SWIM: A NEW MULTICAST ROUTING ALGORITHM FOR WIRELESS NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALPER SİNAN AKYÜREK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2011

Approval of the thesis:

SWIM: A NEW MULTICAST ROUTING ALGORITHM FOR WIRELESS NETWORKS

submitted by **ALPER SİNAN AKYÜREK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ________________
Head of Department, **Electrical and Electronics Eng. Dept., METU**

Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu ________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Gözde Bozdağı-Akar
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Cüneyt Bazlamaçcı
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Bülent Tavlı
Electrical and Electronics Engineering Dept., TOBB ETU ________________

M.Sc. Arda Akman
Türk Telekom ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ALPER SİNAN AKYÜREK

Signature :

ABSTRACT

SWIM: A NEW MULTICAST ROUTING ALGORITHM FOR WIRELESS NETWORKS

Akyürek, Alper Sinan
M.Sc., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Elif Uysal-Bıyıkoğlu

July 2011, 66 pages

In this work, a new multicast routing algorithm for wireless networks is presented. The algorithm, called SWIM (Source-initiated WIreless Multicast), is a depth-optimal multicast tree formation algorithm. SWIM is fully distributed and has an average computational complexity of $O(N^2)$. SWIM forms a shared tree from the source(s) to destinations; yet, as a by-product, it creates a multicast mesh structure by maintaining alternative paths at every tree node. This makes SWIM suitable for both ad hoc networks and access networks with multiple gateways. An extension to the main algorithm is presented for the use in dynamic networks with mobility and/or dynamic destination group. Performance of SWIM is studied with simulations and is compared to other algorithms in the literature. Due to depth optimality, SWIM achieves a lower average and maximum delay than the compared algorithms. The throughput performance is found to be high. Working capability with rateless codes are also studied.

Keywords: wireless multicast, minimum depth, greedy set cover, mesh network, number of forwarding nodes, routing, ad hoc networks, access networks, multicast tree

ÖZ

SWIM - KABLOSUZ AĞLARDA YENİ BIR ÇOĞAGÖNDERİM ALGORİTMASI

Akyürek, Alper Sinan
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Elif Uysal-Bıyıkoğlu

Temmuz 2011, 66 sayfa

Bu çalışmada, yeni bir çoğagönderim yol atama algoritması sunulmaktadır. SWIM (Kaynak Tetiklemeli Kablosuz Çoğagönderim) adını verdiğimiz bu algoritma, en küçük derinlikte bir çoğagönderim ağacı oluşturma algoritmasıdır. SWIM tamamen dağıtık olarak çalışır ve $O(N^2)$ ortalama hesaplama karmaşıklığına sahiptir. SWIM, kaynaktan veya kaynaklardan hedeflere ortak bir ağaç oluşturur; bunun yanında, bir çoğagönderim mesh yapısını da, ağacın her düğümünde oluşturduğu alternatif rotalarla kurar. Bu özellik, SWIM'i hem devingen hem de çok ağ geçitli erişim ağları için kullanılabilir kılar. Asıl algoritmaya, hareketli veya değişken hedef grubuna sahip dinamik ağlarda çalışmak üzere bir eklenti sunulmuştur. SWIM'in başarımı simulasyonlarla ölçülmüş ve literatürdeki en başarılı çoğagönderim algoritmalarıyla karşılaştırılmıştır. Optimal derinliğe sahip olmasından dolayı, SWIM, bu algoritmalara göre azami ve ortalama gecikme bakımından üstün başarım göstermektedir. Bununla birlikte akış hızı başarımı da yüksek düzeydedir. Oransız kod ile çalışması da incelenmiştir.

Anahtar Kelimeler: kablosuz çoğagönderim, minimum derinlik, küme kapsama, örgü ağları, ileten düğüm sayısı, yol atama, devingen ağlar, erişim ağları, çoğagönderim ağacı

To My Beloved Family and Edi

ACKNOWLEDGMENTS

First, I would like to thank Prof. Elif Uysal-Bıyıkoğlu, my advisor, mentor, inspiration and role model, for supporting, encouraging, helping, teaching and advising for more than 5 years, despite of all my sudden disappearances. Not only this work couldn't have finished but also I couldn't be where and who I am, without her.

I would like to thank Prof. Gözde Bozdağı-Akar, Prof. Cüneyt Bazlamaçcı, Prof. Bülent Tavlı and Mr. Arda Akar for showing the kindness of joining my thesis committee and their useful advices and comments on this work.

I would like thank my dear beloved family for supporting and loving me for 24 years, 9 months and 15 days no matter what happens. I owe the good thing called life and all the good things in life to them.

I would like to thank Özge, for all her help, encouragement, never ending support, strengthening and making me smile (sometimes crazy :)) in the darkest and hardest times.

I would to thank Edi, for being with me, since my childhood, no matter what.

I would like to thank, starting from Mr. Alper Gerçek, Mr. B.G., Tolga, Onur, Sinem and Erman, all my co-workers for all their moral support.

My greatest thanks goes to all the Reviewer 2's out there.

I would like to thank the Dallas Mavericks for winning their first championship, especially in this year.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

SWIM	Source Initiated Wireless Multicast
NFN	Number of Forwarding Nodes
WMA	Wireless Multicast Advantage
WMT	Wireless Multicast Tree
NP-C	Non-deterministic Polynomial-time Complete
MAC	Medium Access Control
BIP	Broadcast Incremental Power
MST	Minimum Spanning Tree
BLU	Broadcast Least Unicast
BliMST	Broadcast Link-based MST
EWMA	Embedding Wireless Multicast Advantage
MAODV	Multicast Ad Hoc On Demand Distance Vector
WMN	Wireless Mesh Network
ODMRP	On Demand Multicast Routing Protocol
ARQ	Automatic Repeat Request
WiFi	Wireless Fidelity
UDP	User Datagram Protocol
ns3	Network Simulator 3
Mbps	Mega Bits per Seconds
Kbps	Kilo Bits per Seconds
ETX	Expected Number of Transmissions

CHAPTER 1

THE WIRELESS MULTICAST PROBLEM

This chapter is intended to be an introduction to the topic of this work and is outlined as follows: In Section 1.1 an introduction to the main topic and area of this work is given. Then in Section 1.2 the main underlying problem is formally defined. Finally in Section 1.3 the purpose of this work and a general outline of the thesis is given.

While the work on this thesis was continuing, the solutions and partial results were published in a journal [3] and two different conferences [1][2].

1.1 Introduction

In multi-hop wireless networks, traffic sessions from a source node, gateway or a set of gateways to a group of destinations often occur. Such sessions are used on different types of networks like Ad Hoc Networks or Mesh Networks. In the case of near-real-time applications, these sessions may be delay sensitive. Rather than sending the data by using multiple unicast sessions to each destination, it is desirable to avoid burdening the network with unnecessary retransmissions, to construct a multicast route.

Usage of a multicast route helps to avoid bottlenecks at multicast source nodes. A good multicast route will make efficient use of the bandwidth constrained or energy constrained links. Also, it should try to maximize the energy efficiency by decreasing the number of forwarding of a packet. This is achieved by trying to keep the number of forwarding nodes (NFN) as low as possible. Finding a tree with minimum number of forwarding nodes requires the solution of the Wireless Multicast Tree problem. The formal definition of the problem is done in Section 1.2.

Another important point, especially for near-real-time applications is to keep the delay low. To achieve a low delay, it is of interest to make the distance of each destination node to a source small. By doing so, the forwarding delay induced by routing would be kept low.

1.2 Wireless Multicast Tree Problem

Despite the disadvantages of the wireless medium, like high attenuation or fading channel characteristics, it has a natural broadcast characteristic.

Any transmission from a wireless source is propagated to every direction and can be received from multiple destinations simultaneously (the propagation delay difference is ignored). With this property, a different routing solution can be created when compared to the wired case, using fewer retransmissions. This property is defined as the Wireless Multicast Advantage (WMA) and it is shown as an example in Figure 1.1.

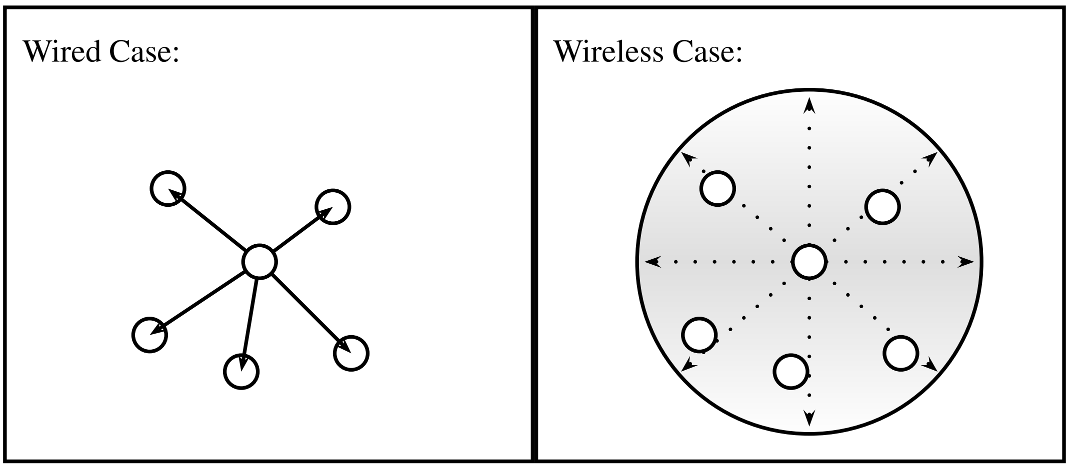


Figure 1.1: Wireless Multicast Advantage Example. In the wired case, the packet needs to be retransmitted to each neighbor, while in the wireless case, a single transmission will reach out to every node in the transmission range.

With the usability of the WMA, the Wireless Multicast Tree (WMT) problem is defined as finding a tree with minimum number of transmissions.

Due to the WMA, the solution tree needs to cover the destination nodes and not necessarily include them.

1.2.1 Steiner Tree Problem

A very similar problem for the wired case is known as the Steiner Tree Problem. The definition is given in Definition 1.2.1.

Definition 1.2.1 (Steiner Tree Problem) *Given a network with nodes V, edges e and a node set S such that $S \in V$, find the minimum set of nodes that connects S.*

A sample network graph and the steiner tree solution is given in Figure 1.2.

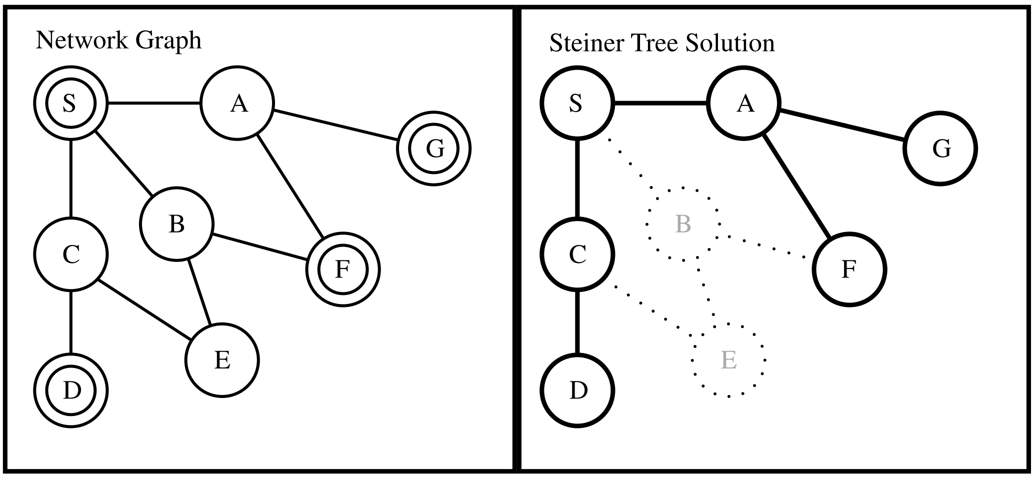


Figure 1.2: Steiner Tree Problem Example. On the left side, an example network graph is given where S is the source and D, F and G are the destinations. The minimum tree connecting the set $\{S, D, F, G\}$, thus the Steiner Tree Problem solution is given on the right side. The solution is the set $\{S, A, C, D, F, G\}$.

The WMT problem is different from the Steiner Tree Problem in the sense that the leaf nodes, the nodes that are farthest away from the source node, are not included in the solution tree. This is due to the WMA, since any broadcast from the neighbor of the leaf node will be sufficient. An example instance of the WMT problem is given in Figure 1.3.

1.2.2 NP-Completeness of WMT Problem

After the WMT Problem has been defined, complexity of the solution needs to be understood. Before making an analysis, the definitions of two complexity related definitions need to be made:

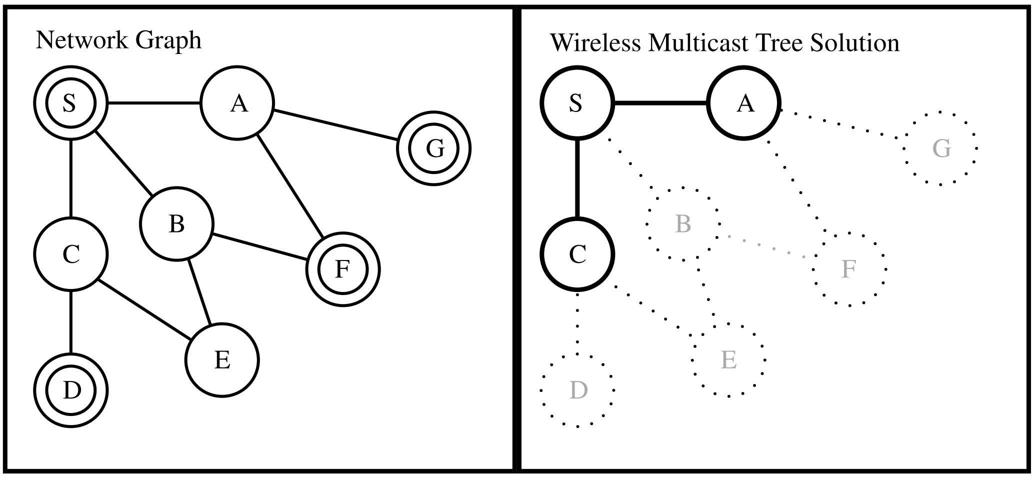


Figure 1.3: Wireless Multicast Tree Problem Example. On the left side, the same example network graph on Figure 1.2 is given where S is the source and D, F and G are the destinations. The minimum set of connected nodes that *cover* the destination set is $\{S, A, C\}$, thus the Wireless Multicast Tree Problem solution is given on the right side. The solution is different from the Steiner Tree Problem, since the broadcast of the solution set is sufficient to reach all the destinations.

1. Class P : Deterministic Polynomial-Time (Definition 1.2.2)

2. Class $NP - C$: Nondeterministic Polynomial-Time Complete (Definition 1.2.3) (Definition 1.2.3)

Definition 1.2.2 (Class P Decision Problem) *Given a decision problem C, C is in P class if it can be solved in deterministic polynomial time.*

Definition 1.2.3 (Class NP-C Decision Problem) *Given a decision problem C, C is in $NP - C$ class if it can be transformed into another $NP - C$ class problem in polynomial time and a given solution can be verified in deterministic polynomial time.*

The Steiner Tree Problem is proved to be a Nondeterministic Polynomial Time (NP) Complete problem [15]. From the Definition 1.2.3, if any NP-Complete problem is transformed into another problem in polynomial time, then the resultant problem is also NP-Complete as long as $NP \neq P$.

The Steiner Tree Problem can be transformed into an instance of the WMT Problem by simply adding dummy nodes as neighbors to each leaf node. The WMT solution to the resulting network is the Steiner Tree of the original network. This concept is shown as an example on Figure 1.4.

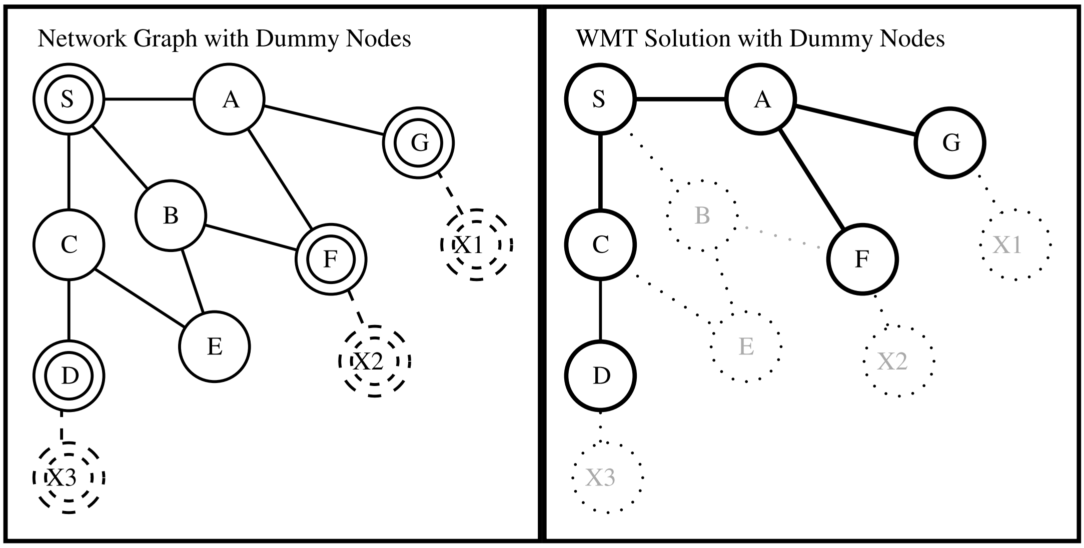


Figure 1.4: WMT - Steiner Relationship Example. On the left side, the same example network graph on Figure 1.2 with dummy nodes attached to the destination nodes that are also leaf nodes, is given. The dummy nodes are included in the destination set. The minimum set of connected nodes that *cover* the destination set is $\{S, A, C, D, F, G\}$, thus the Wireless Multicast Tree Problem solution is given on the right side. The solution is exactly the same for the Steiner Tree Problem without dummy nodes. A linear time transformation is possible.

This linear transformation from the NP-Complete Steiner Tree Problem to the WMT problem shows that the WMT problem is also NP-Complete. Exact solutions of NP-Complete problems are not feasible since they need excessive processing power and memory. In order to find a feasible (deterministic polynomial time) solution, there is a need for a good heuristic.

1.3 Goals of this Thesis

The purpose of this thesis is to develop an algorithm that keeps the distance of each destination node from a source to a minimum, while keeping the NFN as low as possible for different kinds of networks and traffics. Different solutions will be given for different variations of this main problem.

This thesis consists of 7 chapters. In Chapter 2, a review of the literature and descriptions of previous solutions about the stated problem will be given. After that, Chapter 3 explains the basic wireless multicast solution. Chapter 4 extends the basic solution to the case of mesh networks. Then, Chapter 5 extends the solution to dynamic network conditions. After that, Chapter 6 contains various analyses and extensive simulations of the solutions. Finally, Chapter 7 discusses the effect of rateless codes and gives further directions. Chapter 8 concludes this thesis.

CHAPTER 2

LITERATURE REVIEW

There is a broad literature on multicast routing algorithms. The interest in wireless multicast has risen rapidly in the last decade [9].

A number of multicast routing algorithms have been developed, focusing on different priorities such as low latency, energy efficiency and so on [13].

But the literature can be grouped into two groups depending on the area of the solution provided which is also the outline of this chapter.

Section 2.1 explains the tree based solutions and Section 2.2 explains the mesh based solutions and the reason for the need of mesh based solutions, given in the literature.

2.1 Tree Based Solutions

Tree based solutions try to find a solution in the form of a tree. There are a variety of different objectives like latency, throughput or energy-efficiency.

It is desirable, primarily to avoid burdening the network with unnecessary transmissions, to construct a multicast route [7]. An algorithm proposed in [7] is based on the merging of the unicast shortest paths.

A notable multicast tree formation algorithm, for both weighted and non-weighted graphs, appear in [22]. Two different solutions are presented. The first solution, a centralized one, starting from the source node selects the node that minimizes the remaining total distance to the destinations. The second solution, a distributed algorithm is based on merging optimum unicast routes and pruning the resulting subgraph.

A different approach is presented in [33]: the objective is to select the minimum number of nodes in the network that are "on" (and keep others turned "off"), while keeping a communication path from the source to the destinations, by utilizing information about the geographic position of the nodes in the network. A minimum spanning tree is calculated on the final state to further reduce the number of "on" nodes.

Another notable multicast routing heuristic presented in [27] relies on clustering and a certain medium access control (MAC) protocol [26]. The solution is a cross layer solution that combines the MAC layer with the Network layer. The solution starts with a flooding, broadcast by all nodes, then prunes the redundant branches by feedback from the leaf nodes. Also a repair and maintenance algorithm is provided.

There is a richer literature on wireless *broadcast*, which is a special case of the multicast problem. Much of the recent work on broadcast has considered energy efficiency and power control [33].

In [31] three power efficiency based broadcast algorithms are proposed. The algorithms are based on power control. They also try to use the Wireless Multicast Advantage. The first algorithm, Broadcast Incremental Power(BIP) is very similar to Prim's Minimum Spanning Tree (MST) algorithm. Starting from the source, the node that needs the minimum additive power is selected. Although the algorithm seems centralized, a distributed version is also proposed in [32]. The second algorithm is Broadcast Least-Unicast cost (BLU) is a very simple algorithm that is the combination of the best unicast routes. In addition to the combination, the algorithm utilizes the WMA if possible. The last algorithm proposed is Broadcast Link-base MST (BliMST) is the regular MST algorithm. For all three algorithms, an extension is proposed, called the *sweep operation*. It searches for a node, whether the destinations that node is reaching, are already reached by other nodes or combination of nodes. If such a node is present, that node is pruned. Multicast versions of the algorithms are also proposed. The idea is simple, the nodes not in the multicast group and the links used to reach them are pruned from the broadcast solutions.

Another algorithm, called Iterative Maximum-Branch Minimization, proposed in [18], constructs an iterative mechanism for reducing power in a source-initiated wireless broadcast tree.

It starts by adjusting the power of the Source Node to reach all nodes in the network. Then it replaces the maximum length branch with two consecutive transmissions using a forwarding node. Then, the algorithm checks whether another node can be reached through the new forwarding node. If present, those nodes are connected through the new forwarding node. This iteration continues until the power can not be minimized further.

Another iterative algorithm is given in [21]. The algorithm uses integer programming with an energy-efficiency objective.

In [10], the NP-hardness of the minimum energy broadcast problem in metric space was proved. An algorithm is proposed for the case of minimum energy broadcast with power level constraints. The algorithm simple tries all possibilities by brute force.

Later in [28], power-optimal broadcast was proved to be NP-complete under more general conditions. An algorithm, called Embedding Wireless Multicast Advantage (EWMA) is proposed. The algorithm is the MST solution which uses the WMA if possible. The costs for MST are the powers required to reach a given neighbor node.

More recently, an algorithm specifically developed for voice multicasting was proposed in [29]. The algorithm tries to decrease the number of forwarding nodes and claims that this procedure decreases the delay of voice.

One of the most popular tree based algorithms is Multicast ad hoc on demand distance vector (MAODV) given in [24], also stated in [9]. MAODV forms a shared tree, which is a tree connecting the source(s) with the multicast destinations, without explicitly optimizing tree depth by taking advantage of the WMA. Maintenance and repair algorithms have been given. The popularity comes from the fact that an implementation is also given.

2.2 Mesh Based Solutions

Mesh based solutions are solutions based on fault tolerance, multiple source(gateway) usage and wireless access networks to access larger (probably wired network clouds) networks. The multiple source scenario is important with the emergence of Wireless Mesh Networks(WMN) as access networks for widespread wireless networking, with all the self-organization, self-configuration and self-healing properties of this architecture [5, 4].

In the mesh network scenario, outside access is provided by several gateway nodes in the network. Clearly, it may be advantageous for different network nodes to be accessing different gateways depending on their respective proximity to these gateways. In [4], this property of the WMNs is considered. There is a wireless network with gateway nodes, connected through a wired infrastructure. The algorithm tries to select the best gateway and proposes a scheme for seamless handover between gateway regions.

The WMN structure promotes reliable, scalable and cost-effective network deployment under diverse environments and can serve stationary and mobile users alive, in city-wide broadband WiFi networks, private business networks, rural access networks and so on [12].

In WMNs, the gateway tends to be the bottleneck [30]. In [6], an algorithm is proposed for placing the gateways in order to relive the bottlenecks and increase utilization.

Another property of WMNs is the fault tolerance and redundancy. This need is especially important in the cases of high mobility [16] and in multimedia streaming [8].

Another routing algorithm for WMNs was proposed in [34]. The algorithm uses block addressing to route the packets. First an algorithm to partition the network into address spaces is given. Through block addressing, the routing is simplified. If a packet is addressed to a destination node, whose address in on the address space of the node, the packet is forwarded through the branch on which the node lies. If the address is outside the address space, the packet is forwarded to the parent node. Secondly this addressing scheme is combined with a link state routing metric. The reason of selection of a link state routing algorithm is to use a low amount of memory, since the WMNs may scale to very large sizes.

A popular mesh based algorithm is the On-demand multicast routing protocol (ODMRP) [17]. ODMRP forms a mesh where the redundant routes are not pruned, but kept for reliable transmission in case of a link failure. The algorithm uses a scoped flooding. The current route, among the redundant routes, is selected according to the minimum delay.

CHAPTER 3

SWIM - TREE SOLUTION

As stated in Chapter 1, the main aim of this work is to develop a heuristic solution to the WMT problem which will result in low delay while still maintaining energy efficiency. This chapter explains the working principle of the heuristic solution in detail and is outlined as follows: Section 3.1 explains the steps to the solution and Section 3.2 explains the frequently used definitions in the solution. Then Section 3.3 explains the first of the two phases of the solution and Section 3.4 explains the second phase of the solution.

3.1 Solution Approach

The WMT problem is defined for static networks. A good tree heuristic solution to the WMT problem needs to be found as stated earlier. Rather than adding realistic conditions like mobility or packet loss intolerance to the problem at the beginning, starting from the basic situation is desirable. The realistic conditions will be added in the next chapters.

The steps to the solution contain two points:

1. The delay should be minimized by keeping the depth as small as possible.
2. The WMA should be used in order to increase energy efficiency by keeping the NFN small.

Using the simplifications and the solution step ideas, a wireless multicast routing algorithm called SWIM is developed. SWIM is the abbreviation for Source initiated WIreless Multicast.

SWIM basically partitions the network into levels, by the use of the hop distance of each node to the source node. It allows the forwarding of packets only between neighbor levels. This forces the depth of any destination node to be minimum (which will be proved later).

After leveling the network, SWIM tries to find maximum sized covers to increase energy efficiency and use the WMA effectively. The whole operation is distributed and has a low complexity.

SWIM consists of two major phases:

1. Signaling Phase: The network is divided into levels.
2. Covering Phase: Starting from the source node, a coverage distribution is done.

3.2 Definitions

Before advancing to the explanation of the solution, this section explains the frequently used terms and terminologies.

3.2.1 Leveling Terms

Since SWIM uses a leveling procedure to construct a solution tree, a necessity to define node relationships in terms of levels arises.

The distance metric between two nodes is defined in number of hops and shown in Definition 3.2.1.

Definition 3.2.1 (Hop Distance) *Let A and B be two nodes. Then the distance is defined as:* $d(A, B) = NumberOfHopsBetween(A, B)$

If the distance of a node's neighbor to the source node is greater than the node's distance to the source node, then that neighbor is defined as the *Child* of that node. Using the distance in Definition 3.2.1 a *Child Neighbor* is defined in Definition 3.2.2.

Definition 3.2.2 (Child Neighbor) *Let A be a node, B be its neighbor and S is the source. Then:*

if $d(A, S) < d(B, S)$ *then*

B is a Child *of A*.

If the distance of node's neighbor to the source node is equal to the node's distance to the source node, then that neighbor is defined as the *Sibling* of that node. Using the distance in Definition 3.2.1 a *Sibling Neighbor* is defined in Definition 3.2.3.

Definition 3.2.3 (Sibling Neighbor) *Let A be a node, B be its neighbor and S is the source. Then:*

if $d(A, S) = d(B, S)$ *then*

B is a Sibling *of A*.

If the distance of node's neighbor to the source node is smaller than the node's distance to the source node, then that neighbor is defined as the *Parent* of that node. Using the distance in Definition 3.2.1 a *Parent Neighbor* is defined in Definition 3.2.4.

Definition 3.2.4 (Parent Neighbor) *Let A be a node, B be its neighbor and S is the source. Then:*

if $d(A, S) > d(B, S)$ *then*

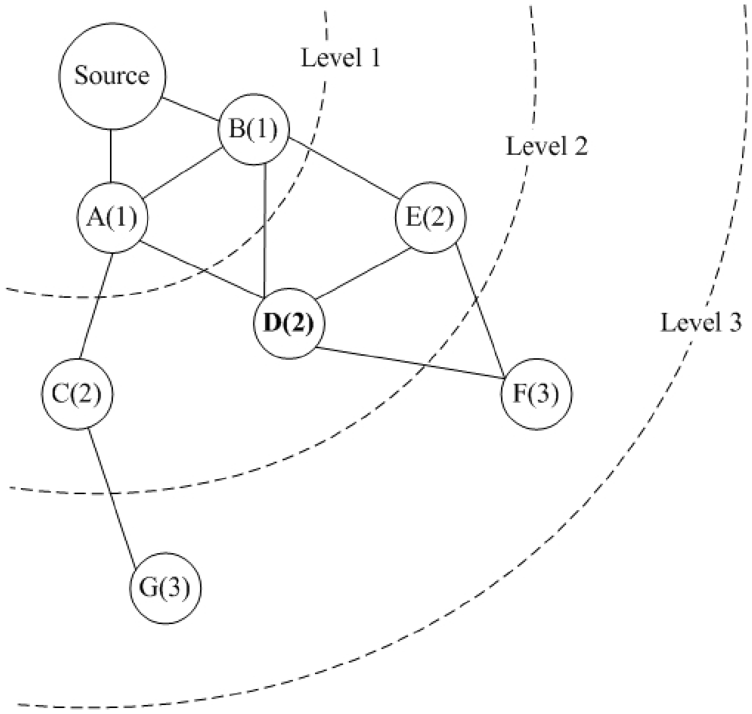
B is a Parent *of A*.

To summarize the definitions, an example is given in Figure 3.1.

Since these definitions will be commonly used throughout this work, an abbreviation table is given in Table 3.1 to simplify reading.

Table 3.1: Abbreviations of the Neighbor Classifications commonly used in this work.

Local Definitions (Scope is a particular node)
$\mathbb{P}$: Parent Node Set
$\mathbb{S}$: Sibling Node Set
$\mathbb{C}$: Child Node Set
System Definitions (Scope is the whole network)
$\mathbb{P}_i$: Parent Node Set of Node i
$\mathbb{S}_i$: Sibling Node Set of Node i
$\mathbb{C}_i$: Child Node Set of Node i



Distance of Node D to Source is 2

Neighbor Name	Distance to Source	Comparison	Classification
A	1	1 < 2	*Parent*
B	1	1 < 2	*Parent*
E	2	2 = 2	*Sibling*
F	3	3 > 2	*Child*

Figure 3.1: Leveling Definitions Example. For node *D*, *A* and *B* are *Parent* nodes, *E* is a *Sibling* and *F* is a *Child*.

3.3 Leveling Phase

As stated in Section 3.1, the first phase of SWIM is to partition the network into levels of hop distances to the source. To achieve this, a leveling algorithm has been developed. The principle is as follows:

The source node starts to transmit a broadcast message, saying that its hop distance to itself is 0. When a neighbor receives the message, it increments the counter by 1 and rebroadcasts it, saying that its hop distance to the source node is 1. Every node in the network continues until the information in each node is consistent with its neighbors, thus have come to a steady state. This means that if a node's distance is D then each neighbor of that node should have transmitted $D - 1,\ D\ or\ D + 1$ as its distance. Any other value is not possible in steady state.

This is proven under Lemma 3.3.1.

Lemma 3.3.1 *Given node A, its neighbor B and source node S:*
if $d(A, S) = D$ *then*
$d(B, S) = (D - 1),\ (D)$ *or* $(D + 1)$ *and no value else.*

Proof. The proof comes from contradiction:
Let $d(B, S) = (D - 1 - \epsilon)$, where $\epsilon > 0$.

Since A and B are neighbors, A can reach the source through B, thus it will have a minimum distance of:
$\min(((D - 1 - \epsilon) + 1, (D)) = (D - \epsilon) < (D)$

This is contradicting with the initial assumption of $d(A, S) = (D)$.
For the other case, let $d(B, S) = (D + 1 + \epsilon)$, where $\epsilon > 0$.

Since A and B are neighbors, B can reach the source through A, thus it will have a minimum distance of:
$\min((D) + 1, (D + 1 + \epsilon)) = (D + 1) \neq (D + 1 + \epsilon)$

This is contradicting with the initial assumption of $d(B, S) = (D + 1)$. ∎

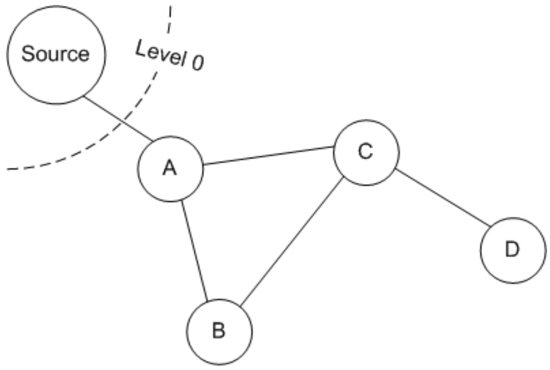
When the system reaches a steady state, all nodes in the network will know:

1. Own hop distance to the source node

2. Hop distance of each neighbor to the source node

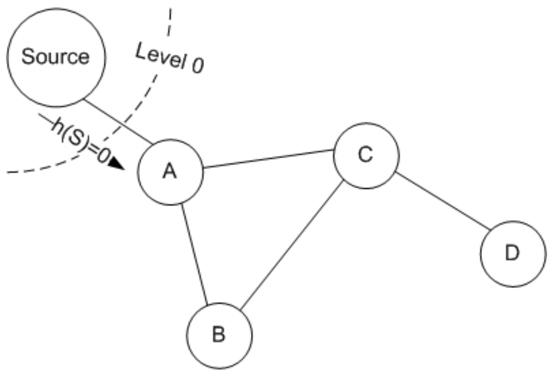
Using this information, each node can partition its neighbors into groups of *Child*, *Sibling* and *Parent* nodes, thus knowing its own level and the levels of its neighbors, which was the main aim of the first phase.

The first phase is explained in Figures 3.2, 3.3, 3.4, 3.5 and 3.6 with an example network.



	Source	A	B	C	D
Source	-	-	-	-	-
A	-	-	-	-	-
B	-	-	-	-	-
C	-	-	-	-	-
D	-	-	-	-	-

Figure 3.2: Initial state of the network. There is no information at the beginning.

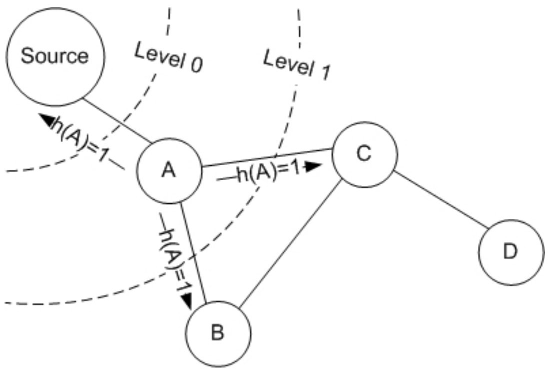


	Source	A	B	C	D
Source	**0**	-	-	-	-
A	0	-	-	-	-
B	-	-	-	-	-
C	-	-	-	-	-
D	-	-	-	-	-

Figure 3.3: Source starts the process by sending its distance as a broadcast message.

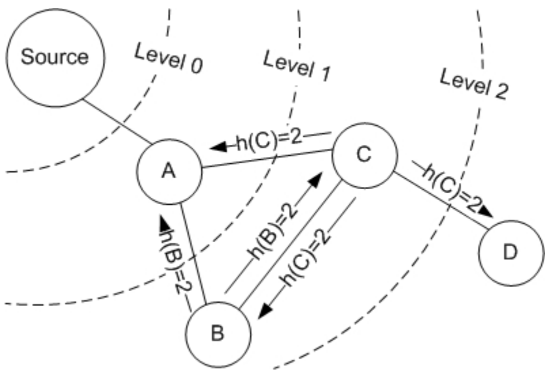
3.4 Covering Phase

In Section 3.3 the network has been divided virtually into levels using the hop distance information to the source. In the Covering Phase, the solution tree will be constructed using the level information.



	Source	A	B	C	D
Source	**0**	1	-	-	-
A	0	**1**	-	-	-
B	-	1	-	-	-
C	-	1	-	-	-
D	-	-	-	-	-

Figure 3.4: Node A sets its distance to 1 and broadcasts this information.

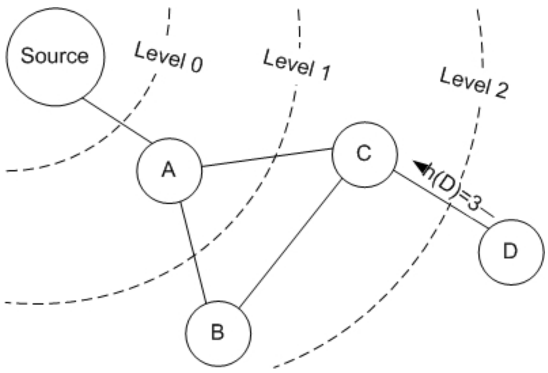


	Source	A	B	C	D
Source	**0**	1	-	-	-
A	0	**1**	2	2	-
B	-	1	**2**	2	-
C	-	1	2	**2**	-
D	-	-	2	2	-

Figure 3.5: Node B and C set their distances to 2 and broadcast this information.

The main idea is to select a neighbor that covers most of the nodes in the destination set.

SWIM only allows the communication of a level with another level that is farther to the source node. This guarantees that every destination will have a minimum distance to the source node, which is proved in Lemma 3.4.2 which is based on Lemma 3.4.1.



	Source	A	B	C	D
Source	**0**	1	-	-	-
A	0	**1**	2	2	-
B	-	1	**2**	2	-
C	-	1	2	**2**	3
D	-	-	2	2	**3**

	ℂ	𝕊	ℙ
Source	A	-	-
A	B, C	-	Source
B	-	C	A
C	D	B	A
D	-	-	C

Figure 3.6: Node D sets its distance to 3 and broadcasts this information.

Lemma 3.4.1 *Given Nodes S, A and B, let the shortest distance between S and A, $d(S,A) = D$, between S and B, $d(S,B) = D+1$ and between A and B, $d(A,B) = 1$. Then, Node A is on one of the shortest paths between S and B. (shown on Figure 3.7)*

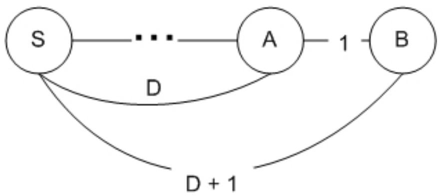


Figure 3.7: Illustration of Lemma 3.4.1. Node A is on the shortest path from Node S to Node B

Proof. Since $d(S,B) = D+1$, any next step on the shortest path must decrease the distance by one. Since Node A is a next step ($d(A,B) = 1$) and decreases the distance by one ($d(S,A) = D$), it is on the shortest path. ■

Lemma 3.4.2 *If a packet is forwarded only from a node at Level* $-(i-1)$*, through a node at Level* $-(i)$*, to a node at Level* $-(i+1)$*, then the packet traverses only through the shortest path between end points.*

Proof. Shortest distance from the source node to a node at *Level* $-(i)$ is i from the definition of *Level*. Applying Lemma 3.4.1 to Source Node, *Level* $-(i-1)$ Node and *Level* $-(i)$ Node gives that the three nodes are on the shortest path from Source Node to the *Level* $-(i)$ Node. Applying the same concept to Source Node, *Level* $-(i)$ Node and *Level* $-(i+1)$ Node gives the same shortest path result. Combining both results, each node on the communication path has optimum depth. ■

Although in a connected network, each node can reach every node, the rule stated restricts the nodes that can be reached through a node. This concept is explained in Figure 3.8.

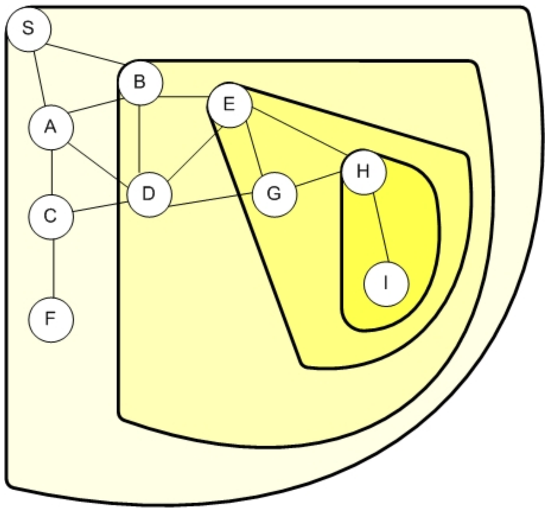


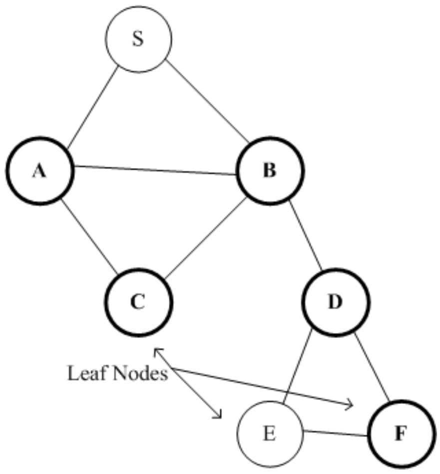
Figure 3.8: Loss of Sight. Due to the restriction of communication between the levels, the number of nodes covered by a neighbor decreases as the process goes further away from the source.

Due to this rule, nodes of the destination set covered by a node is restricted. Using this concept, starting from the source node, each node may try to find the neighbor that covers the largest set of destinations. But to utilize this process, the knowledge of *Destinations Covered by a Neighbor* information is needed by every node.

To distribute this information, the leaf nodes start to broadcast whether they are in the destination set or not. Any parent node, that receives this information, accumulates the information from every child node. When all the child nodes of a node have transmitted their information, that node broadcasts the accumulated information to be received by its parent nodes. This distribution continues towards the source node.

When the distribution is finished, each node in the network has the knowledge of the nodes covered by each of its child neighbors.

This procedure is explained in Figure 3.9, 3.10, 3.11 and 3.12 in an example.



Information Received from Neighbors(W: Waiting, -: Not a Child)

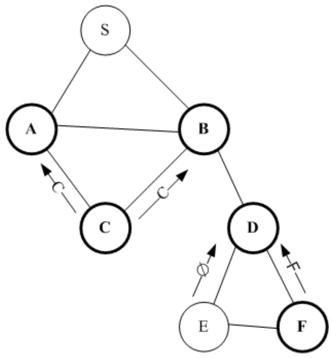
	S	A	B	C	D	E	F	State
S	-	W	W	-	-	-	-	Waiting
A	-	-	-	W	-	-	-	Waiting
B	-	-	-	W	W	-	-	Waiting
C	-	-	-	-	-	-	-	Leaf
D	-	-	-	-	-	W	W	Waiting
E	-	-	-	-	-	-	-	Leaf
F	-	-	-	-	-	-	-	Leaf

Figure 3.9: Nodes Covered Information Distribution Example. The leaf nodes, *C*, *E* and *F* will start the distribution.

Using this information, starting from the source node, each node tries to find the best possible cover. To achieve this, a well known set covering algorithm called Greedy Set-Cover has been used.

The idea of the algorithm is simple. Select the set with maximum number of members and delete the selected members from every remaining sets. Formally the definition is given in Algorithm 3.4.1.

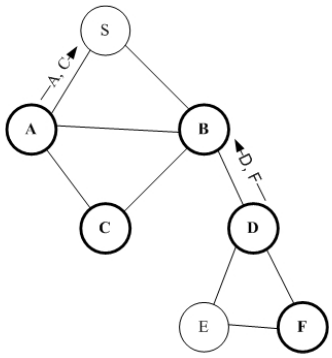
An example explaining the Greedy Set Cover algorithm is given in Figure 3.13.



Information Received from Neighbors(W: Waiting, -: Not a Child)

	S	A	B	C	D	E	F	State
S	-	W	W	-	-	-	-	Waiting
A	-	-	-	C	-	-	-	**Ready**
B	-	-	-	C	W	-	-	Waiting
C	-	-	-	-	-	-	-	*Finished*
D	-	-	-	-	-	∅	F	**Ready**
E	-	-	-	-	-	-	-	*Finished*
F	-	-	-	-	-	-	-	*Finished*

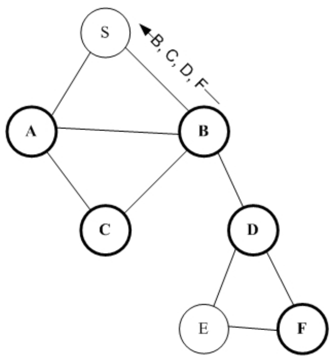
Figure 3.10: Nodes Covered Information Distribution Example. The leaf nodes, *C*, *E* and *F* start the distribution. Only the parent nodes receive these informations. Since Node *A* and *D* have received from all their child nodes, they will broadcast their accumulated informations.



Information Received from Neighbors(W: Waiting, -: Not a Child)

	S	A	B	C	D	E	F	State
S	-	A, C	W	-	-	-	-	Waiting
A	-	-	-	C	-	-	-	*Finished*
B	-	-	-	C	D, F	-	-	**Ready**
C	-	-	-	-	-	-	-	*Finished*
D	-	-	-	-	-		F	*Finished*
E	-	-	-	-	-	-	-	*Finished*
F	-	-	-	-	-	-	-	*Finished*

Figure 3.11: Nodes Covered Information Distribution Example. The ready nodes *A* and *D* broadcast their accumulated information. Node *B* becomes ready since all of its child nodes have transmitted. Source Node is still waiting for *B*.



Information Received from Neighbors(W: Waiting, -: Not a Child)								
	S	A	B	C	D	E	F	State
S	-	A, C	B, C, D, F	-	-	-	-	**Ready**
A	-	-	-	C	-	-	-	*Finished*
B	-	-	-	C	D, F	-	-	*Finished*
C	-	-	-	-	-	-	-	*Finished*
D	-	-	-	-	-		F	*Finished*
E	-	-	-	-	-	-	-	*Finished*
F	-	-	-	-	-	-	-	*Finished*

Figure 3.12: Nodes Covered Information Distribution Example. The ready node *B* broadcasts its information. The Source Node has received all the necessary information to advance to the next step.

Algorithm 3.4.1 Greedy Set Cover Algorithm

Let $\mathbb{N}_1,\ \mathbb{N}_2, \dots,\ \mathbb{N}_n$ be the set of destinations covered by each child node respectively.

Let $\mathbb{M}$ be the set of destinations to be covered.

Let $\mathbb{C} \leftarrow \emptyset$ be the set of nodes covered so far.

while $\mathbb{C} \neq \mathbb{M}$ **do**

  $maxIndex \leftarrow \arg\max_i |\mathbb{N}_i|$

  $\mathbb{C} \leftarrow \mathbb{C} \cup \mathbb{N}_{maxIndex}$

  for $\forall i$ **do**

    $\mathbb{N}_i \leftarrow \mathbb{N}_i - \mathbb{N}_{maxIndex}$

  end for

end while

The source node starts the Greedy Set Cover and sends the results of the algorithm to its neighbors, telling them which destination nodes to cover. This procedure is repeated in each node that receives the command to cover a destination subset.

This step of the second phase is explained in Figures 3.14, 3.15 and 3.16.

Nodes	Covered Nodes			
A	E	F	G	
B	E	F		
C		F	G	
D	E			H
Covered:	-	-	-	-

⇓ Select A, update neighbors

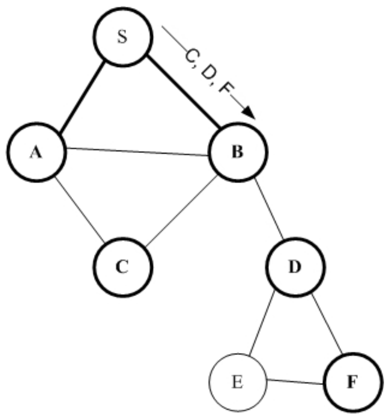
Nodes	Covered Nodes			
A	-	-	-	
B	-	-		
C		-	-	
D	-			H
Covered:	E(by A)	F(by A)	G(by A)	-

⇓ Select D, update neighbors

Nodes	Covered Nodes			
A	-	-	-	
B	-	-		
C		-	-	
D	-			-
Covered:	E(by A)	F(by A)	G(by A)	H(by D)

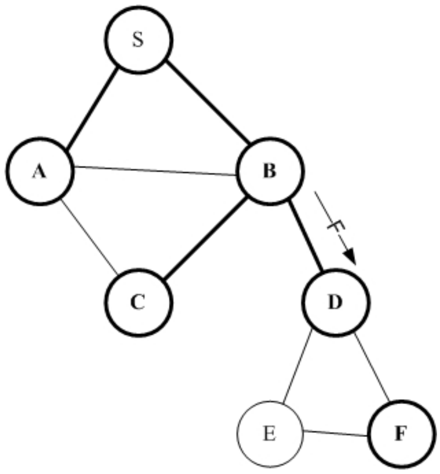
Solution: $\mathbf{A} \rightarrow \{\mathbf{E}, \mathbf{F}, \mathbf{G}\}$ and $\mathbf{D} \rightarrow \{\mathbf{H}\}$

Figure 3.13: Greedy Set Cover Example. In this example the node decides which neighbors(A, B, C, D) to select to reach the destinations(E, F, G, H).



Set Cover for Source Node	
Destinations Covered by A	Destinations Covered by B
C	C, D, F
Result: $B \rightarrow \{C, D, F\}$	

Figure 3.14: Covering Phase Example(1/3). The Source Node starts the Set Cover algorithm and sends the result to Node B.



Set Cover for Node B	
Destinations Covered by C	Destinations Covered by D
$\emptyset$	F
Result: $D \rightarrow \{F\}$	

Figure 3.15: Covering Phase Example(2/3). Node B receives the command, starts the Set Cover algorithm and sends the result to Node D.

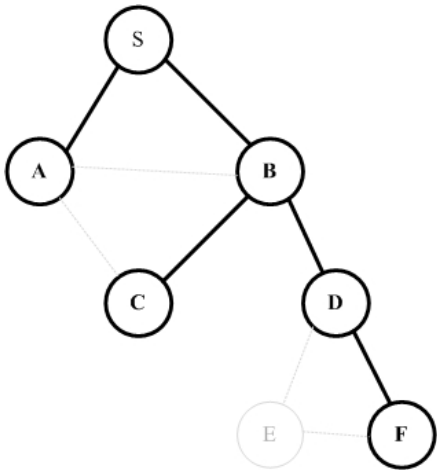


Figure 3.16: Covering Phase Example(3/3). Resulting tree of SWIM. This is also the optimal solution in terms of NFN for this example case.

The result of SWIM is always a tree. The proof is given in Lemma 3.4.3.

Lemma 3.4.3 *Resulting path of SWIM is a tree.*

Proof. SWIM starts from the Source Node and relies on Greedy Set-Cover algorithm. The result of Set-Cover is a group of disjoint sets. Since each intermediate node works on disjoint sets and produces disjoint sets, there can be only one distinct path between each pair of nodes, which is the definition of a tree. ■

This concludes the two phases of SWIM for the static network with loss tolerant case.

CHAPTER 4

SWIM - MESH SOLUTION

A new solution for the Wireless Multicast Tree Problem is given for the simple case of static networks with loss tolerant traffic in Chapter 3. The solution creates a routing tree. The definition of a tree states that when any edge of the tree is broken, the network becomes a disconnected graph. This means a single point of failure and decreases robustness. To extend the solution to mesh networks, where robustness is important, there is a need for alternative paths to destinations, if possible.

These alternative paths can be used as a backup route or to increase throughput through redundancy. The other single point of failure in the routing tree is the source. Allowing multiple sources on a network will increase robustness and also can decrease delay since there is a possibility that the other sources may have a smaller distance to some destinations than the original source.

This chapter explains the extension of alternative paths and the ability to use multiple sources to SWIM, in order to extend the usage to Mesh Networks. The outline is as follows: Section 4.1 explains the extension to the leveling phase (Section 3.3) of SWIM and Section 4.2 explains the generation of alternative paths, which is an extension to the covering phase (Section 3.4) of SWIM.

4.1 Multiple Source Handling

The leveling phase of SWIM partitions the network into layers of levels according to the distance to the source. In case of multiple sources, this concept needs to be changed. Since the main aim of SWIM is to achieve minimum depth, the depth of any destination to a source

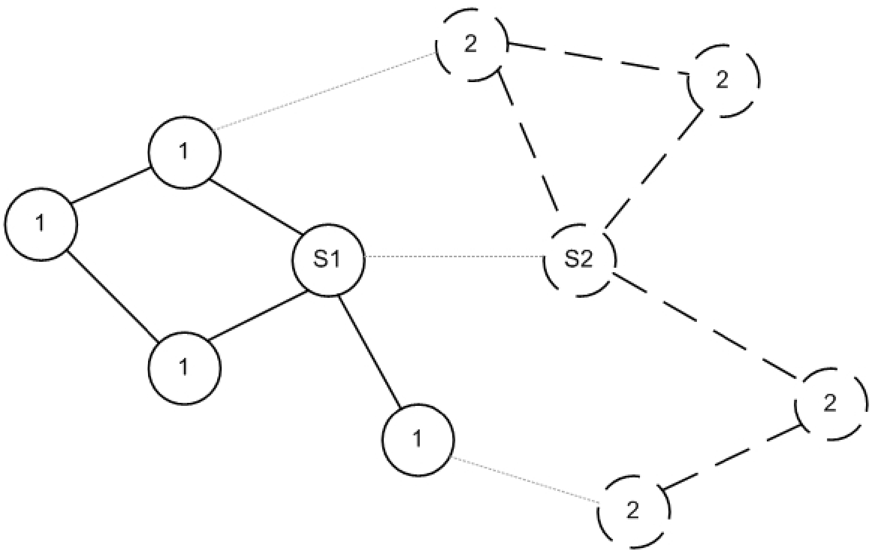


Figure 4.1: Sub-networks due to Multiple Sources. This example shows that the network is divided into two disjoint sub-networks due to two sources.

needs to be minimum. So, leveling the nodes by their distances to the nearest source node will achieve minimum depth for any node.

This idea automatically partitions the network into subnetworks, where the task of reaching a destination node will be assigned to a particular source node. This rises from the fact that the distance of a destination node to a source node need not be minimum for all sources. Using the source that has a greater distance than the minimum one will violate the concept of depth optimality. This concept of subnetworks is explained on Figure 4.1 and proved in Lemma 4.1.1. The Lemma also proves the depth optimality of the multiple source case.

Lemma 4.1.1 *Given multiple source nodes in a connected network, if the nodes in the network are classified into groups according to the minimum distance to any source node (equal distance case is resolved by randomly picking a source while maintaining connectivity) and if these groups are not allowed to communicate, the resulting graph consists of a set of disjoint connected sub-graphs with optimal depths.*

Proof. If a minimum distance to a source node exists for a node, then that node and all other nodes on the shortest path will be connected to that source node by applying Lemma 3.4.1 recursively. Since all the selected nodes belong to one unique source, the resultant sub-graphs are disjoint. The only mean of connecting these sub-graphs is through the nodes having equal distance to multiple sources.Since these nodes are assigned to one source only and the communication links to the group of other sources are not allowed, the connections within the sub-graphs are preserved while keeping the sub-graphs disjoint. Since only the minimum distance to each source is selected, the depth optimality is also preserved for all nodes. ■

This concept can easily be realized in a distributed fashion. Instead of only a single source broadcasting the initial leveling information, all sources broadcast simultaneously the initial level value of 0. If a node receives multiple values, it selects the minimum one and rebroadcasts its minimum level information. This operation is explained on Figures 4.2 and 4.3.

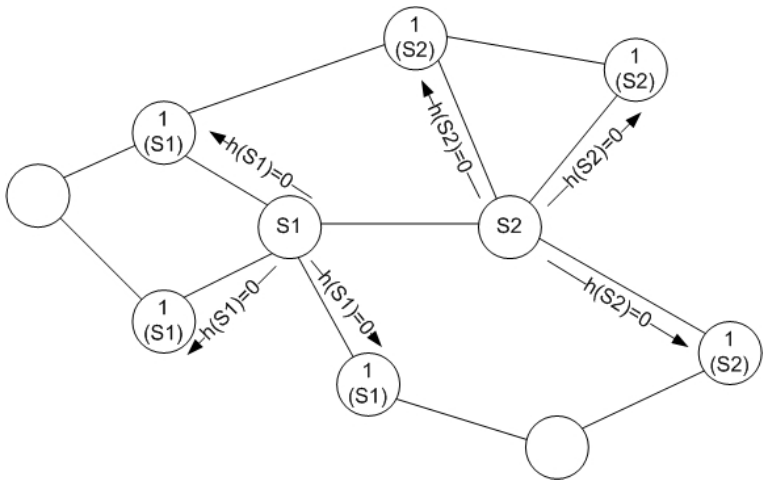


Figure 4.2: Multi Source Leveling Example. Both sources broadcast their level information to their neighbors.

After the leveling is finished and each node is leveled according to its minimum destination to a source, the information of the destinations seen by each node needs to be accumulated by the sources. This procedure is unchanged since the information from each node to the source is conveyed from child nodes to parent nodes. Since the definition of child and parent node concepts are valid only for a single source node, each node will convey its information to the source that is closest to itself. An example is given on Figures 4.4 4.5 and 4.6 for clarification.

Since the network is divided into subnetworks due to the new leveling definition, the covering phase will also be the same. The only difference is that each source will try to reach the destinations from which the seen destination information has arrived. In other words, each source will cover the destinations in its own subnetwork. This is explained on an example in Figure 4.7.

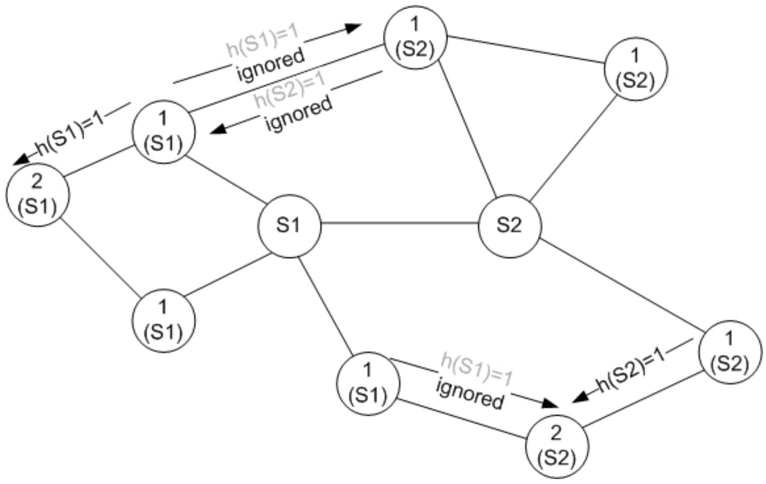
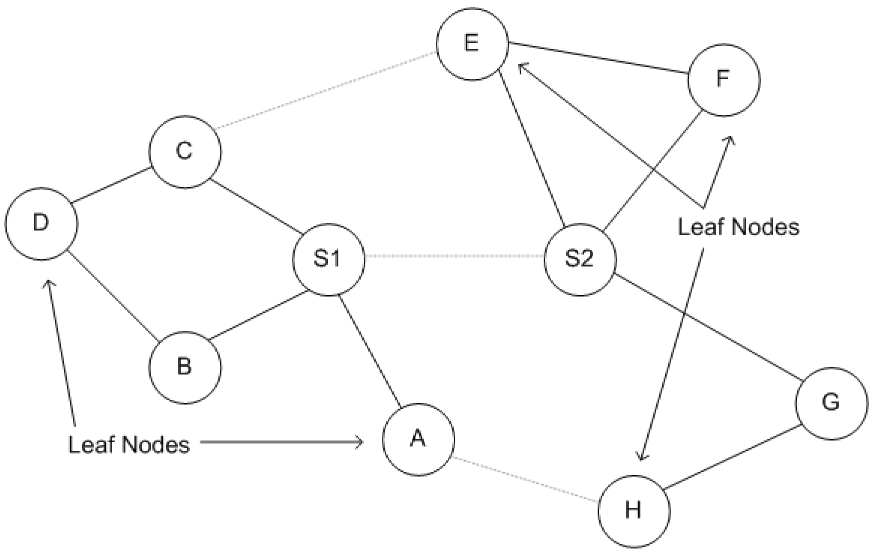


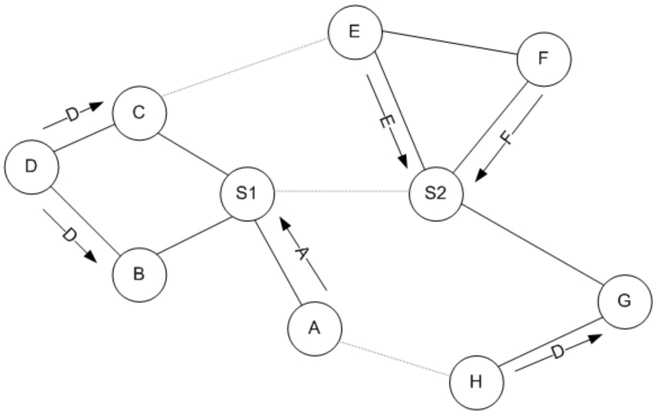
Figure 4.3: Multi Source Leveling Example. Nodes continue to broadcast the level information. If a new information for a different source is received and it has a greater or equal distance, then it is ignored. A node may have equal distance to both sources. The information that arrives earlier is accepted.



Information Received from Neighbors(W: Waiting, -: Not a Child)

	S1	A	B	C	D	State		S2	E	F	G	H	State
S1	-	W	W	W	-	Waiting	S2	-	W	W	W	-	Waiting
A	-	-	-	-	-	Leaf	E	-	-	-	-	-	Leaf
B	-	-	-	-	W	Waiting	F	-	-	-	-	-	Leaf
C	-	-	-	-	W	Waiting	G	-	-	-	-	W	Waiting
D	-	-	-	-	-	Leaf	H	-	-	-	-	-	Leaf

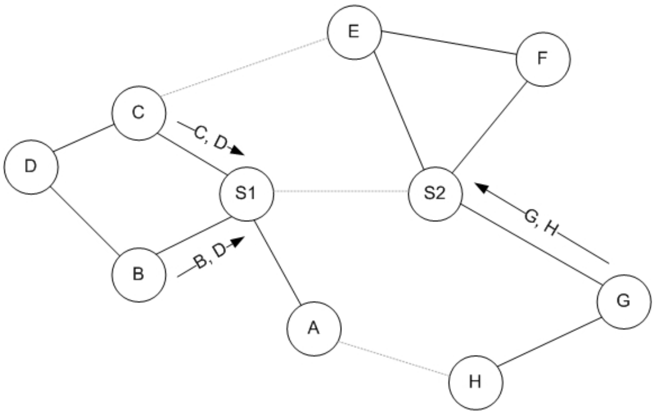
Figure 4.4: Nodes Covered Information Distribution - Multi Source Example(1/3). The leaf nodes of both sub-networks, *A*, *D*, *E*, *F* and *H* will start the distribution.



Information Received from Neighbors(W: Waiting, -: Not a Child)

	S1	A	B	C	D	State		S2	E	F	G	H	State
S1	-	A	W	W	-	Waiting	S2	-	E	F	W	-	Waiting
A	-	-	-	-	-	*Finished*	E	-	-	-	-	-	*Finished*
B	-	-	-	-	D	**Ready**	F	-	-	-	-	-	*Finished*
C	-	-	-	-	D	**Ready**	G	-	-	-	-	H	**Ready**
D	-	-	-	-	-	*Finished*	H	-	-	-	-	-	*Finished*

Figure 4.5: Nodes Covered Information Distribution - Multi Source Example(2/3). On the first sub-network Nodes *B* and *C* will merge the information and broadcast it. And in the second sub-network Node *G* will merge and broadcast the information.



Information Received from Neighbors(W: Waiting, -: Not a Child)

	S1	A	B	C	D	State		S2	E	F	G	H	State
S1	-	A	B, D	C, D	-	**Ready**	S2	-	E	F	G, H	-	**Ready**
A	-	-	-	-	-	*Finished*	E	-	-	-	-	-	*Finished*
B	-	-	-	-	D	*Finished*	F	-	-	-	-	-	*Finished*
C	-	-	-	-	D	*Finished*	G	-	-	-	-	H	*Finished*
D	-	-	-	-	-	*Finished*	H	-	-	-	-	-	*Finished*

Figure 4.6: Nodes Covered Information Distribution - Multi Source Example(3/3). Both sub-networks finished the collecting of the covered nodes information.

For S1			For S2	
Nodes	Covered Nodes		Nodes	Covered Nodes
A		E		
B	D	F		
C	D	G	H	
Covered:	-		-	

⇓ Select B for S1 and G for S1, update neighbors

For S1			For S2	
Nodes	Covered Nodes		Nodes	Covered Nodes
A	-	E	-	
B	-	F	-	
C	-	G	-	
Covered:	D(by B)		H(by G)	

Solution: B → {**D**} for S1 and **G** → {**H**} for S2

Figure 4.7: Greedy Set Cover - Multi Source Example. In this example the node decides which neighbors(A, B, C for S1, E, F, G for S2) to select to reach the destinations(A, B, C, D for S1, E, F, G, H for S2).

4.2 Alternative Path Generation

In this section, the main aim is to generate alternative paths to destinations. Since the main property of SWIM is the minimum depth of each destination to the source node, usage of alternative paths must not violate this property.

Due to this minimum depth property, the leveling phase cannot be changed. So, to generate an alternative path, the covering algorithm needs to be modified. There is a very simple way to adjust it. Since the new path needs to be an alternative, it should not have an intersection with the old path, if possible. To avoid the intersection, the nodes selected on the main tree are simply discarded in the calculation of the set cover and a new cover is calculated, if possible. This gives the second best output of the set cover without any intersection with the main output. This procedure of the alternative set cover is given in Algorithm 4.2.1.

Algorithm 4.2.1 Alternative Set Cover

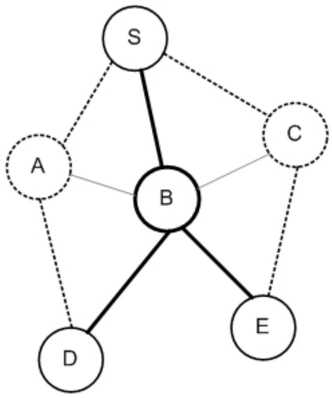
Let SC_{old} be the results of Set Cover algorithm

Let N be the set of neighbors

$N \leftarrow N - SC_{old}$

Re-run Set Cover algorithm

Figure 4.8 shows the main and alternative solutions on an example network.



Main Tree Solution

	A	B	C
Destinations Covered	D	D, E	E

Solution: $\mathbf{B} \rightarrow \{\mathbf{D}, \mathbf{E}\}$

Alternative Tree Solution

	A	-	C
Destinations Covered	D	-	E

Solution: $\mathbf{A} \rightarrow \{\mathbf{D}\}$ and $\mathbf{C} \rightarrow \{\mathbf{E}\}$

Figure 4.8: Alternative Path Example. Source tries to reach the destination set $\{D, E\}$. The main solution is B. When node B is excluded and the set cover algorithm is re-run, the alternative solution is A, C.

The resultant alternative tree is also depth optimal which is proved in Lemma 4.2.1.

Lemma 4.2.1 *Alternative Tree generated by SWIM is also depth optimal.*

Proof. Alternative Set-Cover operation is still done on the same conditions as the normal Set-Cover. Since the same condition of communication is still valid, applying the Lemma 3.4.2 is sufficient. ■

These two extensions provide SWIM the ability to be used in a Mesh network.

CHAPTER 5

SWIM - DYNAMIC NETWORK SOLUTION

Chapters 3 and 4 explained the multicast routing algorithm as a solution to the WMT problem for static networks. Even though there are many uses of static (or low mobility) networks, an extension for dynamic networks would allow SWIM to cover a greater range of applications. This chapter explains the solution for dynamic networks which covers both mobility and multicast group member changes. The outline of the chapter is as follows: Section 5.1 introduces the solution steps followed, while Section 5.2 explains a simpler method to be used for the dynamic case. Finally Section 5.3 combines the steps and explains the main solution.

5.1 Solution Approach

In a dynamic network, there are two major problems to be solved. The first one arises due to the change in the topology caused by mobility and the second one is the change in the destination group.

To solve the problem arising from topology change, the information about the topology needed to construct the tree needs to be updated. The most basic solution would be to periodically transmit the level information and the destinations seen information. This solution would be too costly with a high overhead and a probability of an instable solution due to frequent changes of the topology.

There is a need for an alternative solution that will distribute and obtain the leveling information needed for the covering phase of SWIM.

Fortunately, this information can be obtained from any proactive routing algorithm that obtains the costs to each node. In our solution the cost metric will be hop distance. This solution is explained in detail in Section 5.2.

5.2 Alternative Solution Using a Proactive Routing Algorithm

SWIM requires the information of which node to use to cover the desired group of destinations without violating the leveling rule (communication with parents and children only). The leveling information is automatically obtained in a proactive routing algorithm since each node knows the hop distance to each other node. Also, since it receives the same information from its neighbors, the node knows the same information of its neighbors. 5.2.1

Through this information any node can construct its groups of parents, siblings and children. The steps of construction is given in Algorithm 5.2.1.

Algorithm 5.2.1 Leveling by using a proactive routing information

Let $N_i(j)$ be the hop distance of Node i to Node j obtained from its table.

Let $N(j)$ be the hop distance of the current node to Node j.

Let S be the source node.

Let $\mathbb{P}$, $\mathbb{S}$ and $\mathbb{C}$ be the parent, sibling and child node sets of the node respectively.

if $N(S) = N_i(S) + 1$ **then**

  $\mathbb{P} \Leftarrow \mathbb{P} \cup i$

else if $N(S) = N_i(S)$ **then**

  $\mathbb{S} \Leftarrow \mathbb{S} \cup i$

else

  $\mathbb{C} \Leftarrow \mathbb{C} \cup i$

end if

The tricky part is to use this information to obtain the destinations covered by each child neighbor. The algorithm to find the destination set covered by a child node is given in Algorithm 5.2.2.

The proof of this algorithm is given in Lemma 5.2.1, which is base on Lemma 3.4.1.

Lemma 5.2.1 *Given Nodes S, A, B and C, let B be the child of A and the shortest distance between A and C, $d(A,C) = D$ and between B and C, $d(B,C) = D + 1$. Then, Node A covers*

C over B. (shown on Figure 5.1)

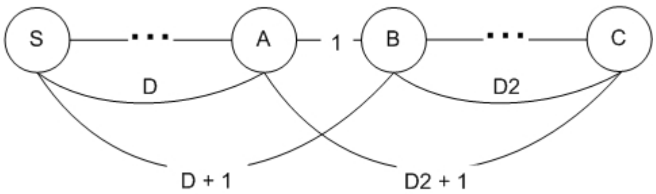


Figure 5.1: Illustration of Lemma 5.2.1. Node A covers Node C over Node B

Proof. Node B being the child of A means $d(S, A) + 1 = d(S, B)$ and $d(A, B) = 1$. A covers C over B means that C is reachable over a path through B without violating the minimum depth property. This means that B is on one of the shortest paths between A and C. Applying Lemma 3.4.1 first on S, A and B results in A is on one of the shortest paths from S to B, secondly on A, B and C results in B is on one of the shortest paths from A to C. Combining both results, A and B are on one of the shortest paths from S to C. Since the minimum depth property is not violated, A knows that it covers C over B with the given conditions. ■

Algorithms 5.2.1 and 5.2.2 are used together to gather the required information to start the second phase of SWIM, the covering part. This shows that a proactive routing algorithm can be used to obtain updated leveling information in a mobile network. The remaining problem is to update the tree which is explained in Section 5.3.

Algorithm 5.2.2 Algorithm to find the destinations covered by a child node

Let $\mathbb{C}$ be the child node set of the node.

Let $N_i(j)$ be the hop distance of Node i to Node j obtained from its table.

Let $N(j)$ be the hop distance of the current node to Node j.

Let $i \in \mathbb{C}$

if $N(j) = N_i(j) + 1$ **then**

  j is covered by i

end if

5.3 Dynamic Network Solution

Section 5.2 explained how to use a proactive routing algorithm in order to find the leveling information. The information will be updated with a selectable rate which can be set according to the mobility level.

With an up-to-date leveling information, the covering phase of SWIM can be used to construct the solution tree. Since the constructed tree can become obsolete due to mobility or network dynamics, it needs to be updated too. To update, the covering phase can be re-run so that the tree is basically reconstructed. This process also ensures that the tree covers the newly joined destinations or cuts the unnecessary transmissions to cover the obsolete destinations that left the group.

The rate can be selected according to the mobility level of the network.

CHAPTER 6

ANALYSIS AND SIMULATION

In the previous chapters, SWIM, a multicast routing algorithm has been explained in detail for various usage purposes. In this chapter, the performance of SWIM will be analyzed through various simulations. In some cases, comparisons to other multicast routing algorithms in the literature will be done. The outline of this chapter is as follows: Section 6.1 analyses SWIM in terms of computational complexity and signaling overhead. Then in Section 6.2 various performance simulations of SWIM are given.

6.1 Complexity Analysis

This section analyses the computational complexity and the signaling overhead of SWIM to give an insight into the practical usage.

6.1.1 Computational Complexity Analysis

Since SWIM is a heuristic solution to an NP-Complete problem, it needs to be computationally efficient in order to be worthy of usage. There are three major computations in SWIM:

1. Leveling
2. Merging of Destinations Seen Information
3. Set-Covering

The leveling part requires only a basic comparison of the received level of a neighbor with the previously received level informations in order to classify into groups. This search operation within a set of maximum size N has a worst case computational complexity of $O(N)$.

The merging of destinations seen information part needs to find the duplicate destinations in the previously received informations and the currently received on. This is again a search operation within a set of maximum size N and has a worst case computational complexity of $O(N)$.

Actually the last part has the biggest impact on the complexity. The Greedy Set-Cover algorithm finds the maximum sized set within its neighbors, which is $O(N)$ and it needs to repeat this until all the destinations are reached which is a repeat of $O(N)$ times, resulting in a node worst case computational complexity of $O(N^2)$. This result is also proved in [14]. Since there are N nodes in the network, the network worst case computational complexity of SWIM is $O(N^3)$.

This result seems to be a loose bound and extensive set of simulations confirm that the average computational complexity is in fact $O(N^2)$, network wide. This result is shown in Figure 6.1. In the simulations, the computational complexity (y-axis) was computed by counting the number of assembly level instructions for each network instance, over a sufficiently large number of instances, in order for the running average of the value to stabilize. This operation was repeated for various network sizes.

Even though the simulations show otherwise, there is a counter example on Figure 6.2 that exhibits a computational complexity of $O(N^3)$.

6.1.2 Signaling Overhead

Another important performance metric is the signaling overhead of an algorithm. Two signaling overhead measures are considered. The first measure is the number of messages sent to construct a tree from initial startup state of the network.

There are three main messaging requiring operations in SWIM. The first task is the leveling part where the neighbors are categorized. In nominal operation (ignoring link failures), each node makes a single transmission since all the transmissions are broadcasts, sufficient for all

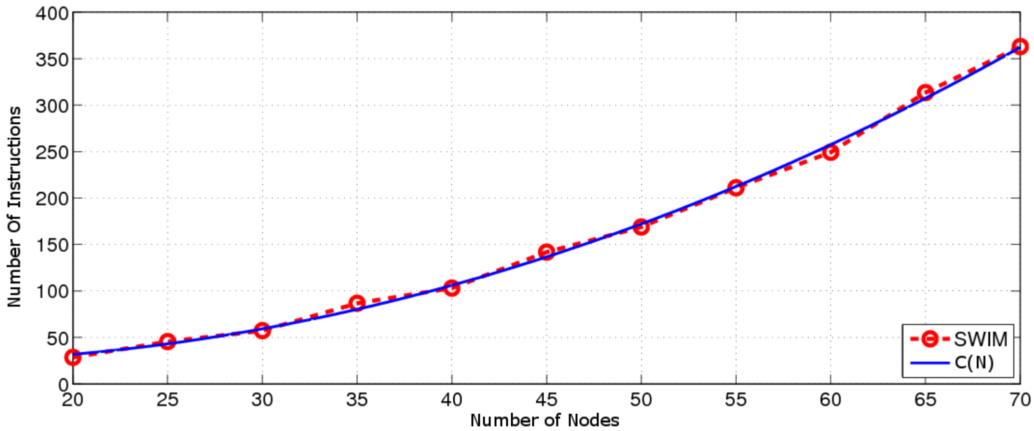


Figure 6.1: Average Computational Complexity. The computational complexity for the broadcast case of SWIM is given as *SWIM* and the curve fitted to the simulation result is given as $C(N)$. The equation of the curve is found as $C(N) = 0.096N^2 - 2.1N + 35$ which shows that the average computational complexity can be modeled $O(N^2)$.

nodes to reach all of its neighbors. Considering all the nodes in the network, this task entails N messages in total.

The second task is the distribution of the information of seen destinations. This information is sent once by each node except the sources. If S represents the number of sources in the network, a total of $N - S$ transmissions is needed for this task.

The last task is to send the result of the Greedy Set-Cover to the respective neighbor node. This number is exactly equal to the size of the tree, in other words, NFN which is represented by T.

Overall, combining all messages, the number of messages is upper-bounded by $2N + T - S$. Since all these messages need to be sent protected against link failures, an Automatic Repeat Request (ARQ) mechanism is needed. But the ARQ will effect as a scaling constant and the order of number of messages will not be effected. The number of messages is upper-bounded by $O(N)$.

An actual count of messages is done in a simulation for various network sizes. 10000 instances have been averaged for a network size to get a stable result. The result is given in Figure 6.3 and seems to be consistent with the theoretical result.

The other signaling overhead is the length of messages sent. For this purpose, an actual

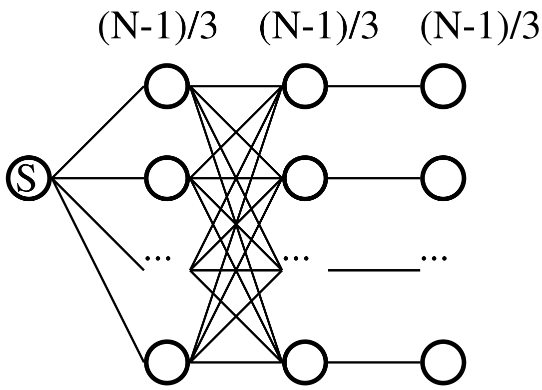


Figure 6.2: Upper Bound Example for Computational Complexity. The first hop neighbors of the Source Node will run the Greedy Set-Cover algorithm with computational complexity $O(N^2)$. Since $(N-1)/3$ number of nodes run the algorithm, the total computational complexity is $O(N^3)$ for this example.

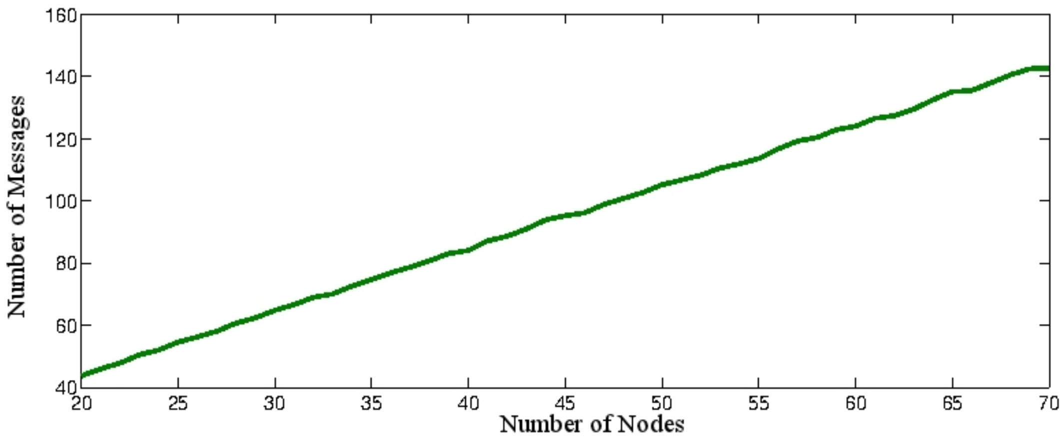


Figure 6.3: Number of Messages Simulation Results. The linear behavior of the curve supports the result of $O(N)$.

realization of the messages needs to be done. Three messages have been defined for the three tasks.

The message for the leveling part is given in Table 6.1.2.

The message to send the information of seen destinations is given in Table 6.1.2.

For the final part, the message to send the result of Set-Cover is given in Table 6.1.2.

Overall, multiplying the message lengths with the respective number of messages:

1. $N \times O(\log_2(N)) = O(N \log_2(N))$

Table 6.1: Leveling Message. An example realization of the message used to do the leveling.

Variable Name	Number of Bits	Explanation
Packet Type	2	Packet type to differentiate SWIM control packets.
Sender Address	$\log_2(N)$	Sender Address
H Value	$\log_2(N)$	Shortest Distance to the nearest source. (Level of sender)
Source Address	$\log_2(N)$	Source Node for which the H Value is valid.

Table 6.2: Seen Destinations Message. An example realization of the message used to send the seen destinations information.

Variable Name	Number of Bits	Explanation
Packet Type	2	Packet type to differentiate SWIM control packets.
Sender Address	$\log_2(N)$	Sender Address
Bitmap	N	Bitmap of the destinations seen. 1 if seen.

Table 6.3: Set-Cover Request Message. An example realization of the message used to send the request result of the Set-Cover operation.

Variable Name	Number of Bits	Explanation
Packet Type	2	Packet type to differentiate SWIM control packets.
Destination Address	$\log_2(N)$	Intended Receiver's Address
Bitmap	N	Bitmap of the destinations to be covered.

2. $O(N) \times O(N) = O(N^2)$
3. $O(N) \times O(N) = O(N^2)$

The message length is upper-bounded by $O(N^2)$.

6.2 Simulations

The performance of SWIM is studied with respect to four different aspects. Since SWIM is depth optimal, the first part of this section studies the *Depth* aspect. The motivation behind selecting the depth to be optimal was to keep the forwarding delay low. So, the second part studies the *Delay* aspect. A Greedy Set-Cover algorithm is used in SWIM in order to keep the number of forwarding nodes low. The third part studies the *NFN* aspect of SWIM. Finally, the dynamic case of SWIM is studied under mobility.

6.2.1 Tree Depth Simulations

The most powerful area of SWIM is the depth property, because it is depth-optimal. There are two different depth values for comparison:

1. Maximum Tree Depth: Maximum depth of a node on the multicast routing tree. It is expected that this property will effect the maximum delay or the upper-bound of delay of a multicast packet.
2. Average Tree Depth: Average of the depths of all the nodes on the multicast routing tree. This property is expected to effect the average delay of a multicast packet.

SWIM is compared to MAODV, one of the most popular and competitive multicast routing algorithms in the literature. For the simulations, random topologies have been generated and both algorithms are run on the same topology. The topology is generated according to the following settings:

1. Nodes are placed independently according to the uniform distribution in a unit square region of side length 1.
2. Transmission range, which is defined as the maximum distance between to nodes, such that they are assumed to be connected is 0.286. This number has been picked because it is the same value that was used in [22]. It corresponds to nodes having a transmission radius of 100 units located in a square area with 350 unit sides.

The algorithms were run on the same independent randomly created topologies for two different scenarios:

1. Multicast Simulation: In a network with 70 nodes ($N = 70$), the size of the multicast destination group is increased from 1(unicast) to 69(broadcast). For each size, 10000 different topologies have been created and the average of the results of these instances

have been taken. The multicast destinations are selected randomly over a uniform distribution.

2. Broadcast Simulation: The network size is varied from 20 to 70 while the multicast destination group is always all the nodes in the network. For each size, 10000 different topologies have been created and the average of the results of these instances have been taken.

The result of the multicast simulation is given in Figure 6.4.

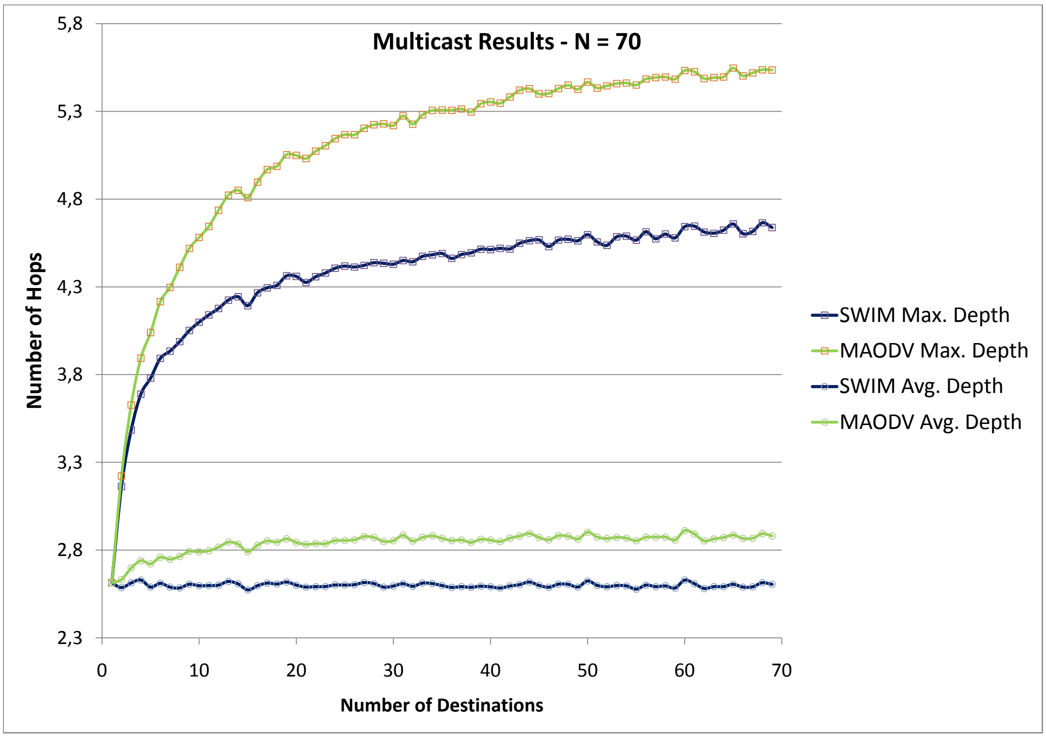


Figure 6.4: Multicast Depth Simulation Results. The results shows the impact of the depth optimality property of SWIM which has both a lower Maximum Depth and Average Depth when compared to MAODV.

The result of the broadcast simulation is given in Figure 6.5.

Another depth simulation scenario has been studied for multiple source case. Since multiple sources can be utilized by SWIM to its advantage, the impact of multiple sources to the depth performance needs to be observed. For this purpose, the previous topology settings have been used (0.286 transmission range on a unit square). The scenario is a broadcast scenario where the number of nodes in the network is varied from 20 to 70. Among these nodes, the number

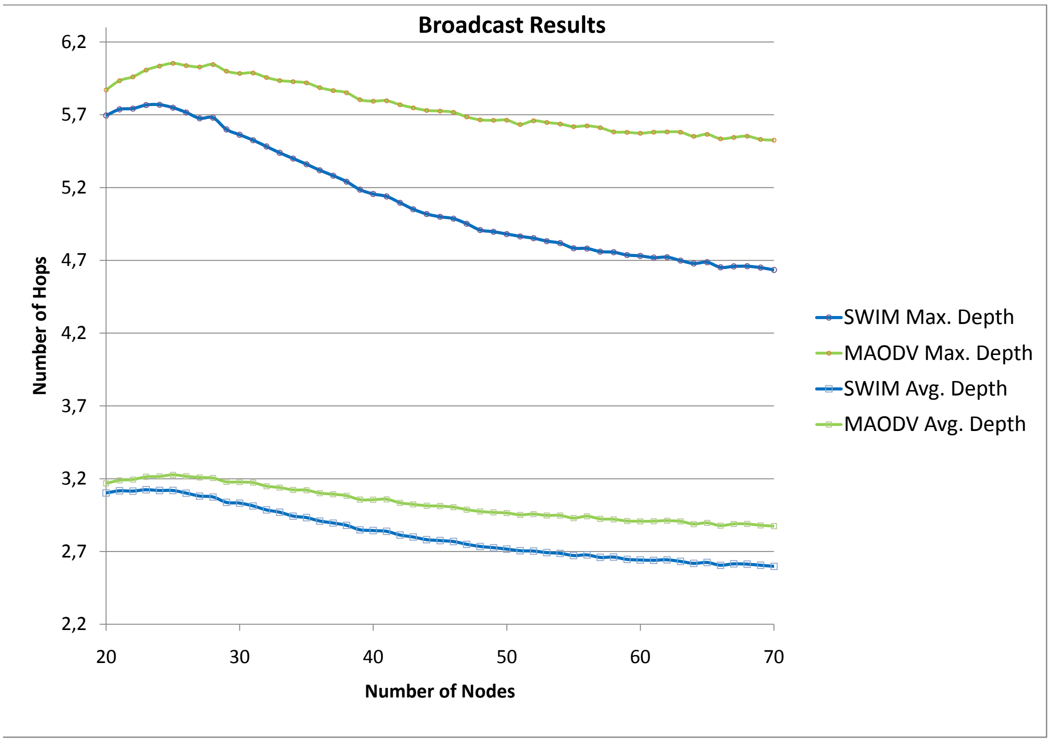


Figure 6.5: Broadcast Depth Simulation Results. The results shows the impact of the depth optimality property of SWIM which has both a lower Maximum Depth and Average Depth when compared to MAODV.

of sources has been selected 1, 2 or 3 for three different cases. For each network size, 10000 instances have been averaged. The results are given in Figure 6.6.

In overall, the depth optimality of SWIM can be clearly observed. The ability to use multiple sources provides another way to decrease the depth even further.

6.2.2 Delay and Delivery Ratio Simulations

Delay (latency) is an important aspect of multicast traffic, since multicast traffic is an efficient way to distribute delay sensitive traffic. The delay is expected to be another strong property of SWIM since the depth optimality was selected to decrease the forwarding delay. Two kinds of delay are studied in this work:

1. Average Delay: This performance metric takes the average of the delay of all packets received by any multicast destination. This means that a single packet will contribute

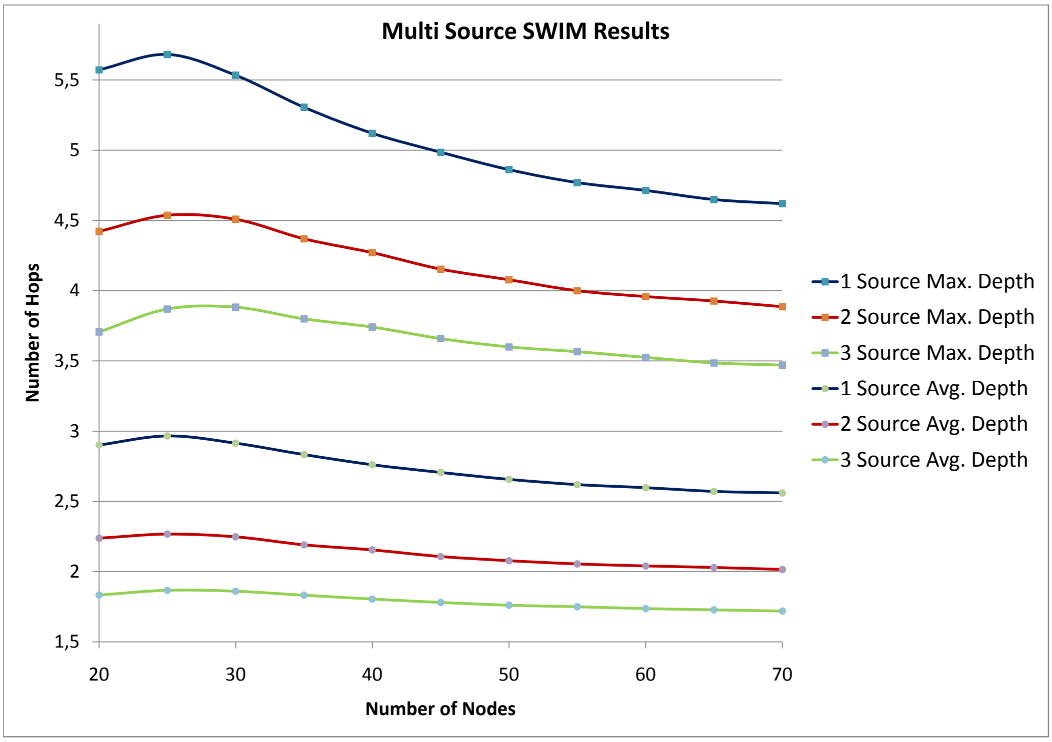


Figure 6.6: Multi Source Depth Results. This figure shows the impact of multiple sources on depth. Using multiple sources decreases the depth significantly. This shows that SWIM can take advantage of the availability of multiple sources in order to decrease the depth.

multiple points to the average since each multicast destination is expected to have a different delay. This result gives how much time it takes for a node in the network to receive a packet on average.

2. Maximum Delay: Maximum delay gives the maximum of a packet's delay among all destinations. This means that a single packet will have a single maximum delay throughout the network. This result gives how much it can take for a node at maximum to receive a packet.

Besides delay, the delivery ratio is also another performance metric to be studied. The delivery ratio is defined as the ratio of the average number of packets received by the destinations to the total number of sent packets.

In the simulations, SWIM is compared to MAODV. The simulations were performed in the popular simulation tool ns3. The settings of the simulation setup are given as:

- 802.11b Wireless Fidelity (WiFi) MAC Protocol - Ad Hoc mode.
- 1 Mbps bandwidth.
- Number of nodes in the network increased from 35 to 70.
- For each network size, results averaged over 100 topologies.
- Source Traffic: User Datagram Protocol (UDP) flow with $100kb/s$.
- Log Distance Propagation Model.
- Each instance run for 150 seconds.
- Destination group is broadcast.

The average delay result is given in Figure 6.7.

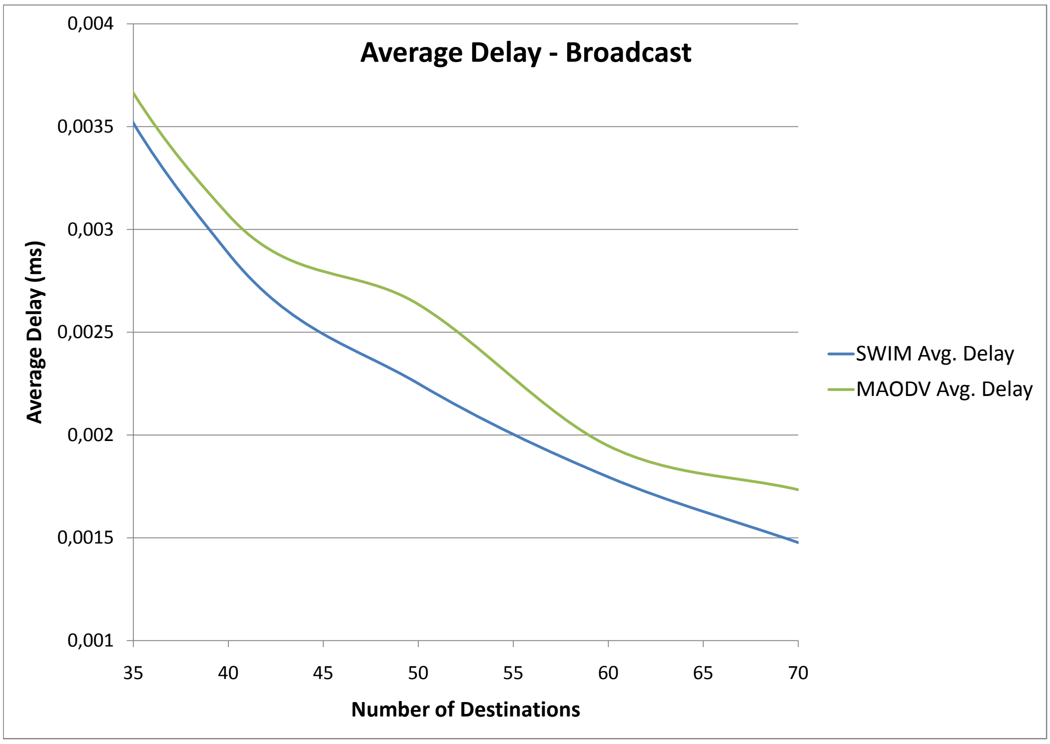


Figure 6.7: Average Delay Results. SWIM exhibits a lower average delay, as expected due to its depth optimality.

The maximum delay result is given in Figure 6.8.

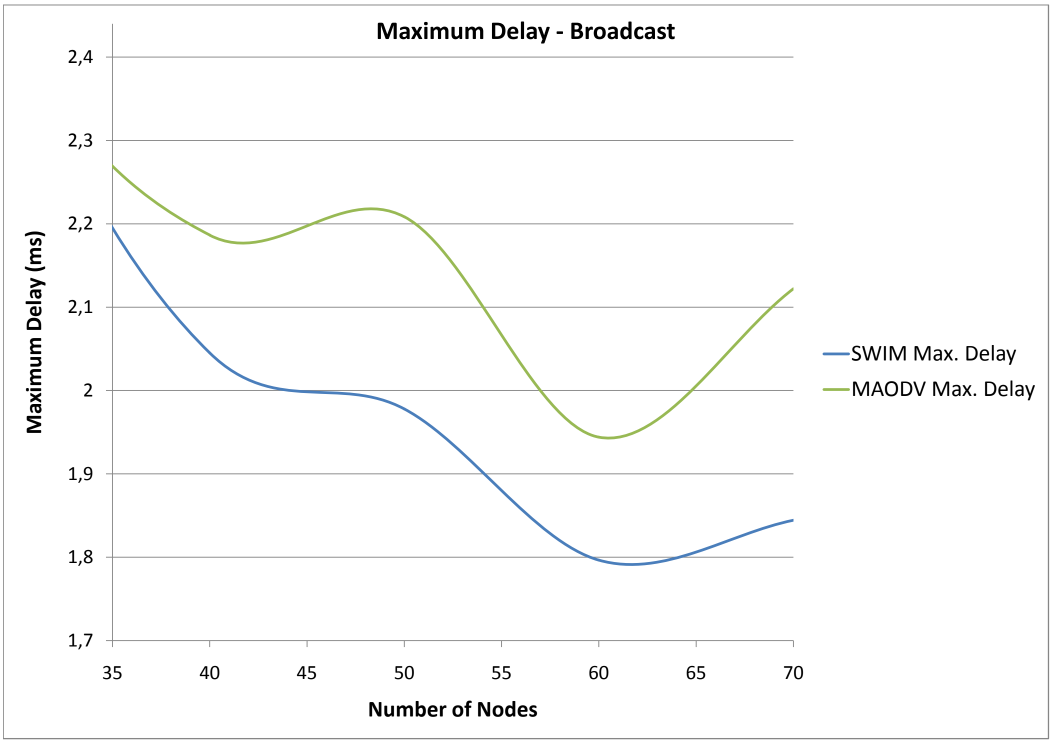


Figure 6.8: Maximum Delay Results. SWIM has a lower maximum delay, again as expected from depth optimality.

The results show that SWIM has a better performance in terms of delay. This result was expected since SWIM is designed to be depth optimal. This property is expected to decrease the forwarding delay which is a major component in total delay.

The delivery ratio result is given in Figure 6.9.

The delivery ratio is observed to be close to MAODV. But the delivery ratio of SWIM can be improved by using alternative paths, without losing the depth optimality in exchange for energy efficiency.

6.2.3 Number of Forwarding Nodes Simulations

Although it is not the first concern of SWIM, number of forwarding nodes, thus energy efficiency is still an important aspect. SWIM, tries to keep the NFN low by using the Greedy Set-Cover algorithm which tries to find the best covers by keeping the number of used nodes low. Two different simulations have been done for the NFN case:

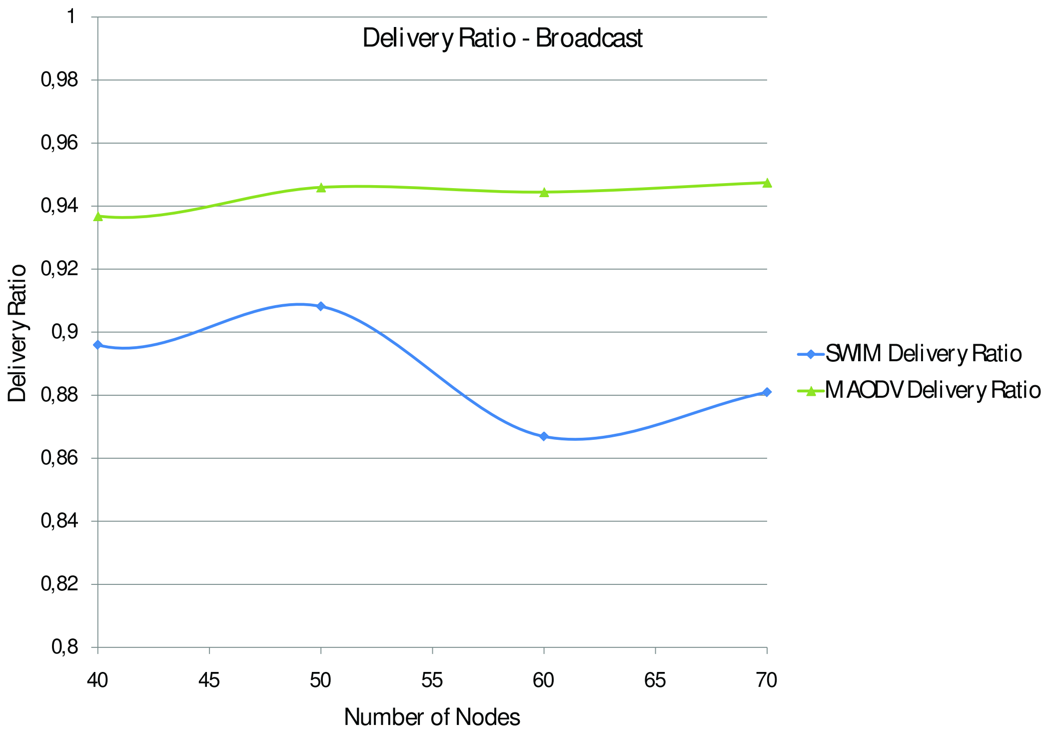


Figure 6.9: Delivery Ratio Results. SWIM has a lower delivery ratio but is still close to MAODV.

1. Broadcast NFN: This simulation tries to measure the NFN for the broadcast case, which is the worst case in terms of NFN. SWIM is compared to different algorithms in the literature.

2. Multicast NFN: This simulation finds the NFN for different destination group sizes including the unicast case. The result is compared to the optimum result.

For the broadcast simulation, the simulation setup is given as:

- Nodes placed independently with uniform distribution.
- Unit square area, side length = 1.
- The transmission range is 0.286.
- Number of nodes in the network is changed from 20 to 70.
- The average of 10000 instances taken for SWIM and MAODV.

- Other results taken from [22] and [11].

The results for the broadcast scenario is given in Figure 6.10.

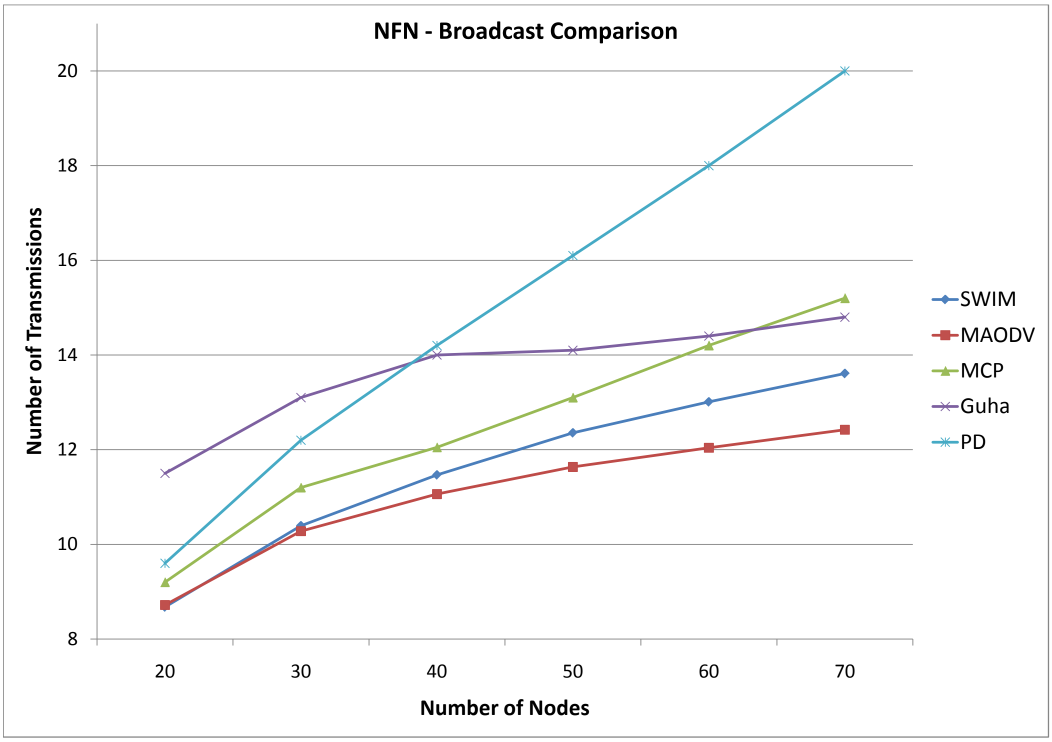


Figure 6.10: Broadcast NFN Results. SWIM is compared with various algorithms in the literature. Although not the best in terms of NFN, SWIM performs better than most of the given algorithms, while being close to MAODV. But NFN, thus energy efficiency is only a secondary concern.

For the multicast case, the simulation setup is as follows:

- Nodes placed independently with uniform distribution.
- Unit square area, side length = 1.
- The transmission range is 0.286.
- Destination group size help constant for 1, 2, 5, and 10.
- Number of nodes in the network changed from 20 to 70.
- The average of 10000 instances taken.
- Optimum found by brute force

The results for the broadcast scenario is given in Figure 6.10.

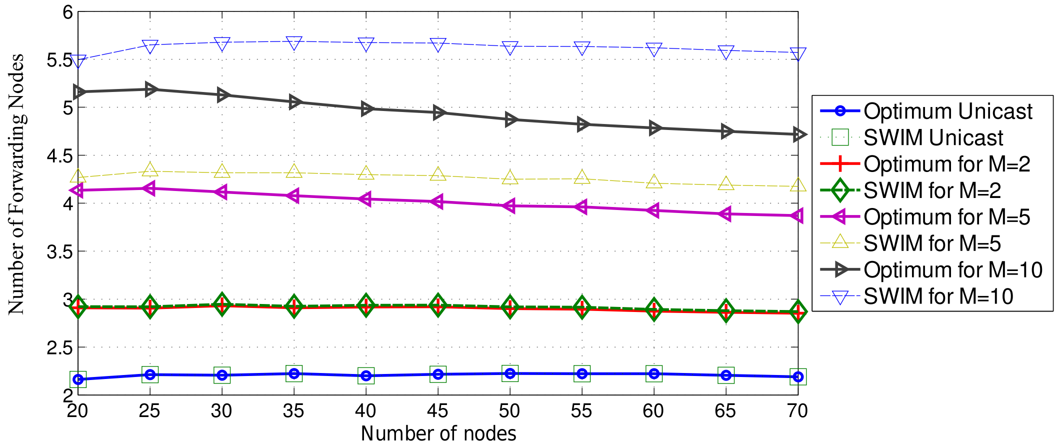


Figure 6.11: Multicast NFN Results. Results for different sized destinations sets are given. SWIM performs close to optimal. For the unicast case, it is optimal.

6.2.4 Dynamic Network Simulations

Another property of SWIM is that it can be used in dynamic network conditions, such as mobility. The performance of SWIM under mobility is studied. Under dynamic conditions, the level information and the tree need to be updated. The update rates can be selected differently for required network conditions.

The effect of mobility on the performance has been observed by simulations, with settings as follows:

- The simulations are performed on Network Simulator 3 (ns3).
- Brownian Motion Model (2d Random Way-point) is used for mobility.
- Node speed is distributed between 3 and $5m/s$.
- 802.11b WiFi MAC with $1Mb/s$ rate is used.
- Source traffic is UDP with $100kb/s$ rate.
- Number of nodes in the network varied from 10 to 40.
- Multicast group size fixed at 9.

- $2500m \times 2500m$ area.

The performance degradation under mobility can be made arbitrarily small by making the updates sufficiently frequent. Of course, the price paid is the extra overhead.

The area takes approximately 12 minutes to cross the largest distance in the field. This duration is selected as a benchmark to select the different update periods.

As explained in Section 5.3, there are two different update rates.

- Neighbor Update Period: Time between two consecutive proactive routing table broadcasts to update the neighbor information.
- Tree Update Period: Time between two consecutive runs of the Covering part of SWIM. The tree will be recalculated.

For the simulations, three different rate pair choices have been used:

1. Low Rate: Neighbor Update Period = 3 minutes, Tree Update Period = 30 minutes
2. Medium Rate: Neighbor Update Period = 30 seconds, Tree Update Period = 6 minutes
3. High Rate: Neighbor Update Period = 1 second, Tree Update Period = 1 minute

The first simulation is a comparison between the mobile case with a High update rate and a static network. The average delay of both cases are studied. For this simulation only, the destination group is every node (broadcast). The results are given in Figure 6.12.

The effect of mobility is seen to be minimal in terms of average delay when a sufficiently high update rate is selected.

The second simulation is to study the degradation in delivery ratio with different update rates. The results are given in Figure 6.13.

The effect of the selection of the update rates on the delivery ratio is observed. The appropriate rate can be selected according to the needs of the traffic characteristics.

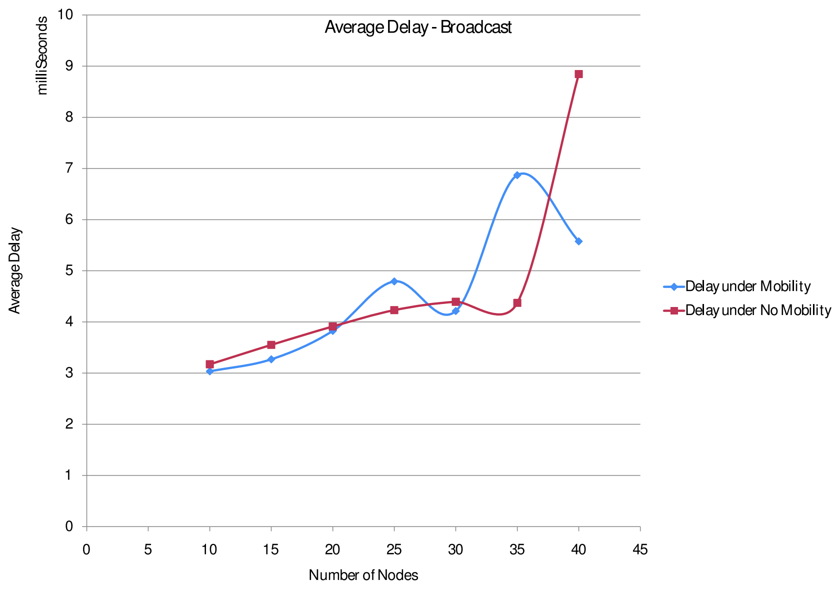


Figure 6.12: Average Delay Comparison under Mobility and No Mobility. The effect of mobility on the average delay performance is seen to be minimal when a High Rate pair is selected.

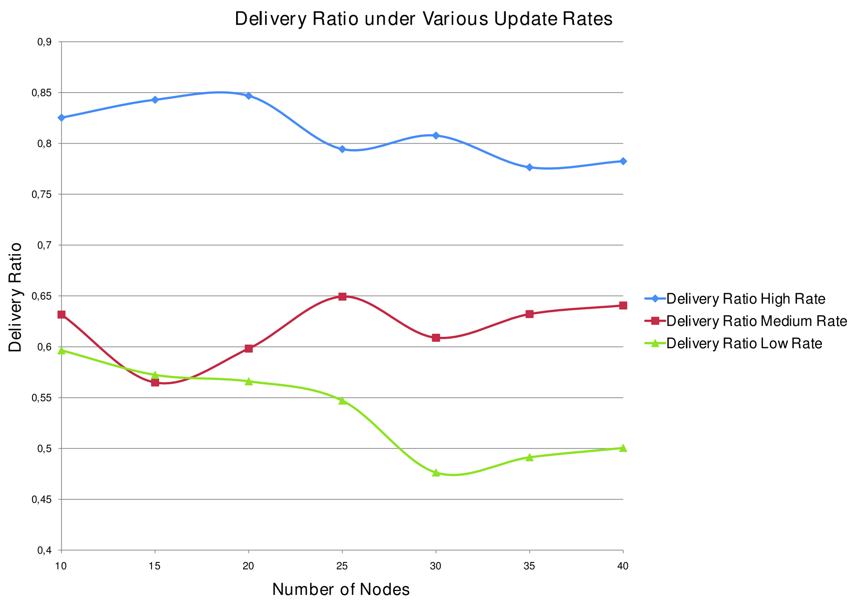


Figure 6.13: Delivery Ratio Results for the Mobile Case. The delivery ratio is observed to increase with increasing update rate. The selection of the update rates can be done according to the traffic needs.

A higher update rate, results in a better performance, but with a higher overhead cost. The final simulation studies the overhead of SWIM under the three different update rates. The results are given in Figure 6.14.

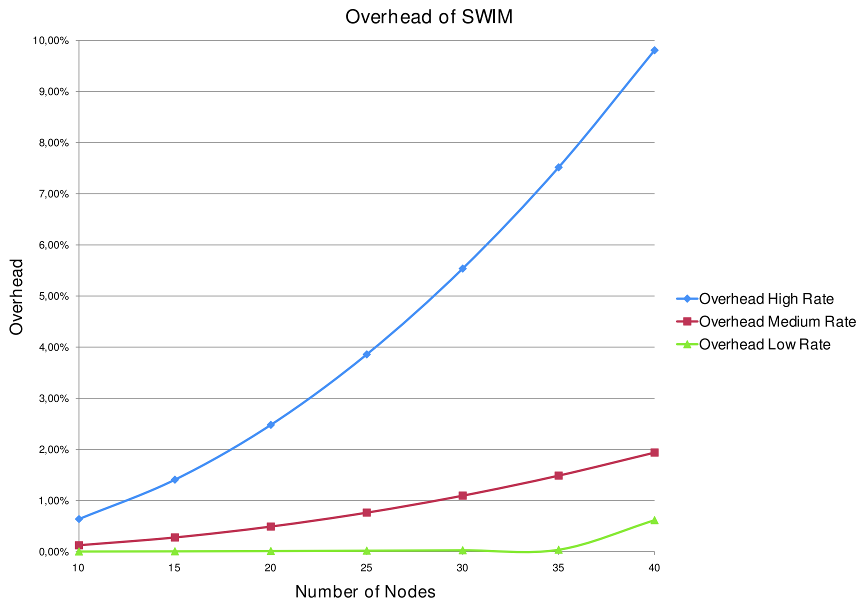


Figure 6.14: Overhead Results for the Mobile Case. Overall protocol overhead of SWIM is observed. The overhead is simply the ratio of SWIM data to the total data. The overhead can be selected from %1 to %10 according to the needs.

The overall protocol overhead of SWIM can be adjusted according to the needs. The selection of the rates depend on traffic needs, topology properties and bandwidth.

CHAPTER 7

DISCUSSION AND FURTHER DIRECTIONS

Until this chapter, packet drop rate was not considered as the main concern. This chapter makes a simplified analysis on a simplified network model and tries to find an expression on throughput. In this chapter a new solution, Rateless Codes, is introduced. The analysis is also applied to the new solution and the performances of both cases are compared.

The outline of this chapter is as follows: First, in Section 7.1 a background information on rateless codes will be given and methods to use this method in both reliable and unreliable communications will be explained. In Section 7.2, a simple analysis will be given for the cases with and without rateless codes.

7.1 Background on Rateless Codes

A new class of error control codes has been invented, which is called Rateless Codes [23]. LT codes [19], Raptor Codes [25] and Online Codes [20] are examples to this class. These codes have very simple encoding and decoding algorithms. The idea is simple. The packets to be sent are coded together into special codewords. Then, these codewords are transmitted until the receiver has sufficient number of packets enough to decode the original packets. Actually, the coding process distributes the overall information to the entire message (information and parity) as much as possible, such that the original message can be recovered from these packets. The required number of packets to receive is close to the number of information packets sent, which means an ignorable overhead.

Rateless codes can be used for both reliable and unreliable communications. For reliable communication, it can be used as an ARQ mechanism, where the destinations need only to acknowledge if they have received the packets. This decreases the amount of both retransmissions and acknowledge messages. For unreliable communication, rateless codes can be used to increase the delivery ratio of the packets by sending redundant packets from the parity part of the coded message.

7.2 A Simple Analysis on Expected Number of Transmissions

Packet drop rate was not considered as a main concern until this chapter. In this section, two simple analyses on a simplified network model are done. SWIM tries to utilize the WMA by the usage of the Greedy Set Cover algorithm, it is still not known if the WMA has a cost in terms of throughput. The purpose of the first analysis is to get an idea about what the effect of using more links on throughput is.

The second analysis studies the case in which the rateless codes are used. The analysis will not only give an idea about the effect of rateless codes on throughput compared to the uncoded case, but it may also give an insight on how to use the rateless codes together with SWIM as a future direction.

In the simplified network model, all links have a constant packet success probability. The packet success probability is assumed to be equal to p for all links. This is not a realistic assumption, but it is sufficient to give an idea about the overall performance. In both cases, the expected number of transmissions (ETX) is analyzed.

7.2.1 ETX in the uncoded case

The expected number of transmissions needed to deliver k packets successfully in the uncoded case is of concern. For this purpose a simple analysis is done on the simple network model. We assume that there are L outgoing links to the L neighbors.

Let $\mathbb{X}_i$ be defined as the number of trials until a success event of link i. Then

$$P(\mathbb{X}_i < j) = 1 - (1 - p)^{j-1}. \tag{7.1}$$

Let $\mathbb{X}$ be defined as the number of trials until all links are successful. Assuming the links are independent

$$
\begin{aligned}
P(\mathbb{X} < j) &= P(\mathbb{X}_i < j, \forall j \in [1, L]) \\
&= (P(\mathbb{X}_i < j))^L \\
&= (1 - (1 - p)^{j-1})^L
\end{aligned}
\tag{7.2}
$$

Since the expected number of trials needed to reach all neighbors is of interest, this can be found by taking the expectation of $\mathbb{X}$.

$$
\begin{aligned}
E(\mathbb{X}) &= \sum_{j=1}^{\infty} j\,(P(\mathbb{X} < j + 1) - P(\mathbb{X} < j)) \\
&= \sum_{j=1}^{\infty} j((1 - (1 - p)^j)^L - (1 - (1 - p)^{j-1})^L)
\end{aligned}
\tag{7.3}
$$

In order to do the summation, we need to get rid of the power terms. This can be done by applying the binomial expansion. Let $q_j = (1 - p)^j$ for simplification. Observe that

$$
\begin{aligned}
& 1 - (1 - p)^j = 1 - q_j = 1 - (1 - p)^{j-1}(1 - p) \\
= & 1 - (1 - p)^{j-1} + p(1 - p)^{j-1} \\
= & 1 - q_{j-1}(1 - p)
\end{aligned}
\tag{7.4}
$$

Substituting the results in Equation 7.4 into the expectation:

$$
\begin{aligned}
E(\mathbb{X}) &= \sum_{j=1}^{\infty} j((1 - q_{j-1}(1 - p))^L - (1 - q_{j-1})^L) \\
&= \sum_{j=1}^{\infty} j \sum_{i=1}^{L} \binom{L}{i} (-q_{j-1}(1 - p))^i - (-q_{j-1})^i \\
&= \sum_{j=1}^{\infty} j \sum_{i=1}^{L} \binom{L}{i} (-q_{j-1})^i((1 - p)^i - 1)
\end{aligned}
\tag{7.5}
$$

To finish the derivation, observe the fact:

$$\sum_{j=1}^{\infty} jq_{j-1}^{i} = \sum_{j=1}^{\infty} j((1-p)^{j-1})^{i} = \frac{1}{(1-(1-p)^{i})^{2}} \tag{7.6}$$

Interchanging the order of summation and applying (7.6):

$$\begin{aligned} & \sum_{j=1}^{\infty} j \sum_{i=1}^{L} \binom{L}{i} (-q_{j-1})^{i}((1-p)^{i}-1) \\ = & \sum_{i=1}^{L} \binom{L}{i} ((1-p)^{i}-1)(-1)^{i} \sum_{j=1}^{\infty} jq_{j-1}^{i} \\ = & \sum_{i=1}^{L} \binom{L}{i} ((1-p)^{i}-1)(-1)^{i}(\frac{1}{(1-(1-p)^{i})^{2}}) \\ = & \sum_{i=1}^{L} \binom{L}{i} \frac{(1-(1-p)^{i})(-1)^{i+1}}{(1-(1-p)^{i})^{2}} \\ = & \sum_{i=1}^{L} \binom{L}{i} \frac{(-1)^{i+1}}{(1-(1-p)^{i})} \end{aligned} \tag{7.7}$$

Generalizing the result for k packets, the expected number of retransmissions will be scaled by k.

$$ETX^{WMA} = k \sum_{i=1}^{L} \binom{L}{i} \frac{(-1)^{i+1}}{(1-(1-p)^{i})} \tag{7.8}$$

(7.8) can be approximated by a simpler expression. Assuming p is large, the term $(1-p)^{i}$ can be ignored for $i > 1$.

$$\begin{aligned} ETX^{WMA} &= \frac{kL}{p} + k \sum_{i=2}^{L} \binom{L}{i} \frac{(-1)^{i+1}}{(1-(1-p)^{i})} \\ &\approx \frac{kL}{p} + k \sum_{i=2}^{L} \binom{L}{i} \frac{(-1)^{i+1}}{1} \\ &= \frac{kL}{p} - kL + k + \sum_{i=0}^{L} \binom{L}{i} (-1)^{i+1} \\ &= \frac{kL}{p} - kL + k \end{aligned} \tag{7.9}$$

(7.9) shows that, when p is large, ETX increases linearly with L as expected, however the good news is that the scaling coefficient is small in particular. If the WMA was not used and all outgoing links are used by unicasting the ETX would become:

$$ETX = \frac{kL}{p} \tag{7.10}$$

The WMA has an ETX advantage of:

$$\begin{aligned} \frac{ETX - ETX^{WMA}}{ETX} &= \frac{p(L-1)}{L} \\ &= p\left(1 - \frac{1}{L}\right) \end{aligned} \tag{7.11}$$

(7.11) leads to the conclusion that the throughput and efficiency increases with increasing number of outgoing links, used by a forwarding node. In order to increase the outgoing links of forwarding nodes, the NFN needs to be decreased. This fact leads to the important conclusion that in order to increase throughput and WMA utilization, the NFN needs to be decreased.

As SWIM was shown to have a relatively small NFN as most of the significant solutions in the literature, the throughput of SWIM is expected to be better due to (7.8). Note that this effect is observed from the comparison results between SWIM and MAODV, where MAODV had a slightly better NFN and throughput, which supports this result.

7.2.2 ETX in the rateless coded case

In the previous section we analyzed the ETX for the uncoded case. The disadvantage of the uncoded case is that the reliable communication of multicast is costly due to the fact that each node has to receive the sent packet. Even if only a single link fails, the packet has to be retransmitted. Rateless codes, distribute the information to be sent, into each packet. This provides the freedom that receiving any additional packet is useful in the decoding process of the information. To see this advantage, a simple analysis is done for the ETX in the rateless coded case.

Let $\mathbb{X}_i$ be defined as the number of transmissions until k packets are successful on link i. Then

$$P(\mathbb{X}_i < j) = \sum_{m=k}^{j-1} \binom{m-1}{k-1} p^k (1-p)^{m-k} \tag{7.12}$$

We assume that there are L outgoing links to the L neighbors. Let $\mathbb{X}$ be defined as the number of transmissions until k packets are successful for all outgoing links. We assume that the link success probabilities are independent.

$$\begin{aligned} P(\mathbb{X} < j) &= P(\mathbb{X}_i < j, \forall i \in [1, L]) \\ &= (P(\mathbb{X}_i < j))^L \\ &= \left(\sum_{m=k}^{j-1} \binom{m-1}{k-1} p^k (1-p)^{m-k} \right)^L \end{aligned} \tag{7.13}$$

The ETX needed to send k successful packets on all outgoing links is of interest. This can be found by taking the expectation of $\mathbb{X}$.

$$\begin{aligned} E(\mathbb{X}) &= \sum_{j=1}^{\infty} j(P(\mathbb{X} < j+1) - P(\mathbb{X} < j)) \\ ETX^{RC} &= \sum_{j=k+1}^{\infty} j \left(\left(\sum_{m=k}^{j} \binom{m-1}{k-1} p^k (1-p)^{m-k} \right)^L - \left(\sum_{m=k}^{j-1} \binom{m-1}{k-1} p^k (1-p)^{m-k} \right)^L \right) \end{aligned} \tag{7.14}$$

In order to see the advantage of rateless codes over the uncoded case, (7.7) and (7.14) have been numerically calculated and plotted in Figure 7.1.

Another important result is the change of ETX per outgoing link with respect to the number of outgoing links. This numerical result is shown on Figure 7.2

7.2.3 Comparison of ETX in the uncoded and rateless coded cases

Figure 7.1, plotted in the previous section leads to two major conclusions.

The throughput is increased: In the case of reliable communication, instead of retransmitting a packet until it is received by each neighbor node, the transmitter is only responsible of delivering the required number of packets to each destination. This property of

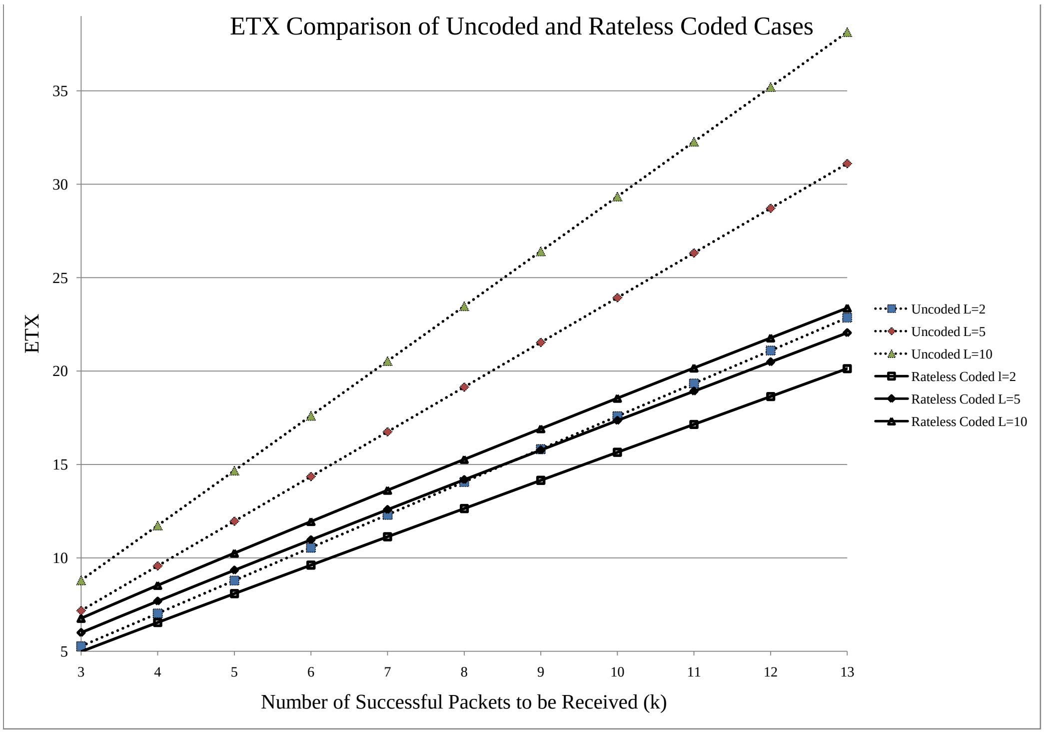


Figure 7.1: Uncoded and Rateless Coded Case Comparison. Various numerical results of the ETX for different number of links($l = 2, 5, 10$) vs. different number of successful packets($k = 3 \rightarrow 13$) have been plotted for link success rate($p = 0.7$). Rateless Coded case performs better than the uncoded case and the performance margin widens as the number of links used is increased. Rateless Coded case has a smaller slope, thus is closer to being constant than the uncoded case.

rateless codes decreases the number of retransmissions. It also simplifies the structure of the acknowledge packets thus decreases their lengths.

In the case of unreliable communication, a lost packet can be decoded from redundant transmissions, if the requirements are satisfied.

The cost of utilizing the WMA can be ignored: In the uncoded case, there was a small constant increase in cost with increasing number of outgoing links. In the Rateless Coded case, the margin between different number of links is even smaller. This means that, with increasing number of outgoing links by a forwarding node, the throughput and efficiency is increased. Increasing the number of outgoing links used by a forwarding node is realized by decreasing the NFN. This property is to the advantage of SWIM, since in the covering part, each node tries to select the neighbor which can reach the maximum number of destinations, thus using the maximum number of outgoing links

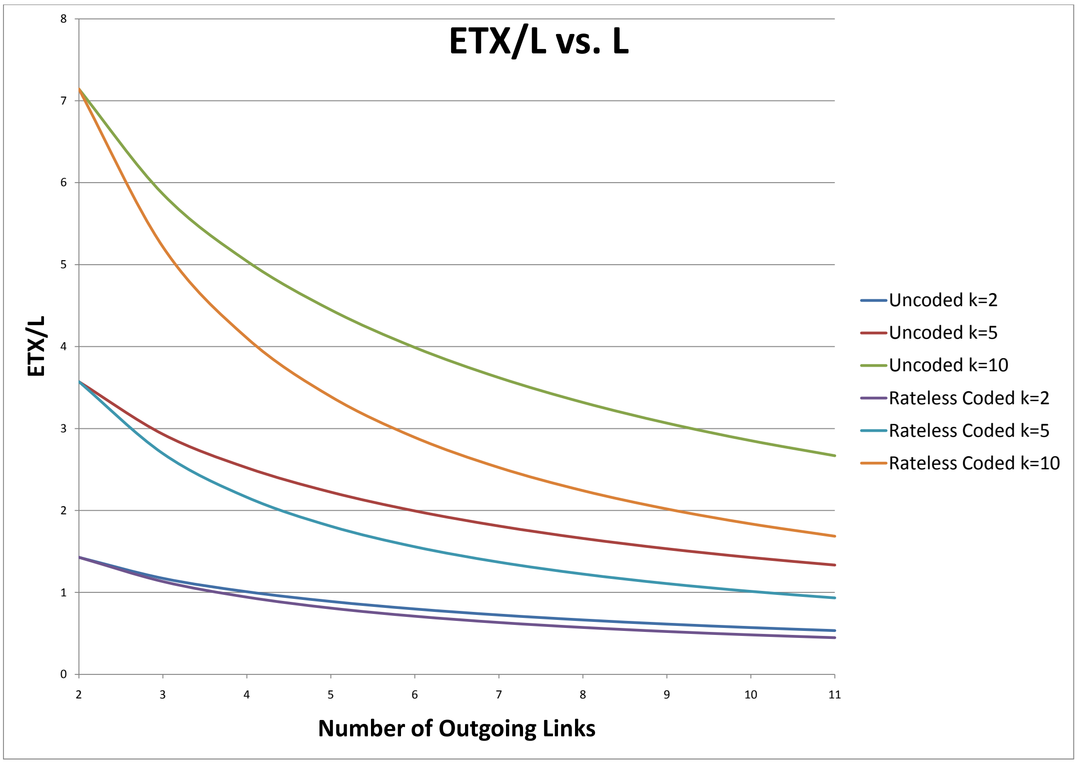


Figure 7.2: ETX per Link for Uncoded and Rateless Coded Cases. The ETX per link has been plotted for various number of successful packets($k = 2, 5, 10$) vs. different number of outgoing links($L = 2 \rightarrow 11$). Rateless Coded case performs better in terms of ETX. Also for both cases, ETX per link decreases with the number of outgoing links. This means that using more links, decreases the ETX per link.

and tries to minimize the NFN.

As a conclusion, rateless coding can be used for both reliable and unreliable communication with an ignorable overhead. It not only utilizes the WMA better which was the starting point of SWIM, but is expected to work better with the Greedy Set Cover algorithm used in SWIM which would be a future research direction.

CHAPTER 8

CONCLUSIONS AND FUTURE DIRECTIONS

This thesis presents a novel solution for the area of wireless multicast routing. The proposed solution, SWIM, is a distributed multicast-tree algorithm and can be used as an implementable solution since it achieves an average computational complexity of only $O(N^2)$ in the number of nodes N. In wireless networks, the number of forwarding nodes is an important aspect for many performance criteria. It directly effects energy efficiency and has a large impact on throughput. Despite its low complexity, SWIM is depth-optimal and obtains a low number of forwarding nodes. The reason for the good performance with low complexity is that SWIM uses a Greedy Set Cover algorithm at each level to reach the destinations. Moreover the number of messages to construct the tree and thus the protocol overhead are minimal.

The ability to use multiple-sources, which is a very important criteria for Wireless Access Networks and the ability to generate alternative paths for redundancy, throughput increase and fault tolerance makes SWIM a good candidate for Wireless Mesh Networks. Because SWIM also creates a multicast mesh as a side product. The formed multicast mesh is also depth optimal.

SWIM also has the ability to work in dynamic network conditions. SWIM can be integrated into any system with a previous unicast proactive routing algorithm or any new unicast proactive routing algorithm can be selected. SWIM uses this information to construct the solution trees. The update rate of tree construction and information distribution can be selected according to the needs of the traffic, the bandwidth of the network and the mobility levels of the nodes.

In summary an implementable solution that can answer the problems of relieving the bottlenecks at the gateways or increase the utilization of the available bandwidth is proposed.

Future directions from here would be the weighted case, where the objective when constructing the routing tree can be the link weights, rather than hop-count. Another direction would be to theoretically bound the capacity of the solution tree. This aspect is important to show the gain on capacity when compared to the unicast alternative. There is also a need for a repair and maintenance algorithm for the static network case, since this is a requirement for an implementable solution. The analysis of SWIM, when used with rateless codes is an important matter of subject. The leveling phase can be modified in order to take advantage of rateless codes. A final direction would be to address the reliable multicast case. This is partially related with the rateless code direction, since rateless codes can be used to increase the throughput and efficiency of the ARQ, which would be a great solution for the reliable transmission problem.

REFERENCES

[1] A. S. Akyurek and E. Uysal-Biyikoglu. A low complexity distributed algorithm for computing minimum-depth multicast trees in wireless networks. In *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pages 1918–1923, 31 2010-nov. 3 2010.

[2] A. Sinan Akyurek and Elif Uysal-Biyikoglu. Kablosuz ağlarda Çoğagönderim için yeni bir yol atama algoritması. In *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, pages 1–4, april 2008.

[3] A. Sinan Akyurek and Elif Uysal-Biyikoglu. A depth-optimal low-complexity distributed wireless multicast algorithm. *The Computer Journal*, 2011.

[4] Yair Amir, Claudiu Danilov, Raluca Musaloiu-Elefteri, and Nilo Rivera. An inter-domain routing protocol for multi-homed wireless mesh networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–10, june 2007.

[5] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, SOSP '01, pages 131–145, New York, NY, USA, 2001. ACM.

[6] B. Aoun, R. Boutaba, Youssef Iraqi, and G. Kenward. Gateway placement optimization in wireless mesh networks with qos constraints. *Selected Areas in Communications, IEEE Journal on*, 24(11):2127–2136, november 2006.

[7] Baruch Awerbuch and Yossi Azar. Competitive multicast routing. *Wireless Networks*, pages 11–4, 1995.

[8] A. Banchs, W. Effelsberg, C. Tschudin, and V. Turau. Multicasting multimedia streams with active networks. In *Local Computer Networks, 1998. LCN '98. Proceedings., 23rd Annual Conference on*, pages 150–159, oct 1998.

[9] C. de Morais Cordeiro, H. Gossain, and D.P. Agrawal. Multicast over wireless mobile ad hoc networks: present and future directions. *Network, IEEE*, 17(1):52–59, jan/feb 2003.

[10] Ömer Egecioglu, Teofilo F. Gonzalez, and Teo Lo F. Gonzalez. Minimum-energy broadcast in simple graphs with limited node power. In *in Proc. IASTED Int. Conf. on Parallel and Distributed Computing and Systems*, pages 334–338, 2001.

[11] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, April 1998.

[12] Odd Inge Hillestad, Andrew Perkis, Vasken Genc, Seán Murphy, and John Murphy. Delivery of on-demand video services in rural areas via ieee 802.16 broadband wireless access networks. In *Proceedings of the 2nd ACM international workshop on Wireless*

multimedia networking and performance modeling, WMuNeP '06, pages 43–52, New York, NY, USA, 2006. ACM.

[13] Weijia Jia, Leung Cheng, and Gaochao Xu. Efficient multicast routing algorithms on mesh networks. In *Algorithms and Architectures for Parallel Processing, 2002. Proceedings. Fifth International Conference on*, pages 110–117, 2002.

[14] David S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, STOC '73, pages 38–49, New York, NY, USA, 1973. ACM.

[15] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[16] R. Langar, N. Bouabdallah, and R. Boutaba. A comprehensive analysis of mobility management in mpls-based wireless access networks. *Networking, IEEE/ACM Transactions on*, 16(4):918–931, aug. 2008.

[17] Sung Ju Lee, William Su, and Mario Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *Mob. Netw. Appl.*, 7:441–453, December 2002.

[18] Fulu Li and L. Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pages 193–202, 2001.

[19] Michael Luby. Lt codes. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, pages 271–, Washington, DC, USA, 2002. IEEE Computer Society.

[20] Petar Maymounkov. Online codes (extended abstract), 2002.

[21] Manki Min, Oleg Prokopyev, and Panos Pardalos. Optimal solutions to minimum total energy broadcasting problem in wireless ad hoc networks. *Journal of Combinatorial Optimization*, 11:59–69, 2006. 10.1007/s10878-006-5977-8.

[22] A. Penttinen. Minimum cost multicast trees in ad hoc networks. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 8, pages 3676–3681, june 2006.

[23] Nazanin Rahnavard, Badri N. Vellambi, and Faramarz Fekri. Rateless codes with unequal error protection property. *IEEE Transactions on Information Theory*, 53:1521–1532, 2007.

[24] Elizabeth M. Royer and Charles E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, pages 207–218, New York, NY, USA, 1999. ACM.

[25] Amin Shokrollahi. Raptor codes. *IEEE/ACM Trans. Netw.*, 14:2551–2567, June 2006.

[26] B. Tavli and W.B. Heinzelman. Mh-trace: multihop time reservation using adaptive control for energy efficiency. *Selected Areas in Communications, IEEE Journal on*, 22(5):942–953, june 2004.

[27] B. Tavli and W.B. Heinzelman. Mc-trace: multicasting through time reservation using adaptive control for energy efficiency. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 2672–2678 Vol. 4, oct. 2005.

[28] Mario Čagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 172–182, New York, NY, USA, 2002. ACM.

[29] G. Venkat Raju, T. Bheemarjuna Reddy, and C. Siva Ram Murthy. A near optimal localized heuristic for voice multicasting over ad hoc wireless networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 1648–1653, june 2007.

[30] Xudong Wang, Edward Knightly, Marco Conti, and Anthony Ephremides. Guest editorial: A special issue on "wireless mesh networks". *Ad Hoc Netw.*, 5:649–651, August 2007.

[31] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 585–594 vol.2, 2000.

[32] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides. The energy efficiency of distributed algorithms for broadcasting in ad hoc networks. In *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, volume 2, pages 499–503 vol.2, oct. 2002.

[33] Jennifer Wong, Giacamino Veltri, and Miodrag Potkonjak. Energy-efficient data multicast in multi-hop wireless networks. In Ramesh Karri and David Goodman, editors, *System-Level Power Optimization for Wireless Multimedia Communication*, pages 69–85. Springer US, 2002. 10.1007/0-306-47720-3_5.

[34] Jianliang Zheng and Myung J. Lee. A resource-efficient and scalable wireless mesh routing protocol. *Ad Hoc Netw.*, 5:704–718, August 2007.