV protocol is on the tree formation, maintenance, repair the tree and tree merging. There are four types of packets in MAODV: RREQ, RREP, Multicast Activation (MACT) and (Group HELLO) GRPH. RREQ and RREP are also packets in AODV. A node broadcasts a RREQ when

i. it is a member node and want to join the tree, or
ii. it is a non-member node and has a data packet targeted to the group.

When a node in the tree receives a RREQ, it responses with RREP using unicast. Since RREQ is broadcasted, there may be multiple RREPs received by the originating node. The originating node should select one RREP that has the shortest distance to the tree and unicast a MACT along the path to set up a new branch to the tree. GRPH is periodically broadcasted by group leader to allow the nodes in the tree to update their distance to the group leader. More detailed information on MAODV can be found in [17].

2.6 Motivation

Although there are different multicast protocols in both wired as well ad hoc networks, most of the protocols tradeoff overhead and reliability. For improving reliability, protocols are designed to maintain network states at each host, thereby, increasing the complexity. This works well with wired networks, but in MANETs, maintaining a network state at each host is in itself a bigger problem due to the high degree of mobility involved and high rate of link variations.

In general, no single multicast routing protocol is efficient for all mobile network scenarios. Hence, it is always desirable to design multicast protocols that will adapt well to all types of network conditions like mobility, traffic, density etc.

CHAPTER 3

Scalable Simple Multicast Forwarding (SSMF)

The need for different multicast forwarding schemes for different MANETs or topology scenarios has been presented in the previous chapters. The requirement for multicast algorithm's independency with respect to any unicast protocol was also discussed previously. There is a need for protocols which can integrate well with already existing multicast protocols in the wired network.

3.1 Problem Formulation

A single multicast protocol is not suitable for all types of networks and aims to meet maximum possible requirements for different network conditions. Section 2.4 described some features required by a multicast algorithm to function efficiently in MANETs. For the efficient function of multicast algorithms, there is a need to separate multicast data dissemination for different network scenarios. The two basic network scenarios or conditions can be the following:

i. Localized scenario: that has a very dense network of multicast receivers.
ii. Scattered Scenario: that has a few and a scattered network of multicast receivers.

A protocol able to adapt to different network conditions can make it highly efficient. Two different broadcasting schemes can be proposed for the above two network scenarios:

i. For the first condition (the localized scenario), a proposal of a broadcasting mechanism is limited flooding. The scope of flooding can be limited by appropriately choosing a TTL of the packets to be flooded.
ii. For the second scenario (scattered network), flooding can be combined with any underlying unicast protocol to reach all the scattered hosts.

The limited flooding combined with any MANET unicast protocol will eliminate unnecessary dissemination of data as well as its dependency with respect to any specific unicast protocol. The combination not only allows flooding to reach dense part of a network, but also allows sources to reach scattered hosts very sparsely located, through unicast, ensuring reliability of packet delivery as well as saves lot of bandwidth with limited flooding.

3.2 Design

The proposed approach goals and definitions defining various stages of a multicast protocol life cycle are presented in this section.

3.2.1 Goal

IETF currently considers Simple Multicast Forwarding (SMF) scheme [28] (presented in section 2.5.2) for forwarding multicast packets in MANETs. SMF uses MPR-flooding as one of its options. The major advantage of this scheme is its simplicity and efficiency with respect to node movement. However, SMF also has some disadvantages like needless data duplication i.e., sources even with no receivers flood the entire network. SMF also does not properly integrate with existing wired protocols like PIM. There is still ongoing research on the integration of SMF with other exterior multicast protocols. Our goal is to modify SMF so that it can overcome aforesaid problems.

3.2.2 SSMF

In our proposed multicast protocol SSMF:

i. All the multicast receivers in a group have to register to all potential multicast sources in the multicast group of interest. The registration packet(s) can be unicasted and/or MPR-flooded (limited flooding) to the sources;

ii. The sources store a list of all interested receivers in their multicast group in a receiver table. Each receiver entry is accompanied by its distance (in terms of hop count) from this particular source;

iii. For each list of receivers, a source will compute a combination of limited scope flooding (with a limited TTL) and unicast to reach all the multicast receivers. Hence, the scheme aims to minimize the flooding overhead compared to that of SMF, by choosing the best combination of limited flooding (with TTL) and unicasting.

3.2.3 SSMF Definition

The above proposed scheme is named as Scalable Multicast Forwarding (SSMF) and this section provides detailed SSMF protocol definition.

When a receiver joins a multicast group (and periodically thereafter), the node simply broadcast JOIN requests to a multicast group; the multicast sources for a particular group, on receiving these requests, add them to their receiver tables. Since registration is initiated by the receivers, SSMF is a *receiver-initiated* approach.

For *de-registration*, two methods can be employed:

i. An explicit deregistration message can be sent to the multicast source by its multicast group's receiver. Upon reception of this explicit message, the source simply removes that receiver from its receiver table.
ii. Since receivers periodically broadcast their group membership to their multicast group(s) source(s), these sources(s), store the active receiver(s) and their hop count. A timer can be employed on the sources receiver tables to periodically flush the inactive receivers upon timeout.

For the implementation of SSMF in this thesis both of the above methods are used.

In this thesis, the timeout value is chosen as three periodic updates plus a small guard time of about (0.05s), that is, if a source does not get three consecutive periodic updates, then source will wait some extra guard time period, before eliminating the receiver from its tables.

The multicast sources store in the receiver tables the distance or number of hops to the respective receivers. The sources will update the hop counts for each receiver by one of these methods:

i. if a proactive unicast routing protocol is being used, the hop count can be directly obtained from the unicast routing table, or
ii. from the periodic updates messages from the receivers (the periodic SSMF JOIN messages will carry this hop count values); if multiple updates are received, the minimum value is used.

In this thesis for robustness, the maximum of hop count values obtained from the above two ways are used.

Proceeding to SSMF's *dissemination of multicast data*, a source will have to choose between a combination of limited scope SMF flooding, and unicasting. The mechanism for choosing the correct combination of flooding and unicast is explained in section 3.3.2

In the broadcasting mechanism of SSMF, decision of flooding with TTL and unicast depends on knowing the current TTL values. To guard against an increase in the TTL values due to changes in the links, limited scope flooding should be carried out with an additional safety factor. Depending on the rate of link changes, the rate of updating the TTL values and an appropriate 'safety' factor can be chosen. However, if the TTL is obtained from a responsive proactive unicast routing protocol, then, only a small safety is required as updating the TTL value is accomplished by the underlying unicast protocol. The safety value added to the optimum TTL value will represent a trade-off between registration and data overhead.

3.3 Protocol Specification

A SSMF protocol specification that includes SSMF message format is presented in this section.

3.3.1 SSMF Message Format

The basic layout of any packet in SSMF from a receiver (omitting IP and UDP headers) is shown in Figure 2.5:

a) <u>Join Request:</u>

0 7 8 15 16 31

<table>
<tr><td>Message Type</td><td>Periodic update time-out</td><td>Message Sequence Number</td></tr>
<tr><td colspan="3">Multicast Group Address</td></tr>
</table>

Figure 2.5: The JOIN request packet format from an SSMF receiver

The packet body consists of one or more SSMF messages which are preceded by a message header for each included message. The message header contains the following fields:

i. ***Message type*** {JOIN Request}: A receiver initially sends a Join request to join a multicast group and then sends periodic JOIN messages to let the multicast group sender(s) know its hop count or to maintain its group membership.

ii. ***Periodic update timeout***: timeout for a particular entry allows different receivers to have different timeout periods.

iii. ***The multicast group address:*** Address field contains field contains the main address of the group, the receiver wants to join.

iv. ***Message sequence number:*** To ensure message freshness, each registration messages has a sequence number, and only new messages are accepted.

3.3.2 Adaptive TTL Flooding

This section presents the calculation of the optimum TTL for limited flooding in SSMF.

Assume that N nodes are uniformly distributed in a rectangular lattice. Given a node i in the lattice, there are 4k nodes at a distance of k hops from i. A flood with a TTL=k will have N_k transmissions, where,

$$N_k = 0.45(1 + 2k(k+1)) \qquad (1)$$

Since the flooding mechanism in SSMF is SMF, the total number of forwarding nodes will be less than N_k. The worst case for this scenario is pure flooding, where, every node in the relay set is an MPR. According to the literature [57-58], the number of total number of forwarding nodes by using MPR flooding will be approximately 45% of N_k. The validation of this approximation is presented in Chapter 4.

Similarly, there is a need to calculate the number of forwarding nodes or overhead data packets for unicast messages in order to quantify the tradeoff between the SMF flooding and unicasting to a group of SSMF multicast receivers. For unicast delivery, the forwarding hops will be same as the value of hop count or TTL in the unicast tables of senders.

Based on the overhead for a certain TTL and unicast, the optimum TTL (the one that minimized the number of transmissions) can be computed.

For example, let there be a source and five receivers located at one, two and three hops away from the source shown in the Figure 3.1.

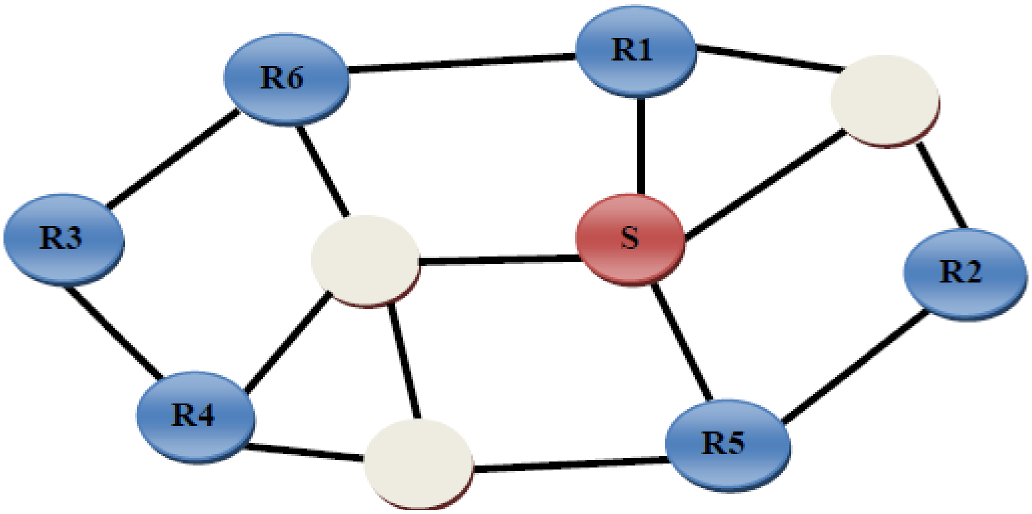


Figure 3.1: An example MANET with a source and multicast group consisting five receivers.

Table 3.1 shows the receiver table for the network topology shown in Figure 3.1. Table 3.2 shows how an optimum TTL can be chosen based on the information in the receiver table.

Table 3.1: SSMF receiver table at the SSMF source for topology in Figure 3.1.

Receiver	**Hop Count**
R1	1
R2	2
R3	3
R4	2
R5	1
R6	2

Table 3.2: Overhead calculations for optimum TTL

TTL *{k}*	**Data overhead due to SMF (*Packets)*** **{*.45 * [1+2k (k+1)]* }**	**Data overhead due to Unicast (*Packets)*** ***{Hop count from periodic update table}***	**Total Data Overhead {*Packets}***
0	0.45	11	11.45
1	2.25	9	11.25
2	5.85	3	8.85
3	11.25	0	11.25

The flooding TTL =0 i.e., k = 0 represents a case, when no limited flooding occurs and only unicast to all the group members will take place. If TTL is set to 1 for limited scope flooding using SMF, since, receivers R1 and R5 are one hop away from source, SMF flooding will cover them. The other receivers that are more than a hop away need the data to be unicasted to them. Since R2, R4 and R6 are two hops away and R3 is three hops away, total unicast packets generated is the sum of their hop counts i.e. in this case it is 2+2+2+3 = 9 packets. A similar calculation can be performed for each TTL. In this example, the optimum TTL is equal to two, as the combination of unicast and SMF flooding generates less data overhead for this TTL value, and the remaining receivers which are not covered by limited SMF flooding are unicasted.

CHAPTER 4

Results and Analysis

We used network simulations to verify and evaluate the performance of SSMF in a wide range of scenarios. This chapter describes the implementation and presents the results and the analysis by using NS-2.

4.1 Evaluation Techniques for MANET Multicast Protocols

Simulation is an established and widely used method for conducting performance evaluation of network protocols [33]. Simulators such as NS-2 [34] or OMNeT++ [35] come with built-in support for the most popular MANET network protocols such as OLSR [32], AODV[37], DSDV[38], DSR [39] etc., as well as IP Multicast protocols like PIM and DVMRP[15]. The entire protocol stack can thus be simulated, enabling validation of new protocols or algorithms implemented as additional code or scripts. Certain approximations and simplifications are, however, often made when simulation models are used, which leads to biased conclusions. Simulations of ad-hoc networks, have for example, been criticized for not using valid mobility models [40] or for relying on only one specific scenario [41]. The network simulator itself may also include errors or assumptions such as an unrealistic propagation model [41]. Despite many pitfalls and possible errors when performing simulations, performance evaluation by simulation is very common for validating the protocol design. There are alternative evaluation methods. Some multicast protocols are only evaluated analytically or by emulations [42, 43]. Others are implemented for small real-world experiments [33]. Different multicast protocols often take advantage of completely different scenarios in their simulation and experiments [44]. Hence, studies are difficult to compare.

4.1.1 Evaluation of SSMF

In this thesis, the implementation of SSMF protocol is done by extending the models in NS-2. Using NS-2, the performance of the SSMF multicast protocol for MANETs is studied and compared with some of the existing MANET multicast protocols. The SSMF protocol is implemented in NS-2 by integrating NRL-OLSR [45] and extending its existing SMF code (as both OLSR and SMF use MPR-forwarding as their broadcasting mechanism).

4.2 NS-2 Simulation Environment

SSMF implementation details are presented in this section.

4.2.1 SSMF Implementation

The SSMF is implemented as an extension to SMF from NRL-OLSR. The extension is made both by altering the original code and by adding new source code files to NRL-OLSR. Using the new code, NRL-OLSR is able to handle new SSMF messages such as periodic updates as if they are original OLSR messages.

The major building blocks involved in the transmission of the update messages are: the original NS-2 code, SMF and the SSMF extension. The original NRL-OLSR files that were modified for implementing SSMF are Nrlolsr, ProtolibManetKernel and NrlolsrAgent. SSMF code extension files include MultiQueue and limited flooding. The main parts of NRL-OLSR and SSMF code extensions to NRL-OLSR are shown in the Figure 4.1

.

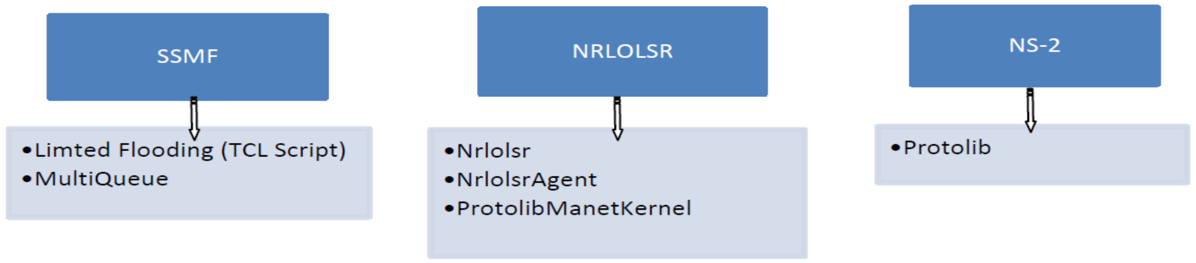


Figure 4.1: NRL-OLSR and its SSMF code extensions

The NRL-OLSR source code was extended to include receiver tables with the help of additional C++ class - ‘*MultiQueue’* source code files.

4.3 Simulation Settings

In this section, the simulation environment and settings are described. The network simulator version 2.27 is used, as NRL-OLSR has designed the protocol package for this version. All the simulations are run on Linux with a 2.6.18-238.1.1.el5 kernel version.

For the physical layer: transmission distance and carrier sense distance values are selected, such that, they cover 250m and 550m distance respectively. A Rayleigh propagation model is used. For the MAC layer 802.11g, broadcast and unicast rates are set to 6Mbps and 54Mbps respectively. In the network, OLSR is used as the unicast routing protocol and SMF used as forwarding protocol for SSMF. UDP is used for packet generation with packet size set to 1500 bytes. The UDP data rate is periodic with a jitter of 0.001 seconds. The offered traffic load in the network determines the periodicity of the UDP traffic and accordingly jitter is also varied. The simulation time is set to 5400s. The mobility model is set to random way point model (RWP) , in which, nodes move at any random speed between (5, 15) m/sec. The simulation field is 1350m X 1350m.

4.4 Performance Evaluation Metrics

The following performance evaluation metrics are used in the evaluation of SSMF [11].

Packet Delivery Ratio is the ratio of the number of packets delivered without duplicates to the destinations and the number of data packets that should be received. This ratio represents the effectiveness and throughput of a protocol in delivering data to the intended receivers (group members) within the network. The total number of received packets by all receivers is divided by the number of packets sent from senders multiplied with the total number of receivers.

Average end-to-end delay*:* is the average delay of data packets from application layer of multicast source to application layer of the multicast group members. This delay includes all the queuing and protocol processing delays as well as propagation and transmission delays. The lost packets are not accounted in the average delay.

Control Overhead: is the total number of control bytes transmitted. Control packets do not carry any user payload. The control overhead shows how efficiently control packets are utilized in delivering data. In computing the overhead the following kinds of packets are considered: the OLSR Hello, TC messages, the periodic updates of multicast receivers to multicast sources.

4.5 Simulation Parameters

The performance is evaluated by varying parameter one at a time. The parameters chosen for the simulation are shown in the Table 4.1, with the default values shown in square brackets.

Table 4.1: NS-2 parameter settings; the default values are shown in square brackets

Parameter	**Value**
Pause times (s)	5400 ,4200,3600,3000 ,[2400],1200,600
Offered Load (packets/s)	1,2,5,[10],20,50,100,200,400
Network size (nodes)	10,20,[50],100
Multicast Group Size (nodes)	1, 2, 5,[10], 50,100
Number of Senders (nodes)	[1], 2, 3,4,5
Rate of joins/leaves (per s)	[0],0.01,0.02,0.04,1,2,5,10,20
Hello intervals (s)	[50],60,100,150,250
TC and Periodic Intervals (s)	150,[250],350,550,650
Safety Factor (hops)	0,[1],2,3,4,5

4.6 Simulation Results

4.6.1 Varying HELLO Interval

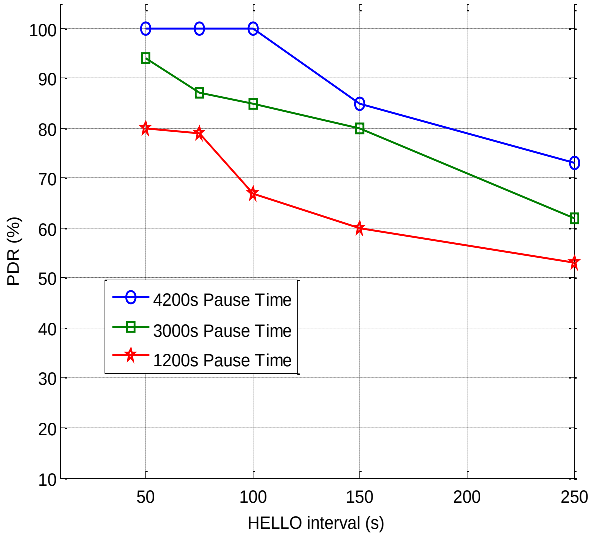


Figure 4.2: Packet delivery ratio as a function of HELLO interval

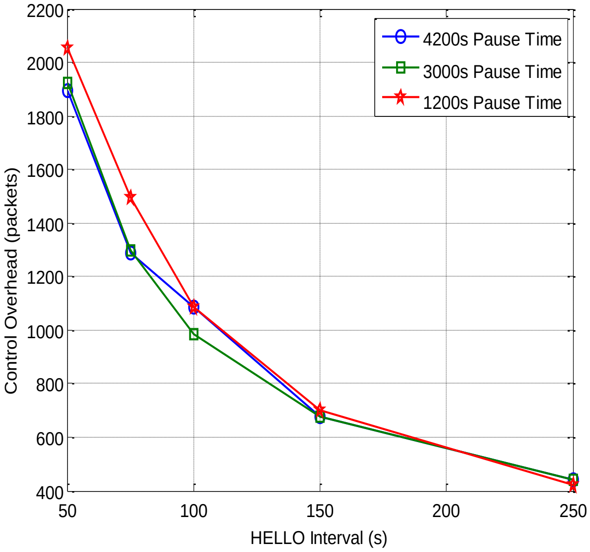


Figure 4.3: Control overhead as a function of HELLO interval

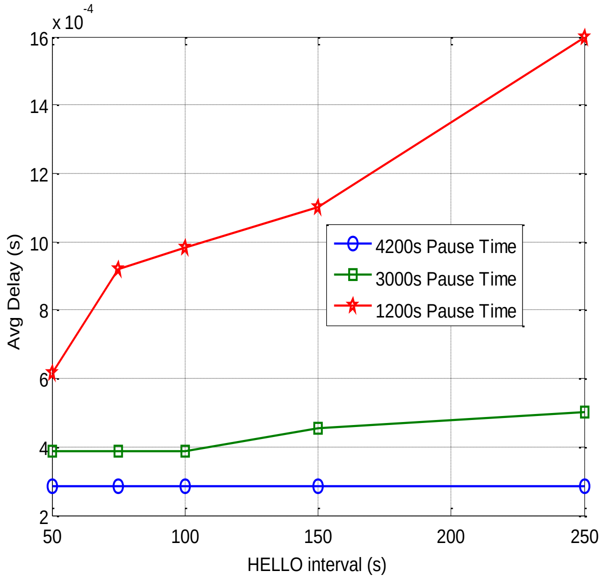


Figure 4.4: Average end-end delay as a function of HELLO interval

Figures 4.2, 4.3 and 4.4 depict the PDR, control overhead and delay for different HELLO intervals. Smaller HELLO intervals speed up neighbor and link failure detection leading to smaller end to end delays and increasing PDR, however, control overhead also increases leading to congestion in the network.

4.6.2 Varying TC and Periodic Update Intervals

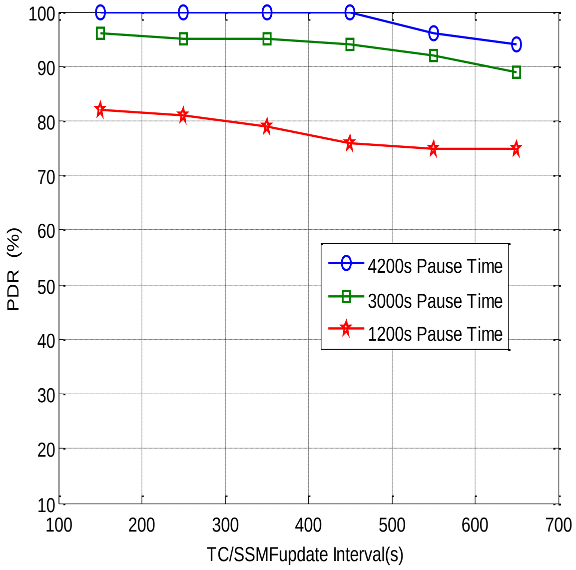


Figure 4.5: Packet delivery ratio as a function of TC and Update interval

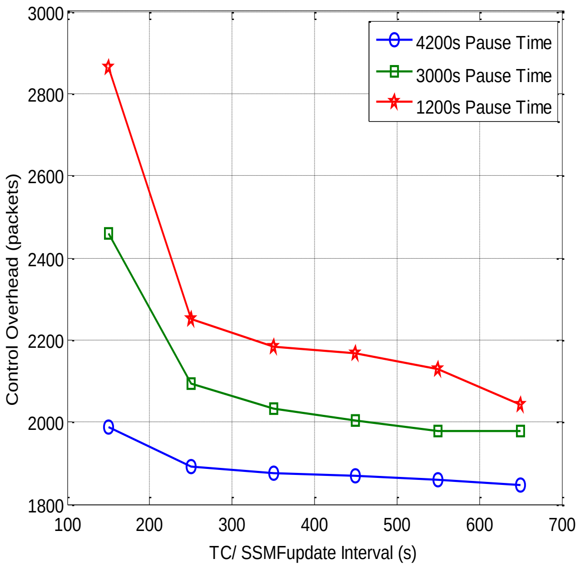


Figure 4.6: Control overhead as function of TC and Update interval

The Figures 4.5, 4.6 and 4.7 shows the PDR, control overhead and delay for different TC and SSMF update intervals. The smaller is the TC and update interval, the higher is the frequency of TC and update message exchanges, the more current the topology information on every mobile multicast source and receiver, the lower is the delay and the higher is the control overhead.

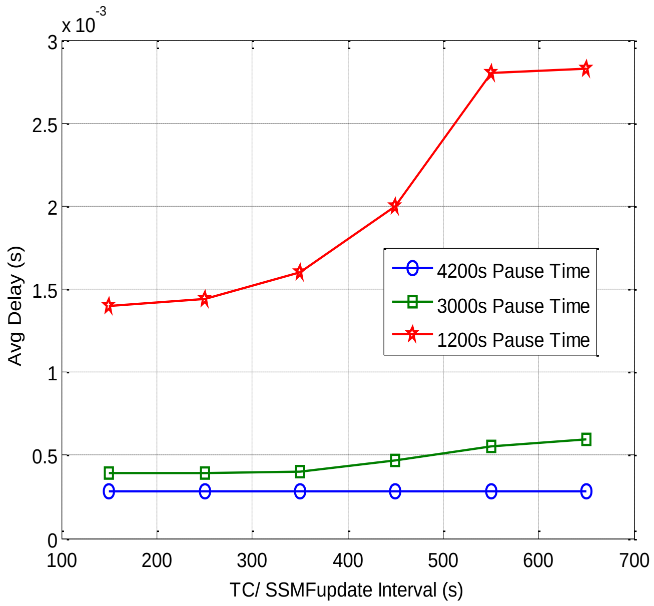


Figure 4.7: Average end-end delay as a function of TC/ SSMF update interval

4.6.3 Varying the Safety Factor

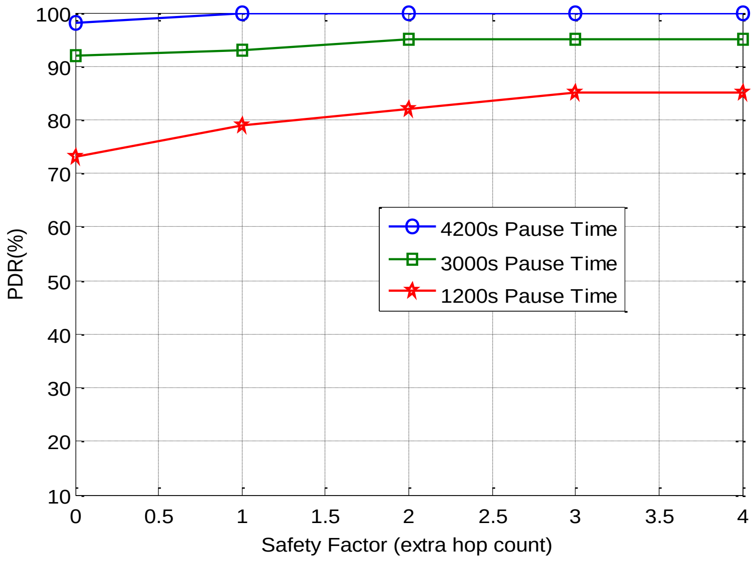


Figure 4.8: PDR vs. TTL Safety (extra safety for TTL during limited flooding)

Figure 4.8, 4.9 and 4.10 shows the performance of PDR, control overhead and delay for different safety factor hops. The TTL of the SSMF multicast broadcast packet increases when increasing safety factor, leading to an increase in the PDR. The graphs show that the low mobility cases (higher pause times) do not benefit from this safety factor. However, in high mobility cases the impact is greater, that is, increasing safety factor results in increasing of the PDR as the hop count increases because these extra safety hops act as guard hops against the mobility of the node. The delay and overhead are almost constant in low mobility cases, but in high mobility cases, when they are already affected by high congestion, the delay and overhead increase due to some amount of congestion added by packets with high TTL values, before they can be discarded.

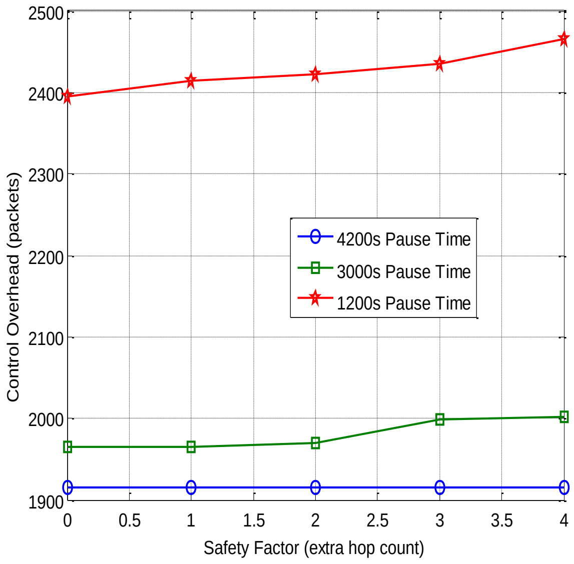


Figure 4.9: Control overhead as a function of (extra safety for TTL during limited flooding)

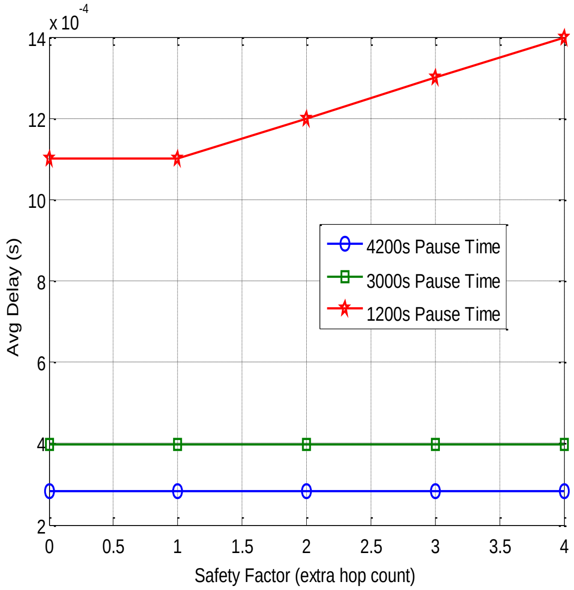


Figure 4.10: Average end-end delay as a function (extra safety for TTL during limited flooding)

4.6.4 Validation of Control Overhead Calculation

In the Figure 4.11, the number of expected transmission (equation number: (1) presented in Section 3.3.2) as a function of the TTL of the flooded packets. Considering the large range of mobility the formula is a reasonable approximation of the overhead as function of TTL.

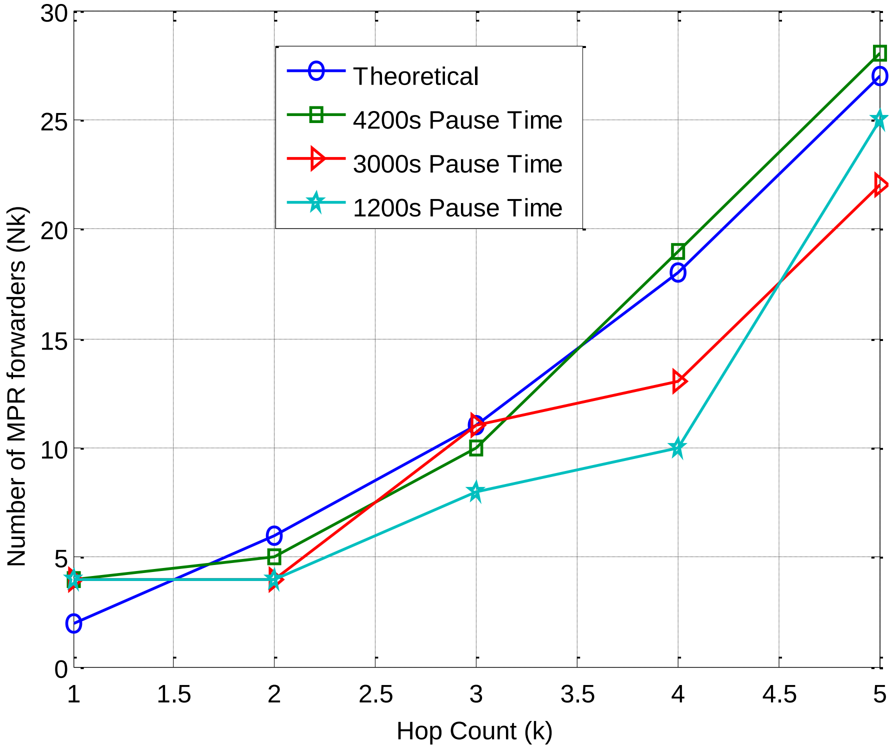


Figure 4.11: Validation of formula in Section 3.3.2 for different hop counts for both fixed and mobile networks

For all the results in the following sections, the following four protocols are compared:

i. SSMF - SMPR: *uses OLSR for unicasting and SMF (S-MPR) scheme for limited (by TTL) flooding.*

ii. SSMF - SIMPLE : uses *OLSR for unicasting and simple flooding (limited by TTL)*

iii. SMF: *uses SMF (S-MPR) scheme.*

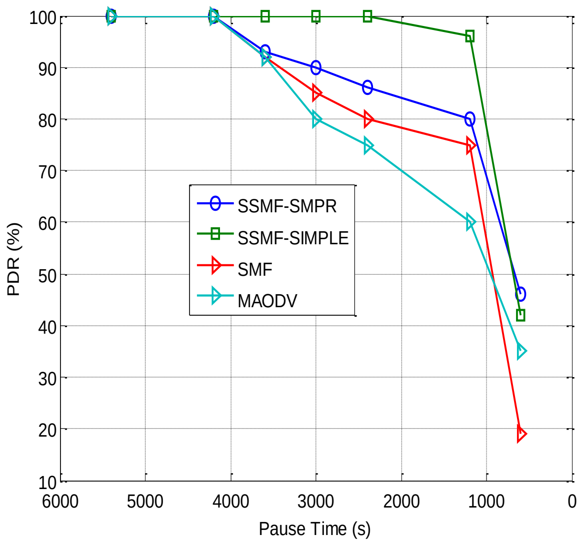
iv. MAODV: *uses MAODV (presented in Section 2.5.5) for multicasting.*

4.6.5 Mobility Simulation Results

These simulations are performed by varying the pause times in the RWP mobility model. The graphs for different performance evaluation metrics are show in Figures 4.12, 4.13 and 4.14. Figure 4.12 compares the packet delivery ratio of the four protocols at varying mobility conditions. The graph demonstrates that both the variants of SSMF (S-MPR flooding or simple flooding) performed better than MAODV and SMF. Increased mobility causes frequent link changes, due to which MAODV has to reconfigure the multicast tree more frequently resulting in a decreasing PDR. SSMF with SIMPLE flooding performs better than SSMF with S-MPR flooding as it is resistant to mobility, whereas in S-MPR, mobility affects selection of relay nodes and until convergence there is a possibility of sub-optimal routing and missed packets.

Figure 4.13 shows that due to increase in the mobility, there is a frequent change in the link state and this causes changes in MPR node list, which in turn, results in periodic broadcast of HELLO message and Topology Control (TC) messages in order to discover neighborhood nodes and hence an increasing trend of control overhead of SSMF. AODV on the other hand has lower network load due to the fewer routing information packets kept in its cache. Since AODV (MAODV) is a reactive protocol, it has greater overhead when compared with the

OLSR (SSMF) at higher mobility or lower pause times due to sudden increase in transmission of RREQ and RREP to maintain multicast trees.



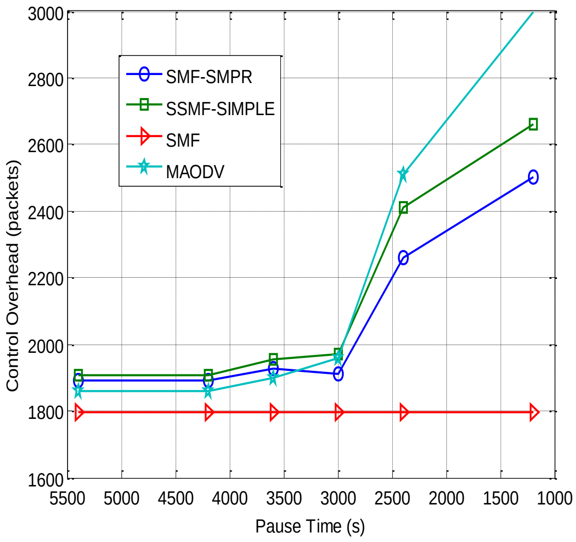


Figure 4.12: Packet delivery ratio as a function of pause times

Figure 4.13: Control overhead as a function of pause times

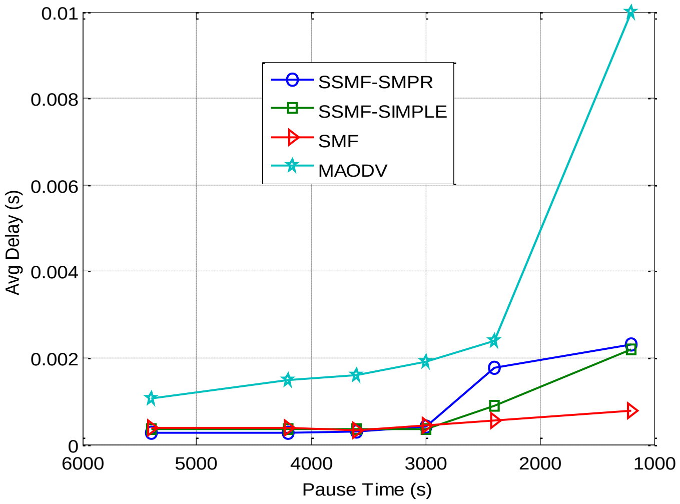


Figure 4.14: Average end-end delay as a function of pause times

Figure 4.14 shows the performance of the end-to-end delay under increasing mobility. Higher mobility causes more links to be broken, which results in frequent re-routing and thus larger end-to-end delays. MAODV shows the worst delays due to the process of reinitializing the route flooding process every time it detects a change in topology (due to mobility) to discover new routes. SMF has better medium access delay as it uses MPR-flooding, which does not need any route acquisition eliminating latency, whereas SSMF, along with flooding uses simultaneous OLSR unicast resulting in slightly higher delays.

4.6.6 Offered Load Simulation Results

The offered load simulations are performed by varying the offered load to the network. The graphs for different performance evaluation metrics are shown in Figures 4.15, 4.16 and 4.17. In Figure 4.15 the PDR decrease is due to the high contention the network experiences as the traffic rate and network load grow, resulting in increased packet loss. The graph shows that SSMF with the combination simple flooding performed better than the other three protocols, whereas, SSMF with S-MPR when compared to pure SMF or MAODV, performed better in low traffic conditions and are comparable in high traffic scenarios.

Figures 4.16 and 4.17 show the performance of the control overhead as well as end-end delay under increasing network load. A high offered load causes more collisions and frequent loss of packets and thus causes larger end-to-end delays. MAODV responds by injecting three kinds of packets, i.e., RREQ, RREP and MACT packets. As a result, many RREQ packets may be flooded if a RREP packet is not received soon enough. The injection of these packets may lead to more link breaks due to the loss of more HELLO packets in collisions, which in turn leads to the injection of more RREQ, RREP and MACT packets, in an attempt to fix these new link breaks. As a result of this cyclic nature of congestion, there is sharp increase in control overhead and delay of MAODV. In case of SMF and SSMF, MPR flooding of overhead packets limits its control overhead and delay. However, for OLSR as the traffic load increases the rate at which HELLO and TC messages are sent also increases due to link breaks resulting in the increase of both the metrics on network saturation but almost constant on lower traffic loads.

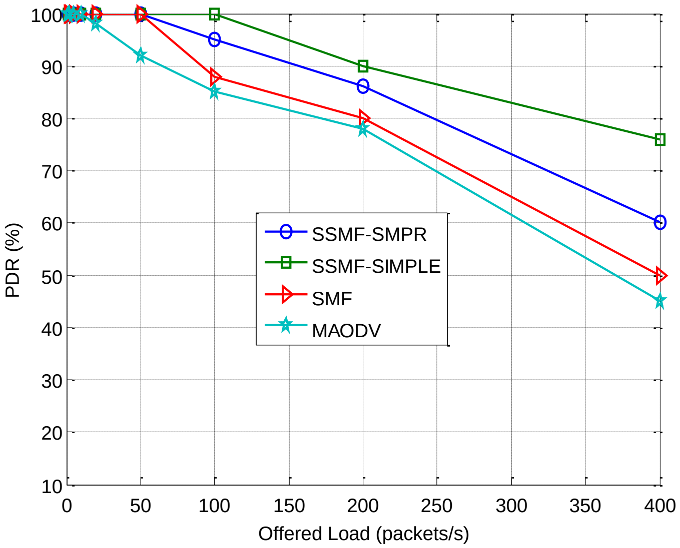
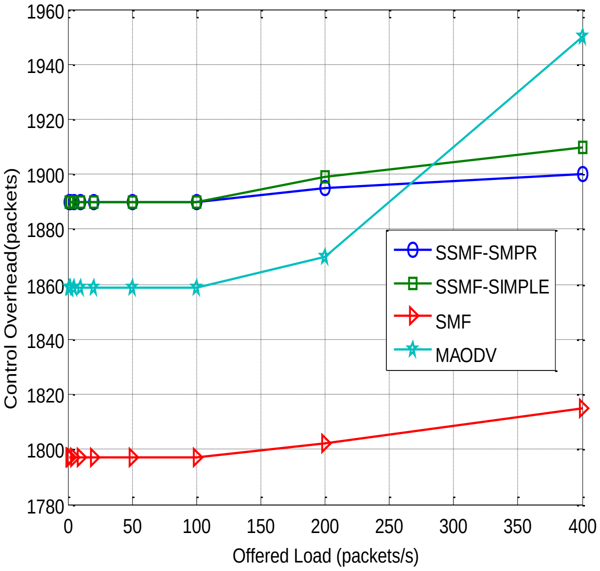


Figure 4.15: PDR as a function of offered load



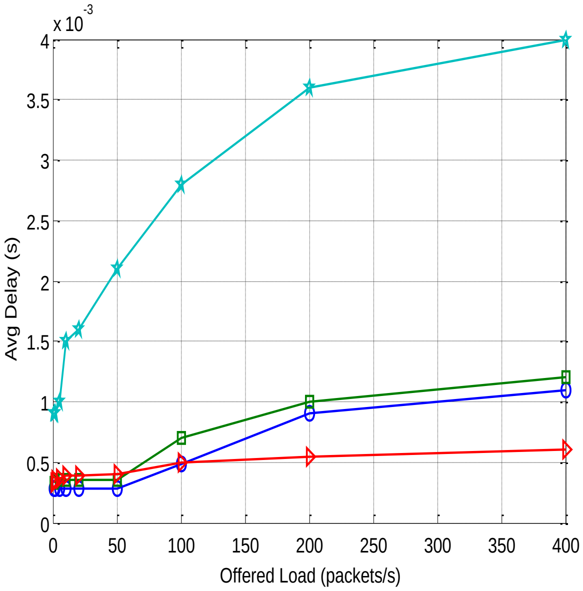


Figure 4.16: Control overhead as a function of offered load

Figure 4.17: Average end-end delay as a function of offered load

4.6.7 Network Size Simulation Results

For performing these simulations the number of nodes in the topology is varied. The graphs for different performance evaluation metrics are shown in Figures 4.18, 4.19 and 4.20.

Figures 4.18 and 4.19 depict the PDR and the control overhead when the node density or network size is increasing. The graph shows that both the variations of SSMF (S-MPR flooding or Simple Flooding) outperformed MAODV and simply SMF. The main reason for this performance is the SSMF broadcasting mechanism, that employs S-MPR or simple limited flooding, both of which are robust for increased network density as they involve no route calculations. On the other hand frequent multicast tree re-construction results in more network congestion and hence decreasing trend in the PDR and increasing trend of control overhead. In Figure 4.20 MAODV has worst delays due to route latency as result of delay in construction of its multicast delivery trees, whereas, SMF has almost constant delays since it uses MPR flooding purely which does not depend on network size. SSMF on the other hand has almost constant to linear increase as the limited simple or MPR-flooding approaches have no route latencies with OLSR always having updated routes available due to its pro-active nature.

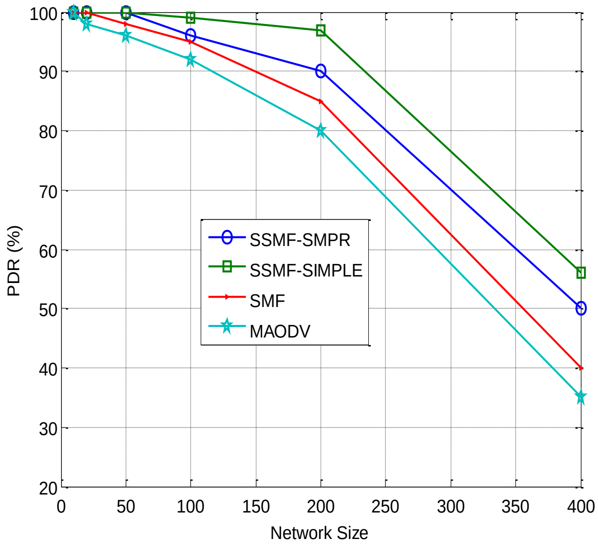


Figure 4.18: Packet delivery ratio as a function of network size

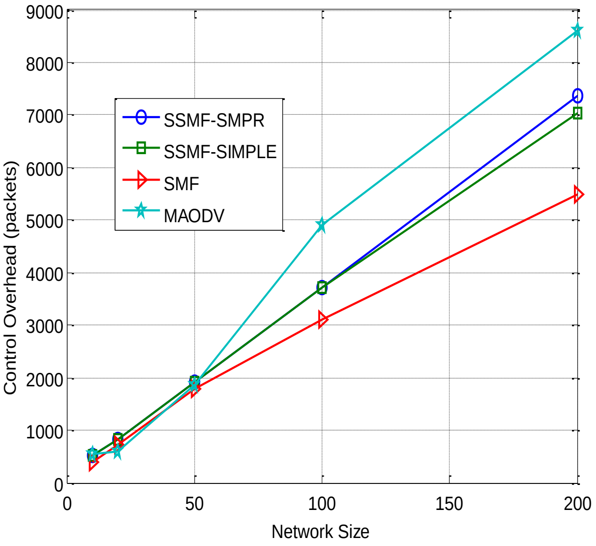


Figure 4.19: Control overhead as a function of network size

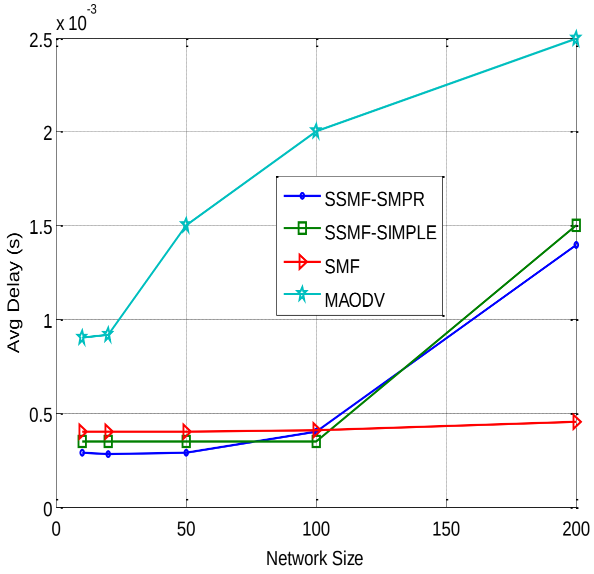


Figure 4.20: Average end-end delay as a function of network size

4.6.8 Multicast Group Size Simulation Results

The multicast group size or number of receivers in the topology is varied for these set of simulations. The graphs for different performance evaluation metrics are shown in Figures 4.21, 4.22 and 4.23.

In Figure 4.21, the PDR of SSMF outperforms both pure SMF and MAODV. The increase in the group size results in a larger number of links, which results in more congestion due to an increase in the number of broadcasts. An increase in the group size also results in a higher latency due to larger routing tables or multicast trees. Since proactive OLSR always has the updated routing tables the decrease in PDR of SSMF is lower when compared with MAODV which depends on a reactive AODV protocol for updated routes. Figure 4.23 shows that as group size increases, the increase in latency during SMF is less when compared to the latency of both MAODV and SSMF, as, the time required to reach all the group members increase with the number of members or larger multicast tree computation in MAODV and increase in the OLSR routing tables in case of SSMF. Similarly, the control overhead seen in Figure 4.22, increases with the increase in group size as more and more control packets are generated by multicast group members to maintain multicast group. In case of MAODV, more GRPH and MACT (described in Section 2.5.5) messages are generated along the paths in the multicast tree to maintain multicast group, and in case of SSMF and SMF, more HELLO and TC/SSMF update messages are generated and broadcasted using MPR flooding.

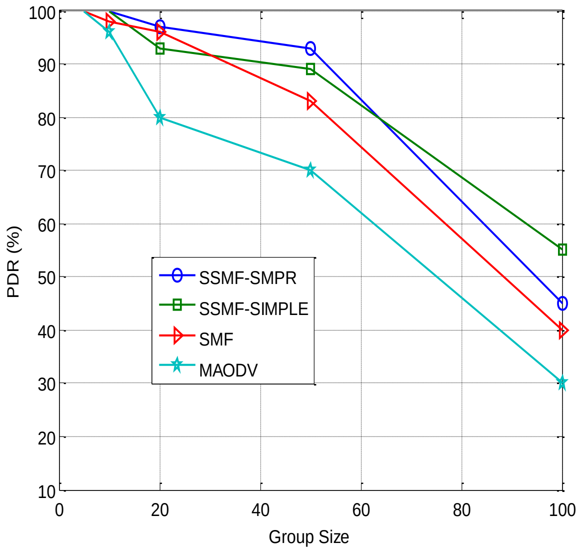


Figure 4.21: Packet delivery ratio as a function of group size

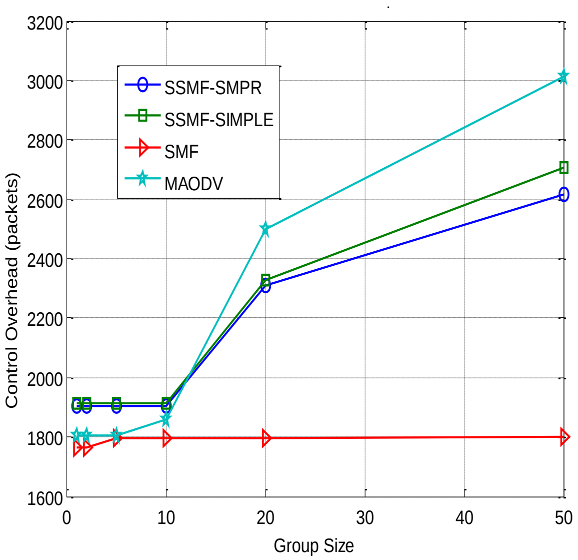


Figure 4.22: Control overhead as a function of group size

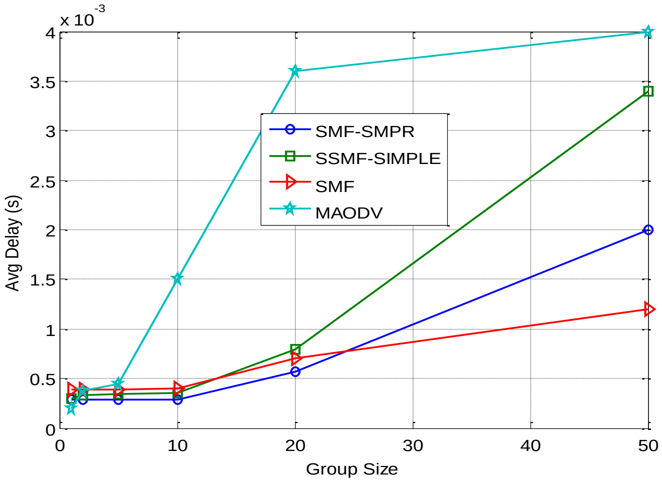


Figure 4.23: Average end-end delay as a function of group size

4.6.9 Number of Senders Simulation Results

For this set of simulations the number of multicast senders in the topology is varied. The graphs for different performance evaluation metrics are shown in Figures 4.24, 4.25 and 4.26.

In Figure 4.24, the PDR SSMF and its variants outperform both SMF and MAODV, proving that SSMF is scalable with respect to number of senders. An increase in number of senders also results in an increase in control overhead as shown in Figure 4.25. Both MAODV and SSMF will have to handle the control packets to maintain multicast trees and OLSR routing tables respectively from multiple multicast senders. MAODV has worst overhead when compared with SSMF. However overhead of SMF is almost constant because it need not maintain any multicast topology trees or tables. Similarly, the delays shown in Figure 4.26, increase with increase in number of senders, which causes increase in the congestion, and hence increased route latency.

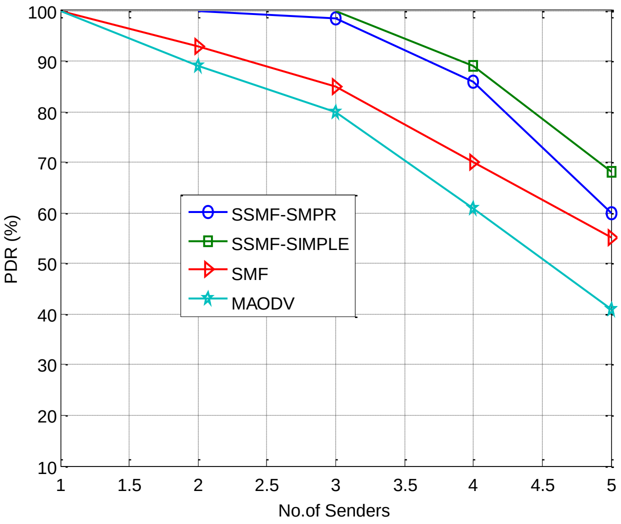


Figure 4.24: Packet delivery ratio as a function of number of senders

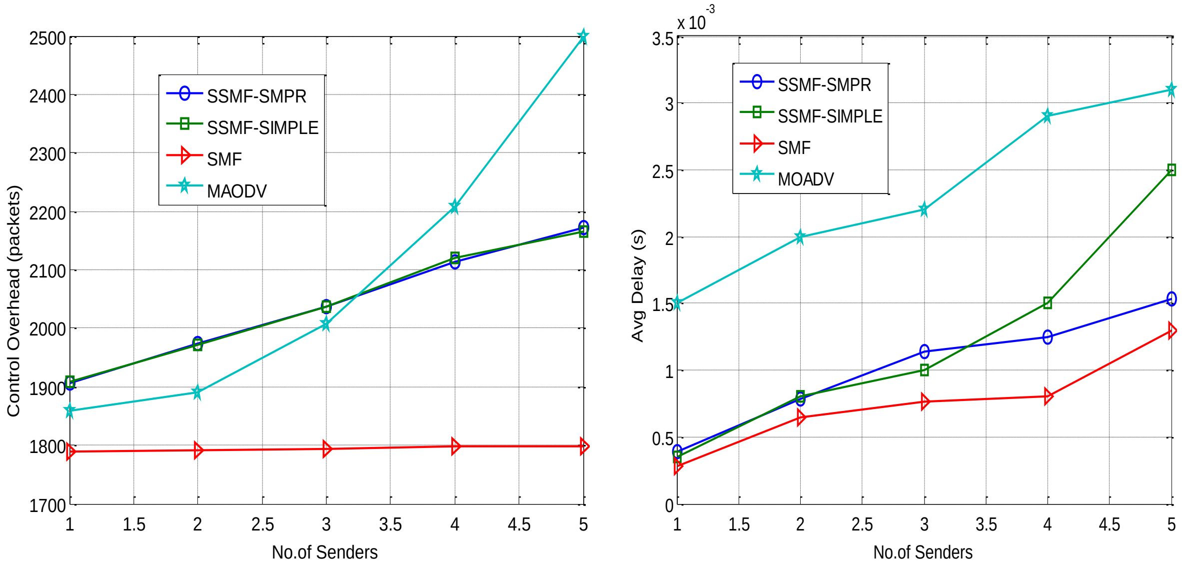


Figure 4.25: Control overhead as a function of number of senders

Figure 4.26: Average end-end delay as a function of number of senders

4.6.10 Rate of Leaves and Joins Simulation Results

Figure 4.27 depicts the PDR as a function of the rate of joins and leaves from the multicast group. In all the three protocols (SSMF, SMF and MAODV), as the multicast group members join and leave at periodic intervals at increasing rate, the routes become less and less stable. The protocols cannot keep up forwarding tree maintenance, link update, and membership management etc., resulting in a decreased PDR and increase in the control overhead and average end-end latency as shown in Figures 4.28 and 4.29 respectively. MAODV relies greatly on shared multicast delivery tree and hence has the worst performance measures compared with both SSMF and SMF. Multicast delivery trees become highly unstable if joining/leaving frequency increases, as frequent tree re-construction (route discovery and establishment) introduces lot of latency and hence the greater decrease in PDR and increase in the control overhead. In case of SSMF and SMF, since the broadcasting mechanism does not depend on construction of trees and mostly uses flooding, both SSMF and SMF fare better than MAODV. SSMF however, also uses OLSR for its broadcasting mechanism, the delays and overhead are slightly higher and PDR lower than SMF, as construction of routing tables of OLSR introduces some latency and overhead control packets.

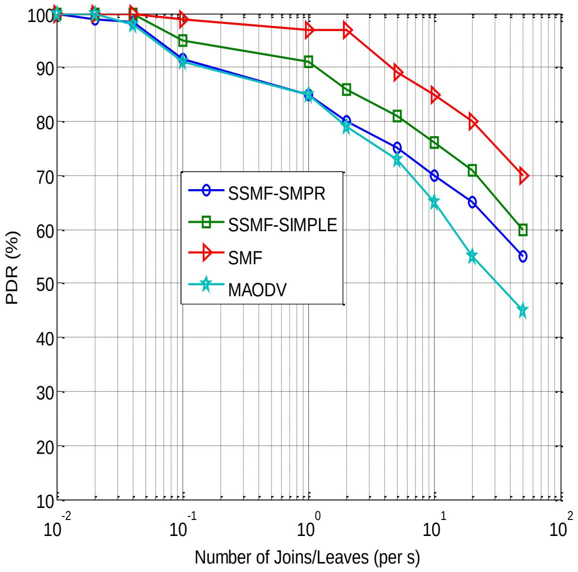


Figure 4.27: PDR as a function of rate of Joins/Leaves

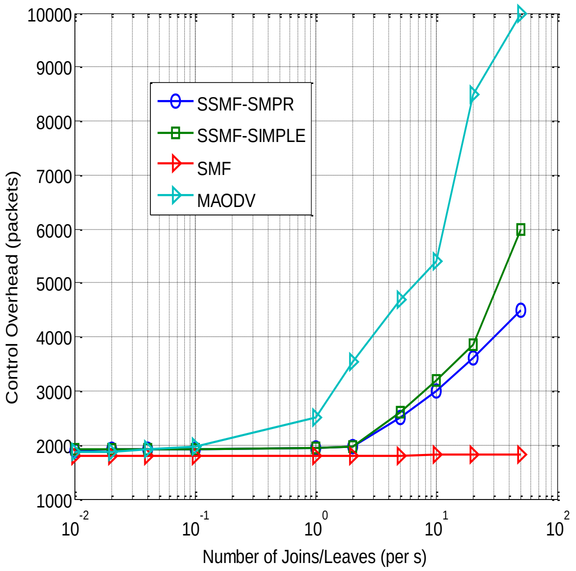


Figure 4.28: Control overhead as a rate of Joins/Leaves

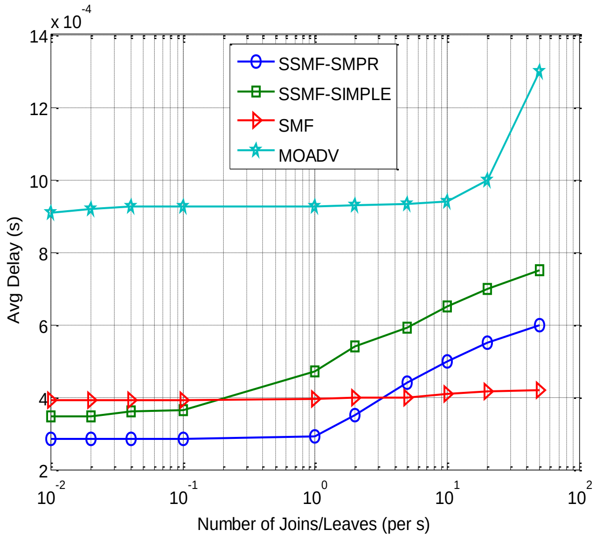


Figure 4.29: Average end-end delay as a function of rate of Joins/Leaves.

CHAPTER 5

Conclusion and Future Work

In this thesis we presented several existing multicast protocols for MANETs and then motivated the need for a new multicast protocol we call SSMF. We showed that SSMF was robust at high offered loads, mobility, network size and group size. The scalability of SSMF is good in multiple multicast sender cases.

SSMF variant of simple flooding proved much more resilient than SSMF variant of SMF (S-MPR) flooding, especially during high mobility, offered load, network and group size, number senders. Hence other than S-MPR, simple flooding provides a good option during high stress conditions on networks, even if it results in greater delays and more control overhead when compared with S-MPR.

SSMF is a very simple protocol with very little state maintenance. However, SSMF's performance depends on the underlying MANET unicast protocol. For this thesis, OLSR unicast protocol is used, and Sections 4.6.1 and 4.6.2 show how the performance of OLSR depends on its HELLO and TC intervals, which indirectly affected SSMF's performance (PDR, latency and overhead). As future work, other unicast protocols should be considered.

Furthermore, there is a need for a more robust formula for calculating the number of MPR-forwarders and, thus, the overhead packets for calculating optimal TTL for SSMF broadcasting mechanism. In this thesis, the formula in Section 3.3.2 is very simple.

Interoperation of SSMF with other existing widely deployed multicast protocols like PIM is also left as future work.

The flexibility of SSMF, in terms of both unicast and multicast protocols will help towards deploying it in real scenarios.

LIST OF REFERENCES

[1] D.P. Agrawal and Q.A. Zeng, Introduction to wireless and mobile systems, Brooks/Cole (2003).

[2] Luo Junhai and Ye Danxia et al., Research on topology discovery for IPv6 networks, *IEEE, SNPD (2007)*, pp. 804-809.

[3] S. Toumpis, Wireless ad-hoc networks, in: *Vienna Sarnoff Symposium, Telecommunications Research Center,* April 2004. Available from: http://www.eng.ucy.ac.cy/toumpis/publications/sarnoff04.pdf.

[4] S. Paul. “Multicasting on the Internet and its Applications”. *Kluwer Academic Publishers*, ISBN 0-792-38200-5, 1998.

[5] IETF MANET Working Group, http://www.ietf.org.

[6] C.-K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice-Hall, Englewood Clis, NJ, USA, 2002.

[7] C. Perkins, Ad-Hoc Networking, Addison-Wesley, Reading, Mass, USA, 2000.

[8] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". *Introduction to Algorithms* (Second ed.). MIT Press and McGraw-Hill. pp. 595–601.

[9] T. Ballardie, P. Francis, and J. Crowcroft, Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing, Proc. of ACM SIGCOMM ’93, 1993, p. 85.

[10] Gossain, H.; Cordeiro, C.D.M.; Agrawal, D.P., "Multicast: wired to wireless," *Communications Magazine, IEEE* , vol.40, no.6, pp.116-123, Jun 2002.

[11] Osamah S. Badarneh and Michel Kadoch, “Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy,” EURASIP Journal on Wireless Communications and Networking, vol. 2009, Article ID 764047, 42 pages, 2009.

[12] Yogen K. Dalal and Robert M. Metcalfe, “Reverse path forwarding of broadcast packets”. *Commun. ACM* 21, 12 (December 1978), pp. 1040-1048.

[13] T. Maufer, C. Semeria, "Introduction to IP Multicast Routing". draft-ietf-mboned-intro-multicast- 03.txt. July 1997.

[14] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

[15] Waitzman, D., Partridge, C., and S. Deering, "Distance Vector Multicast Routing Protocol",RFC 1075, November 1988.

[16] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[17] E.M. Royer and C.E. Perkins, Multicast operation of the ad-hoc on-demand distance-vector routing protocol, ACM MOBICOM (1999), pp. 207-218 August.

[18] J.J. Garcia-Luna-Aceves and E.L. Madruga, The core-assisted mesh protocol, IEEE JSAC (1999), pp. 1380-1394 August.

[19] K. Chen and K. Nahrstedt, Effective location-guided tree construction algorithms for small group multicast in MANET, Proceedings of the INFOCOM (2002), pp. 1180-1189.

[20] M. Gerla, S.J. Lee, W. Su, On-Demand Multicast Routing Protocol (ODMRP) for Ad-hoc Networks, Internet draft, draft-ietf-manet-odmrp-02.txt, 2000.

[21] C.C. Chiang, M. Gerla, L. Zhang, Forwarding group multicast protocol (FGMP) for multi-hop, Mobile Wireless Networks, AJ. Cluster Comp, Special Issue on Mobile Computing, vol. 1 (2), 1998, pp. 187-196.

[22] E. Bommaiah et al., AMRoute: Ad-hoc Multicast Routing Protocol, Internet draft, August 1998.

[23] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR: multicast core-extraction distributed ad-hoc routing, in: IEEE Wireless Communications and Networking Conference, September 1999, pp. 1313-1317.

[24] L. Ji and M.S. Corson, Differential destination multicast-a MANET multicast routing protocol for small groups, Proc. INFOCOM (2001), pp. 1191-1201.

[25] Xiaojing Xiang, Xin Wang, Yuanyuan Yang, "Stateless Multicasting in Mobile Ad Hoc Networks," IEEE Transactions on Computers, vol. 59, no. 8, pp. 1076-1090, Apr. 2010.

[26] T. Clausen, P. Minet, and C. Perkins, "*Multipoint Relay Flooding for Manets*", Internet-Draft, IETF MANET working group, February 2004.

[27] B.Williams and T. Camp, "Comparison of broadcasting techniques for Mobile Ad Hoc Networks",Mobihoc'03.

[28] J. Macker, SMF Design Team, IETF MANET Working Group. [e-document].Simplified Multicast Forwarding for MANET (draft-ietf-manet-smf-02), 2006[retrieved March 24, 2011]. From: http://tools.ietf.org/html/draft-ietf-manet-smf-02.

[29] Bhattacharyya, S., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, July 2003.

[30] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

[31] P. Jacquet, A. Laouiti, P. Minet, L. Viennot, “Performance of multipoint relaying in ad hoc mobile routing protocols,” Networking, May 2002.

[32] T. Clausen and P. Jacquet. “OLSR RFC3626”, Oct 2003. http://ietf.org/rfc/rfc3626.txt

[33]W. Kiess and M. Mauve. A survey on real-world implementations of mobile ad-hoc networks. Ad Hoc Netw., 5(3):324-339, April 2007.

[34]University of California. ns2 Network Simulator. (http://www.isi.edu.nsnam/ns/), Accessed 2010.

[35] In Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (2008), pp. 1-10. OMNeT++. (http://www.omnetpp.org), Accessed 2010.

[36] C.W.Wu, Y. C. Tay, and C.-K. Toh, “Ad hocMulticast Routing protocol utilizing Increasing id-numberS (AMRIS),” draftietf-manet-amris-spec-00.txt, 2000.

[37] Perkins, C, Belding-Royer, E. Das, S. (July 2003). “*Ad hoc On-Demand Distance Vector (AODV) Routing*”. IETF. RFC 3561. http://tools.ietf.org/html/rfc3561. Retrieved 2011.

[38] Charles E. Perkins, Pravin Bhagwat. 1994. “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”. In *Proceedings of the conference on Communications architectures, protocols and applications* (SIGCOMM '94). ACM, New York, NY, USA, 234-244.

[39] Johnson, D. B., Maltz, D. A., & Broch, J. (2001). “DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks”. http://tools.ietf.org/html/rfc4728. Retrieved 2011.

[40]S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: the incredibles. SIGMOBILE Mob. Comput. Commun. Rev., 9(4):50-61, October 2005.

[41]M. Kropff, T. Krop, M. Hollick, P. S. Mogre, and R. Steinmetz. “A survey on real world and emulation testbeds for mobile ad hoc networks”. Proceedings of the 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TRIDENTCOM 2006, pages 6 pp.+, 2006.

[42] Laurent Dairaine, Guillaume Jourjon, Emmanuel Lochin, and Sebastien Ardon. 2007. IREEL: remote experimentation with real protocols and applications over an emulated network. *SIGCSE Bull.* 39, 2 (June 2007), 92-96.

[43] http://cs.itd.nrl.navy.mil/work/proteantools/mne.php. “MobileNetwork Emulator (MNE)”, Accessed 2011.

[44] Damien Magoni and Jean-Jacques Pansiot. 2001. “Influence of Network Topology on Protocol Simulation”. In *Proceedings of the First International Conference on Networking-Part 1* (ICN '01), Pascal Lorenz (Ed.). Springer-Verlag, London, UK, UK, 762-770.

[45] "Naval Research Lab OLSR (NRLOLSR)." http://pf.itd.nr1.navy.mil. Accessed October, 2010.

[46] Zeng, X. Bagrodia, R. Gerla, M , "GloMoSim: a library for parallel simulation of large-scale wireless networks," *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on* , vol., no., pp.154-161, 26-29 May 1998.

[47] Dijkstra, E W. “A note on two problems in connection with graphs”. Numer. Math .1 (1959) pg. 269-271.

[48] Bellman, R. 1958. On a routing problem. *Quart. Appl. Math.* Pg 87–90.

[49] Francisco J. Ros, *“UM-OLSR”,* http://masimum.inf.um.es/?Software:UM-OLSR.

[50] Royer, E.M. and C.E. Perkins, 1999. “Multicast operation of the ad-hoc on-demand distance vector routing protocol”. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Aug. 15-19, ACM Press, Seattle, Washington,USA,pp:207-218.

[51] David Cavin, Yoav Sasson, and Andre Schiper. 2002. “On the accuracy of MANET simulators”. In *Proceedings of the second ACM international workshop on Principles of mobile computing* (POMC '02). ACM, New York, NY, USA, 38-43.

[52] OPNET Users' Manual, *OPNET Architecture, OV.*415.http://forums.opnet.com. Accessed October, 2010.

[53] E. Nordstrom, P. Gunningberg, C. Rohner, O. Wibling, "*A Comprehensive Comparison of MANET Routing Protocols in Simulation, Emulation and the Real World*", Uppsala University, pp. 1–12, 2006.

[54] Luc Hogie, Pascal Bouvry, and Frederic Guinand. "An Overview of MANETs Simulation". *Electron. Notes Theor. Comput. Sci.* 150, 1 (March 2006), 81-101.

[55] Wang, J., Abolhasan, M., Franklin, D. R. & Safaei, F. (2009). OLSR-R^3: Optimised link state routing with reactive route recovery. Proceedings of the 15th Asia-Pacific Conference on Communication (APCC 2009) (pp. 359-362). USA: IEEE.

[56] F.A. Tobagi and L. Kleinrock, Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution, IEEE Transactions on Communications 23 (1975), no. 12, 1400–1416.

[57] Kamaljit I. Lakhtaria and Prof. Bhaskar N. Patel, "*Comparing Different Gateway Discovery Mechanism for Connectivity of Internet & MANET",* International Journal of Wireless Communication and Simulation Volume 2 Number 1 (2010), pp. 51–63.

[58] E. Bacelli, P.Jacquet, "Flooding Techniques in Mobile Ad Hoc Networks", http://hal.inria.fr/inria-00077040/en/ , 2003.

[59] Standard IEEE 802.1D - Spanning Tree protocol.

[60] H. Lundgren, E. Nordstrom, and C. Tschudin. "Coping with communication gray zones in IEEE 802.11b based ad hoc74 networks. In Proceedings of The Fifth ACM International Workshop On Wireless Mobile Multimedia (WoWMoM)", September 2002.

[61] I. D. Chakeres and E. M. Belding-Royer. "The utility of hello messages for determining link connectivity". In 5th International Symposium on Wireless Personal Multimedia Communications (WPMC), October 2002.

[62] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, "Multicast Optimized Link State Routing," IETF manet, draft-ietf-manet-olsr-molsr-01.txt, 2002.

[63] Ogier, R., "MANET Extension of OSPF Using CDS Flooding", Proceedings of the 62nd IETF, March 2005. http://www3.ietf.org/proceedings/05mar/slides/ospf-5/sld1.htm

[64] C. Adjih, P. Jacquet, L. Viennot. "Computing Connected Dominating Sets with Multipoint Relays", INRIA RR-4597. August 2004.

[65] B. Adamson, J, Dean. ”Simplified Multicast Forwarding (SMF) Update”, Proceedings of the 64th IETF, November 2005. http://www3.ietf.org/proceedings/05nov/slides/manet-6.pdf

[66]L.V.A Qayyum and A. Laouiti, “Multipoint relaying for flooding broadcast messages in mobile wireless networks,” in Proc. HICSS, Big Island, HI, USA, Jan 2002.

APPENDIX

A.Summary of NS-2 Modifications

This section presents a brief description of all files that were modified and created in NS-2.27 for SSMF simulation model. To have a clear understanding of entire SSMF implementation, all files involved are hierarchically categorized on the basis of functionalities and internal relationships, logically following the order laid out in the main body of thesis. The following files have to be imported into the NS distribution and added to the Makefile for compilation. Although most of the files discussed below are originally available in the NRL's OLSR implementation, and the readme file provided by them allows us to make the necessary changes for installation of NRL-OLSR largely. However for SSMF implementation, since some of the NRL-OLSR files are modified and also since some of the NRL-OLSR unmodified files are necessary for its functioning as a whole, we have presented all the important modified and unmodified NRL-OLSR files along with the new files created specifically for SSMF.

Filename	NrlolsrAgent.cc & .h
Location	/nrlolsr/ns
Description	Implements all nrlolsr specific parts. It handles, duplicate detection and forwards packets to routing layer (protolibmanetkernel) for rebroadcasting. It allows different broadcasting schemes like SIMPLE, S-MPR, NS-MPR, ECDS etc. This file was modified to handle the TTL value passed down from TCL script. This particular TTL value is passed down to protolibmanetkernel (routing layer).

Filename	ProtolibManetKernel.cc & .h
Location	/nrlolsr/ns
Description	These files existing in NS-2 NRL-OLSR were modified to define limited

	flooding. Implements broadcasting of both control and UDP packets. The TTL value passed down from the TCL script to NrlolsrAgent and then to ProtolibManetKernel to assign the TTL value to IP header in this file before broadcasting.

Filename	nrlolsr.cc & .h
Location	/nrlolsr/common
Description	These files existing in NS-2 NRL-OLSR were modified to add periodic update messages to already existing framework of control messages like HELLO, TC, HNA etc. These files maintain or update MPR selector lists, routing tables, neighbor tables, two-hop neighbor tables etc. These files were modified to maintain/update the periodic update lists on the receivers and pass them down to MultiQueue files.

Filename	MultiQueue.cc & .h
Location	/nrlolsr/common
Description	These files were created to implement the SSMF broadcasting scheme. They create and update receiver tables on multicast sources with the help of periodic update lists passed down to them by nrlolsr files. These files handle limited flooding by calculating and passing the optimal TTL value to the TCL script which in turn passes it down to routing layer (ProtolibManetKernel) to set the TTL value in IP header of the UDP packets.

Filename	olsr_packet_types.cc & .h
Location	/nrlolsr/common
Description	These files define all the OLSR message types like HELLO, TC, HNA

	etc. These files were modified to handle periodic update packet types. These files define the messages structures and are used to pack and unpack the packet/message fields, i.e. buffer the message fields before broadcasting and extract the message fields from the buffer on receiving broadcast packets.

Filename	nbr_queue.cc & .h
Location	/nrlolsr/common
Description	These files define the queues and lists data structures for the entire NRL-OLSR protocol. They are used to model mprselector lists, duplicate lists, routing table, neighbor list, two-hop neighbor list, periodic update lists, HELLO, TC, HNA message lists.