ENHANCED RESOURCE-AWARE VIDEO MULTICASTING VIA ACCESS GATEWAYS IN MULTI-HOP WIRELESS MESH NETWORKS

by

Gürkan Rakanoğlu

B.S., Electrical and Electronics Engineering, Ege University, 2007

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2012

ENHANCED RESOURCE-AWARE VIDEO MULTICASTING VIA ACCESS GATEWAYS IN MULTI-HOP WIRELESS MESH NETWORKS

APPROVED BY:

Prof. Emin Anarım (Thesis Supervisor)	. . . .
Assist. Prof. Mutlu Koca	. . . .
Assoc. Prof. Fatih Alagöz	. . . .

DATE OF APPROVAL: 09.05.2012

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my advisor, Prof. Emin Anarım, whose encouragement, guidance and support from the initial to the final level enabled me to develop a clear understanding of the subject.

This thesis would not have been possible without the precious help and support of Tolga Kurt. He has contributed to every stage of the thesis work with his invaluable advices and guidance.

I am grateful to all of my lecturers, who has equipped us with invaluable technical knowledge. Particularly, I would like to thank Prof. Bülent Sankur, Prof. Fikret Gürgen, Assoc. Prof. Fatih Alagöz, Assoc. Prof. Kıvanç Mıhçak, Assist. Prof. Mutlu Koca, Assist. Prof. Kerem Harmancı for their teachings. Especially, I would also like to show my gratitude to my Bachelor of Science final term project advisor Assist. Prof. Radosveta Sokullu and Assoc. Prof. Musa Alci who has introduced me to C++ programming language.

My family has been extremely supportive in many ways all my life and they deserve a special mention. Especially, I would like to thank to my fiancee for her patience, encouragement and confidence about my capability.

I would also like to thank TÜBİTAK for supporting my whole Master of Science work with the BIDEB 2210 scholarship programme.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

ABSTRACT

ENHANCED RESOURCE-AWARE VIDEO MULTICASTING VIA ACCESS GATEWAYS IN MULTI-HOP WIRELESS MESH NETWORKS

Wireless mesh networks (WMN) are the communication networks made up of radio nodes organized in a mesh topology. In recent years there are many applications like video survelliance, video conferencing that have been implemented over these kind of networks. As wireless mesh networks become more popular and their size and complexity continues to grow, mesh networks that contain multiple hops become increasingly vulnerable to problems such as bandwidth degredation, radio interference and network latency. In extreme cases, where voice and video applications are heavily at work, a complete connection loss can occur when latency and RF interference reach unacceptable levels. This thesis work studies video multicasting in large scale areas using wireless mesh networks. The focus is on the use of internet access gateways that allow a choice of alternative routes to avoid potentially lengthy multi-hop wireless paths with low capacity. We mainly get inspired from the set of heuristic-based algorithms that were described at the paper named “Resource aware video multicasting via Access Gateways in Wireless Mesh Networks”: the two-tier integrated architecture algorithm (TIA), the weighted gateway uploading algorithm (WGU) and the link-controlled routing tree algorithm (LCRT). Two enhancements were developed for both TIA and LCRT, and afterwards performance analysis have been made to obtain the contribution of the new scheme.

ÖZET

ÇOKLU SEKMELİ KABLOSUZ ÖRGÜ AĞLARINDA ERİŞİM GİRİŞ KAPISI VASITASIYLA GELİŞTİRİLMİŞ KAYNAKTAN HABERDAR VİDEO ÇOKLU GÖNDERİMİ

Kablosuz Örgü Ağları örgü şeklinde birbirlerine bağlanmış radyo düğümlerinden oluşan haberleşme ağlarıdır. Son yıllarda bu tip ağlar üzerinde görüntülü denetleme, video konferans gibi pek çok uygulama gerçeklenmiştir. Kablosuz örgü ağları daha popüler hale geldiği ve boyutları ve karmaşıklığı artmaya devam ettiği için çoklu sekmesi olan örgü ağları, bant genişliği bozulması, radyo karışma ve ağ gecikmesi problemleri gibi sorunlara karşı daha hassas hale gelmiştir. Öyle ki ses ve video uygulamalarının yoğun olarak çalıştığı uç örneklerde gecikme ve RF karışımı kabul edilemez seviyelere ulaştığında tam bağlantı kopması meydana gelebilmektedir. Tez çalışmamızda geniş ölçekli alanlarda kablosuz örgü ağları kullanılarak video çoklu-gönderimi üzerinde çalışılmıştır. Çalışmanın odağı düşük kapasiteli muhtemel çoklu sekmeli kablosuz yollardan sakınmak için alternatif rota seçeneği sunan internet erişim giriş kapılarını kullanmaktır. Çalışmada genel olarak “Kablosuz Örgü Ağlarında Erişim Giriş Kapısı Vasıtasıyla Kaynaktan Haberdar Video Çoklu Gönderimi” makalesindeki deneyimsel algoritma kümesinden ilham alınmıştır; iki aşamalı tümleştirilmiş yapı algoritması (TIA), ağırlıklandırılmış erişim giriş kapısı yükleme algoritması (WGU), bağlantı kontrollü yönlendirme ağacı algoritması (LCRT). TIA ve LCRT algoritmalarının her ikisi için de ikişer algoritma iyileştirmesi geliştirilmiş ve sonrasında yeni yapının katkılarını tespit edebilmek için performans analizi yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{C}_i$	Current available capacity of the ith gateway
$D(p)$	Delay value of the pth packet
G’	Gateways that replied to the gateway search request of the source node
$\mathbf{h}_i$	Wireless hop distance of the ith gateway to s
i	Index of gateway
K	Maximum permitted hop distance threshold
k	Number of wireless hops from s to its uploading gateway
L	Largest node level
l	Node level
m	Number of members who are in the $(l + 1)$-th level and covered by the selected forwarder from the uncovered member set
n	Number of nodes
n_s	Number of multicast sources
n_d	Number of destinations
$P_{x,y}$	Probability to change the state from x to y
p	Index of packet
s	Source node
T	Waiting acknowledgement period for selecting a new area gateway
T’	Calculating metric period for weighted gateway uploading algorithm
uml	Number of uncovered members in the $(l + 1)$th level
V	Video stream
x	State of one
y	State of zero
δ_D	Delay cost

δ_J	Delay jitter cost
δ_T	Throughput cost

LIST OF ACRONYMS/ABBREVIATIONS

ACK	Acknowledgement
AG	Area Gateway
CCI	Co-Channel Interference
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
CTTF	Cumulative Transmission Time Fraction
DIFS	Distributed Inter Frame Space
DSL	Digital Subscriber Line
FIFO	First In First Out
GUI	Graphical User Interface
GW	Gateway
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LCRT	Link Controlled Routing Tree
LCRT-ENH-1	Link Controlled Routing Tree Enhancement 1
LCRT-ENH-2	Link Controlled Routing Tree Enhancement 2
MAC	Medium Access Control
NAV	Network Allocation Vector
NLoS	Non Line of Sight
QoS	Quality of Service
RCAM	Rate and Contention Aware Multicast
RF	Radio Frequency
RTS	Request To Send
RX	Receiver
SIFS	Short Inter Frame Space

TCP	Transport Control Protocol
TIA	Two-Tier Integrated Architecture
TIA-ENH-1	Two-Tier Integrated Architecture Enhancement 1
TIA-ENH-2	Two-Tier Integrated Architecture Enhancement 2
TTL	Time To Live
TX	Transmitter
UDP	User Datagram Protocol
UGW	Uploading Gateway
WBA	Wireless Broadcast Advantage
WGU	Weighted Gateway Uploading
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network

1. INTRODUCTION

1.1. Wireless Mesh Networks

A wireless mesh network (WMN) is an emerging technology to provide open access to the internet with high bandwidth and low cost while covering a wide area without wiring. Compared to ad-hoc networks, traditionally focusing on military and specialized civilian applications, WMNs can be thought as the generalized one for fulfilling the actual user requirements. Accordingly, WMNs have become one of the most prospective candidates for the future internet technology [1].

Wireless mesh networks can easily, effectively and wirelessly connect entire cities using inexpensive, existing technology. Traditional networks rely on a small number of wired access points or wireless hotspots to connect users. In a wireless mesh network, the network connection is spread out among dozens or even hundreds of wireless mesh nodes that “talk” to each other to share the network connection across a large area [2].

WMNs consist of two types of nodes: mesh routers and mesh clients. Other than the routing capability for gateway/repeater functions as in a conventional wireless router, a wireless mesh router contains additional routing functions to support mesh networking. To further improve the flexibility of mesh networking, a mesh router is usually equipped with multiple wireless interfaces built on either the same or different wireless access technologies. Compared with a conventional wireless router, a wireless mesh router can achieve the same coverage with much lower transmission power through multi-hop communications. Optionally, the medium access control (MAC) protocol in a mesh router is enhanced with better scalability in a multi-hop mesh environment [3].

The architecture of WMNs can be classified into three types:

(i) Infrastructure/Backbone WMNs
(ii) Client WMNs

(iii) Hybrid WMNs

1.1.1. Types of WMNs

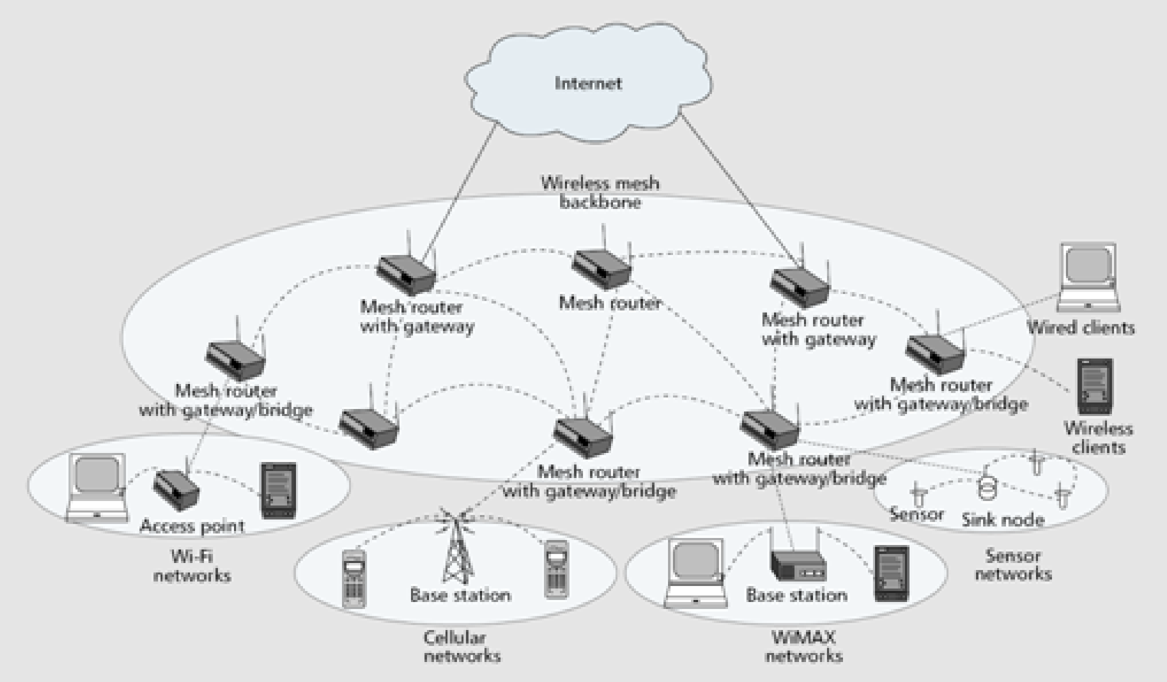


Figure 1.1. Infrastructure/backbone WMNs.

<u>1.1.1.1. Infrastructure/Backbone WMNs.</u> In this architecture, mesh routers form an infrastructure for clients, as shown in Figure 1.1, where dashed and solid lines indicate wireless and wired links, respectively. The WMN infrastructure/backbone can be built using various types of radio technologies, in addition to the mostly used IEEE 802.11 technologies. The mesh routers form a mesh of self-configuring, self-healing links among themselves. With gateway functionality, mesh routers can be connected to the internet. This approach, also referred to as infrastructure meshing, provides a backbone for conventional clients and enables integration of WMNs with existing wireless networks, through gateway/bridge functionalities in mesh routers. Conventional clients with an Ethernet interface can be connected to mesh routers via Ethernet links. For conventional clients with the same radio technologies as mesh routers, they can directly communicate with mesh routers. If different radio technologies are used, clients must communicate with their base stations that have Ethernet connections to mesh

routers [4].

1.1.1.2. Client WMNs. Client meshing provides peer-to-peer networks among client devices. In this type of architecture, client nodes constitute the actual network to perform routing and configuration functionalities as well as providing end-user applications to customers. Hence, a mesh router is not required for these types of networks. Client WMNs are usually formed using one type of radios on devices. Thus, a Client WMN is actually the same as a conventional ad hoc network. However, the requirements on end-user devices is increased when compared to infrastructure meshing, since in Client WMNs the end-users must perform additional functions such as routing and self-configuration [4].

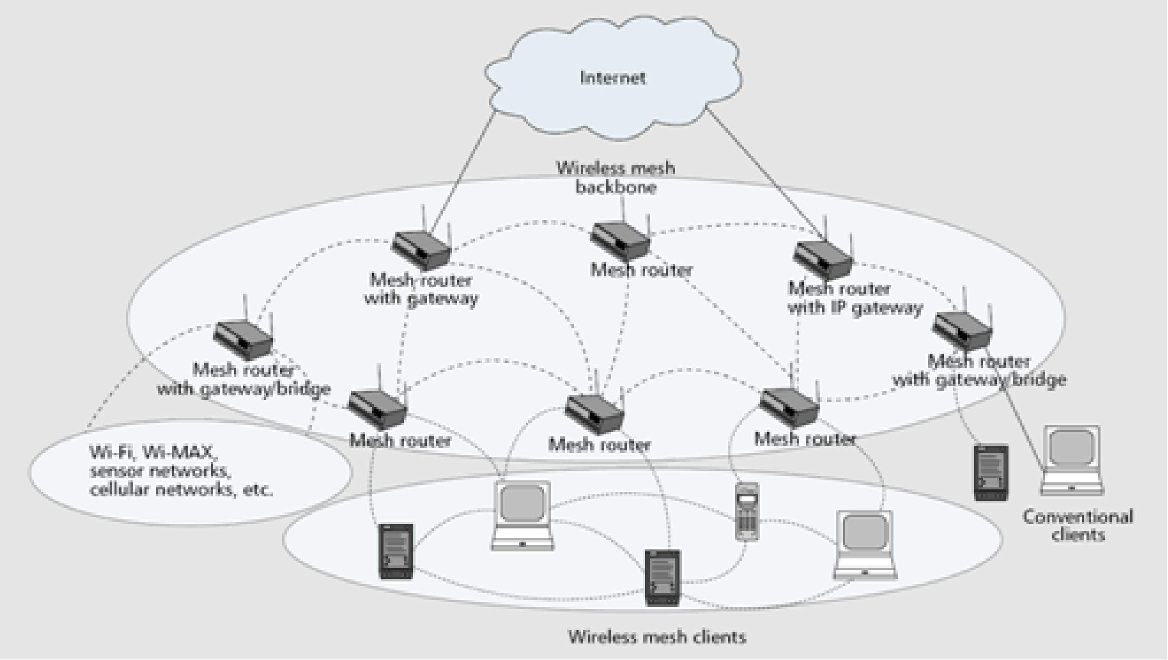


Figure 1.2. Hybrid WMNs.

1.1.1.3. Hybrid WMNs. This architecture is the combination of infrastructure and client meshing, as shown in Figure 1.2. Mesh clients can access the network through mesh routers as well as directly meshing with other mesh clients. While the infrastructure provides connectivity to other networks such as the Internet, Wi-Fi, WiMAX, cellular, and sensor networks, the routing capabilities of clients provide improved connectivity and coverage inside WMNs [4].

1.1.2. How Wireless Mesh Networks Work

Mesh nodes are small radio transmitters that function in the same way as a wireless router. Nodes use the common WiFi standards known as 802.11a, b and g to communicate wirelessly with users, and, more importantly, with each other [2].

Nodes are programmed with software that tells them how to interact within the larger network. Information travels across the network from point A to point B by hopping wirelessly from one mesh node to the next. The nodes automatically choose the quickest and safest path in a process known as dynamic routing [2].

In a wireless mesh network, only one node needs to be physically wired to a network connection like a DSL Internet modem. That one wired node then shares its Internet connection wirelessly with all other nodes in its vicinity. Those nodes then share the connection wirelessly with the nodes closest to them. The more nodes, the further the connection spreads, creating a wireless “cloud of connectivity” that can serve a small office or a city of millions [2].

The advantages of wireless mesh networks are [2]:

- Using fewer wires means it costs less to set up a network, particularly for large areas of coverage.
- The more nodes you install, the bigger and faster your wireless network becomes.
- They rely on the same WiFi standards (802.11a, b and g) already in place for most wireless networks.
- They are convenient where Ethernet wall connections are lacking – for instance, in outdoor concert venues, warehouses or transportation settings.
- They are useful for Non-Line-of-Sight (NLoS) network configurations where wireless signals are intermittently blocked. For example, in an amusement park a Ferris wheel occasionally blocks the signal from a wireless access point. If there

are dozens or hundreds of other nodes around, the mesh network will adjust to find a clear signal.

- Mesh networks are “self configuring”; the network automatically incorporates a new node into the existing structure without needing any adjustments by a network administrator.
- Mesh networks are “self healing”, since the network automatically finds the fastest and most reliable paths to send data, even if nodes are blocked or lose their signal.
- Wireless mesh configurations allow local networks to run faster, because local packets don’t have to travel back to a central server.
- Wireless mesh nodes are easy to install and uninstall, making the network extremely adaptable and expandable as more or less coverage is needed.

1.2. Multicast Video Delivery over Wireless Networks

In multicast video delivery, the same video content is streamed to a group of interested receivers via multicast trees. This can be achieved either at the network layer via IP multicast, or by application-layer relaying in a peer-to-peer fashion. With the advance of wireless networking technologies, video multicast over wireless mesh networks is also gaining popularity, for potential applications such as broadband multimedia content sharing within a residential community or among cell phone users [5].

Multimedia transport typically requires stringent QoS metrics (bandwidth and delay and jitter guarantees).However, in addition to unreliable wireless channel effects, it is very hard to maintain an end-to-end route which is both stable and has enough bandwidth in an ad hoc network. The rapid growth of wireless communications and networking protocols will ultimately bring video to our lives anytime, anywhere, and on any device [6].

Video streaming over wireless networks is compelling for many applications, and an increasing number of systems are being deployed. Video streaming of news and entertainment clips to mobile phones are now widely available. For surveillance applications, cameras can be flexibly and cheaply installed, if a wireless network provides connectivity. A wireless local area network (WLAN) might connect various audiovisual entertainment devices in a home. Last, but not least, in search-and-rescue operations, real-time audiovisual communication over wireless ad-hoc networks can save lives [7].

As is well known, video streaming is very different from data communication due to the inherent delay constraints, as late arriving data is not useful to the video decoder. In fact, it is better to drop such data at the sender rather than attempt to send it after the deadline has passed. For this reason, TCP/IP, which is designed to deliver data reliably (but asynchronously) and works very well for data communication, is not always the best solution for real-time video streaming [8].

Additionally, interesting technical challenges arise when the unpredictable nature of the wireless radio channel meets the requirements of high data rate and low latency for video transport [7].

1.2.1. Challenges of Multicast Video Delivery over Wireless Networks

Wireless links introduce bottlenecks for a number of reasons:

First, communication over a wireless channel is simply not able to achieve the same quality (throughput, error rate, etc.) as its wired counterpart, which reduces the quality of the multimedia content that can be delivered.

Second, in a mobile environment the channel conditions can change rapidly due to changing distance between the stations (user mobility), Rayleigh fading, and interference. Since multimedia streaming applications must deliver their content in real time, they are very sensitive to jitter in packet delivery caused by retransmissions in the underlying transport protocols.

Third, multimedia streaming may be done over a shared medium like 802.11 and interfere with other users who are, for instance, downloading files [9].

1.3. Multicast Video Delivery Researches for Wireless Networks

Research in the area of video multicasting using wireless mesh networks can be classified as being either intra-mesh, where the focus is on the optimisation of wireless links and interfaces, or integrated where the use of Internet gateways is assumed [10].

Intra-mesh video multicasting utilizes modern wireless techniques such as multiple rate transmission, multiple channels, wireless broadcast advantage, etc. B. Liu *et al.* [11] proposed the Rate and Contention Aware Multicast (RCAM) scheme that exploits link-rate diversity to construct a multicast forwarding tree, based on the link transmission rates and the associated congestion load expressed via a cumulative transmission time fraction (CTTF) metric. For efficient wireless broadcasting, C. Chou *et al.* [12] suggested the use of a metric that optimizes the product of the link rate and the coverage area. Subsequently, Wang *et al.* [13] proposed a broadcast tree construction algorithm that reduces start-up delays, while exploiting the relationship between transmission rates and their coverage range. Apart from these papers, it is analytically well-known [14] that the per-node multicast throughput of a random multi-hop network with n nodes, n_s multicast sources and n_d destinations is fundamentally bounded by

$$O\left(min\left(1, \frac{\sqrt{n}}{n_s\sqrt{n_d.log_n}}\right)\right)$$

with a high probability, implying that the multicast throughput is a decreasing function of the size of the network [10].

Integrated wireless transmission has been studied by B. Liu *et al.* in [15]. This paper analyzes unicast flows and shows that using Internet shortcuts would significantly increase network capacity. For integrated multicasting in WMNs, P. Ruiz *et al.* [16] proposed a routing mechanism where mesh nodes connect to their “closest” gateway by the procedures described in [17]. Mesh nodes that form an “island” with prefix continuity connect to the Internet through a shared “closest” gateway. The selection of gateways is based purely on topology and fails to consider the tradeoff between the selection of a closer (i.e., less hops) but more congested gateway vs. the use of a farther, less utilized gateway, especially when the end-to-end path consists of additional intra-WMN hops. Y. Amir *et al.* [18] presented a hybrid routing protocol for multi-homed wireless mesh networks that provides uninterrupted connectivity and fast handoffs, rather than load-based multicast dissemination. In general, these papers do not consider load-balanced access via the Internet and use wired resources irrespective of the relative merits of intra-mesh vs. gatewaybased paths. In [10], a multi-step integrated procedure that makes a judicious use of Internet resources while cooperatively sharing intra-WMN wireless bandwidth for single-source video multicasting applications is presented [19].

1.4. Outline of the Thesis

This thesis addresses the problem of video multicasting via access gateways over wireless mesh networks with resource-aware video multicasting techniques. Rest of the thesis is organized as follows; motivation and problem definition of the thesis is stated at Chapter 2. Chapter 3 includes the details of resource-aware multi-gateway wireless mesh network video multicasting scheme. The description of the proposed algorithm enhancements for the same scheme mentioned at Chapter 4. At Chapter 5, the simulation model of the written simulation software is explained. The results of the simulation runs are described at Chapter 6 which also includes the performance analysis of the proposed enhancement. And finally Chapter 7 concludes the thesis.

2. MOTIVATION AND PROBLEM STATEMENT

Video streaming represents one of the fastest growing segments of traffic in the Internet today [10]. And Internet is becoming extremely reachable from anywhere thanks to the cellular networks. On the other hand, the number of wireless hot spots are growing fast to serve people much more capacity. By thinking the advantages of the wireless mesh networks over cellular networks, it seems wireless mesh could become more common. Although WMNs have advantages like low construction cost even if we create a large scale network, it is still a wireless access technology and thus video multicasting with high throughput, low latency and jitter is still a problem because of the consistently changing wireless medium conditions.

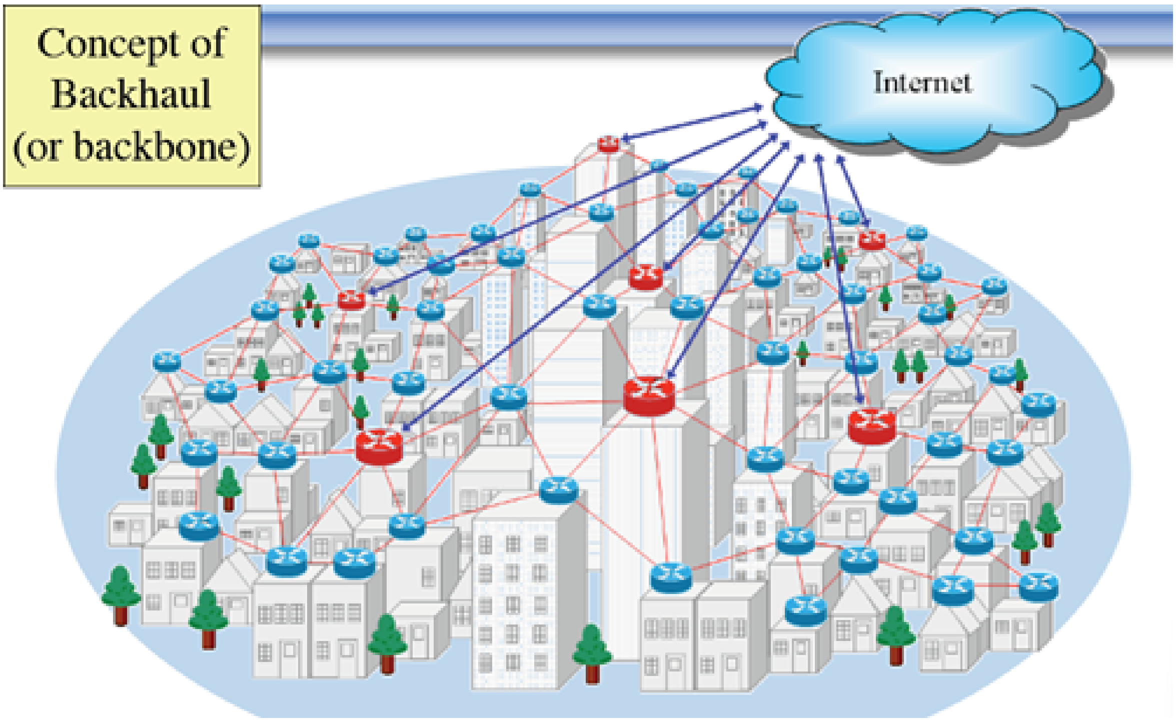


Figure 2.1. Citywide WiFi Network.

Even though, there exists several researches about video multicasting over wireless mesh networks, a few deals with the video multicasting via access gateways. However, WMNs with access gateways are becoming more common day by day with the increasing number of wireless hotspots of the Internet Service Providers (ISPs). Figure 2.1 shows a Citywide 802.11 network which enables the use of WiFi almost everywhere at the city and motivates us to deal video multicasting problem over it with its charming capacity offerings.

For this purpose, first the litreture was investigated and found that the paper named “Resource-aware Video Multicasting via Access Gateways in Wireless Mesh Networks” [10] is dealing with the same problem but some deficiencies with its algorithms. Hence, we decide to enhance the algorithms of this paper to improve its performance. In order to do this, we first write a simulation programme and then implement both the original algorithms of the paper and the enhanced ones.

3. RESOURCE-AWARE MULTI-GATEWAY WMN VIDEO MULTICASTING

At this chapter, “resource-aware multi-gateway WMN video multicasting scheme” [10], which combines a hierarchical architecture with a set of load-aware routing protocols to improve the overall video traffic capacity in large-scale WMNs, is presented in general. In our project, we used the TIA, WGU and LCRT algorithms of this paper and enhance TIA and LCRT.

Various papers, e.g. [11–13], have examined the use of multiple wireless transmission rates, multiple radio interfaces, multiple paths, and wireless broadcast advantage (WBA) to improve wireless network capacity. These approaches are intra-mesh schemes, in that they aim to support video transmission in WMNs by using wireless resources more effectively. An alternative approach is to enhance WMNs with access gateways that can provide alternative routing paths via the Internet. This allows a reduction in wireless transmission distance (the number of wireless hops traversed) and hence an improvement in residual capacity. This gateway-based approach is called as the integrated architecture and in Figure 3.1 its potential is illustrated. The intra-mesh communication from S to R experiences at least 6 hops, while the integrated path (shown by the arrow lines) traverse only 2 wireless hops (to or from the gateways) by taking advantage of an Internet shortcut among the gateways. Besides reducing the hop count, an additional set of advantages accrue from the higher bandwidth and lower loss rates that the Internet (wired) paths offer, compared to the WMN wireless links [19].

3.1. Two-Tier Integrated Architecture (TIA)

Considering the already existed Internet traffic burden, the reasonable and efficient utilization of Internet shortcuts is necessary as it can be seen in Figure 3.1. Distant users will likely benefit from employing an Internet shortcut but close users will

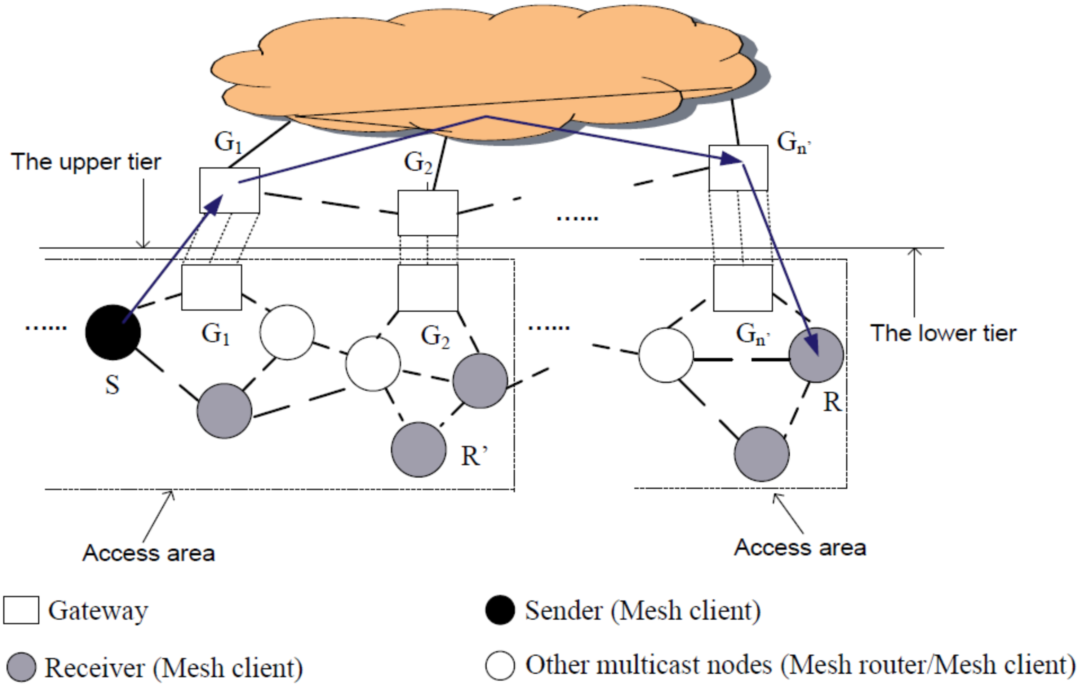


Figure 3.1. An example of two-tier integrated architecture.

likely prefer intra-mesh routing. For a group of multicasting members who distribute in large-scale areas, the path selection should consider the preference of individual members with different distances to the senders. The two-tier integrated architecture is designed to facilitate this selection of paths. As illustrated in Figure 3.1, all of the group members join in the lower tier where they are separated into access areas. In a given access area there is one area gateway that is selected from the gateways covered by that area. Area gateways form the upper tier in which group members belonging to the same access areas share the same wired links to communicate with group members in other areas [10].

3.1.1. Access Area Construction

The access area construction algorithm clusters WMN nodes into individual access areas, such that the diameter (hop distance between nodes within the access area) of each access area does not exceed a maximum permitted threshold, K. In practical systems, K is determined by the choice of radio interfaces/channels, the propagation environment, the video data rate, and the QoS desired by the application [10]. In our simulation, K is choosen to cover all of the multicast group members since we assume all of the video members are fully covered.

The first access area is constructed by the source (denoted as s) of the video stream V . The protocol works as follows. s broadcasts a AREA_CONSTRUCTION request packet that includes three fields: the area id of the constructing access area, the address of the packet sender/forwarder, and the packet's TTL. TTL is initially set to K by s. Group members who receive AREA_CONSTRUCTION set their area ids as the same one in the request packet, and then check the value of TTL. If TTL $>$ 1, they update TTL $=$ TTL $-$ 1 in AREA_CONSTRUCTION and rebroadcast the packet, thereby flooding over the K $-$ *hop* neighborhood. At the same time, these members send a JOIN REPORT packet to s, including its node type (i.e., gateways or mesh nodes), address and hop distance to s. The feedback allows s to learn about the newly joined members [10].

3.1.2. Other Gateway-Initiated Access Areas

After the first access area (i.e., the source access area) is constructed, s uses the weighted gateway uploading algorithm to select an uploading gateway from all its candidate gateways. This gateway is responsible for selecting area gateways before it multicasts V to the gateways in the group through wired connections. An area gateway acts as a virtual source injecting traffic into the mesh, and thus constructs its own access area (similar to that constructed by s).

To select area gateways, the uploading gateway runs a protocol that broadcasts SEARCH packets that include the IP address of the packet sender, V 's group id and the revised threshold of the number of wireless hops which is (K $-$ k), where k is the number of wireless hops from s to its uploading gateway; this ensures that the number of wireless hops is always restricted to K. The destinations who have V 's group id but have not joined in any access area acknowledge their states to the SEARCH sender (the uploading gateway). After a period (say T), the SEARCH sender stops waiting for acknowledgement and selects the closest acknowledged gateway as a new area gateway. This new area gateway implements two procedures successively:

(i) using the algorithm that s does to construct a new access area with (K −k) as the threshold of the number of wireless hops.
(ii) searching another area gateway by using the same way that the uploading gateway does.

At each time when a new area gateway is found, it feedbacks SEARCH_ACK to the uploading gateway along the reverse path from which it receives SEARCH. SEARCH_ACK includes the states of the gateways covered by the area gateway's access area. The above procedure continues until all of the group members join in an access area. The construction of the upper tier is completed when the uploading gateway knows about all of the group gateways [10].

In order to improve TIA, we propose the following enhancements:

(i) At [10], TIA adds only nodes which belong no access area to the new access area. Nevertheless, we thought that if a node belongs to an access area but with higher hop count to source, we should add it to the new access area. By doing that, we could be able to reduce its hop count to source which will probably lead a performance improvement.
(ii) The other improvement could be possible by extracting the gateways which are not the creator of that access area. In other words, the procedure can be expressed as:
    - For source access area the access area is created with permitting multiple gateways to join;
    - Then we run WGU algorithm to obtain the uploading gateway;
    - After acquiring uploading gateway extract the gateways other than UGW (Uploading Gateway) from the source access area and finally create other

access areas without permitting multiple gateways in them.

3.2. Weighted Gateway Uploading (WGU)

The weighted gateway uploading selects an uploading gateway through which a source can send its video streams to other access areas. The algorithm assigns each reachable gateway (i.e. a gateway in the sending access area) a weight and selects the gateway with the largest weight as the uploading gateway. In consideration of a desire for real-time and high-throughput uploading, the weight is a function of the gateways' available capacity and their distance from the video source.

To obtain the available capacity of each reachable gateway, s operates a protocol which broadcasts a gateway selection request GATEWAY_SEARCH within its access area. Each gateway who receives request of s replies with a CAPACITY_REPORT with three fields: the IP address of the gateway, the available capacity of the gateway's wireless link, and the distance of the gateway to s. The available capacity is assessed by each gateway based on the transmission rate that the gateway currently provides. After a period (say T'), s calculates the weights for the G' gateways that replied to its request by using the following equation:

$$\mathbf{w}_i = \frac{\mathbf{C}_i}{\mathbf{h}_i} \; for \; \mathbf{i} \; \epsilon \; [0, G' - 1] \tag{3.1}$$

where $\mathbf{C}_i$ is the current available capacity of the ith gateway, and $\mathbf{h}_i$ is the distance (represented by the number of wireless hops) of the ith gateway to s.

Equation 3.1 shows that the selection of uploading gateway uses a load-distance balanced metric to find a "non-busy" and close gateway. This approch helps the sender to route its video traffic around 'bottleneck' gateway nodes that might be overloaded handling traffic for other flows. It shows that the weighted gateway uploading algorithm is able to deliver video traffic between access areas with low δ_T and δ_J [10].

3.3. Link-Controlled Routing Tree (LCRT)

The link-controlled routing tree algorithm is run by the source in the sending access area or by the area gateways in other access areas to construct a routing tree that multicasts packets in that access area. The motivation of the algorithm is to decrease interference by selecting the least number of forwarding nodes that can connect all of the multicast receivers in the access area. Before describing the algorithm, the following terms should be defined.

- Node level is defined as the least number of wireless hops from the node to its closest gateway. A l-level node has at least l wireless hops to its closest gateway. As an example, in Figure 3.2, A, B, C and D are 1-level nodes because they only need one hop to their closest gateways, and E, F and G are 2-level nodes because of their 2 hops to the closest gateways.
- A node's uncovered out degree refers to the node's number of direct child members who have not found their forwarders. For example, in Figure 3.2, the uncovered out degree of G_2 is 0.

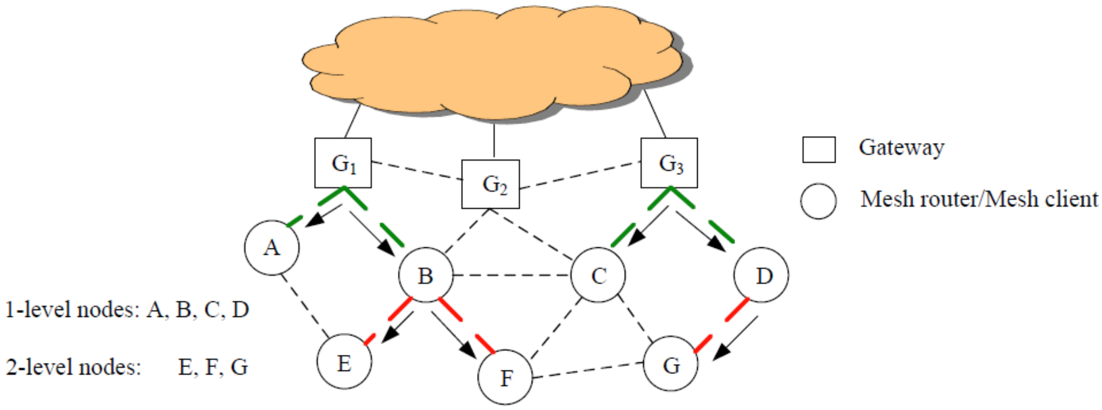


Figure 3.2. An example of the link controlled routing tree.

The link-controlled routing tree is constructed based on each node's uncovered out degree and available output capacity. The tree in the sending access area has one root (i.e., the source s), but in the non-sending access areas has multiple roots (i.e., more than one gateway covered by the access area). As we have introduced, during the access area construction, the source s of the area gateway obtains the information of the members covered by its access area, including the node level for each node. The tree construction begins at the members who have the largest levels (say L). All of the L-level members are on-tree leaf members. To search the forwarders for these leaf members, the algorithm checks all of the (L-1)-level members. At each time, a (L - 1)-level non-forwarder member who 1) has the largest uncovered out degree and the available capacity to output V and 2) is not a neighbor of any selected (L-1)-level forwarder has priority to be selected as a forwarder.When the selected (L-1)-level forwarders cover all L-level members, the algorithm stops searching for additional (L-1)-level forwarding nodes. If the (L-1)-level forwarders selected based on the two conditions cannot cover all L-level members, the algorithm will select more forwarders only based on the first condition to guarantee that each L-level member has a forwarder [10].

For example, in Figure 3.2, nodes E, F, and G are leaf nodes because they have the largest level values. Their forwarders will be selected from the 1-level members. Among the 1-level members, B and C has the same uncovered out degrees which are all 2. A node with larger available output capacity (B in this example) will be selected as a forwarder. Then, D is selected as another forwarder by the algorithm because C is B's neighbor. When the selected forwarders connect all of the members in their immediate lower level, the algorithm goes to search forwarders for the (L - 1)-level forwarders among all (L - 2)-level members. In Figure 3.2, G_1 and G_3 are selected as forwarders. When each area member has its on-tree forwarder, the link-controlled routing tree is constructed [10].

By choosing the nodes with the largest uncovered out degree as forwarders, the link-controlled routing tree algorithm exploits wireless broadcast advantage to reduce the resource usage of the network. Meanwhile, the neighborhood of sibling forwarders is greatly avoided and therefore the interference/conflict when multicasting video in parallel. It is good for real-time (i.e., low δ_D) wireless video multicasting [10].

```
Require An access area, the source s in the sending access area, or the area gateway AG in a non-sending access area.
s/AG obtains each area member's address information and its hop distance to s/AG during the access area construction;
s/AG assigns a node level to each member according to its hop distance;
s/AG sets the members that have the highest levels as leaf nodes, and $l = \mathrm{L} - 1$ (where L is the value of the highest level);
while $l > 0$ do
  while $uml \neq 0$ ($uml$ is the number of uncovered members in the $(l+1)$th level) do
    s/AG searches a $l$-level member who has a) the largest uncovered out degree and b) the least connections to the selected $l$-level forwarders, and then informs the $l$-level member to become a forwarder;
    s/AG removes m members who are in the $(l + 1)$-th level and covered by the selected forwarder from the uncovered member set, and updates $uml = uml - m$;
  end while
  $l = l - 1$;
end while
return The link-controlled routing tree in the access area;
```

Figure 3.3. Link-Controlled Routing Tree Algorithm.

On the other hand, the original algorithm could has better performance with the following modification and usage:

(i) Since we know that if an uncovered node has only one parent, we should choose it as a forwarder sooner or later. However, choosing it later may cause a degradation by the means of LCRT performance. Thus, we proposed to first set these kind of parents as forwarders and then run the LCRT algorithm as usual.

(ii) As a second enhancement for LCRT, we could change the method of LCRT usage. At [10], it is run for every access area and if the access area has multiple gateways, then a multiroot multicasting tree is being constructed. To improve the performance by avoiding interference, we suggest to again use LCRT for source access area but for the other access areas, running the LCRT algorithm as if they are all at the same access area. By doing that, we will get a multiroot multicast routing tree with maximum avoidance of interference.

4. ENHANCED RESOURCE-AWARE MULTI-GATEWAY WMN VIDEO MULTICASTING

Enhanced Resource-Aware Multi-Gateway WMN Video Multicasting consists of the TIA, WGU, LCRT algorithms of [10] with below mentioned enhancements. In order to make the "Resource-Aware Multi-Gateway WMN Video Multicasting" algorithm much more powerful in terms of handling video traffic with better QoS metrics like average throughput, delay and jitter, both TIA and LCRT algorithms were improved with two enhancements for each. The enhancements are described in brief at the following sections.

4.1. Two-Tier Integrated Architecture Algorithm Enhancements

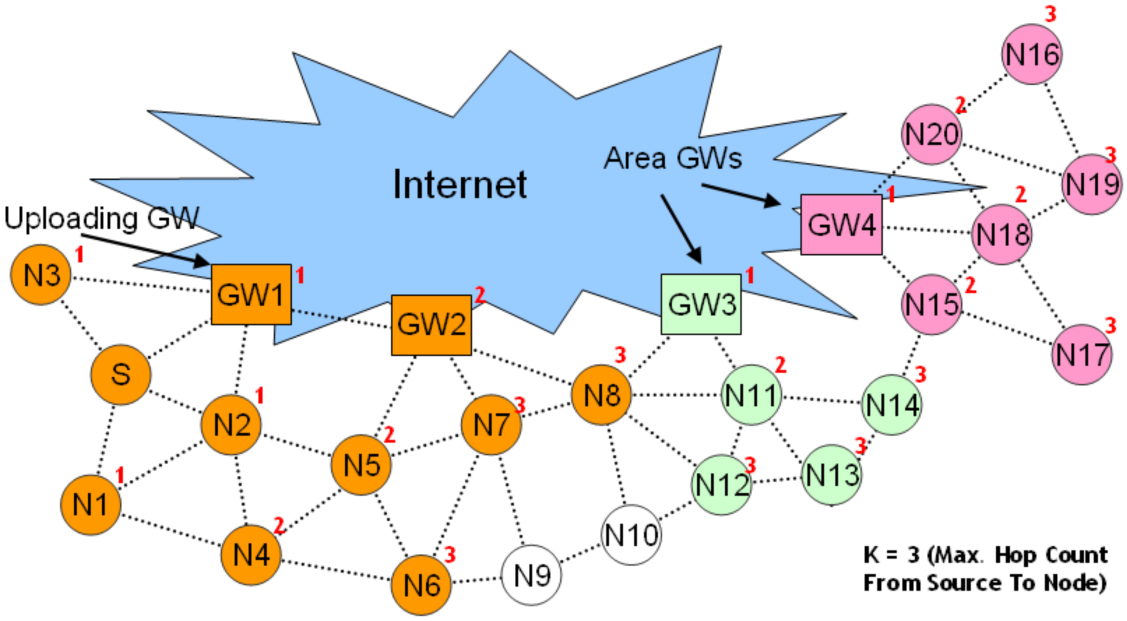


Figure 4.1. TIA implementation over an example reference network.

In order to explain the enhancements, let us first obtain an example network topology as shown in Figure 4.1. While describing the improvements, we will refer to that topology.

If we run the TIA algorithm over our reference network, we get three access areas one of which is the source access area with colour light orange, and the others with light green and levander as we could see in Figure 4.1.

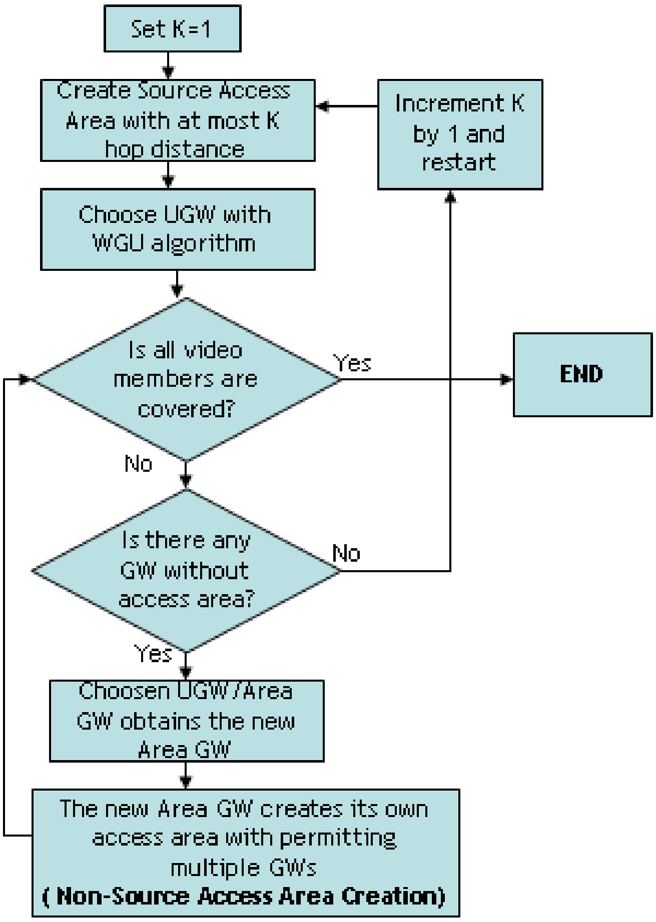


Figure 4.2. Flow Chart of TIA Algorithm.

In addition to the description of TIA at Section 3.1, the flow chart of the TIA algorithm could be found at Figure 4.4. Next subsections are for explaining the TIA algorithm improvements which are named TIA-ENH-1 for the first enhancement and TIA-ENH-2 for the second. Additionally, we named the combined version of these two algorithms as Enhanced TIA algorithm.

4.1.1. Cropping Node Enabled TIA (TIA-ENH-1)

Besides the advantages of TIA, it seems it has some deficiencies like it has no node crop capability from an access area. While constructing the new access area, it only includes nodes that are not belong to an existing access area. This behaviour would reduce performance by increasing average hop count to the source node, which lead higher average delay and lower average throughput values. To avoid this, TIA is enhanced as; even if the node already has an access area, it will check the hop count of the node if it is higher than the one offered by the constructing access area. If so, the TIA-ENH-1 algorithm crops this node and its child nodes from its previous access area and adds it to the constructing access area.

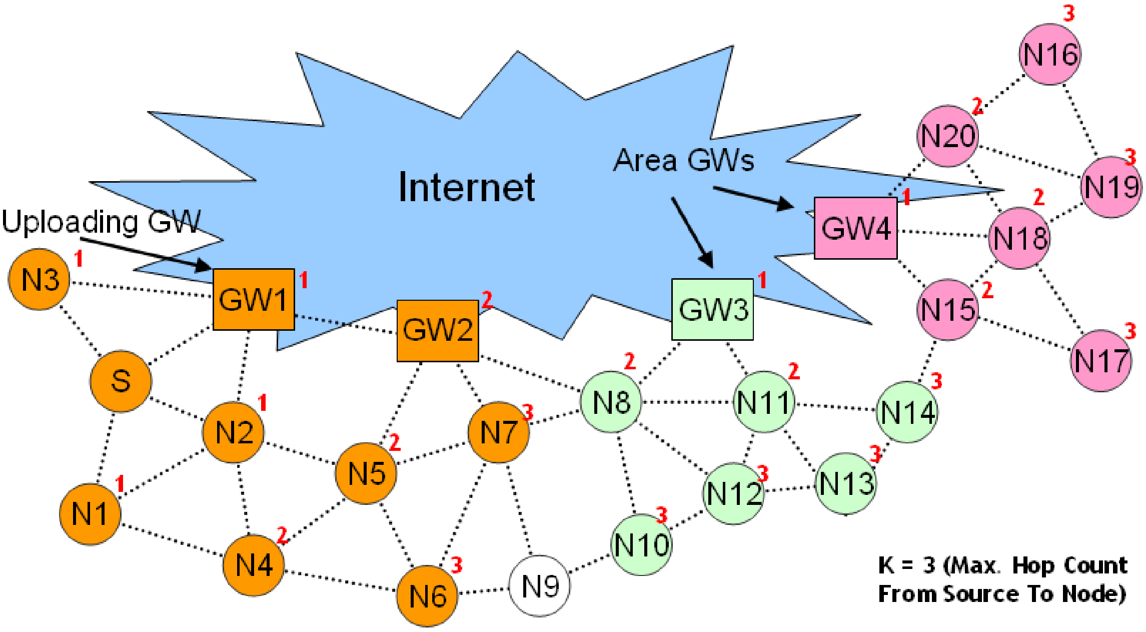


Figure 4.3. TIA-ENH-1 implementation over the reference network.

Figure 4.3 shows the access area distribution after applying the TIA-ENH-1 over the reference network at Figure 4.1. At a glance, it can easily be seen that by using the first enhancement, we could be able to include Node-10 to the multicasting tree. On the other hand, hop count of the Node-8 is reduced from 3 to 2 which leads reduction in average hop count value.

In order to implement this enhancement (TIA-ENH-1), some modifications has been made at the non-source access area creation code. Before enhancement, TIA algorithm was handling the create non-source access area operation by following the flow chart which is shown at Figure 4.4. As we can see, after selecting a candidate node to add the access area, the algorithm is just checking if the node has already joined any access area. If it has not, algorithm adds it to the access area; otherwise, the node is skipped to be added. On the other hand, at the TIA-ENH-1; if node is belonging to an existing access area, previous hop count of the node is additionally checked if it is bigger than the offered one. If so, TIA-ENH-1 crops the node from its previous access area and adds it to the new access area. To be able to do that, flow chart of the TIA-ENH-1 non-source access area creation becomes like the one at Figure 4.5.

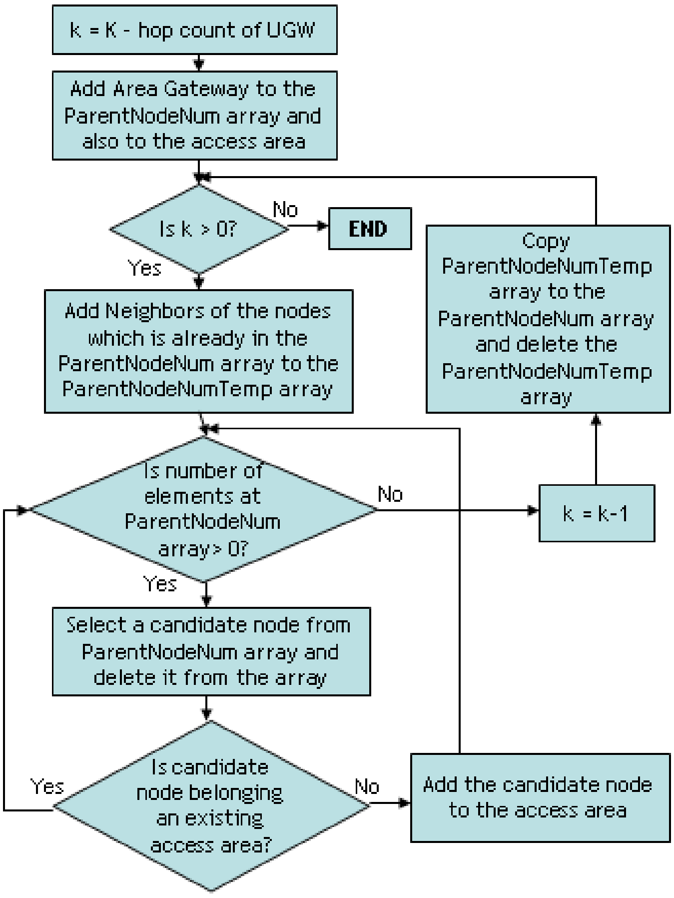


Figure 4.4. Flow Chart of TIA Non-Source Access Area Creation Algorithm.

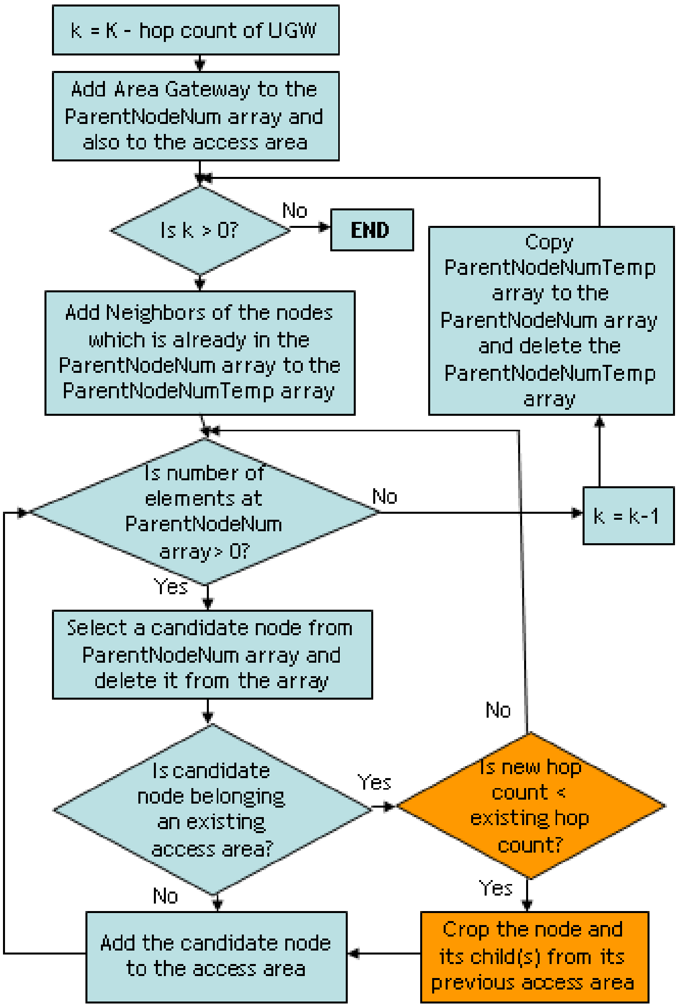


Figure 4.5. Flow Chart of TIA-ENH-1 Non-Source Access Area Creation Algorithm.

4.1.2. TIA With One Gateway Per Access Area (TIA-ENH-2)

Another drawback of the TIA algorithm seems that it locks gateways other than uploading gateway and constructing gateways (Area Gateways) within the access areas. For instance; at Figure 4.1, since GW1 is the uploading gateway, source node has just locked the GW2 and prevent it to create its own access area. However, if source released GW2 after obtaining the uploading gateway, GW2 would be able to create its own access area and could reach i.e. Node-10 without exceeding the maximum allowed hop count. This approach seems it has more capability to cover more nodes with smaller hop count values.

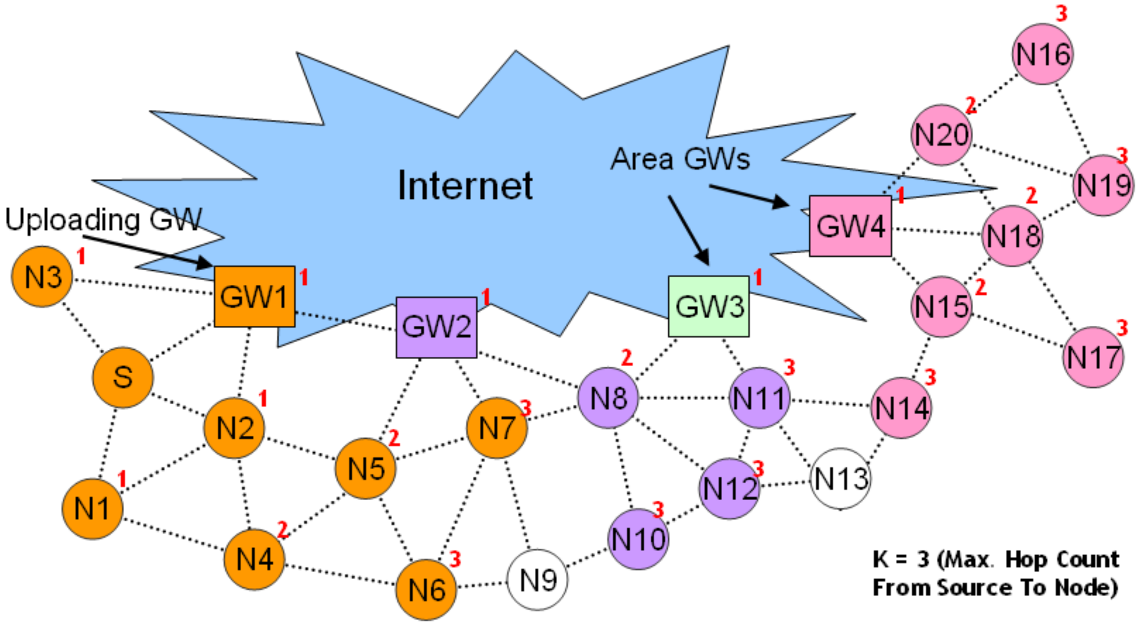


Figure 4.6. TIA-ENH-2 implementation over the reference network.

Although TIA-ENH-2 could be able to cover new nodes different than covered by TIA, TIA-ENH-2 might affect reachability of some nodes negatively if it is used without TIA-ENH-1. What is intended to say is; at Figure 4.6, Node-13 becomes unreachable after implementing TIA-ENH-2 since the members of the access area which is constructed by GW2 is blocking the way of GW3. However, if we had used also TIA-ENH-1 in a combination with TIA-ENH-2 (which constitutes Enhanced TIA algorithm), we would be able to cover also Node-13 which is shown at Figure 4.7.

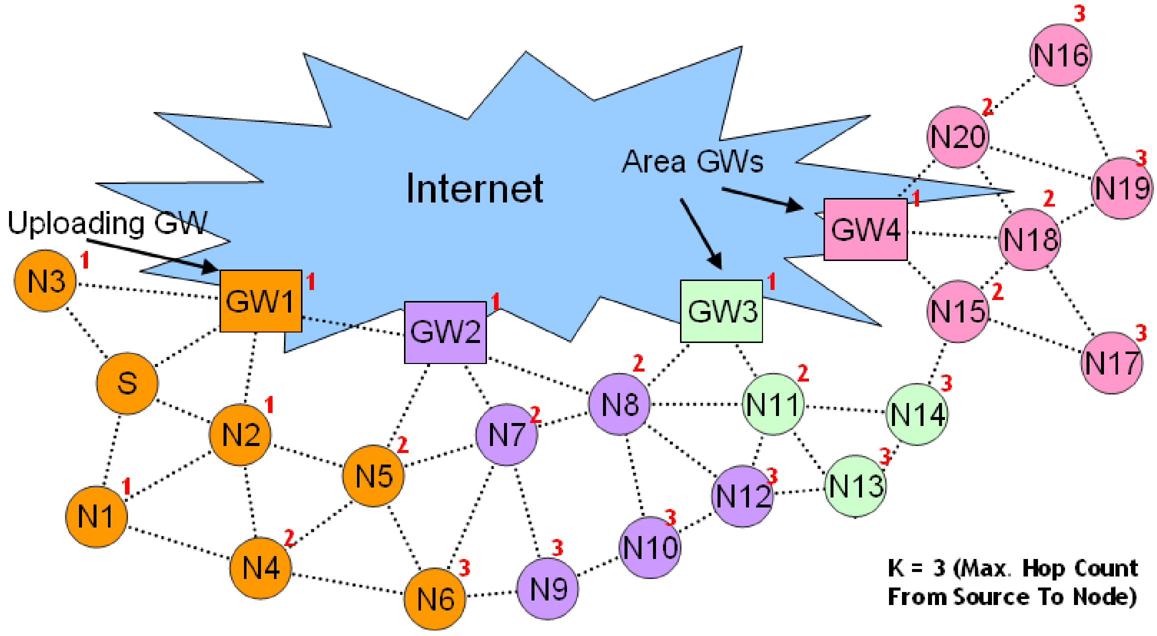


Figure 4.7. Enhanced TIA implementation over the reference network.

According to the code point of view, what we did to enhance the TIA algorithm can be recognized by viewing the content of the new added or modified operations indicated with colour light orange at Figure 4.8. Differently from the original TIA flow, we are checking the source access area that it has gateway(s) other than uploadig gateway. If there exists, TIA-ENH-2 crops it and its children from the source access area and then go on with its next operation. Another modification to the TIA flow is that multiple gateway addition to the currently constructing access area is no longer permitted. By the way, TIA-ENH-1 guarantees that every access area has only one gateway which makes every gateway either uploading gateway or area gateway (constructer gateway of the access area). Additionally, it can be said that maximum reachability of the gateways could be utilized by using TIA-ENH-2.

As it is mentioned above; we also put all the TIA enhancements together at the algorithm named Enhanced TIA in order to make use of both of our enhancements (TIA-ENH-1 and TIA-ENH-2). By implementing Enhanced TIA over the reference network, we could be able to cover Node-10 without loosing Node-13 as it can be seen at Figure 4.7.

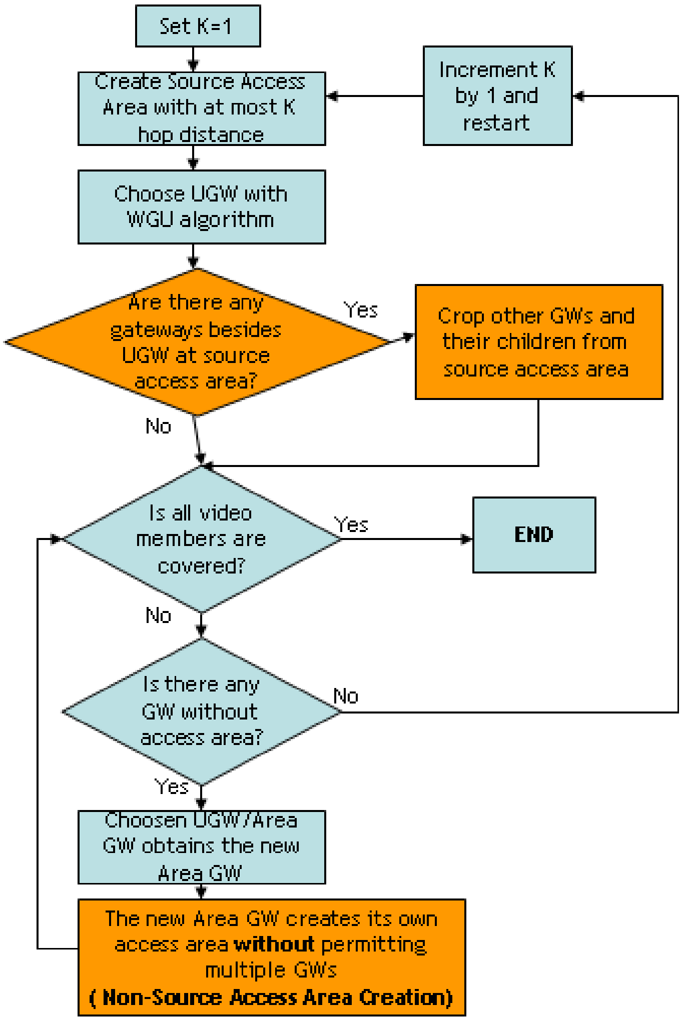


Figure 4.8. Flow Chart of TIA-ENH-2 Algorithm.

4.2. Link-Controlled Routing Tree Algorithm Enhancements

Before starting to describe the LCRT enhancements, it could be better to describe LCRT over its flow chart. For this purpose, the flow chart of the LCRT algorithm has been shown at Figure 4.9.

In order to construct video multicast tree by prunning the initial access area tree, LCRT first determines the node levels according to their hop distance to the root(s) of the LCRT video multicast tree. Afterwards, LCRT obtains the maximum node level and set the nodes of that level as leaf nodes. Then, LCRT states the uncovered out degree of the parents of these leaf nodes by counting the child nodes which have not joined the video multicast tree yet. Thereafter, LCRT searches for a node with largest uncovered out degree and the least connections to the selected forwarders. If it find one, LCRT selects it as a forwarder and removes its childs from uncovered member list. If LCRT could not be able to find any node with the above mentioned specifications, it then eliminates the "least connections to the selected forwarders" criteria and obtains the new forwarder accordingly.

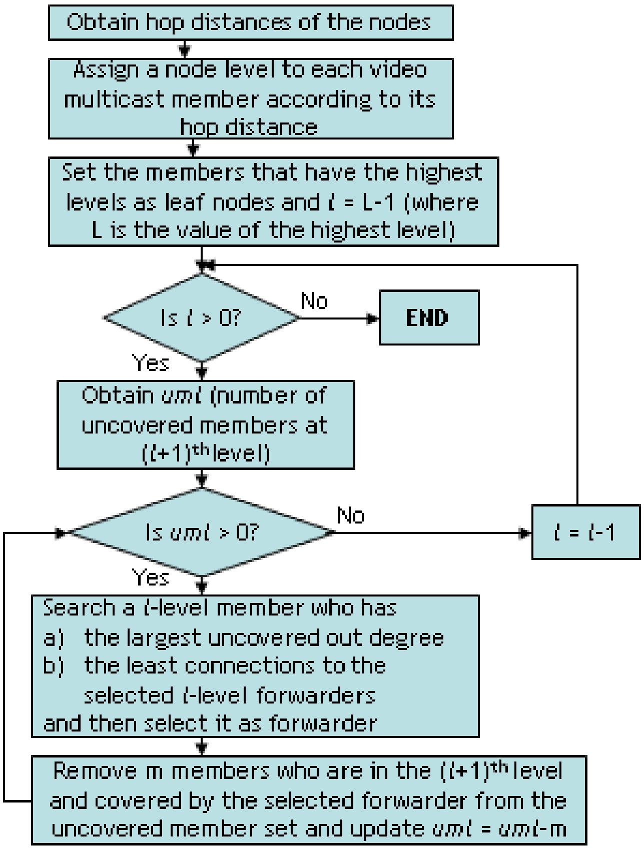


Figure 4.9. Flow Chart of LCRT Algorithm.

4.2.1. Cover Nodes Which Has Only A Parent First LCRT Algorithm (LCRT-ENH-1)

As a first improvement over LCRT algorithm, it is proposed to first set the nodes which has uncovered members with only one parent as a forwarder. After adding these kind of nodes to the video multicast tree, LCRT algorithm is being continued with its regular flow as it can be seen at Figure 4.9. In order to find out the difference between LCRT and the LCRT-ENH-1, Figure 4.10 could be examined. At that figure, the added part is stated with light orange to make the distinction more clear.

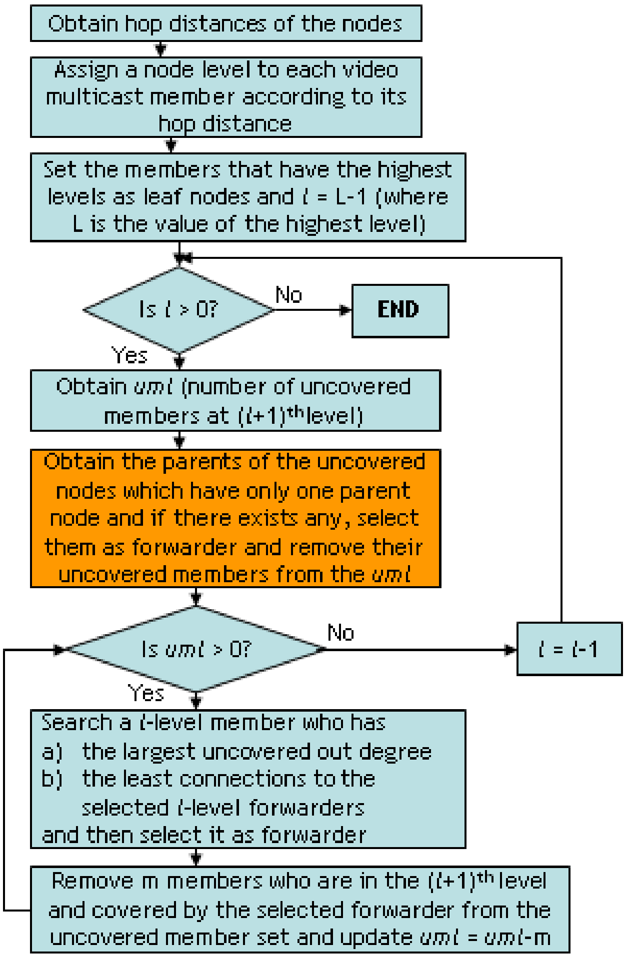


Figure 4.10. Flow Chart of LCRT-ENH-1 Algorithm.

The inspiration of LCRT-ENH-1 algorithm is that nodes which have uncovered children that have only one parent node should be added to the video multicast tree as forwarders sooner or later. Since we are trying to avoid to set two neighboring nodes in order to minimize the interference between video transmissions, it is evident that we would better first add the nodes that we would have to include sooner or later.

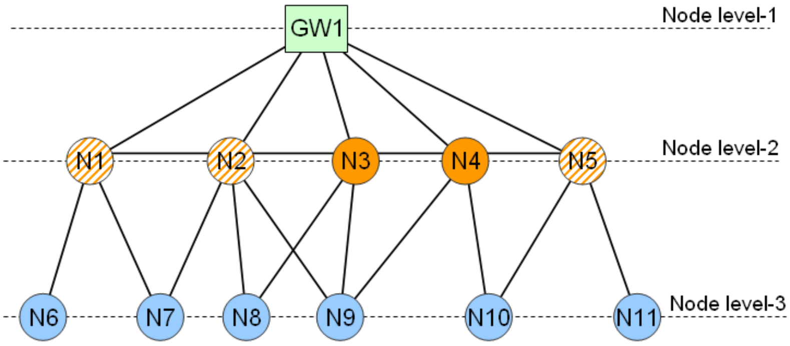


Figure 4.11. LCRT implementation over an example network to describe LCRT-ENH-1.

For the sake of clarity, the best thing to do is giving an example. For this purpose let us have an access area with the node levels mentioned at Figure 4.11. If we run ordinary LCRT over the network indicated at Figure 4.11, it selects first N2 as a forwarder since it has the most uncovered members with three uncovered nodes. Afterwards, amongst the remaining level-2 nodes, N5 has the largest uncovered out degree number with two. Then, there remains only one uncovered node to be covered which has only one parent named N1.

However, if we apply LCRT-ENH-1 to the same topology, we could get better results in terms of minimal interference. Figure 4.12 shows the forwarder state after implementing LCRT-ENH-1 on the example network for LCRT-ENH-1 description. As we can see, the LCRT-ENH-1 algorithm has first selected the N1 and N5 nodes as forwarder, then it will conclude the forwarder selection operation for node level-2 with adding N3 to the video multicast tree as a forwarder. By choosing this combination, we could be able to avoid the interference between node N2 and N1.

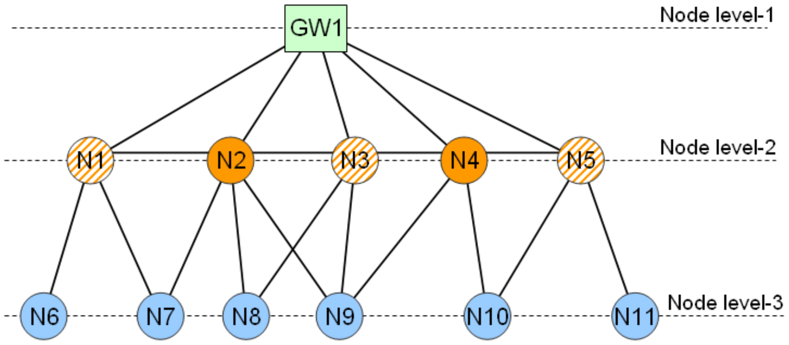


Figure 4.12. LCRT-ENH-1 implementation over the example network to describe LCRT-ENH-1.

4.2.2. LCRT For The Union Of The Non-Source Access Areas (LCRT-ENH-2)

Although the LCRT algorithm considers the interference between two neighboring nodes within the same access area, it is overlooking the interference between the neighbors which belong different access areas. To illustrate this situation, let us dig in the topology mentioned at Figure 4.13.

At Figure 4.13, if N2 has higher avaliable capacity, LCRT would choose N2 to be the forwarder node for N4 and N5. This is just because even though N1 and N2 has same uncovered out degree, N2 has also superior avaliable capacity value. Nevertheless, if we look at the network from an upper layer, we recognize that N2 is interfering with another forwarder node of different access area.

On the other hand, if we use LCRT-ENH-2 algorithm instead of LCRT for that network, we would get a video multicast tree illustrated at Figure 4.14. Since we are combining the non-source access areas within a virtual access area, LCRT will start to consider the inter access area neighbor relationships. By that way, LCRT-ENH-2 algorithm selects N3 first because of its larger number of uncovered members. Afterwards, the algorithm will choose N1 as the other forwarder instead of node N2 in order to avoid the interference of neighboring nodes.

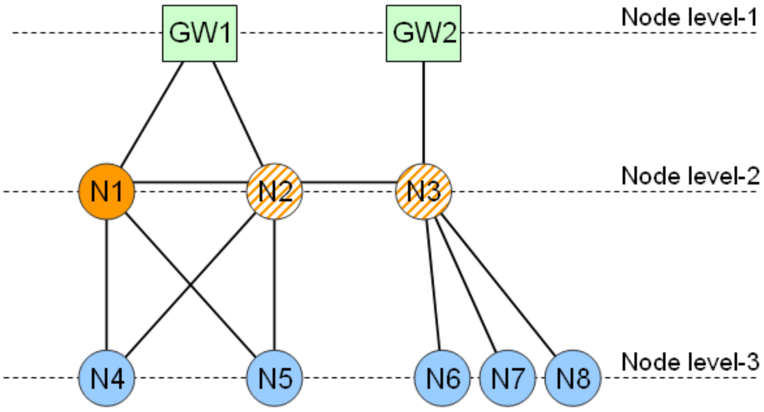


Figure 4.13. LCRT implementation over an example network to describe LCRT-ENH-2.

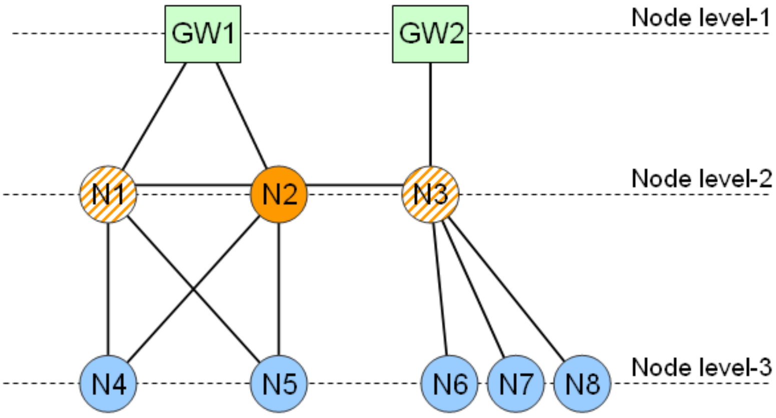


Figure 4.14. LCRT-ENH-2 implementation over the example network to describe LCRT-ENH-2.

In order to implement this enhancement, one additional task was added to be done before running LCRT. The task is simply adding all members of the non-source access areas to another virtual access area. It is said “virtual” since the access area is just used for LCRT-ENH-2 implementation.

The flow chart of the LCRT-ENH-2 algorithm is stated at Figure 4.15.

In addition to these two enhancements over LCRT algorithm, we also implement another algorithm improvement named Enhanced LCRT which is the combination of LCRT-ENH-1 and LCRT-ENH-2. All of these three algorithms have been simulated in order to see the effects of them more precisely and to be able to interpret the results in a more clear way.

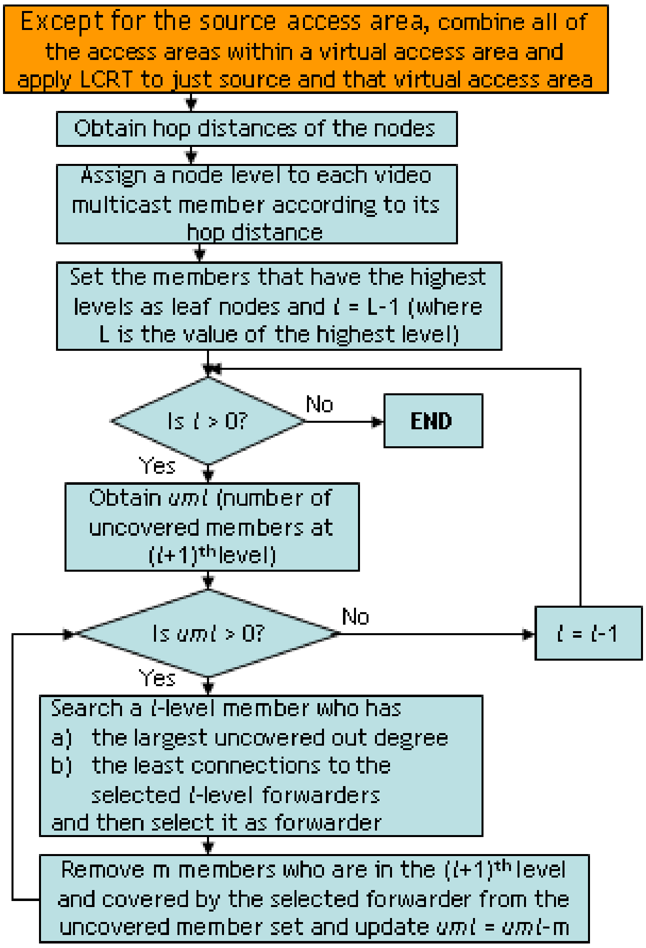


Figure 4.15. Flow Chart of LCRT-ENH-2 Algorithm.

5. SIMULATION MODELS

In order to simulate the algorithms proposed at this thesis work, a simulation programme has been written by using C++ programming language [20,21] and Borland C++ Builder 6 [22] as a compiler. The programme constructed over a simulation model that is mainly consists of three parts as it could be seen at Figure 5.1.

First, the video multicast tree is constructed by using appropriate TIA, WGU and LCRT algorithms. By implementing TIA at first time, we get the source access area and then use the gateways of this access area to maintain uploading gateway. Afterwards, the non-source access areas are constructed by the help of that uploading gateway. After completing the access area creation, the create video multicast tree part of the simulation will continue with applying LCRT over the obtained access areas.

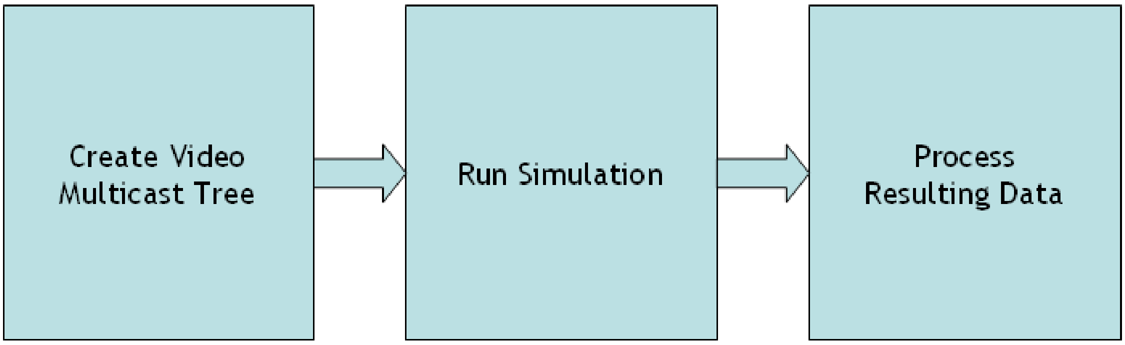


Figure 5.1. Overview of the Simulation Model.

As the second task, simulation model starts running the simulation itself. In order to acheive that, simulation programme uses several other models to simulate network, interference relationships of the transmissions, medium access to transmit data and create data traffic. These models will be briefly described at Section 5.2.

Finally, simulation programme continues with the calculations to present the performance of the algorithms by the means of average throughput, delay and jitter which are crutial metrics for video transmission over Wireless Mesh Networks like they are for any other kind of computer networks. At Section 5.3, the formulas for the metric calculations will be stated in short.

5.1. Video Multicast Tree Creation Model

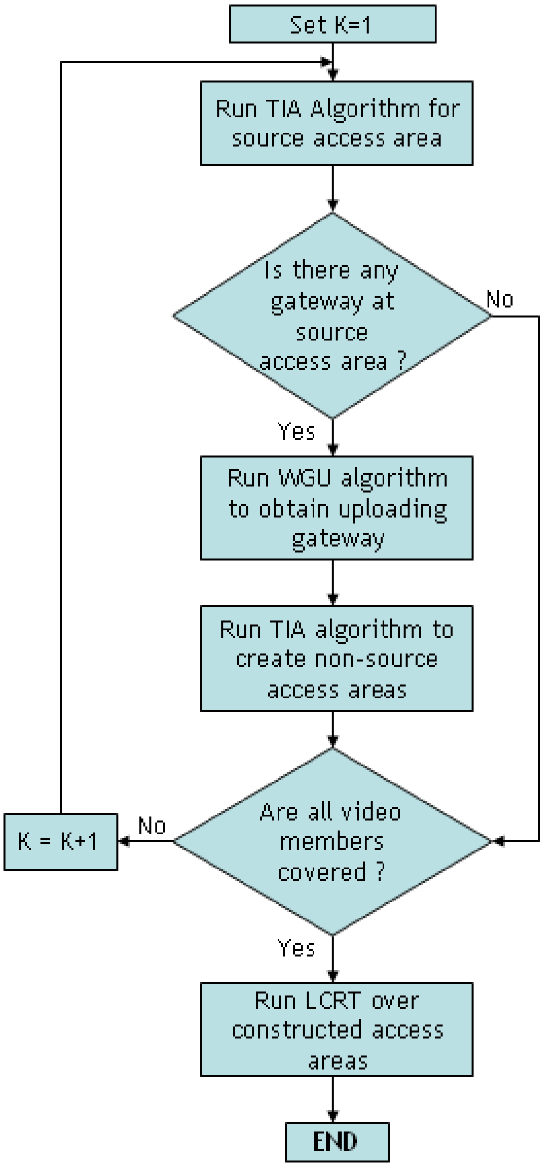


Figure 5.2. Flow Chart of Video Multicast Tree Creation Algorithm.

While implementing TIA, it is needed to obtain K value first to begin to create an access area. At our scheme the K value is maintained as the minimum value that makes TIA capable of covering all video members since in our scheme, it is assumed that every node belongs to an access area. By this way we could be able to serve video to all of the video multicasting members.

Hence, video multicast tree creation starts K with value one and increments it if it could not be able to cover all video multicast members after maintaining multicast tree. According to the flow chart that is shown at Figure 5.2, video multicasting tree creation continues with creating the source access area and obtaining the uploading gateway by using WGU algorithm. Only after maintaining the uploading gateway, it could be possible to create non-source access areas by selecting an area gateway (constructor gateway of the related access area) and running TIA for it. As soon as the access area creation has been finished the video multicast tree creation algorithm checks if all video members are covered or not. If so, it goes further to the next step which is applying the LCRT algorithm to the constructed access areas. Otherwise, it increments K by one and return back to the running TIA to create source access area stage.

5.2. Running Simulation Model

Running simulation model is the main part of the simulation programme and consists of network, interference, transmission and traffic model which is described at the following subsections.

As it can be seen at the flow chart of the running simulation, which is indicated at Figure 5.3, since the running simulation function is called several times from the main function, it begins the operation with first clearing all buffers. Afterwards, simulation run is started with time zero and checks if the video start time is reached. If so it adds video data to the video buffer of the source and update the active transmissions; otherwise, it updates the active transmissions directly. After checking time for video data, programme checks if it is time to add video data to the nodes. If so, it runs the

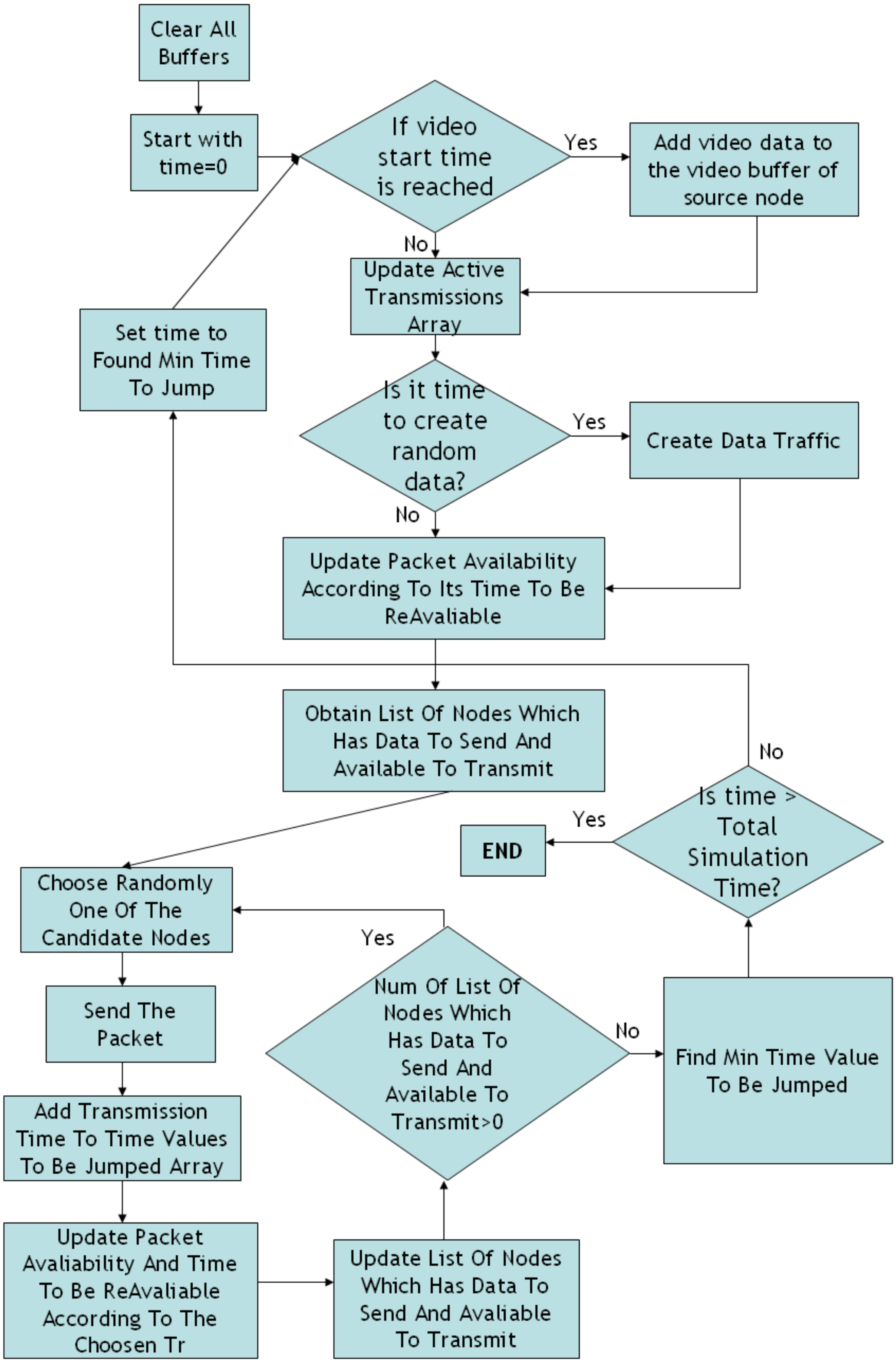


Figure 5.3. Flow Chart of the Running Simulation.

random traffic generation algorithm and includes related data to the data buffers of the related nodes.

After updating buffers, the algorithm obtains the candidate nodes for transmission and choose one of these contenting nodes randomly. After choosing the transmission, it sends the data to the next buffer if it is video data and blocks the affected area for the transmission time. In order to do so, it implements the transmission rules described in Section 5.2.2. By the way, the release time, which is sum of the actual time and the transmission time, is recorded to the appropriate array named TimeValuesToBeJumped. Afterwards the programme updates the contenting node list for that moment and if there exists it again choose one of them and do the same operations since there will exist no contenting node. When there remains no contenting node, programme goes for next step which is the task of jumping the next time value. It picks and deletes the minimum time value from TimeValuesToBeJumped array and if the time to be jumped is not exceeding the total simulation time, program goes to the beginning of the flow and do the same tasks for the new time. When time reaches a value bigger than the total simulation time, programme finishes the running simulation and starts calculating the average metrics (average throughput, delay and jitter).

5.2.1. Network Model

The network scheme consists of wireless mesh nodes and gateways which makes getting use of wired resources possible. In general manner, the topology is like the one that is shown at Figure 5.4.

For small scale topology, we have fully connected wireless mesh nodes and gateways with their wireless interfaces. Additionally, gateways has also wired connections with very high speed that can allow us to neglect the transmission time. Hence, we are assuming all of the gateways have shortcuts to each other.

On the other hand, at the large scale topology scheme, it is like we have group of wireless mesh networks which have gateway(s) to reach the internet.

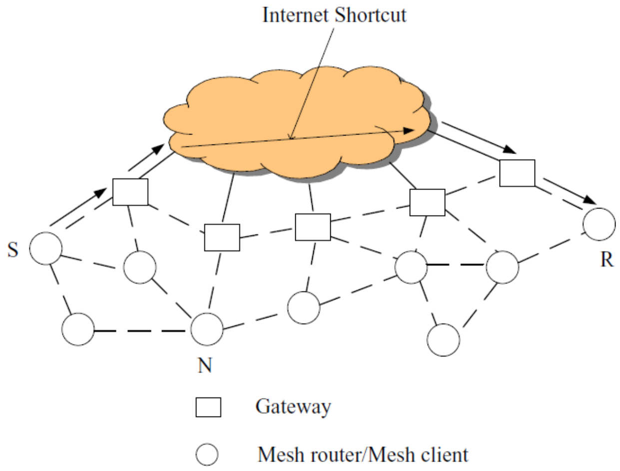


Figure 5.4. Network Model.

5.2.2. Interference Model

In communications and electronics, especially in telecommunications, interference is anything which alters, modifies, or disrupts a signal as it travels along a channel between a source and a receiver. The term typically refers to the addition of unwanted signals to a useful signal. There are several kinds of interference like co-channel interference (CCI) which is considered at our simulation. CCI is by definition a crosstalk from two different radio transmitters using the same frequency and affects the transmission quality in a negative manner.

As co-channel interference takes place at wireless transmissions and affect the performance of the transmission, it should also be modeled at our simulation programme. At our simulation it is assumed that all nodes transmitting with same power; thus, they all have the same interference range which is the area that it affects when it is transmitting.

On the other hand, in order to simplify the simulation it is assumed that there is no phsical layer error like bit or symbol errors (i.e. caused by fading).

In addition, although 802.11b has different data rates as 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps; data rate of 1 Mbps is choosen to reach much more users.

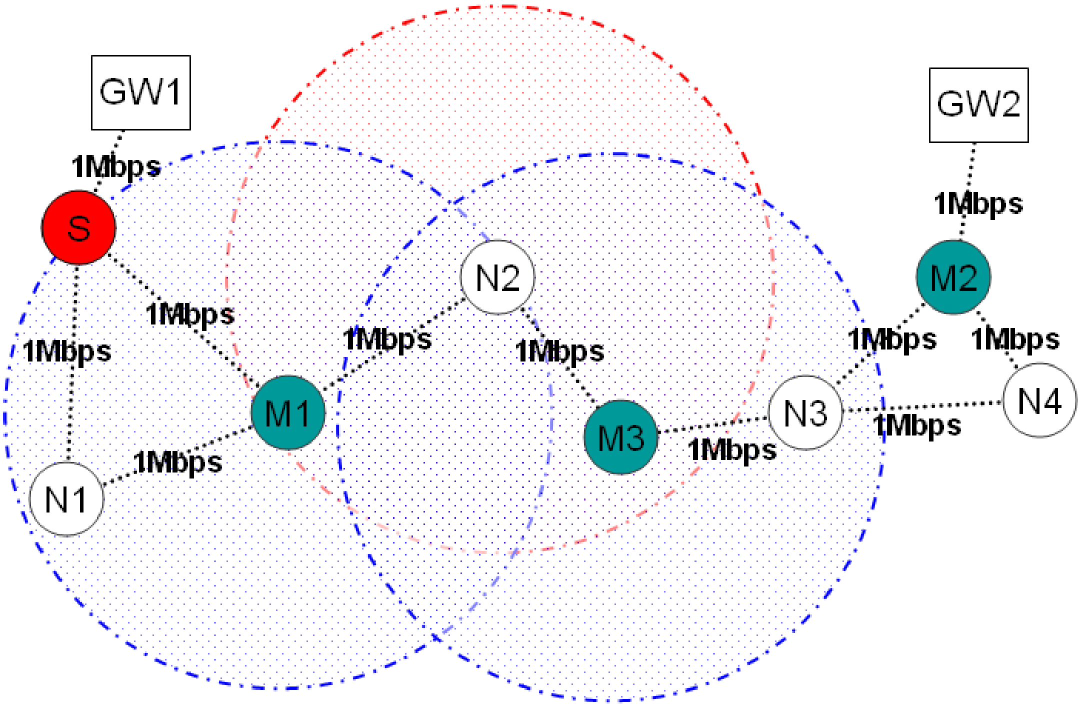


Figure 5.5. Interference Model.

As rules of thumbs for transmission, the following items could be stated:

(i) A node can not receive a packet while it is transmitting [23].

(ii) While transmission being occured, no node at the interference range of the TX could be a receiver.

(iii) While transmission being occured, no node at the interference range of the RX could be a transmitter.

5.2.3. Transmission Model (CSMA/CA Implementation)

As a media access protocol, 802.11 uses Carrier Sense Multiple Access with Collision avoidance which is better to be defined first to describe the transmission model.

A CSMA protocol works as follows: A station desiring to transmit senses the medium, if the medium is busy (i.e. some other station is transmitting) then the station will defer its transmission to a later time, if the medium is sensed free then the station is allowed to transmit [24].

These kind of protocols are very effective when the medium is not heavily loaded, since it allows stations to transmit with minimum delay, but there is always a chance of stations transmitting at the same time (collision), caused by the fact that the stations sensed the medium free and decided transmit at once [24].

These collision situations must be identified, so the MAC layer can retransmit the packet by itself and not by upper layers, which would cause significant delay. In the ethernet case this collision is recognized by the transmitting stations which go to a restransmission phase based on an exponential random backoff algorithm [24, 25].

While these Collision Detection mechanisms are a good idea on a wired LAN, they cannot be used on a Wireless LAN environment, because of two reasons [24]:

(i) Implementing a Collision Detection Mechanism would require the implementation of a Full Duplex radio, capable of transmitting and receiving at once, an approach that would increase the price significantly.
(ii) On a Wireless environment we cannot assume that all stations hear each other (which is the basic assumption of the Collision Detection scheme), and the fact that a station willing to transmit and senses the medium free, does not necessarily mean that the medium is free around the receiver area.

In order to overcome these problems, the 802.11 uses a Collision Avoidance mechanism [26] together with a Positive Acknowledge scheme, as follows:

A station willing to transmit senses the medium, if the medium is busy then it defers. If the medium is free for a specified time (called Distributed Inter Frame Space (DIFS) in the standard) then the station is allowed to transmit, the receiving station will check the Cyclic Redundancy Check (CRC) of the received packet and send an acknowledgement packet (ACK). Receipt of the acknowledgement will indicate the transmitter that no collision occured. If the receiver does not receive the acknowledgement then it will retransmit the fragment until it gets acknowledged or thrown away after a given number of retransmissions [24].

In order to reduce the probability of two stations colliding because they cannot hear each other, the standard defines a Virtual Carrier Sense mechanism which is described as follows:

A station willing to transmit a packet will first transmit a short control packet called RTS (Request to Send), which will include the source, destination, and the duration of the following transaction (i.e. the packet and the respective ACK), the destination station will respond (if the medium is free) with a response control Packet called CTS (Clear to Send), which will include the same duration information [24].

All stations receiving either the RTS and/or the CTS, will set their Virtual Carrier Sense indicator (called NAV, for Network Allocation Vector), for the given duration, and will use this information together with the Physical Carrier Sense when sensing the medium [24].

This mechanism reduces the the probability of a collision on the receiver area by a station that is “hidden” from the transmitter, to the short duration of the RTS transmission, because the station will hear the CTS and “reserve” the medium as busy until the end of the transaction. The duration information on the RTS also protects the transmitter area from collisions during the ACK (by stations that are out of range from the acknowledging station) [24].

In our scheme, RTS/CTS mechanism is proposed to be used in order to avoid collisions. By utilizing the RTS/CTS mechanism, it is assumed that there will be no collision at the environment. Since we use the data rate of 1 Mbps, data transmission time is needed to be calculated in order to be able to obtain the medium occupancy of

Table 5.1. Transaction times for TCP data (for data rate of 1 Mbps).

	TCP data	**TCP ACK**
DIFS	50 μs	50 μs
RTS	352 μs	352 μs
SIFS	10 μs	10 μs
CTS	304 μs	304 μs
SIFS	10 μs	10 μs
802.11 Data	12480 μs	608 μs
SIFS	10 μs	10 μs
802.11 ACK	304 μs	304 μs

each transmission. According to [27,28], a Transmission Control Protocol (TCP) data transmission takes the time that is shown at Table 5.1. At Table 5.1, it can be seen that for a transaction there are mainly four phases which are TCP data, 802.11 ACK for TCP data, TCP ACK and 802.11 ACK for TCP ACK transmissions [29]. However, the 802.11 ACK and TCP ACK frame transmissions are omitted in order to reduce the complexity. By that way, we converted TCP packet as if it is an UDP packet (since it has no ACK now) and simplified the transaction as it is stated at Table 5.2.

Table 5.2. Reduced transaction times for UDP data (for data rate of 1 Mbps).

	UDP data
DIFS	50 μs
RTS	352 μs
SIFS	10 μs
CTS	304 μs
SIFS	10 μs
802.11 Data	12480 μs
Total transaction time	13206 μs

Consequently, one packet takes 13206 μs at our simulation; thus, other calculations has been done accordingly.

In order to follow the stages of the process the flow chart of CSMA/CA implementation which is stated at Figure 5.6 can be examined.

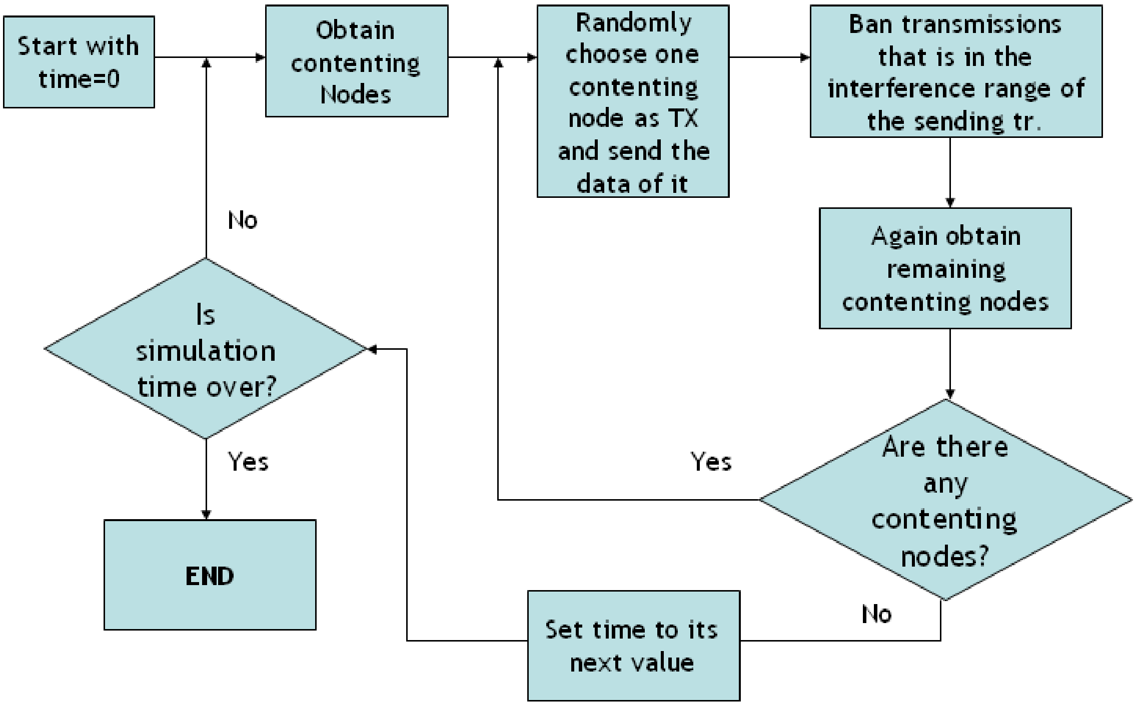


Figure 5.6. Flow Chart of CSMA/CA Implementation.

As a first operation, programme starts with initializing the time value by setting it to zero. Then it obtains the contenting nodes by checking all of the buffers of the nodes if there exists any avaliable data to send. After gathering the contenting nodes list, programme randomly choose one of them and send its data to the related destination. While transmitting data, it bans the transmissions in the interference range of the transmission by setting their avaliability tag to “false”. Then obtaining contenting nodes function is run for the remaining contenting nodes. Afterwards, the list of contenting nodes is checked if it has any nodes that is waiting to send data and avaliable to transmit. If so, the programme jumps the stage that it randomly chooses a contenting node to transmit and goes as it did before. If there remains no contenting node for that moment, programme set time to its next value and unless the simulation time is over, it goes on with the obtaining contenting nodes phase.

5.2.4. Traffic Model

In order to simulate the data traffic, a simple traffic generation method is used. At that traffic model, a two state Markov chain [30, 31] is used to identify the traffic load. State diagram of the Markov chain is shown at Figure 5.7. At the state diagram $P_{x,y}$ indicates the probability to change the state from x to y.

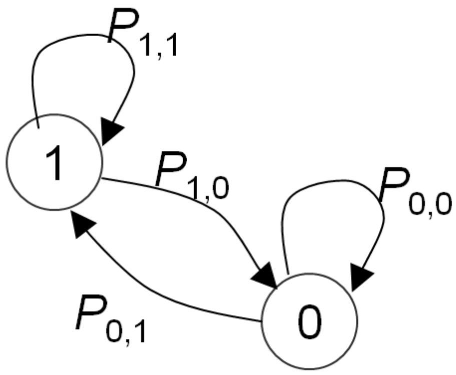
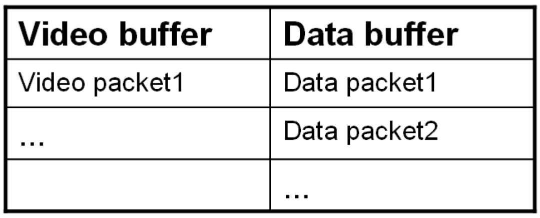


Figure 5.7. Markov Chain State Diagram for Traffic Model.

By using that state diagram, we could be able to create data for a constant time period which is choosen minor than the minimum transmisiion time to provide data continuity. Hence, the period is choosen to be 13000 μs since for 1 Mbps data rate the transmission time is calculated as 13206 μs at Section 5.2.3.



Video buffer	Data buffer
Video packet1	Data packet1
...	Data packet2
	...

Figure 5.8. Video and Data Buffer of a Node.

In order to store the data, it is designed that the nodes seperately have a video and a data buffer as shown in Figure 5.8. The buffers are working by obeying the FIFO (First In First Out) inventory management system [32] which means if a new data comes for i.e. video buffer, it is added at the end of the buffer while if a data will be send from i.e. video buffer, it is selected from the top of the video buffer list.

5.3. Data Processing Model

For each video transmissions, simulation programme records its receiving time to be able to calculate throughput, delay and jitter. After running the simulation for the specified time (80 secs in our case), it processes the packet arrival times and obtain metrics accordingly. Detailed formulas for the metrics are stated at the following sections.

5.3.1. Throughput Calculation

For a specified time interval, the average throughput is calculated as;

$$\mathbf{Avg.throughput} = \frac{Size\ of\ the\ packets\ within\ that\ time\ interval}{Time\ interval} \tag{5.1}$$

5.3.2. Delay Calculation

By differentiating the values of the video sending time and the arrival time, we could maintain delay value for each packet. Afterwards, average delay could easily be calculated as;

$$\textbf{Avg.delay} = \frac{Sum\ of\ all\ delay\ values\ for\ all\ packets}{Number\ of\ packets} \tag{5.2}$$

5.3.3. Jitter Calculation

After obtaining the delay values, the jitter values could be calculated as;

$$\textbf{Avg.jitter} = \frac{\sum_{p=2}^{number\ of\ packets} |D(p)\ -\ D(p-1)|}{Number\ of\ packets\ -\ 1} \tag{5.3}$$

where $D(p)$ is the delay value of the pth packet.

6. SIMULATION RESULTS

Simulations that have been made to investigate the performance of the proposed algorithms can be categorized into two groups as:

(i) Evaluation using a small scale WMN
(ii) Evaluation using a large scale WMN

At simulations for both small and large scale WMNs, there are initial topologies which will then be enlarged with randomly located WMN nodes. Since the new nodes are added randomly, we had to perform the simulation several times in order to reduce the randomness effect of it. On the other hand, there is another random process at creation of the data traffic which is also be considered. Hence, to reduce the effect of our random processes, we enlarged the initial topology 10 times to get 10 topology sets and the simulation was run 10 times for each topology set. Consequently, we get more stable performance graphs as it can be expected.

Table 6.1. Simulation Parameters.

MAC Protocol	802.11 with 1 Mbps data rate
Video transmission rate	120 Kbit/s
Network initialization time	20 s
Simulation time	80 s
Total run time	100 s
Number of simulation runs per topology	10
Number of topology sets	10

The simulation parameters are stated at Table 6.1. As it is shown, simulation is run for 80s although the programme totally runs for 100s. This is because the programme first starts simulating in order to be able to calculate the weight of nodes/GWs correctly. By executing the initial run, program learns the busyness of the nodes and becomes capable of calculating the metric for WGU algorithm. After this initial phase, program begins to add video data to the video buffer of the sender node in order to start the video multicasting simulation. Beyond the end of totally 100s, programme calculates the transmission results for the last 80s.

Next sections of this chapter, the brief descriptions of the simulation environments are stated and at Section 6.3 simulations are evaluated all together. For each topology, all of the 8 algorithm sets were performed in order to be able to observe the effect of each enhancement.

6.1. Results of Simulation for Small Scale WMN

As the initial network of simulation for small scale WMN, the topology which has shown at Figure 6.1 is used.

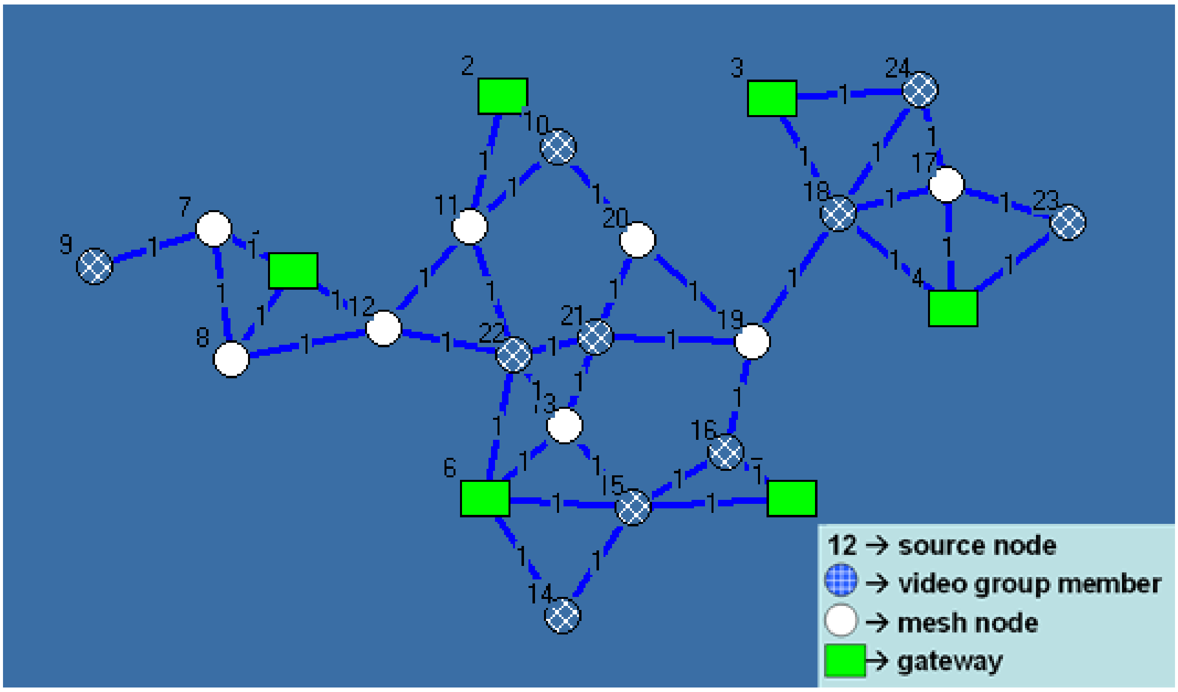


Figure 6.1. Topology of Small Scale WMN.

After initially loading the topology shown at Figure 6.1, programme starts the simulation process for 10 times and calculates average throughput, delay and jitter values. Then, it enlarges the topology with 4 randomly located new nodes and goes on with the previous process (running simulation and calculating the average metrics). This enlarging operation is continued until the network reaches the predefined maximum node number which is 42 for the small scale WMN. After completing the whole simulation for all topologies, program saves the result to a resulting file in order to be processed by Matlab [33]. Afterwards program is doing the same cycle for 10 times in order to reduce the randomness effect of the toplogy enlargement.

After getting all of these results, we processed the resulting data by using a Matlab program to get the performance graphs. In order to evaluate the performance, the following graphs have been drawn for both small and large scale WMNs:

- Average Multicast Throughput vs. Number of WMN nodes for TIA Enhancements
- Average Multicast Delay vs. Number of WMN nodes for TIA Enhancements
- Average Multicast Jitter vs. Number of WMN nodes for TIA Enhancements
- Average Multicast Throughput vs. Number of WMN nodes for LCRT Enhancements
- Average Multicast Delay vs. Number of WMN nodes for LCRT Enhancements
- Average Multicast Jitter vs. Number of WMN nodes for LCRT Enhancements
- Average Multicast Throughput vs. Number of WMN nodes for All Enhancements
- Average Multicast Delay vs. Number of WMN nodes for All Enhancements
- Average Multicast Jitter vs. Number of WMN nodes for All Enhancements

Additionally, in order to be able to interpret the performance, below mentioned improvement percentage graphs have been plotted for all enhancements:

- Average Multicast Throughput Improvement Percentage vs. Number of WMN nodes
- Average Multicast Delay Improvement Percentage vs. Number of WMN nodes
- Average Multicast Jitter Improvement Percentage vs. Number of WMN nodes

Finally, the mean of the performance improvement percentages have been averaged to make enhancement algorithm comparison much more easier.

Results for small scale WMN are shown at Figure 6.2, Figure 6.3, Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9, Figure 6.10, Figure 6.11, Figure 6.12, Figure 6.13. In order to make interpretation easier, the average values of the performance improvements are stated at Table 6.2.

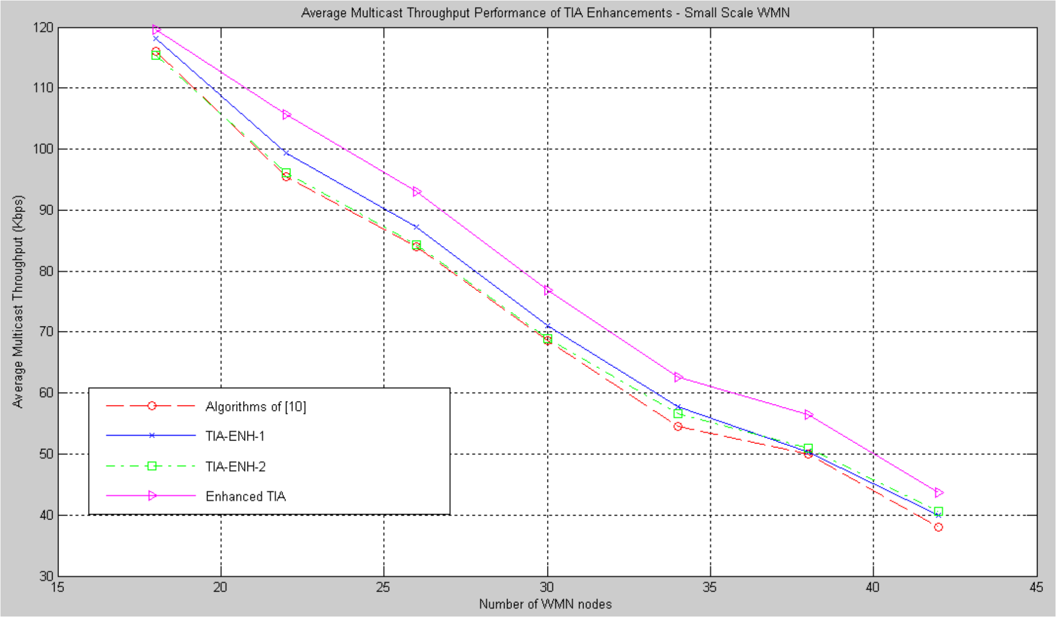


Figure 6.2. Average Multicast Throughput Performance of TIA Enhancements for Small Scale WMN.

For the average multicast throughput and delay graphs of TIA enhancements, it can be said that TIA-ENH-1 has superior performance than TIA-ENH-2. On the other hand, for LCRT there seems not so much improvement since the special case which motivates us to develop these enhancements is not occured for the used small scale network. Hence, we could not be able to see so much improvement.

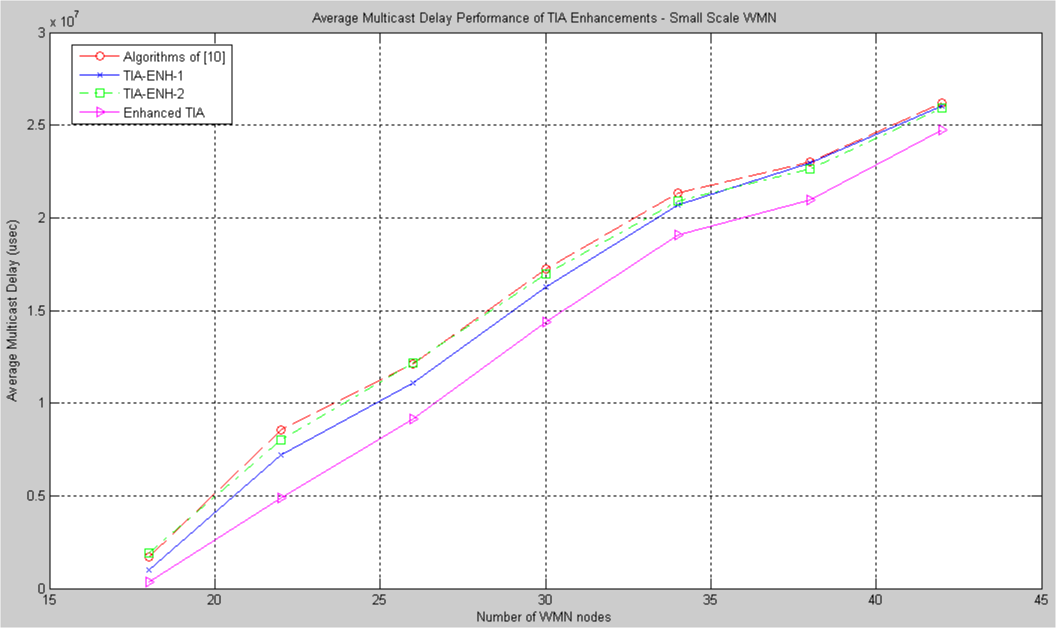


Figure 6.3. Average Multicast Delay Performance of TIA Enhancements for Small Scale WMN.

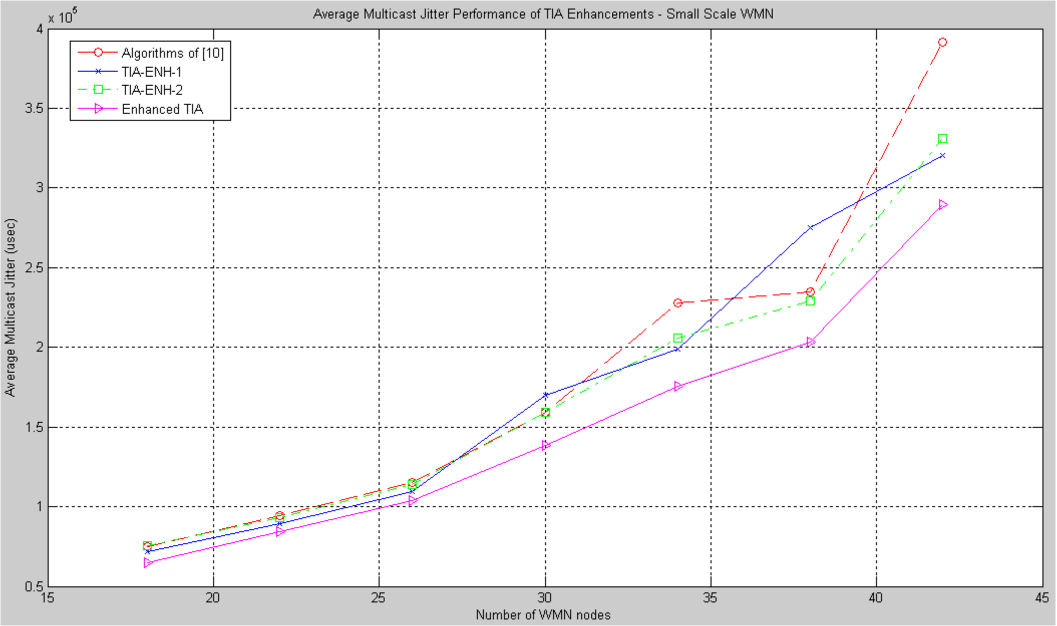


Figure 6.4. Average Multicast Jitter Performance of TIA Enhancements for Small Scale WMN.

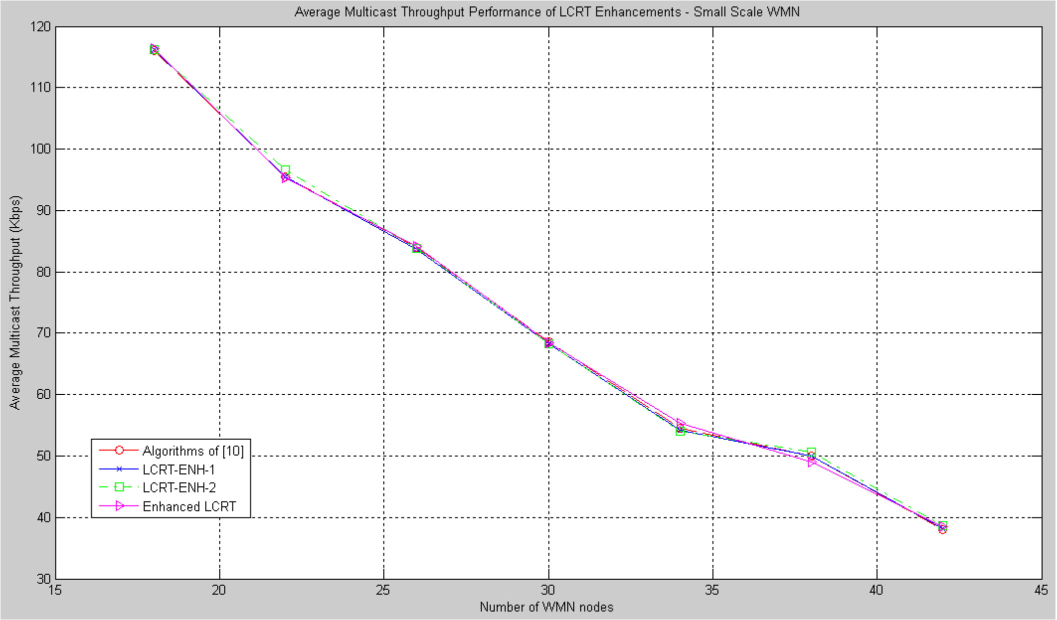


Figure 6.5. Average Multicast Throughput Performance of LCRT Enhancements for Small Scale WMN.

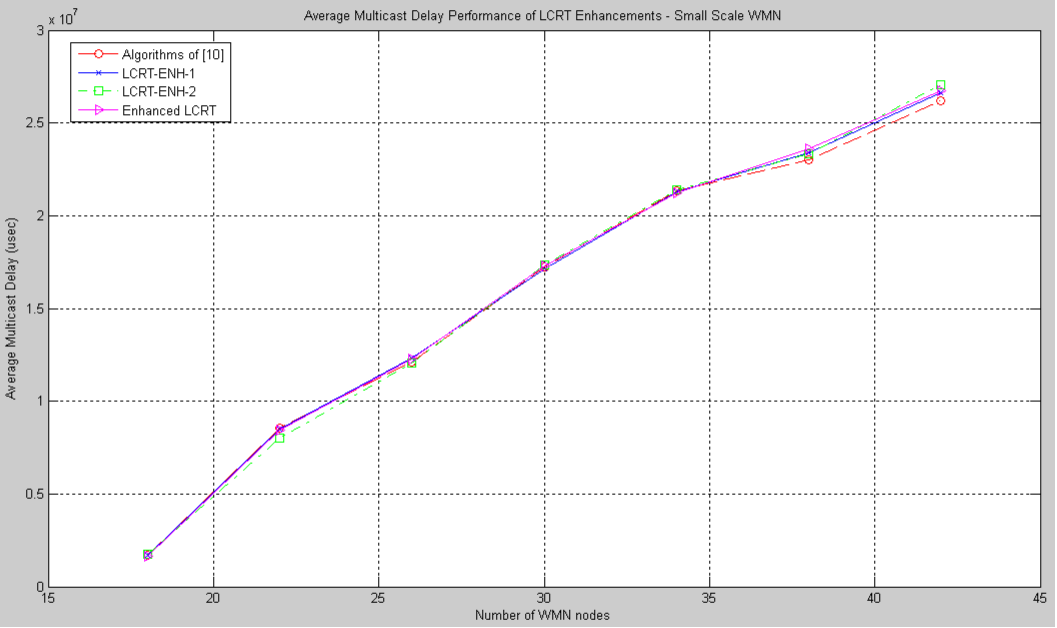


Figure 6.6. Average Multicast Delay Performance of LCRT Enhancements for Small Scale WMN.

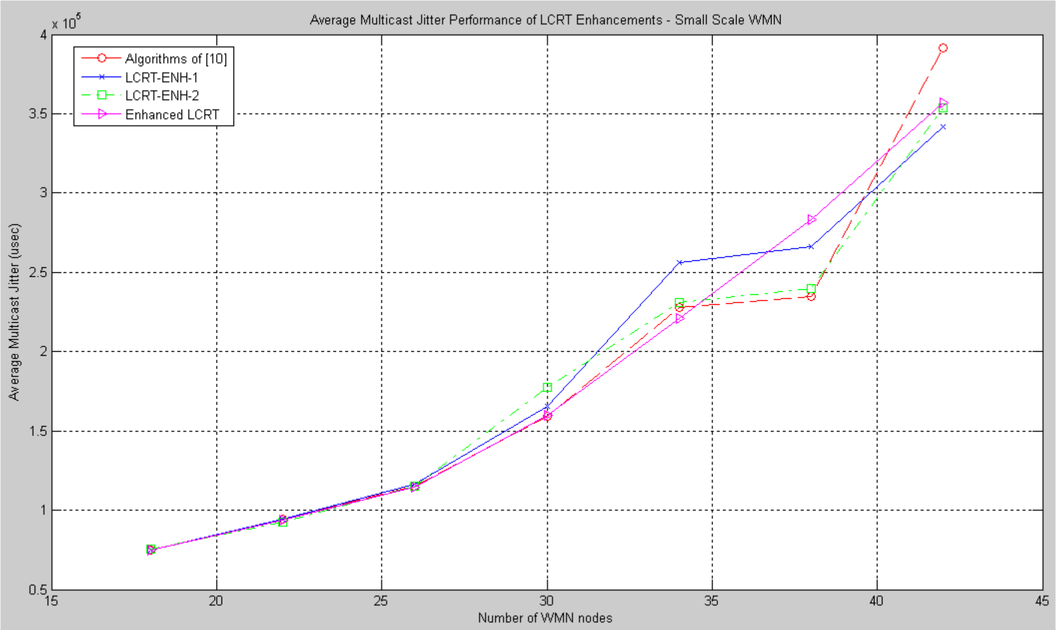


Figure 6.7. Average Multicast Jitter Performance of LCRT Enhancements for Small Scale WMN.

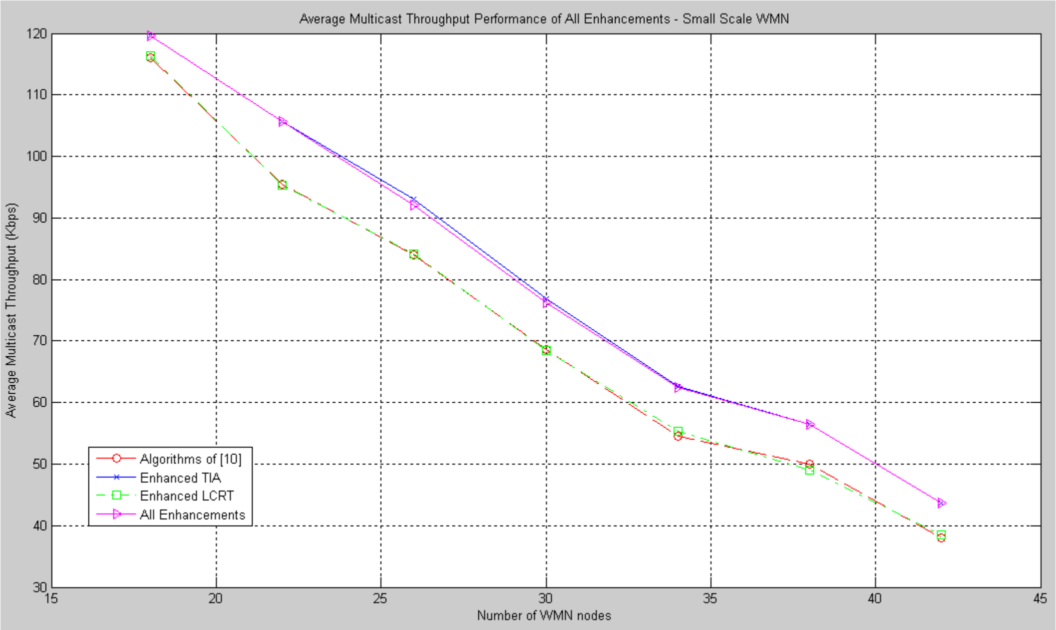


Figure 6.8. Average Multicast Throughput Performance of All Enhancements for Small Scale WMN.

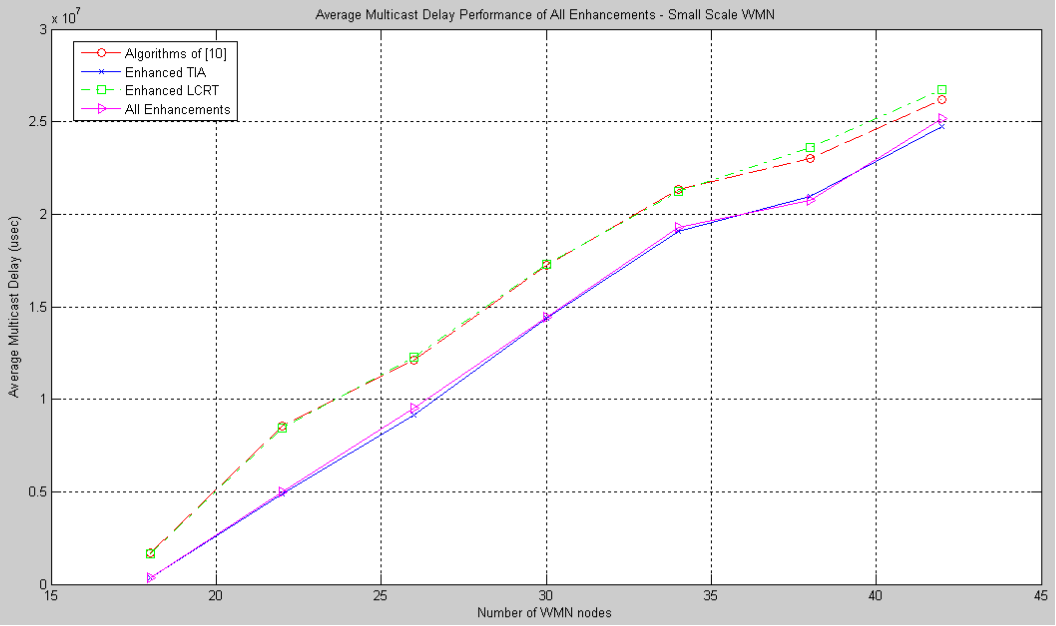


Figure 6.9. Average Multicast Delay Performance of All Enhancements for Small Scale WMN.

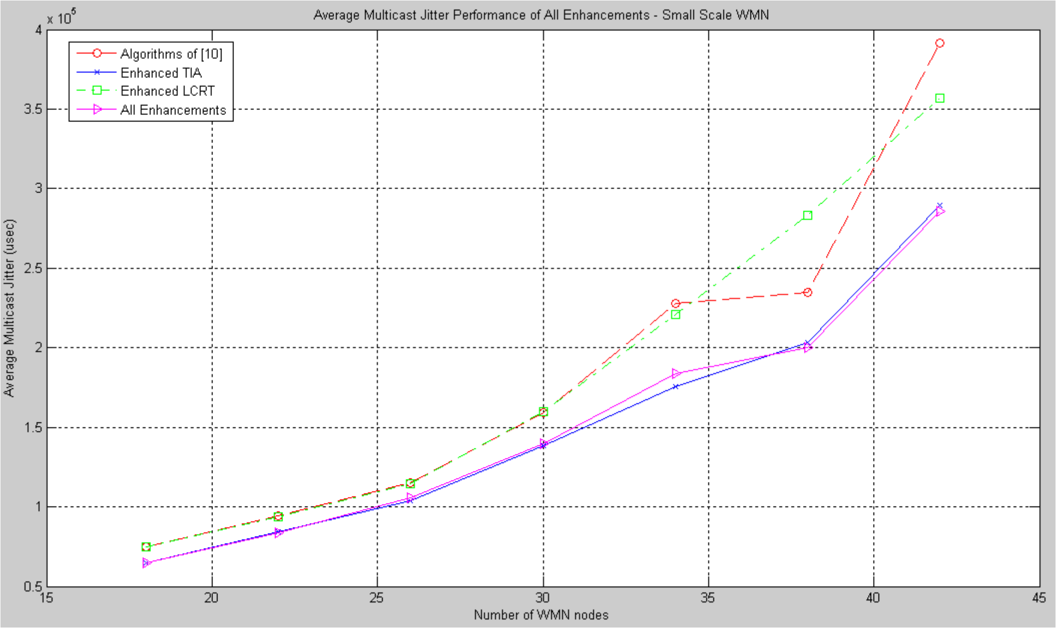


Figure 6.10. Average Multicast Jitter Performance of All Enhancements for Small Scale WMN.

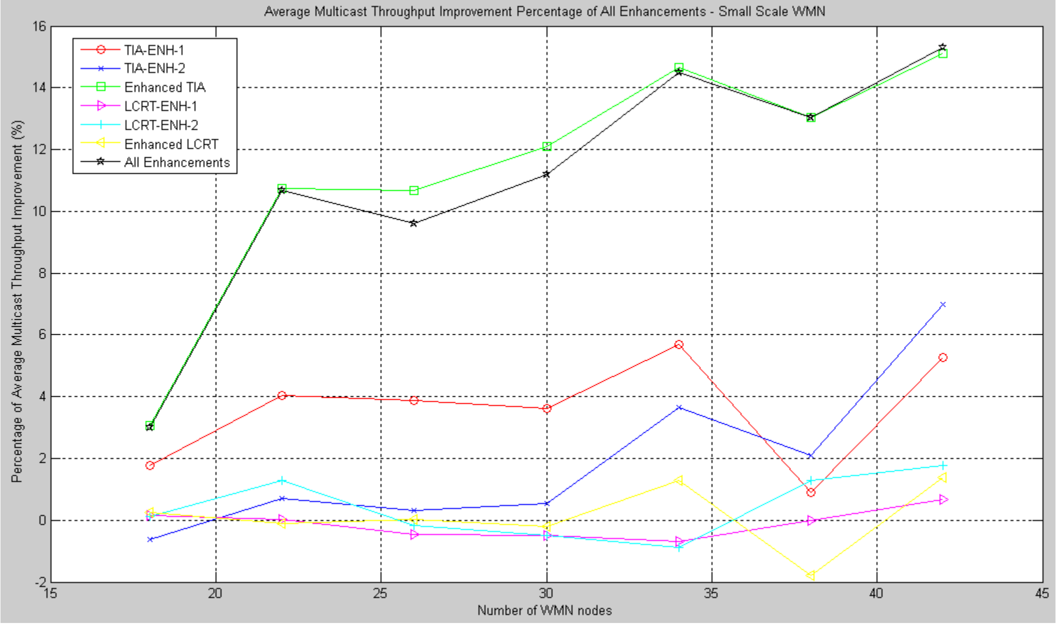


Figure 6.11. Average Multicast Throughput Improvement Performance of All Enhancements for Small Scale WMN.

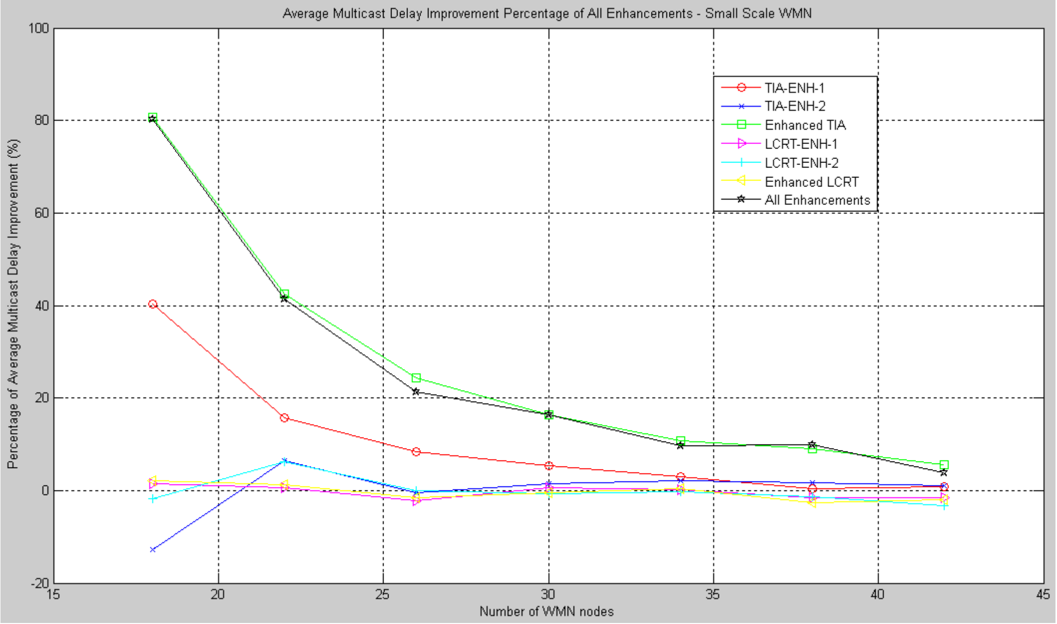


Figure 6.12. Average Multicast Delay Improvement Performance of All Enhancements for Small Scale WMN.

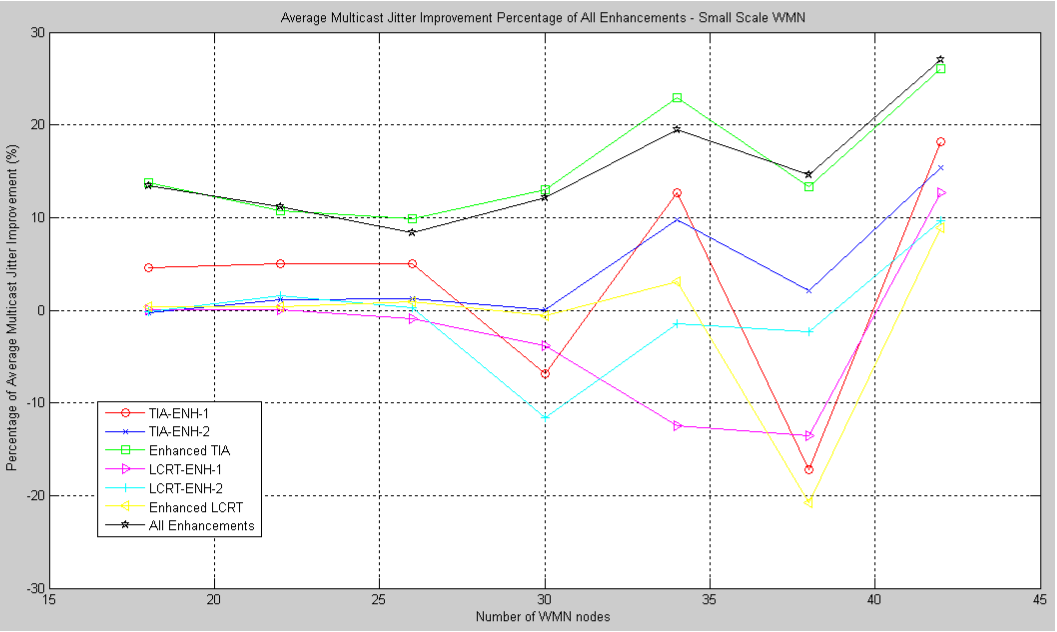


Figure 6.13. Average Multicast Jitter Improvement Performance of All Enhancements for Small Scale WMN.

6.2. Results of Simulation for Large Scale WMN

For the simulation of large scale WMN, the topology shown at Figure 6.14 is used as the initial network. Then that initial network is extended with new nodes four by four until the network size reaches the number of wireless mesh nodes 92. At each step, 10 simulations are run to reduce the randomness effect of the data traffic creation as it is done for small scale network. Like small scale WMN, the whole mentioned operation is repeated another 10 times in order to reduce the fluctuation of the results caused by the random process that is used for topology enlargement.

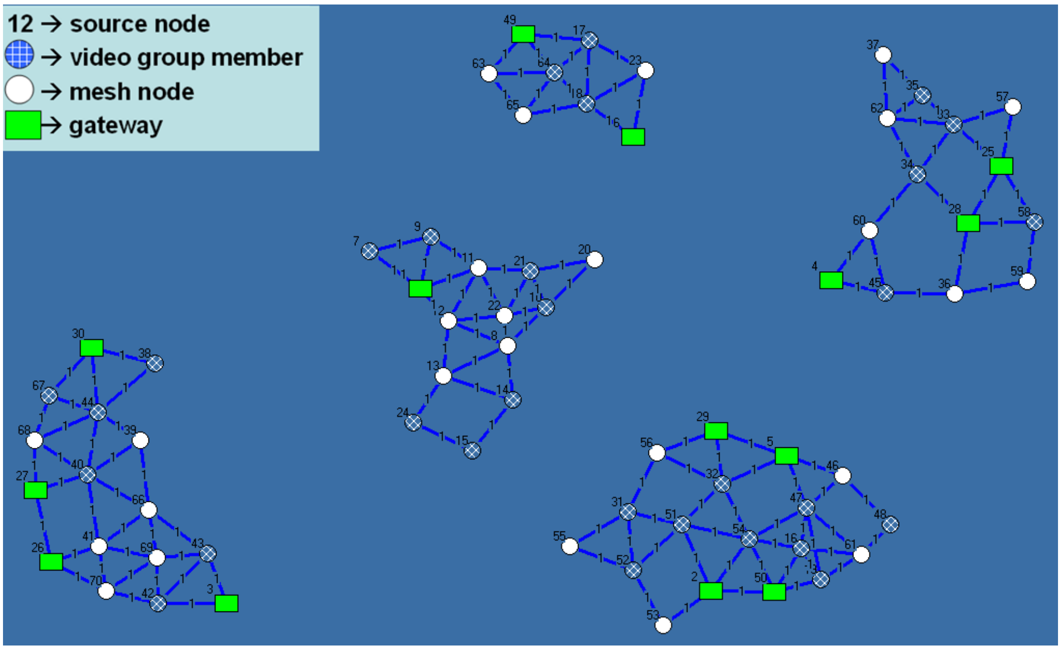


Figure 6.14. Topology of Large Scale WMN.

After performing the simulation, program calculates the metrics that are average multicast throughput, delay and jitter to make it possible to compare the performance of the enhancements and the algorithms of [10]. Then these metrics are plotted versus the number of wireless mesh nodes at Figure 6.15, Figure 6.16, Figure 6.17, Figure 6.18, Figure 6.19, Figure 6.20, Figure 6.21, Figure 6.22, Figure 6.23. In addition to these bunch of graphs, improvement percentage graphs are also drawn and added to the the report as the Figure 6.24, Figure 6.25, Figure 6.26. As for the last step, performance improvement percentages are averaged like it is done for the small scale WMN.

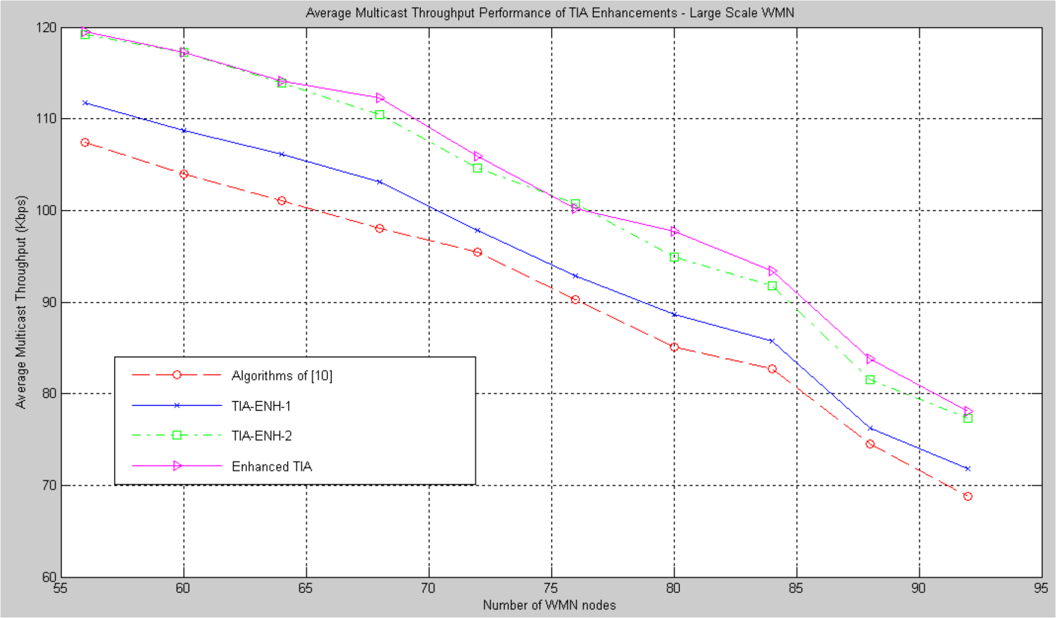


Figure 6.15. Average Multicast Throughput Performance of TIA Enhancements for Large Scale WMN.

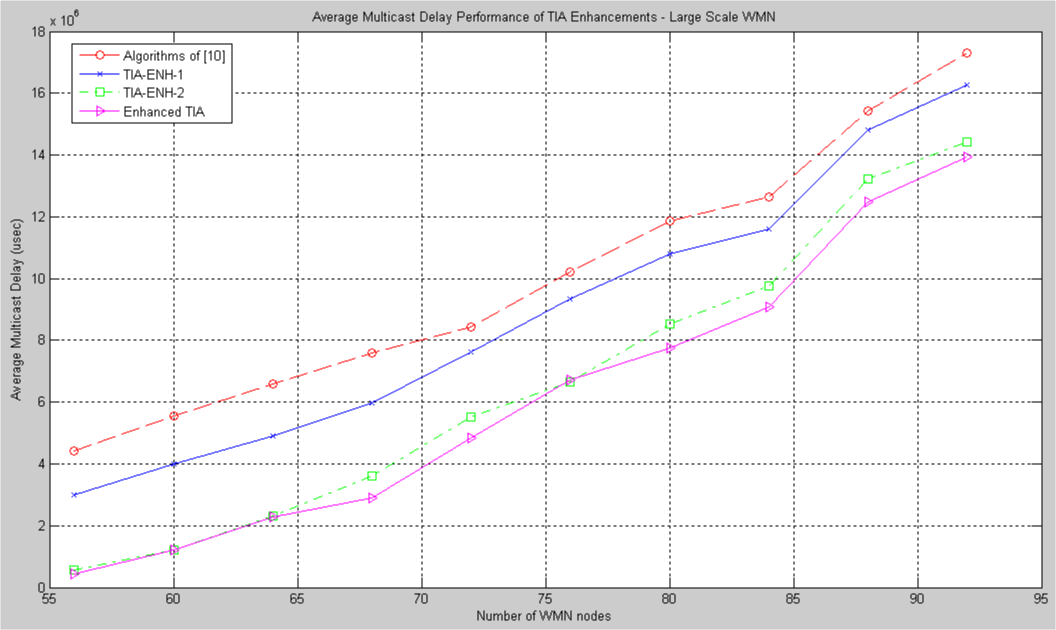


Figure 6.16. Average Multicast Delay Performance of TIA Enhancements for Large Scale WMN.

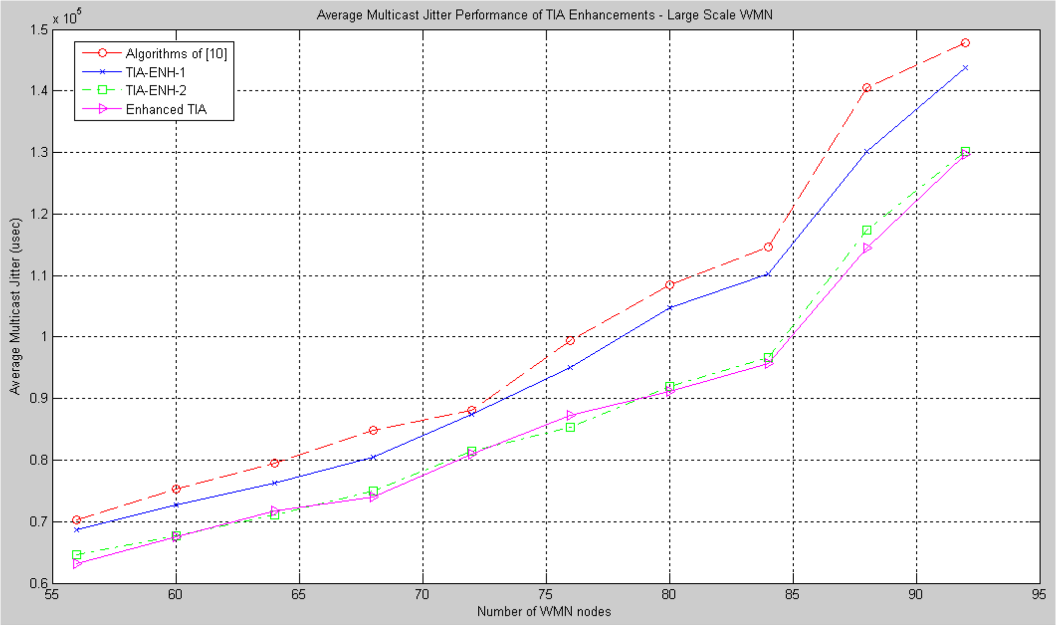


Figure 6.17. Average Multicast Jitter Performance of TIA Enhancements for Large Scale WMN.

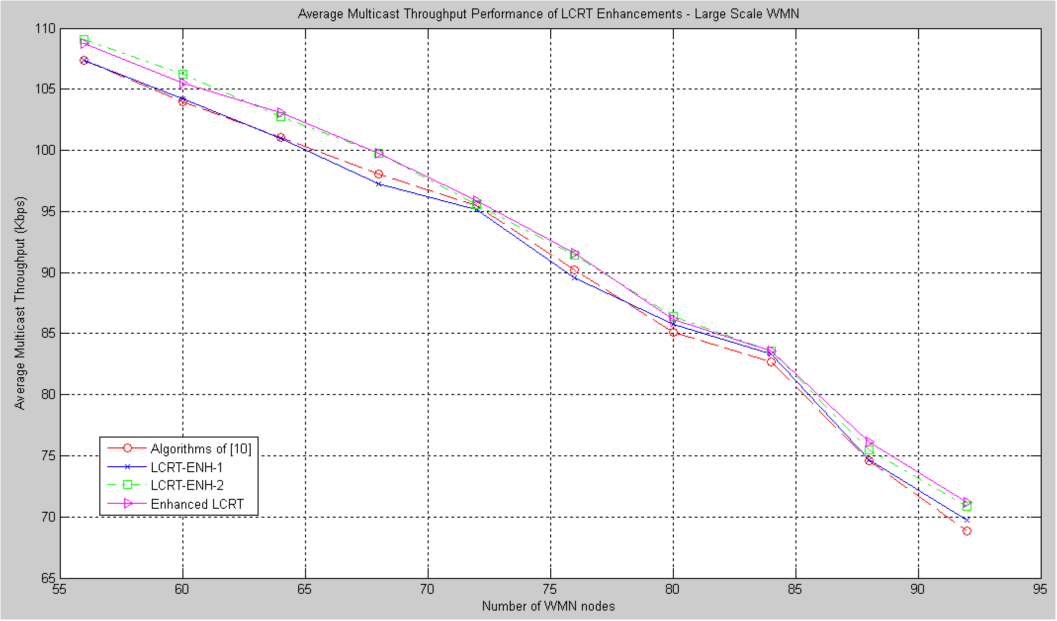


Figure 6.18. Average Multicast Throughput Performance of LCRT Enhancements for Large Scale WMN.

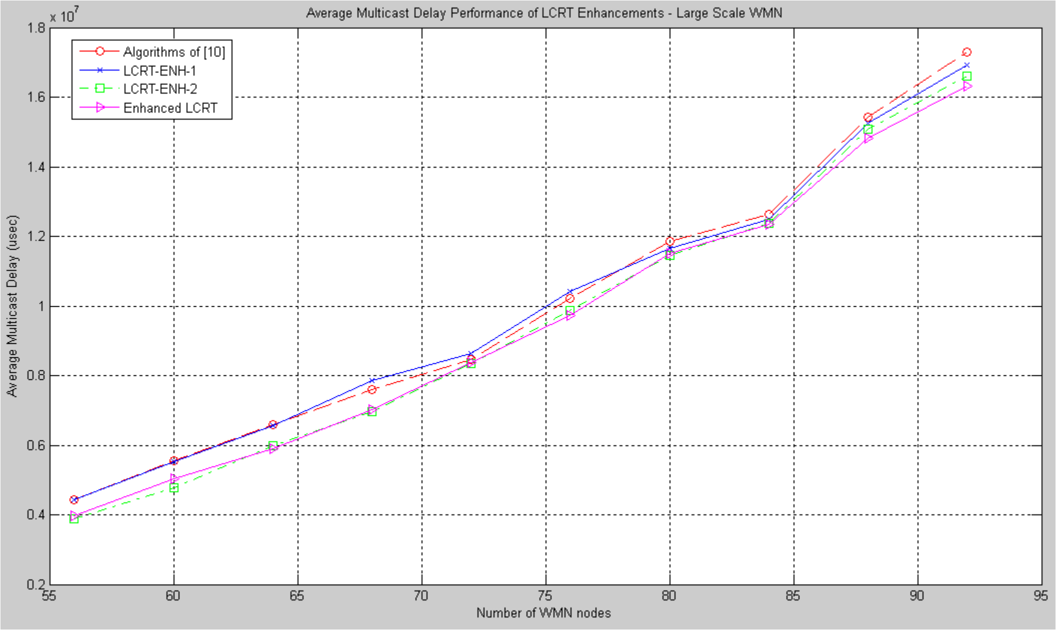


Figure 6.19. Average Multicast Delay Performance of LCRT Enhancements for Large Scale WMN.

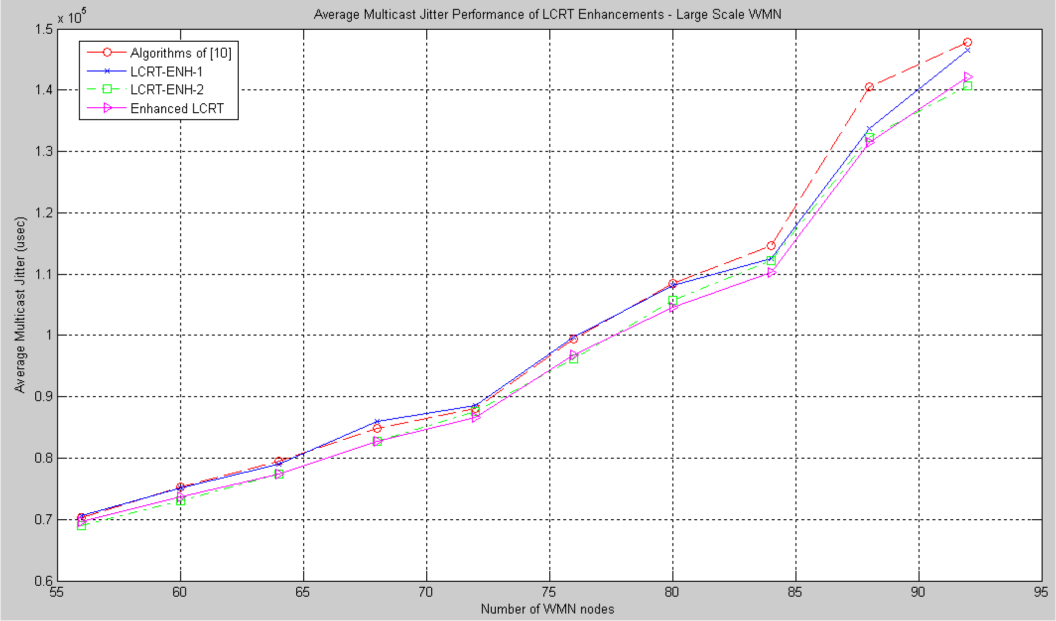


Figure 6.20. Average Multicast Jitter Performance of LCRT Enhancements for Large Scale WMN.

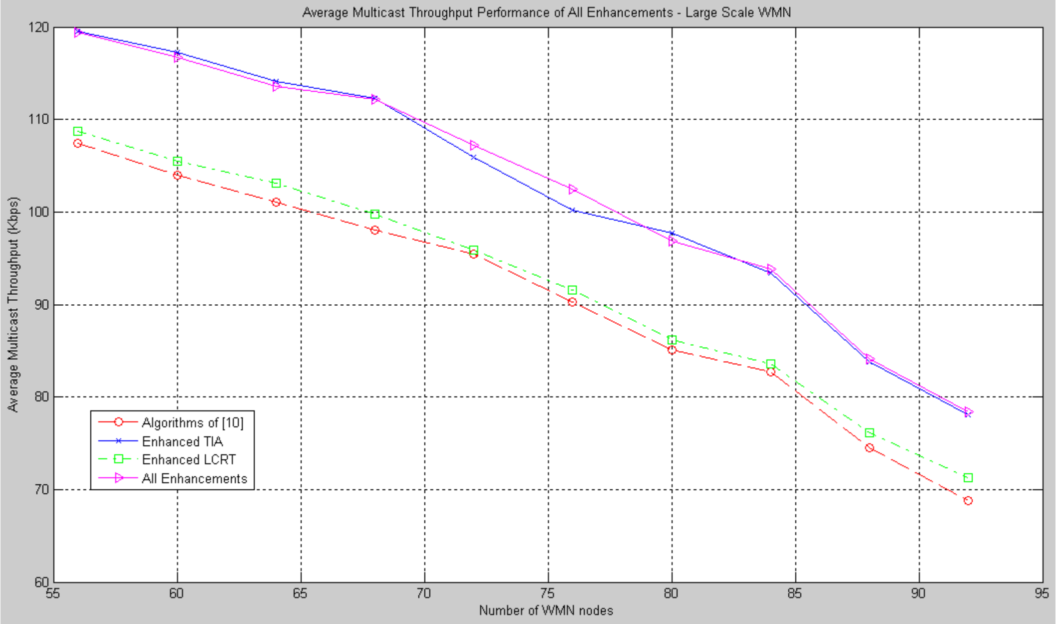


Figure 6.21. Average Multicast Throughput Performance of All Enhancements for Large Scale WMN.

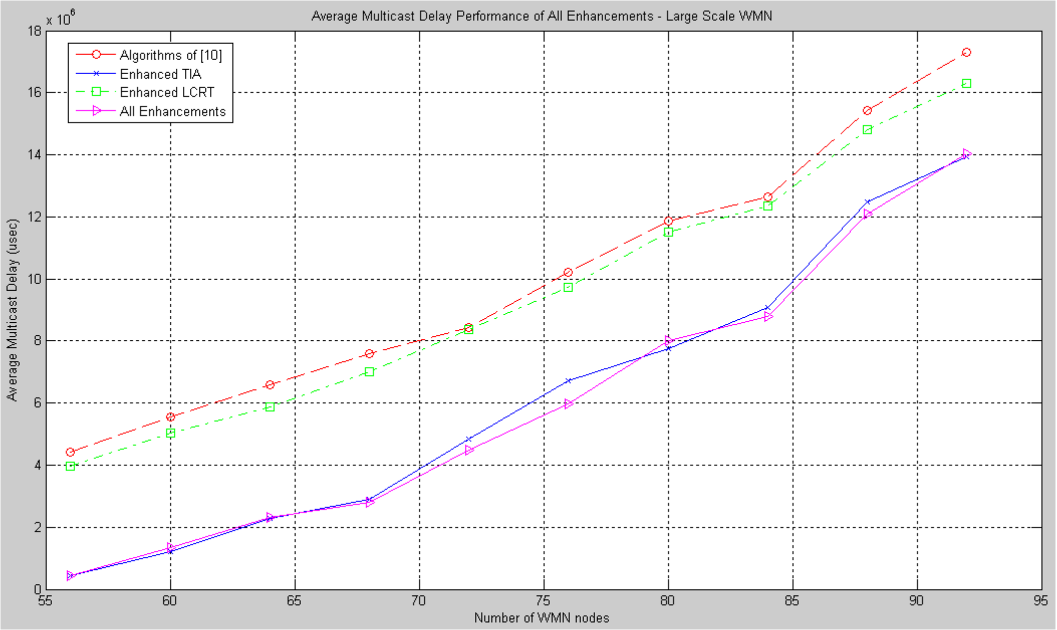


Figure 6.22. Average Multicast Delay Performance of All Enhancements for Large Scale WMN.

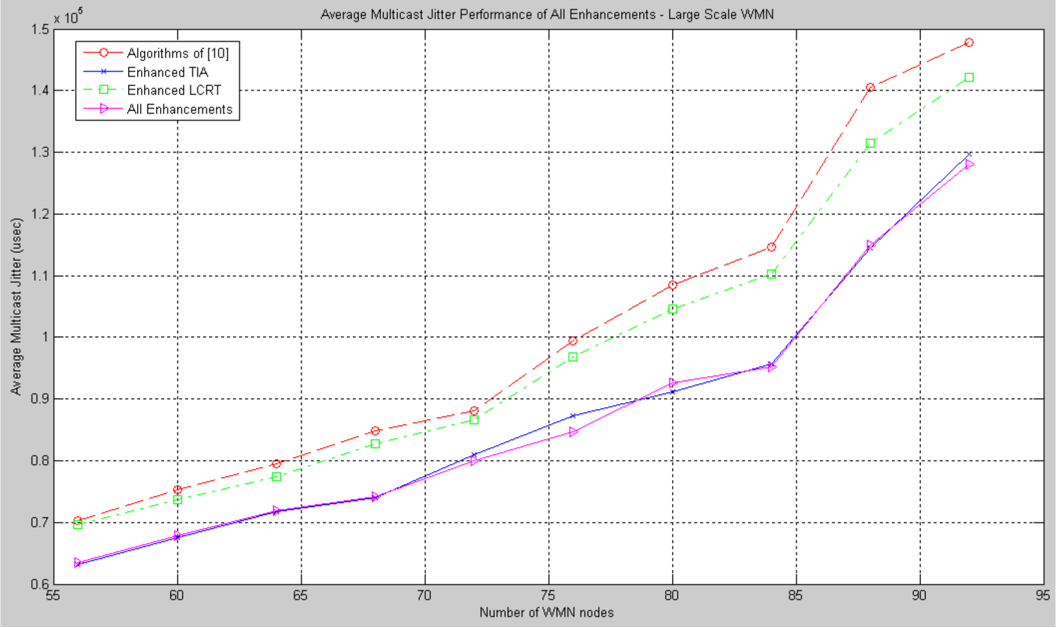


Figure 6.23. Average Multicast Jitter Performance of All Enhancements for Large Scale WMN.

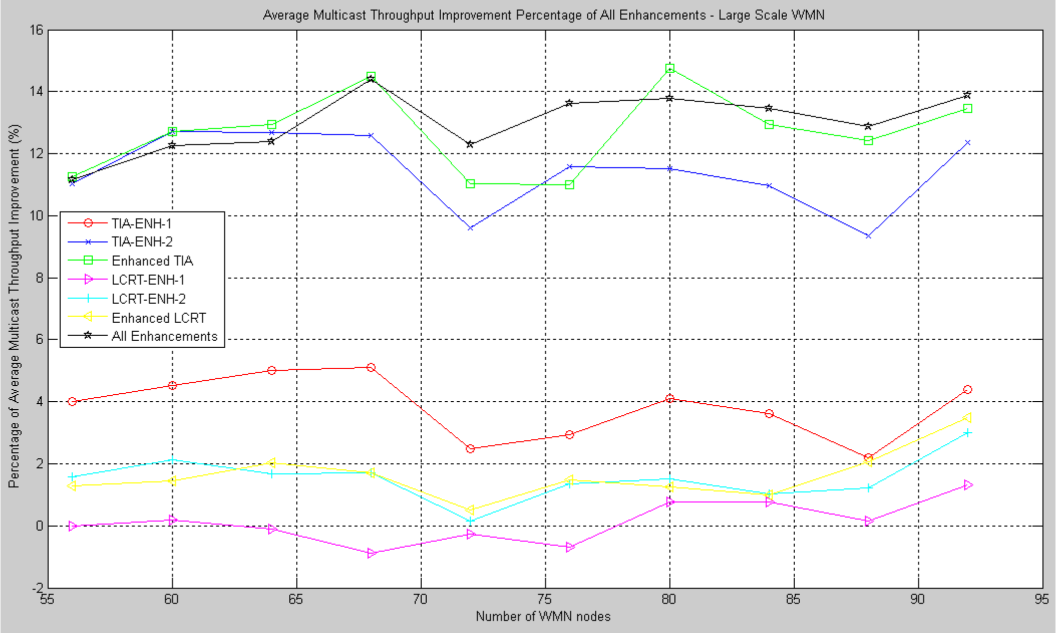


Figure 6.24. Average Multicast Throughput Improvement Performance of All Enhancements for Large Scale WMN.

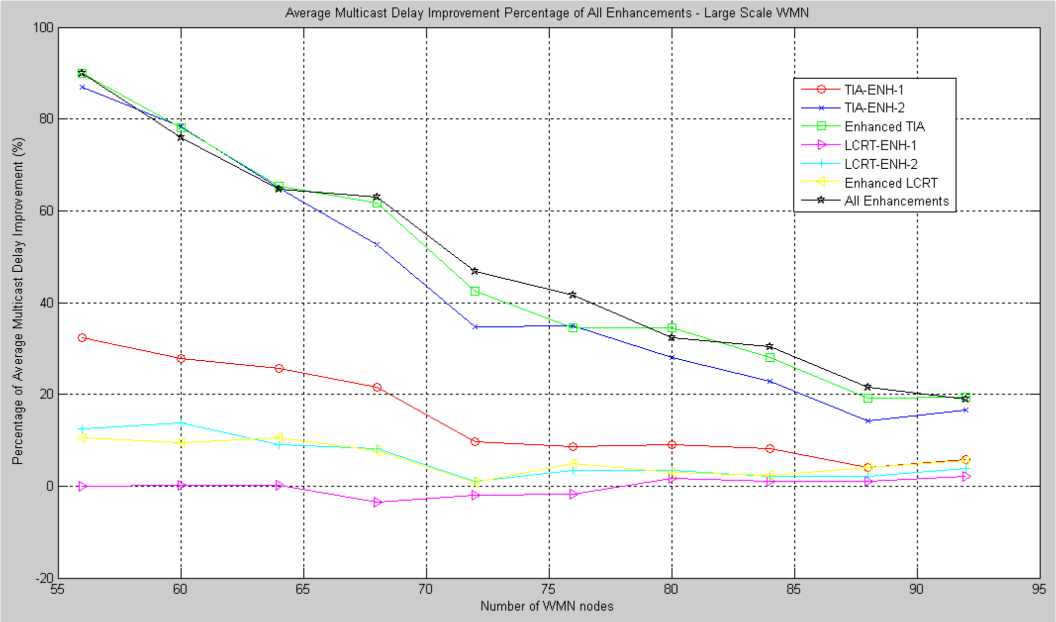


Figure 6.25. Average Multicast Delay Improvement Performance of All Enhancements for Large Scale WMN.

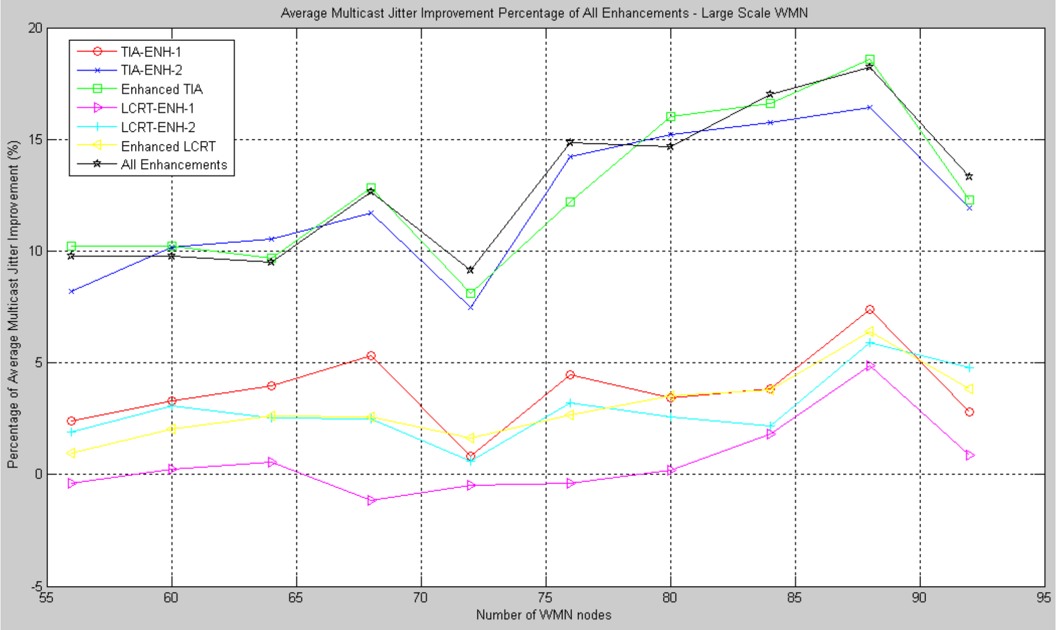


Figure 6.26. Average Multicast Jitter Improvement Performance of All Enhancements for Large Scale WMN.

6.3. Evaluation of the Simulations

The average values of the average throughput, delay, jitter performance improvement percentages of the enhancements for small scale and large scale WMN simulations are stated at Table 6.2 and Table 6.3 respectively.

For small scale simulation, it can easily be seen that TIA algorithms have superior performance over LCRT algorithms. In fact, LCRT algorithm seems that it is not useful at these kind of networks. Although it was developed to resolve some special case deficiencies of the LCRT algorithm, it could not be able to improve the performance of the network because of the lack of that special case. Indeed, we could also be conclude that LCRT algorithms may harm the system if they are used solely.

Table 6.2. Average improvement percentages for small scale WMN.

	Avg.Throughput (%)	**Avg.Delay (%)**	**Avg.Jitter (%)**
TIA-ENH-1	3.5844	10.5581	3.0494
TIA-ENH-2	1.9532	-0.0889	4.2103
Enhanced TIA	11.3342	27.0334	15.6542
LCRT-ENH-1	-0.1179	-0.3513	-2.5496
LCRT-ENH-2	0.4163	-0.1666	-0.5734
Enhanced LCRT	0.1195	-0.4048	-1.1174
All Enhancements	11.0449	26.1032	15.1548

As for the large scale network simulation, unlike the case at the simulation for small scale WMN, LCRT improves the performance. However, if we compare the Enhanced TIA and All Enhancements, we could see that including the LCRT enhancements to the algorithm set slightly reduce the improvement percentage of the Enhanced TIA.

After investigating the results of both simulations, it can also be noted that pro-

Table 6.3. Average improvement percentages for large scale WMN.

	Avg.Throughput (%)	**Avg.Delay (%)**	**Avg.Jitter (%)**
TIA-ENH-1	3.8421	15.2545	3.7755
TIA-ENH-2	11.4407	43.3934	12.1666
Enhanced TIA	12.7063	47.3102	12.6734
LCRT-ENH-1	0.1229	-0.0901	0.6152
LCRT-ENH-2	1.5271	5.9705	2.9189
Enhanced LCRT	1.6257	5.8718	3.0098
All Enhancements	13.0111	48.5400	12.8879

posed algorithms improve the delay performance much more than throughput and jitter metric. That makes them useful for large scale networks which have lower performance because of their relatively high average delay.

7. CONCLUSION

Within the scope of the thesis work, four enhancements have been developed over the algorithms TIA and LCRT of [10]. Additionally, in order to be able to evaluate the performance of the proposed enhancements, a simulation programme is written by using C++ programming language [20, 21] and Borland C++ Builder 6 [22] as the compiler.

After running simulations, it is observed that the enhanced algorithms give superior performance for the special cases that they are developed for. In detail, we can conclude that:

- Enhanced TIA enhancements have the best performance amongst all proposed enhancements.
- Since they are all for some special cases, they do not give same improvement performance for every topology that have same number of WMN nodes.
- Especially, if we are dealing with small scale network, TIA-ENH-1 is performing very well, because of its special condition that access areas should be nested.
- For TIA-ENH-2, it is obvious that the source access area has to have more than one GW and in order it to be efficient there should also be some neighboring access areas.
- For LCRT enhancements, since the conditions seems to happen not so often, it seems it is not so efficient.

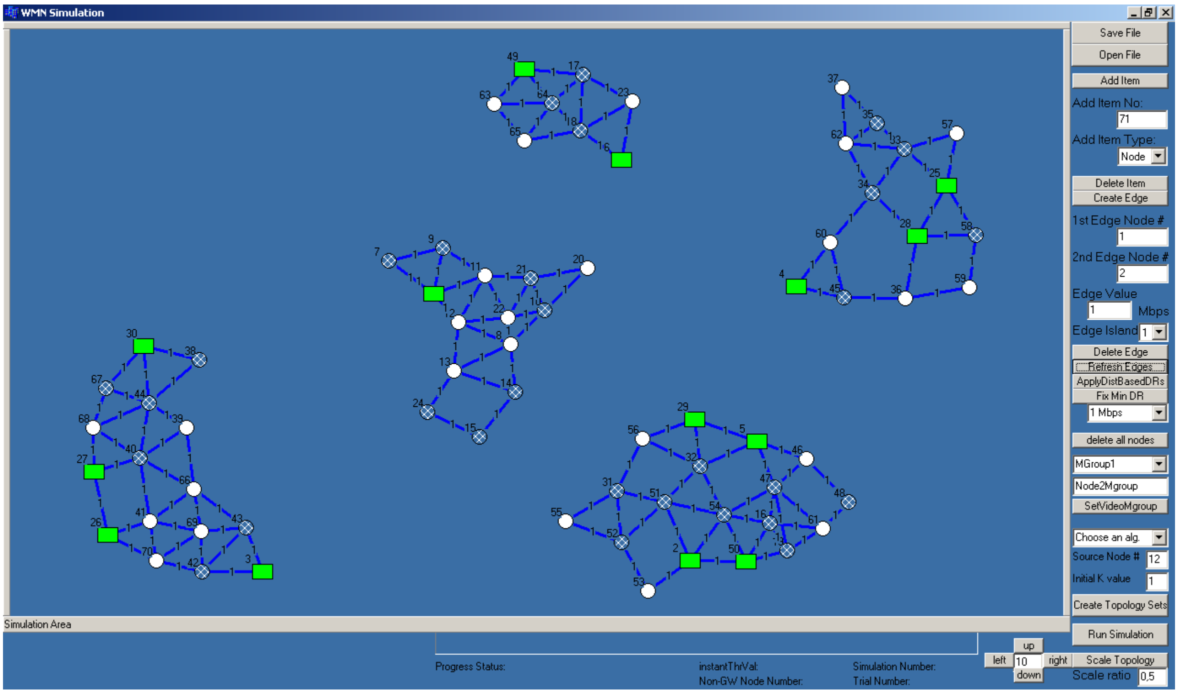
All in all, if we take the mean of the improvement percentages, which can be found at Table 6.2 and Table 6.3, it could be easily observed that implementing just the Enhanced TIA algorithms makes more sense instead of using all of the enhancements. Hence, we can conclude that even though LCRT enhancements have advancements for some of networks which contains the inspiring special case of LCRT (which is described in detail at Section 4.2), it is better to use just the Enhanced TIA algorithm set in order to be able to get the improvement percentages stated at Table 7.1.

Table 7.1. Average improvement percentages for Enhanced TIA algorithm set.

	Avg.Throughput (%)	**Avg.Delay (%)**	**Avg.Jitter (%)**
Small scale WMN	11.3342	27.0334	15.6542
Large scale WMN	12.7063	47.3102	12.6734

APPENDIX A: USER GUIDE OF THE SIMULATION PROGRAMME

At this appendix chapter, the basic user manual for the GUI (Graphical User Interface) of the written simulation programme is stated. General view of the simulation GUI is shown at Figure A.1.



Add Item No: 71

Add Item Type: Node

1st Edge Node # 1

2nd Edge Node # 2

Edge Value 1 Mbps

Edge Island 1

1 Mbps

MGroup1

Node2Mgroup

Choose an alg.

Source Node # 12

Initial K value 1

10

Scale ratio 0,5

Figure A.1. General view of the simulation GUI.

In order to create our own topology, the GUI has some buttons like “Add Item”, “Delete Item”, “Create Edge”, “Delete Edge” and “delete all nodes”.

To add an item to the topology, we should first specify the item number and type of the item by filling the related fields. As a type, an item can either be a node or be a gateway. As for the item number, we could just choose an unused item number; otherwise, we get an error message indicating that the number is already in use.

For deleting an item we should just specify the item number of it and click the

“Delete Item” button. Additionally, if we want to delete all of the items we can use the “delete all nodes” button to achieve that.

Similarly, to create an edge we should specify the end points of the edge and the edge value. For future use, there is also an option to choose a frequency island by selecting an existing island id from the appropriate field. However, it is not necessary for our simulation and is not advised unless the island id is same for all edges.

To delete an existing edge, program needs just the endpoint information regardless of the edge value.

The “FixMinDR” button is to change data rates according to the predefined distance rules.

In addition, we can add a node to a multicast group by entering its item number to the “Node2Mgroup” field, then selecting the “Mgroup” from the related field and clicking the “SetVideoGroup” button. Invertedly, we can also unset an existing video multicast member by choosing “Unset the node” instead of “Mgroup”.

At the buttom right corner part of the GUI, there are some buttons to move the whole topology either upwards, downwards, right side or left side. Additionally, the topology can be scaled with a specified scale ratio by using “Scale Topology” button.

To modify the locations of the items, they can be drag and dropped to any place within the simulation area.

To save the created topology, “Save File” button can be used to write the topology information to a specified text file. After saving our topology, we become able to open it by using “Open File” button at any time.

Finally, in order to run the simulation we should follow the below mentioned two steps:

(i) Create the topology sets.
(ii) Run simulation.

By clicking the "Create Topology Sets" button, program starts to randomly enlarge the active topology with the predefined number of nodes (ADDEDNODENUMFORONESTEPTOPOLOGYINCREMENT) until it reaches the predefined number of WMN nodes (MAXNODENUMFORTOPOLOGYINCREMENTOPERATION). At the same time, it saves the topology sets in order to be able to use them later.

After creating the topologies, now we are ready to implement and simulate our algorithms by using the "Run Simulation" button. After clicking the button, the program starts to run all of the eight algorithm sets one by one and automatically creates the results at the same folder where it gets the topology files. Beyond the completion of the simulation run, resulting files will be ready to be processed by using the written Matlab code in order to get the resulting graphs of the simulation.

REFERENCES

1. Jinsung, L., L. Junhee and C. Song, "A Realizable Cross-Layer Architecture for Wireless Mesh Networks", *3rd International Conference on Future Internet Technologies (CFI'08)*, 2006.

2. Roos, D., "How Wireless Mesh Networks Work", http://computer.howstuffworks.com/how-wireless-mesh-networks-work.htm, accessed at June 2007.

3. Akyildiz, I. F., X. Wang and W. Wang, "Wireless Mesh Networks: A Survey", *Elsevier Computer Networks Journal*, Vol. 47, No. 4, pp. 445–487, 2005.

4. Akyildiz, I. F. and X. Wang, "A Survey on Wireless Mesh Networks", *IEEE Communications Magazine*, Vol. 43, No. 9, pp. 23–30, 2005.

5. Zhu, X., T. Schierl, T. Wiegand and B. Girod, "Video Multicast over Wireless Mesh Networks with Scalable Video Coding (SVC)", *Visual Communications and Image Processing, Proceedings of the SPIE (Society of Photographic Instrumentation Engineers)*, 2008.

6. Psannis, K. P. and Y. Ishibashi, "Impact of Video Coding on Delay and Jitter in 3G Wireless Video Multicast Services", *EURASIP Journal on Wireless Communications and Networking*, Vol. 2006, pp. 1–7, May 2006.

7. Zhu, X. and B. Girod, "Video Streaming over Wireless Networks", *Proceedings of the European Signal Processing Conference (EUSIPCO'07)*, pp. 1462–1466, 2007.

8. Majumdar, A., D. G. Sachs, I. V. Kozintsev, K. Ramchandran and M. M. Yeung, "Multicast and Unicast Real-Time Video Streaming over Wireless LANs", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 6, pp. 524–534, 2002.

9. Haratcherev, I., J. Taal, K. Langendoen, R. Lagendijk and H. Sips, “Optimized Video Streaming over 802.11 by Cross-Layer Signaling”, *IEEE Communications Magazine*, Vol. 44, No. 1, pp. 115–121, 2006.

10. Tu, W., C. J. Sreenan, C. T. Chou, A. Misra and S. Jha, “Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks”, *In Proceedings of IEEE International Conference on Network Protocols (ICNP'08)*, 2008.

11. Liu, B., C. Chou, A. Misra and S. Jha, “Resource-Aware Routing of Broadcast and Multicast in Multi-Rate Wireless Mesh Networks”, *Mobile Networks and Applications*, Vol. 13, No. 1–2, pp. 38–53, 2008.

12. Chou, C., A. Misra and J. Qadir, “Low Latency Broadcast in Multi-Rate Wireless Mesh Networks”, *IEEE JSAC (Journal on Selected Areas in Communications) Special Issue on Multi-hop Wireless Mesh Networks*, 2006.

13. Wang, T., X. Du, W. Cheng, Z. Yang and W. Liu, “A Fast Broadcast Tree Construction in Multi-Rate Wireless Mesh Networks”, *In Proceedings of IEEE International Conference on Communications (ICC'07)*, 2007.

14. Shakkottai, S., X. Liu and R. Srikant, “The Multicast Capacity of Large Multihop Wireless Networks”, *In Proceedings of ACM MobiHoc'07*, pp. 247–255, 2007.

15. Liu, B., P. Thiran and D. Towsley, “Capacity of a Wireless Ad Hoc Network with Infrastructure”, *In Proceedings of ACM MobiHoc'07*, pp. 239–246, 2007.

16. Ruiz, P. M., F. J. Galera, C. Jelger and T. Noel, “Efficient Multicast Routing in Wireless Mesh Networks Connected to Internet”, *In Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, 2006.

17. Jelger, C. and T. Noel, “Proactive Address Autoconfiguration and Prefix Continuity in IPv6 Hybrid Ad Hoc Networks”, *In Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*

(SECON'05), 2005.

18. Amir, Y., C. Danilov, R. M. Elefteri and N. Rivera, "An Inter-Domain Routing Protocol for Multi-Homed Wireless Mesh Networks", *In Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*, 2007.

19. Tu, W., C. J. Sreenan, C. T. Chou, A. Misra and S. Jha, "Resource-Aware Video Multicasting via Access Gateways in Wireless Mesh Networks", *To appear in IEEE Transactions on Mobile Computing*, 2012.

20. Deitel, P. J. and H. M. Deitel, *C++ How to Program*, 4th ed., Prentice Hall, New Jersey, 2002.

21. Dale, N., *C++ Plus Data Structures*, 4th ed., Jones and Bartlett, Mississauga, 2006.

22. Hollingworth, J., B. Swart, M. Cashman and P. Gustavson, *Borland C++ Builder 6 Developer's Guide*, 2nd ed., Sams, Indianapolis, 2002.

23. Modiano, E., "Communication Systems Engineering", http://dspace.mit.edu/handle/1721.1/55894, accessed at February 2003.

24. Brenner, P., "A Technical Tutorial on the IEEE 802.11 Protocol", http://www.sss-mag.com/pdf/802_11tut.pdf, accessed at July 1997.

25. IEEE, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", http://standards.ieee.org/getieee802/download/802.3-2008_section1.pdf, accessed at December 2008.

26. IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", http://standards.ieee.org/getieee802/download/802.11-2007.pdf,

accessed at June 2007.

27. Gast, M., *802.11 Wireless Networks*, 1st ed., O'Reilly Media, Sebastopol, 2002.

28. Jimenez, O. S., "RF Lecture Series, Modulation Fundamentals 3, 802.11b/a/g Modulation Standard", http://www1.verigy.com/cntrprod/groups/public/documents/file/wcmd_001437.pdf, accessed at December 2008.

29. Gast, M., "When is 54 Not Equal to 54? A Look at 802.11a, b, and g Throughput", http://www.oreillynet.com/wireless/2003/08/08/wireless_throughput.html, accessed at August 2003.

30. Stark, H. and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*, 3rd ed., Prentice Hall, New Jersey, 2001.

31. Adas, A., "Traffic Models in Broadband Networks", *IEEE Communications Magazine*, Vol. 35, No. 7, pp. 82–89, 1997.

32. Cummings, C. E. and P. Alfke, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons", *SNUG (Synopsys Users Group Conference)*, 2002.

33. Magrab, E. B., S. Azarm, B. Balanchandran, J. Duncan, K. Herold and G. Walsh, *Engineers Guide to Matlab*, 3rd ed., Prentice Hall, New Jersey, 2010.