Deanship of Graduate Studies

Al-Quds University

InfiniBand-Based Mechanism to Enhance Multipath QoS in MANETs

Mohammad Yousef Mohammad Ismail

M.Sc. Thesis

Jerusalem-Palestine

1438/2017

InfiniBand-Based Mechanism to Enhance Multipath QoS in MANETs

Prepared By:

Mohammad Yosef Mohammad Ismail

B.Sc.: Computer Science, Al-Quds University-Palestine

Supervisor: Dr. Raid Zaghal

Co- supervisor: Dr. Said Salah

A thesis Submitted in Partial fulfillment of requirements for the degree of Master of Computer Science / Department of Computer Science / Faculty of Graduate Studies – Al-Quds University.

1438/2017

Al-Quds University
Deanship of Graduate Studies
Computer Science Department

Thesis Approval

InfiniBand-Based Mechanism to Enhance Multipath QoS in MANETs

Prepared By: Mohammad Yousef Mohammad Ismail

Registration No: 21213054

Supervisor: Dr. Raid Zaghal

Co- supervisor: Dr. Said Salah

Master thesis submitted and accepted. Date: 18 / 6 /2017

The names and signatures of the examining committee members are as follows:

1- Head of Committee: Dr. Raid Zaghal Signature: ....

2- Co-Supervisor: Dr. Saeed Salah, Signature: ....

3- Internal Examiner: Dr. Nidal Kafri Signature: ....

4- External Examiner: Dr. Iyad Tumar Signature: ....

Jerusalem – Palestine
1438/2017

Dedication

This work is dedicated…

To my parents, for giving birth to me at the first place and supporting me spiritually throughout my life.

To my beloved wife for the endless support.

Special dedication

To my grandmother, cousins, uncle and grandfather souls

Mohammad Yousef Mohammad Ismail

Declaration

I certify that this thesis submitted for the degree of Master, is the result of my own research, except where otherwise acknowledged, and that this study (or any part of the same) has not been submitted for a higher degree to any other university or institution.

Signed…………………………………

Mohammad Yousef Mohammad Ismail

Date: 22 / 8 /2017

Acknowledgements

I would like to express my deepest gratitude to Dr. Raid Zaghal and Dr. Said Salah for providing their technical expertise and support throughout the entire thesis process. I appreciate their patience, caring, their motivation and immense knowledge. Furthermore, I would like to thank the rest of the thesis committee and computer science department for all their support during my research.

Abstract

In Mobile Ad-hoc Networks (MANETs), the continuous changes in topology and the big amounts of data exchanged across the network makes it difficult for a single routing algorithm to route data efficiently between nodes. MANETs usually suffer from high packet loss rates and high link failure rates, which also makes it difficult to exchange data in effective and reliable fashion. These challenges usually increase congestion on some links while other links are almost free. In this thesis, we propose a novel mechanism to enhance QoS in multipath routing protocols in MANETs based on the InfiniBand (IB) QoS architecture. The basic idea of our approach is to enhance the path balancing to reduce congestion on overloaded links. This mechanism has enabled us to give critical applications higher priority to send them packet when routing their packets across the network, effectively manage frequent connections and disconnections and thus help reduce link failures and packet loss rates, and reduce the overall power consumption as a consequence of the previous gains. We have tested the scheme on the (IBMGTSim) simulator and achieved significant improvements in QoS parameters compared to two well-known routing protocols: AODV and AOMDV.

تحسين جودة الأداء في شبكات الـ (MANET) من خلال آلية تعتمد على تقنية الـ (InfiniBand)

اعداد: محمد يوسف محمد اسماعيل
المشرف الرئيسي: د. رائد الزغل
المشرف الثاني: د. سعيد صلاح

الملخص:

هناك نوع من الشبكات حيث يكون كل المكونات فيها عبارة عن اجهزة متحركة بدون اي بنية تحتية تسمى "MANET" في هذا النوع من الشبكات تتعاون الاجهزة ذاتيا لتحديد الطرق في ما بينها ولانها متحركة تقوم هذه الاجهزة بحساب اكثر من طريق عوضًا عن حساب طريق واحد لتقليل من احتمالية فشل في الارسال حيث اذا تم فشل في طريق معينة تبقى الطرق الاخرة سليمة.

وفي ناحية اخرى ولتنوع اهمية البرامج والخدمات التي توفرها هذه الاجهزة هناك ما يسمى "بجودى الخدمات Quality of Service "حيث يقوم المستخدم بوضع اولويات للبرامج والخدمات من استهلاك المصادر المتاحة, والطريق الشائعة هي ان يقوم المستخدم بوضع حدود على سرعة استعمال الشبكة من قبل البرامج الاقل اهمية لترك المصادر متاحة للبرامج لاكثر المهمة بشكل اكثر وهذا الحل يحتوي على الكثير من المشاكل في هذا النوع من الشبكات, حيث ان مواصفات الطرق غير معروفة وغير ثابتة وقد تحتوي او تتغير الى قيم اقل من الحدود الموضوعة للبرمج الغير مهمة فتتساوى البرامج والخدمات الاقل اهمية بالبرامج الاكثر اهمية مما يعني فشل في جودة الخدمات.

من خلال بحثنا عن حلول ودراسة انواع مختلفة من الشبكات وجدنا نوع من تطبيق جودة الخدمات في نوع الشبكات المسمى بInfiniBand حيث يتم تطبيق جودة الخدمات من خلال تغيير عدد الرسال المبعثة من قبل البرامج, حيث تقوم الاجهزة بارسال عدد اكبر من الرسال التابعة للبرامج المهمة مقارنة بعدد الرسال التابعة للبرامج الاقل اهمية, ويتم ذلك باستخدام الصفوف, حيث تصطف الرسال من البرامج المهمة بصف يختلف عن الصف الذي يحتوي على رسال البرامج الغير مهمة.

هذا الحل له فائدتان مهمتان الاولى انه لا يوثر عالطريقة التقليدية ويمكن ان يستخدم معها والفائدة الثانية انه وبخلاف الطريقة التقليدية, الطريقة الجديدة لا تتاثر بصفات الطريق المحسوبة او بتغير صفاتها فنسبة عدد الرسال تكون نفسها مهما اختلفت الطرق و صفاتها, بعد تطبيق هذا النوع وجدنا تحسين في كفائة الارسال تصل الى 18% في جودة التوصيل و 10% في سرعة الوصول مع العلم ان جودة الخدمات لم تفشل على غرار الطريقة التقليدية.

Contents

List of Figures

List of Tables

Chapter One – Introduction

This chapter will describe the problem and art of solution, mythology, motivation and contribution of the problem and our proposal to solve it.

1.1 Introduction

A wireless network that consists of heterogeneous nodes without fixed infrastructure is called a Mobile Ad-hoc Network (MANET). It is a self-determined system of mobile routers and related hosts that communicate over wireless links. Such kind of networks suffer from limited bandwidth, packet losses, power constraints and mobility-induced route changes. Nevertheless, MANETs have a good side such as scalability, robustness and independency since nodes themselves cooperate to provide services. MANETs have many important applications such as tactical networks, Wireless Sensor Network (WSNs), Data Network and Device Networks [1].

Routing protocols can be divided into two main types: table driven (proactive) routing and on-demand (reactive) routing, each of which has its own advantage and disadvantages. On one hand, proactive routing protocols maintain fresh lists of destinations and their routes by periodically distributing routing tables throughout the network, and on other hand, reactive routing protocols find a route on demand by flooding the network with route request packets. One of the main disadvantages of proactive routing is path maintenance and slow reaction on restructuring and failures. While reactive protocols have high latency times in route finding and excessive flooding can lead to network congestion. Since MANETs do not use fixed nodes for forwarding packets from source to destination, routing is the biggest challenge in this type of network architectures.

Existing routing protocols can be either based on single-bath or multiple-paths mechanisms. In a single path routing, there is a dedicated path between the source and the destination, in which the packet header contains the complete path with all hops' information until the destination is reached. Moreover, as the MANET has mobile and continuously moving nodes, the single path routing protocols can cause serious performance degradation. These types of protocols suffer from packet loss and congestions which makes the network unreliable. However, a multipath routing scheme which is based on building multiple paths from a source to a destination can achieve better performance and can resolve many of the single path performance issues such as scalability, security and network lifetime. They can also enhance the end-to-end throughput and offer improved load balancing, fix packet loss, and increased throughput of network by using alternative paths [2]. Therefore, achieving high Quality of Service (QoS) attributes in MANETs is a big challenge since they usually contain pre-defined service performance limitations. These include end-to-end delay, available bandwidth, battery limitations, packet loss probability, etc. As previously mentioned, the main issue is the continuous movement of nodes and changing network topology which causes frequent link failures and disconnections. The QoS parameters maintain some metrics to enhance the overall system performance [3].

Routing protocols were designed decades ago when QoS, load balancing, and handling big scale networks were not really relevant as a design goal. IB was introduced to provide a high-performance computing with QoS and load balancing as core features; IB is a centralized network model which combines managers and agents collaborate to provide the best network performance that regular Internet protocols cannot reach [4,5]. By presenting the nodes in the MANET as mesh topology [24], they seem to be more understandable and can be controlled to minimize bandwidth consumption and enhance reliability. The mesh mechanism is well-known and has many features that can be combined with the IB standard to control MANET's natural chaos [24]. Using the IB multipath (Pathbits) and QoS with Virtual Lanes (VLs), we can take control up to the application level and thus improve network performance and enhance bandwidth in MANETs [6,7].

1.2 Motivation

While internet protocol was not designed to support multipath and QoS and suffer from chaotic network such as MANET or high-performance network, the world is trying to have a specific network model to run on specific network types to match the need.

Due to the existence of new network model, there are new approaches which were created as well to solve and enhance network performance in several network layers, actually, InfiniBand has come with significant network performance and with huge mechanisms to run with multipath and QoS and support them as core features.

While studying the QoS multipath, we can see how it suffers under internet protocol and how InfiniBand came with momentous result, so this increase our curiosity of mixing the InfiniBand approaches with MANET and try to compare the result with and without InfiniBand approaches, and figure out the answer of “is internet protocol is the best for MANET?”

1.3 Research Problem

At MANET, there is no network infrastructure and nodes have variant characteristics where there is a real need for multipath to achieve satisfied communication, while QoS become a used basic need that causes a cycled problematic situation since Internet model is not designed to work in such cases, but some other models such as InfiniBand has come with an approach to support QoS and multipath as major features that should be supported with significant result as this model was designed to run on high network performance.

So, some of the questions that maybe asked are:

1. Will using modified InfiniBand going to give better result at MANET?
2. Do QoS rules Achieve less overhead than Internet Model?
3. Does InfiniBand with its organized multipath will achieve a better performance than Internet model?

1.4 Contribution

This thesis takes the first step to compare MANET while using QoS with Multipath in Internet model with InfiniBand model.

This work will focus on comparing the two models modified InfiniBand and Internet, while there are variant number of nodes and nodes have variant rates and how it is affecting the QoS characteristic at which InfiniBand will give a better result than what can be achieved through the Internet model in Packet Delivery Ratio, End-to- End Delay and Control Overhead.

Our work can be counted as a first step of the same way that high performance network walks through to have its own network model, internet model cannot run on all network types while high performance network ‘HPN’ leave the Internet model to achieve a better performance we assume that can be applied on MANET network as well to have its own model that focus on solving only MANET problems leaving any chain that Internet protocol can put it.

In average, we have achieved 8% enhancement in Throughput, 26% enhancement in packets delivery, 45% in end-to-end delay, and about 15% reduction in control overhead and this average has been calculated regardless of the nodes number that exist in the network topology and regardless to the data rate used.

1.5 Thesis Outline

The thesis includes five chapters alongside this; the chapter two includes the background and related work and gives general definitions of the terms that will be used during the thesis, and afterwards it defines related works.

Chapter three describes the thesis methodology and approach that has been used, afterwards the chapter presents the scenarios that will be adopted in the thesis, as well it shows the plan of work of these scenarios. Chapter four presents the experiments and the results, in this chapter the thesis will present the experiments design and all the related plans and work implemented within the experiments, afterwards the chapter shows the results and the analysis on them using graphs and SPSS. Finally, the forth chapter describes other factors and experiments that affect QoS at multipath MANET networks and presents the results. Chapter five shows the conclusion and future work.

Chapter Two- Background and Related Work

In this chapter will describe the InfiniBand solution and how we can adapt it to work with a network as MANET beside of that, we will describe the related work and Taxonomy.

2.1 InfiniBand overview

IB is a network model used in high performance networks and has many applications in High Performance Computers (HPC) and Large-Scale Database Management Systems (DBMS) such as Google, Facebook [19] and NASA [20]. Since IB is a centralize network model that is controlled by managers and agents, it is completely opposite to MANET's nature, since it provides high performance in a completely organized fashion and thus, can be used to help organizing and managing chaotic networks like MANETs. Figure 2.1 [17] represents the layered system architecture of the IB system. It mainly consists of four layers: The upper layer to carry out transactions on the client's level; the transport layer contains que pair which presents the data communication and link application to other lower layers to allow any application allocating many que pair to communicate with other nodes; and below it there is the network layer whose function

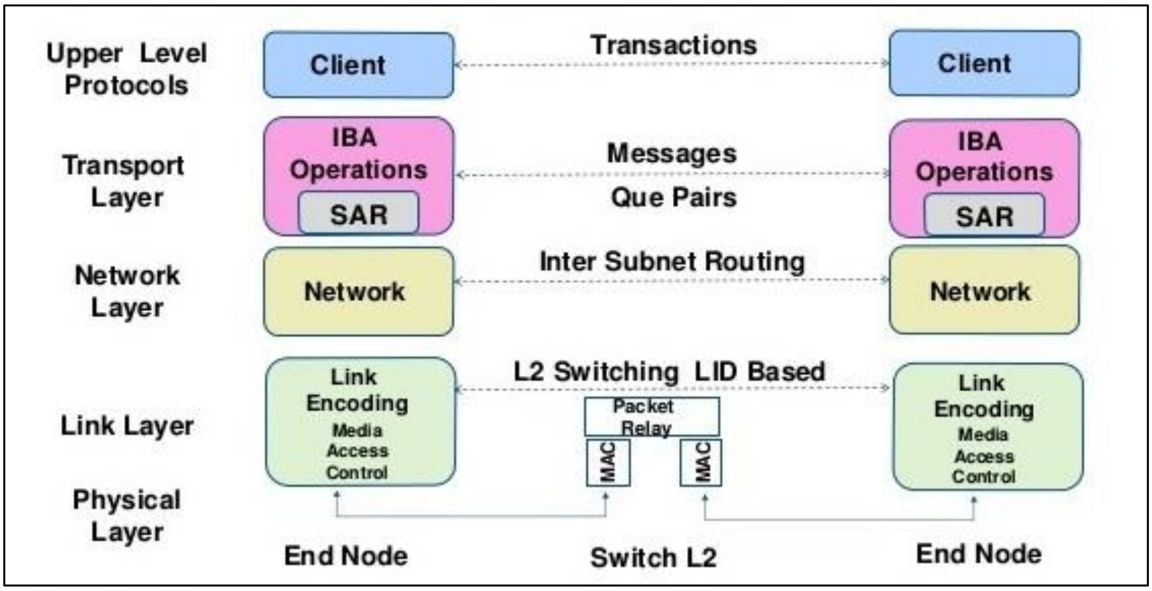

Figure 2.1 InfiniBand overview

is similar to the network layer found in the TCP/IP model. Each routing node contains the linear forwarding table that can manage the exit port of destination Global Identifier (GID) or Local Identifier (LID); and before the physical layer that presents data in physical presentation there exist the data link layer which has also the same functionality of the data link layer in the TCP/IP model suite. This layer is responsible for managing QoS, and contains VLs to manage QoS as well [21, 23].

In its design, IB supports multi-path communication; multi-paths can be used for fault tolerance, Automatic Path Migration (APM) [19], load balancing, and congestion avoidance through the right provisioning of the network, i.e., multiple parallel links can be used to increase the Constant Bisectional Bandwidth (CBB) on potential hot spots. Each node in IB has a GID that is like IPv6 in its structure (128-bit addresses). But since the 128 bits is a large number that exceeds any logical network capacity, and to minimize traffic load there is another identifier called the Local Identifier (LID). The LID is a short description of the GID that consists of a 16-bit number only. In every network, there is a file called lid2gid that maps the relationship between both identifiers: LIDs and GIDs. Each node has a LID Mask Control (LMC) that determines how many paths each node can have. LMC is the Least Significant Bit (LSB) of the LID, and Pathbits (PB) is the maximum number of possible path to reach that node in which the LID assigning process should be done without overlapping. The number of Pathbits for any node (a) can be calculated using equation 1 below [19, 23].

$$Pathbits\ (a) = 2^{LMC(a)} \text{ ....(1)}$$

and the total number of paths between two nodes, i.e., node (a) to node (b) can be calculated using equation 2.

$$total\_paths_{a\_to\_b} = Pathbits(a) * Pathbits(b) \text{ ....(2)}$$

The IB architecture supports built-in QoS mechanisms; this means that QoS services are not extensions as the case in the IEEE 802.3-based Ethernet architectures which can only provide a best-effort delivery service. Because of the inherent inclusion of QoS in the base layer in IB standard specification, there are two levels for QoS implementation: (1) QoS mechanisms that are inherently built into the basic service delivery mechanism supported by the hardware; and (2) the queuing services and management for prioritizing flows and guaranteeing service levels or bandwidths.

It is worth mentioning that a discussion of the bandwidth and latency characteristics is important because inefficiencies in these parameters can lead to congestion in the network, and this will fire congestion-control and other QoS mechanisms. If all interfaces can meet the necessary bandwidth and latency requirements, they can meet the required service levels expected by the application.

The Virtual Lanes (VLs):

In IB, QoS is supported via Virtual Lanes (VLs); these VLs are separate logic communication links (in the Data Link Layer) which share the same physical link. Each data link layer has 16 VLs from VL0 to VL15, such that VL15 is reserved for management to transmit control packets such as routing information and has the highest priority. Therefore, the applications can use VL0 to VL14 to send their data. A Service Level (SL) for a link is defined to ensure its QoS level. Each link along a given path can have a different VL, and the SL defines the desired priority of communication. Each switch/router has an SL to VL mapping table that is set by the subnet manager to keep the proper priority with the number of VLs supported on each link. Therefore, the IB Architecture can ensure end-to-end QoS through switches, routers and over the long haul [6, 21].

VLs from VL0 to VL14 are bound to two quotas: VL-high and VL-low. The VL-high quota sets the high order transmission rate while the VL-low quota sets the low order rate; The VL-low transmit only if the VL-high counter becomes 0 or if all the VL-highs have no packets to send [19, 21]. This mechanism is called VL arbitration. The VL and VL arbitration mechanism is defined by the user to achieve the target QoS level. Each VL is assigned to a specific SL since the relationship between them can be presented by many-to-one relation; which means that each VL can be bound to one SL and/or many VLs can be bound to a single SL – this mapping is stored in a table called the SL/ VL table [19, 21]. At each node, when a packet is received, it looks at the SL/VL table, if it has VL15 it is processed immediately, otherwise, it is inserted into the VL queues to be forwarded to the next hop according to the QoS rules specification [17, 19].

Suppose this quota for example is defined in a QoS configuration file:

VL high: 5VL_0, 2VL_1, 3VL_2, VL_3

VL low:10VL_{11}, 3VL_{12}, 2VL_{13}, 4$VL_{14.}$

VL high counter 1000:10

We conclude:

- The VLs 0, 1, 2 and 3 belong to high VLs quota and VLs 11,12,13 and 14 belong to low VLs quota.
- Then VLs high reserve the physical Layer and transmit packets VL0 should send 5 packets at most and then release transmit to VL1 to send at max 2 packets and so on.
- The ratio of locate traffic between high quota to low quota is 1000:10, after sending 1000 packet from VL high quota, then VL low will send ten packets.

By using this mechanism, we can assign important service level to high quota and low importance service level to low quota and we can achieve the QoS target in additional to downsizing the path capability.

2.2 Related Work

Vidwans et al. [8] proposed a congestion control mechanism for Ad hoc On-demand Multipath Distance Vector Routing (AOMDV). The mechanism uses rate base data sending scheme and queue base congestion control to improve the performance. The results have shown that packet delivery and control overhead are enhanced.

Abhinav L. et al. [9] proposed a methodology to enhance the QoS based on Node-Disjoint Multipath Routing (NDMR) using multiple node-disjoint paths to assign packets to paths in the best possible way, and handle some limitations in MANETs.

Sanguankotchakorn et al. [10] proposed an approach to increase packet delivery ratio by forwarding a smaller number of control packets to maintain route discovery and route failure, and thus managed to reduce control overhead.

Francisco J. et al. [6] came-up with the description of the QoS mechanism in IB using VL with better results. Zhu B. et al. [11] proposed an adaptive load balancing routing algorithm. Unlike other works, this algorithm has relied on the forwarding statistics, nodes distribution, and load status to achieve load balancing in the discovery stage.

Kaixin X. et al. [12] proposed a QoS architecture that is suitable for large scale networks. This architecture was established with the help of Landmark Routing (LANMAR); a scalable routing algorithm to push the work into the source nodes and limit the work being performed in the intermediate nodes without the need for any state information.

Chengyong L. et al. [13] presented a load balancing mechanism that is useful for increasing reliability and network throughput. Their approach relied on the modification of Ad hoc On-Demand Distance Vector Routing (AODV) protocol via an extension to utilize the node's resolvable load information and fairly distribute the load across the network.

Fujian Q. et al. [14] introduced an algorithm based on Dynamic Source Routing (DSR) discovery and maintenance mechanism to deal with route failures and enhance the load balancing without incurring network overhead. Their simulation results have shown that packet delivery ratio was high.

Pradeep B. et al. [15] proposed a detailed packet-layer model in which the physical layer and Media Access Control (MAC) address are used to study the traffic performance of both the AODV and the QoSAODV protocols.

Francisco et al. [16] compared the QoS results after using VLs as a part of QoS mechanism. The result has shown that using VLs gave better results, especially for big topologies.

J. Pelissier [17] described the use of the IB mechanism in enhancing QoS, and showed that IB-based QoS mechanisms provide better performance measures over traditional QoS mechanisms.

Ryan E. et al. [18] introduced the QoS main features for socket-based communication over IB networks, and concluded that IB can provide the opportunity to enact service differentiation for traditional socket-based applications over high-performance networks.

Our work merge between modern QoS mechanism that does not exist in MANET with MANET to perform a better performance, while keep QoS rules applied.

By using queues to achieve QoS, by controlling the number of passing packets, which is not affect by changing paths while transmit unlikely the traditional QoS approach, which is coupling with path characteristics.

2.3 Taxonomy

Current MANET routing algorithms can be categorized in two categories proactive and reactive.

Proactive uses Destination Sequence Distance Vector or DSDV router designed with Bellmann-Ford algorithm. In this protocol, all the nodes maintain the information about the next node. All the mobile nodes of this protocol have to relay its entries to its adjacent nodes. The nodes that lay in the route pass the data packet from one node to the other node after mutual agreement therefore all the nodes must constantly update their position in DSDV protocol so that there is no interruption in the route.

While reactive protocols Ad hoc On-Demand Distance Vector or AODV and Temporary Ordering Routing Algorithm or TORA. In AODV routing protocol the node works independently and does not carry the information of nodes adjacent to it or the information of the other nodes in the network. They work only when a data is delivered to them so as to maintain the route to the destination. These nodes have the information of the route through which data has to be delivered so they pass the packet to the next node in predetermined route. TORA is a very efficient and adaptive algorithm as it works up all the shortest possible routes from source to destination. This protocol is able to ensure the creation of route, journey of the data and erase the route in case there is partition in the network. In this protocol, every node carries the information of its neighboring nodes.

Actually, reactive has many advantages over proactive protocols in following network properties:

- The delivery of packet data is much more efficient in Reactive Protocols than in Proactive Protocols.
- Reactive Protocols are much faster in performance than Proactive protocols.

- Reactive Protocols are much more adaptive and work much better in different topographies than Proactive Protocols.

Our thesis will focus on reactive routing algorithms and especially on ADOV protocol since its most commonly used and reactive routing algorithms suffer more when network is dynamic and QoS rules should be applied.

Our thesis does not present new algorithms. In fact, it studies the effect of using predefined algorithms that exist in InfiniBand network model to solve and enhance multipath and QoS with remarkable results. However, our work can be applied on other algorithms but we use reactive routing algorithm and our target is to study the effect of injection and mixing algorithms and techniques to come up with better performance result.

Chapter Three - Methodology and Approach

In this chapter, we will explain verbosely our mythology and approach and how we merge InfiniBand solution with MANET, also how we divide the VL queues and quotas to serve QoS and explain multipath using LMC.

3.1 Approach

The approach is to use some of IB features and modify them to work with environments such as MANET and to overcome QoS multipath problems since both QoS and multipath are treated better and in a more organized way in IB than in Internet model.

First, we defined a special field on the node that defines the number of supported paths, each path has its own life cycle and supports a specific QoS profile and is managed independently to the others.

Moreover, each path is calculated once and depending on its state, it may need to be recalculated or another path is used to serve QoS file, since the relation between paths and QoS profile is many to many.

Secondly, we defined 16 VLs from VL0 to VL15 for each physical port, these VLs are classified in three quotas:

1- Management VL, which is VL number 15 that has the most priority and should always be served first so that all other VLs should be blocked until this VL become empty.

2- VLs high, which are the VLs belong to important traffic these VLs are able to send packets more than the low VLs.

3- VLs low, which belong to low important traffic in the network.

In order to control the traffic each VL obeys mechanism that is called VL arbitration. The VL arbitration is define by the user who sets the number of packets each VL should send in its quota and how many packets should be sent by VLs low in every transmission cycle.

3.2 Methodology

There are two types of routing algorithms in MANET:

1- Proactive (Table-driven):

This type of protocols maintains fresh lists of destinations and their routes by periodically distributing routing tables throughout the network

2- Reactive (On-demand):

This type of protocol finds the route on demand by flooding the network with Route Request packets.

Our methodology is applied on the second type (reactive) since it suffers from High latency time in route finding and floods the network with routing request packets that affect the entire network performance. We believe that our approach by using VLs can enhance the performance for such kind algorithm that can reduce the effect of routing request packets and do not have influence on QoS rules, since our approach will depend on packets number that through from the VLs which connected to service level, so beside of serving QoS by limitation the traffic proprieties in front of low important application we can minimize the packet through of that applications.

Reactive routing algorithms are mostly applied on MANET and especially Ad hoc On-demand Multi Distance Vector ‘AOMDV’ where the source floods the network with route request packets and collect all received responses to calculate multi paths from it ‘the source’ to the target destination regardless to the QoS rule and then finds paths that can serve the minimal requirement of QoS and send data. Since the data traffic should match the QoS rule, this approach can limit the calculated paths and exclude the ones that are below the requirements and downsize the other paths with higher capabilities to match the rules, and of course this is considered as a wasting of resources.

Multipath is supported by path-bit, which is the number of paths that are calculated from a specific source. It is the least significant bits of IPv6 where we use a shortcut of it called Local Identifier [LID], which is combined of 16bits instead of 128bits ‘IPv6’, and the least significant bits are called LID mask control ‘LMC’, for example if we have a node that has a base LID = 0x8000 and LMC is 2 then the LID range for that node is [0x8000, 0x8003] which means that the node has 4 LIDs, therefore the Pathbits can be calculated as the following equation:

$$pathbit = 2^{LMC} \text{ .... } (1)$$

So, the path between any source to any destination can be presented by the following equation:

$$Total_{paths} = pathbit_{\ source} \times pathbit_{\ destination} \text{ .... } (2)$$

By replacing the Pathbits by its basic component LMC, the total paths can be represented by:

$$Total_{paths} = 2^{LMC\ source} \times 2^{LMC\ destination} \text{ .... } (3)$$

This number is fixed and the paths calculated are maintained after the source finishes sending all the data packets in case these paths are needed to be reused or are requested from other node to reach the same destination. Each path has its own state that is changed as follow:

```
//initiate paths
INIT = 1
ARMNED = 2
ACTIVE = 3
DISABLED = 4
num_paths= pow(2, lmc)
for num_path in range(num_paths):
    path = new(path)
    path.id = num_path
    path.state = INIT
```

```
for path in paths:
    switch (path.state){
        case INIT:
            if calculate(path):
                path.state = ARMED
            else:
                path.state = DISABLED
        case ARMED:
            try:
                if data:
                    path.state = ACTIVE
                    start_transmit(data)
            Except error:
                    path.state = DISABLED
        case ACTIVE:
            try:
                while data:
                    transmit(data)
                path.state = ARMED
            Except error:
                path.state = DISABLED
        case DOWN:
            if time_out == 0:
                path.state = INIT
```

Figure 3.1 Link state

Virtual Lanes and Service Levels that are applied in the IB provides two fields for marking packets with a class of service: the service level (SL) field in the LRH and the traffic class field in the GRH. The SL field is a four-bit field that may be arbitrarily used to indicate a class of service.

Deploying QoS on MANET supports the two a of QoS, both of them depend on service level which the user should set them to targeted application which hop to control.

The ethernet approach which we achieve the QoS rule by minimizing the traffic characteristic for lower priority application to let high important application use resources in sufficing way, so we limited the MTU and rate both speed and width in face of low priority application.

The InfiniBand approach which lays on VLs which combined with:

- SL/VL table
- Virtual lanes (VL),
- Virtual lane arbitration (VLarb).

SL/VL mapping table is existing on all nodes, the table which user define the relation between application SL and data link layer VL with defining the linking between application to high or low VL which serve the QoS goal,

In addition to SL, the LRH contains the VL field that indicates the virtual lane number from which packet should pass VL through.

VLs, since we have 16 VLs from VL0 to VL15 note that VL15 is located to control packets as route packets, the user has a 15 VL from VL0 to VL14 free to use and it is not a condition to use all of them, user can use as much of these VLs depends in his need to achieve the QoS goal, since that he should define all the VLs he need and any packet that hold SL that mapped to undefined VL with be dropped.

With the mapping of SL/VL which define in SL/VL table each node can have different table which can lead us that VL can be changed from node to other in this approach user can control the QoS through the network if he wants to have nodes less calibrate in routing.

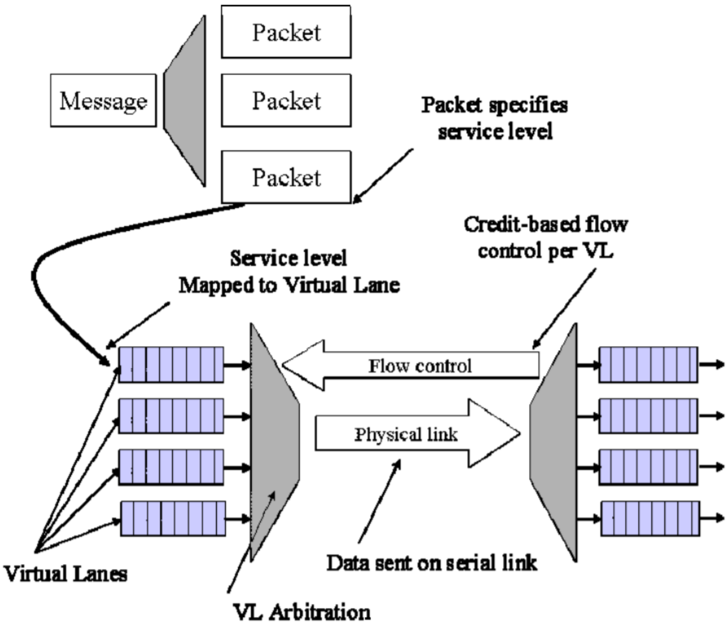

Figure 3.2 Virtual Lane Design.

The last part of IB approach is VLarb, VLarb is the controller of passing VL, VLarb split the VLs in two quotas VL high and VL low, VLarb use round robin scheduling of reserving physical layer by VL.

So since here we have three quotas listed in priority:

- Control quota which is the VL15 has the most priority and should be always empty as much as it can.
- High level quota which are chosen from VL0 to VL14 which have the second most priority after control lane. They have to send packets as much as written in the high-level limit then stop for the low-level quota to send.
- Low level quota which are also chosen from VL0 to VL14 and have the lowest priority. Actually, they can send just it two cases; the first is after the high level VLs reach the given high-level limit, or if the high level VLs are empty and the control VL is empty too.

The VL arbitration is a multiplexer that selects the right VL to reserve the physical layer and transmit its packets each VL belongs to VL high and VL low quotas has a weight from 0 to 255 and that weight emphasize the weight of that VL in that quota.

Loop in VL quota called VL loop, VL loop will affect the VLhighCounter which the counter that prioritize the transmit from VL high and VL low quotas.

Moving the transmit from VL high to VL low depend in VLhighCounter while this counter is not zero all packets transmit are from VL high after this counter set to zero the VLarb start to give the physical layer to VL low depends on their weight and order in quota after one loop of VL low the VLhighCounter will have its value from the QoS configuration file, so VL low send packets in two conditions if there are no packets at VL high quotas or the VLhighCounter is set to zero.

Knowing that if any packet received at control quota VLarb will pause the traffic and give the physical traffic to control quota since the control quota has the ultimate priority.

So, if we have VLhighCounter equal to 10 that mean we have 11 loops first 10 loop for VL high and last one for VL low.

As an Example of VLarb:

If we have following configuration:

- VLhighCounter: 2
- VL high: VL0:2, VL1L3
- VL low: VL2:2, VL3:2

Assuming all VLs have unlimited packets and there is no interrupt from control packets the VLs will reserve physical layer as follow:

- First Loop: VL0 will send 2 packets then VL1 will send 3 packets, VLhighCounter will decrement by 1, so VLhighCounter will be 1.

- Second loop: VL0 will send 2 packets then VL1 will send 3 packets, VLhighCounter will decrement by 1, so VLhighCounter will be 0.
- Third loop: VL2 will send 2 packets then VL3 will send 2 packets, VLhighCounter will set back to 2.

3.3 Scenarios

IBMGTSim is designed to match with IB structure and static node dimension but without tracking traffic, so we added two features to IBMGTSim to add capability to track and simulate traffic and add node dimension, also we used SWIG interface to add capability to control this and observe the topology characteristics.

Before starting the scenario, we must define the nodes speeds and widths to the nodes within the topology at the topo file. And using SWIG to add nodes dimension and control of nodes behavior. Each scenario runs with specific topo file and QoS file, that are parsed. The topo file is parsed to give data to the whole topology and the QoS file is parsed to give data to every node to set calculated QoS for them, nodes behavior can be changed by sending command to IBMGTSim directly or by writing script to simulate the node movement and state.

Nodes will need to send data randomly to random nodes while nodes keep moving and disconnecting a node will cause a recalculate to paths regarding to QoS and then send the data through it. A log file can show everything happening in the network and how the data move from one node to another.

All scenarios have the random seed as an input, so if we use it to have the exact random values we became capable to compare results of the same scenario with different routing algorithms, and we can make sure that we will have same result if we run the same scenario in same machine with the same seed.

3.4 System processing design

Regression is designed to maintain three processes

1. Simulator
2. Diagnostic
3. Test cases

1- Simulator is composed of several types of threads:

- Node Management Thread (NMT) that is responsible of managing nodes and processing packets and each thread of this kind can manage up to 16 nodes.
- The Dispatcher Thread (DT) which is responsible of sending, receiving, forwarding and discarding packets.
- Logger Thread (LT) that is responsible of registering the log messages.

2- Diagnostic process: which is the process that checks the validations of paths and searches for dead end and loops.

3- Test cases process: that is composed of several threads which are responsible to represent the scenario by moving nodes, disconnecting nodes and setting data to the nodes.

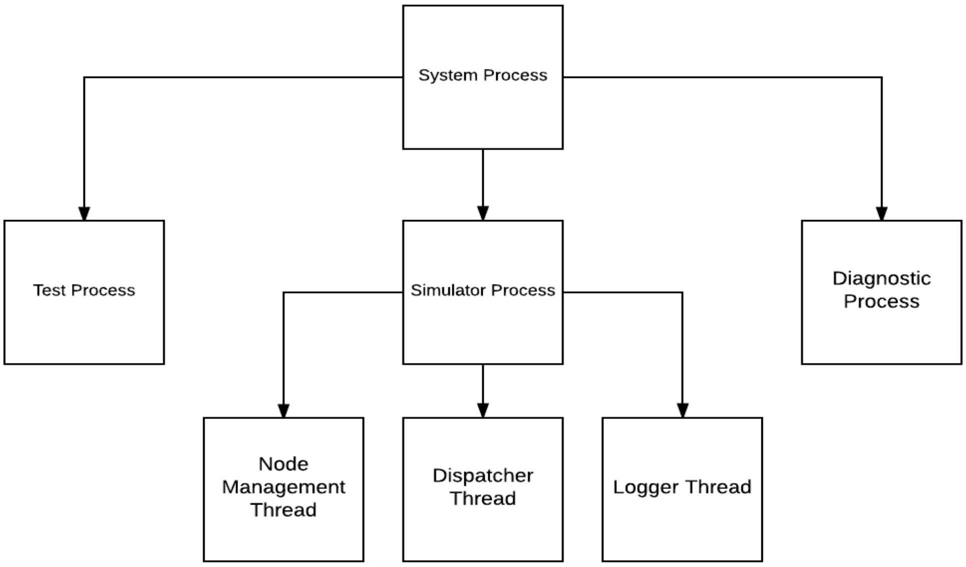

Figure 3.3 system process

Chapter Four - Experiments and Results

Chapter four contains all our experiments, tests subjects, analysis and result first we will go through the device used and test environments then experiments setups and metrices and measurements tools and mechanism.

Our tests can be categorized in two phases first phase that found relation between LMC and number of nodes since node destitution will be studded in future work, second phase is the main phase that study the network performance while changing number of nodes, transport rate and packet size.

4.1 Experimental Design

4.1.1 Device Used

We run IBMGTSim simulator on a server that runs with these specifications that ensure running the same test cases with same inputs, the same topology, the same number of threads and the same random seed on the server that has the same equivalent specification, which gives similar results, not exactly similar since there is no control on thread racing but on an average of a large run, it will come out with the same results, however, running the test cases on other devices will affect the results, it can bring different results and of course the data will be different but the main core and the concept of the results will be almost the same, i.e., if a test runs on a faster machine, it will increase the simulated node processing, which affects the delay and latency, this fact will not be changed on other devices, but the data will vary.

Device specification:

CPU: intel core i7 4790

RAM: Kingston PC3-8500R 16 GB

HDD: western digital WD1003FZEX

OS: Linux Redhat 6.5

4.1.2 Experiments Environment

Table 4.1: IBMGTSim capability

Simulator Parameter	Description
Simulation time	3000s
Number of simulation runs	Up to 255
Number of nodes	Up to 512
Node mobility pattern	From 0 to 1m/s
Topology	Up to 100m x 100m
Transmission range	Up to 25 m
Transmission power	31 Mw
Queue length	100 packets
Routing Protocol	AOMDV
MTU	Up to 5x256
Link Width	Up to 4
Link Speed	Up to10Mb
LMC	Up to 7
packet-life	Up to 10s

IBMGTSim requires the machine to have:

1- OS RedHat with a version of 6.5 up to 7.0.

2- C/C++ compiler as g++ version (GCC) 4.8.

3- swig version should not be under 2.7.3, the required is 3.0.7.

4- Boost C++ version 1.6.0.

Dependent packages needed are:

1- IBUMAD version 1.7

2- IBMAD version 1.7

3- IBVERBS version 1.7

4.1.3 Experiments Setup

Our regression contains three layers.

- IBMGTSim.
- swig interface.
- test case body.

IBMGTSim is the main part of the simulator of IB, which simulates the node and links between the peers, it is written in C++ and a part of OFA solution, both port and node have their own characteristics. Since these layers are not existing on MANET network, we modify the IBMGTSim to support dynamic ports to define the direct link to simulate the connection between the pair nodes and dynamic change.

for example, if we have a node that is connected to four nodes, that node will have four external ports within an internal port to communicate with it though a socket. Knowing that each IBMGTSim changing state is logged in a log file which is separated from the test log file.

SWIG interface is an interfacing tool that comes between the IBMGTSim and the test cases since each one of them is implemented with a different technology. SWIG has the interfaces to all the classes, structures, functions and even variables that maybe used in the test cases.

Test Case: the test case is written in python scripting language since python productivity is better than any other language, which gives us capability to write several scenarios in a short time and all the test cases can use all of the IBMGTSim capabilities which are interfaced by the SWIG.

4.1.4 Experiments Metrics and Measurement

Our main goal is to compare the network performance while using AOMDV with QoS in two cases, one while IB enhancements are used and another one without IB enhancements.

We will study the effect of our proposal on the network performance while assuring that QoS rules are applied, so these metrics should be studied:

- Throughput: which is a measure of how many units of information a system can process in a given amount of time. It is applied broadly to systems ranging from various aspects of computers and network systems to organizations. Related measures of system productivity include the speed with which some specific workload that can be completed, and response time which is the amount of time between a single interactive user request and receipt of the response.
- Bandwidth: which is used as a synonym for data transfer rate, it is the amount of data that can be carried from one point to another in a given period of time. Network bandwidth is usually expressed in bits per second.
- End to End delay: The time between the moment when a packet is transmitted and the moment when it reaches its destination. It of course leads to delay which is caused by slow network links and is a major concern in communication when it comes to call quality. There are two ways latency is measured: the one direction and the round trip. One direction latency is the time taken for the packet to travel one way from the source to the

destination. Round-trip latency is the time taken for the packet to travel to and from the destination, back to the source.

- Control overhead: The control overhead is defined as the total number of routing control packets normalized by the total number of received data packets

All of the above metrics will be affected by the following system properties:

- Nodes Rates: which is the maximum transmit speed that a node can individually support. This propriety has a direct relation between the port speed and width.
- Number of nodes: which is the total number of nodes that exist in the topology, increasing the number of nodes will increase the complexity of the network and negatively affecting the network performance.
- Packet size: which is the size of transmitted packet since splitting packet size after the transition is forbidden and it is not allowed to change the packet size in between nodes.

At our work, we will study the effects of changing the properties on metrics and we keep collecting data samples results while just changing one of the properties at each test run. The following table shows the changed properties and the studied metrics.

Table 4.2 metrics vs properties

Metrics \ Properties	Throughput	Bandwidth	End to End delay	Control overhead
Nodes Rates				
Number of nodes				
Packet size				

4.2 Difficulties and Challenges during the Experiments

Challenge 1:

IBMGTSim only supports a fixed network by a fixed design, actually it only supports a fixed topology with a fixed number of ports and with fixed links connections.

We have overcome movement challenges by adding two variables at a node lat. and long. and these variables explore the current location of the node and can be changed since they are directly affecting the dispatcher by disconnecting links with peers if the distance is more than 100 meters.

We overcome static peers issue by replacing the array of ports with a victor of ports and these ports are logical and they indicate which pair of nodes are currently connected and the values are not pointed to an integer, we manage to use objects to set port to NULL if any peer gets disconnected or its state is changed to DOWN.

Challenge 2:

IBMGTSim has bad logger:

In current implementation of IBMGTSim logger supports only the changing state of the node, so we added counters, which count the send and receive packets. To calculate the traffic, we added fields in the packet called PACKET INFO to collect the data while the packets is being transferred from a specific source to final destination and write the log using the same logger thread to another log file, depending on a flag if log messages come from our module or from existing module.

Challenge 3:

Add user control to simulator:

IBMGTSim should support the user modifications or test modifications, but the implementation has not been done yet, IBMGTSim can be manage by using OpenSM, but in our case, we need to test and run a regression system to modify it, we managed we solved this issue by adding the SWIG interface to export the IBMGTSim functionality to test cases domain. knowing that the test is written in python code since python has a great productivity over C++ so we use same the SWIG interface to export python code control rather than exporting C++ controls.

Challenge 4:

Adding LMC, pathbit, VL and VLarb to the existing AOMDV routing algorithm, in order to solve this issue, we have to understand the implementation of AOMDV algorithm at the code level besides of understanding the AOMDV algorithm design, then adding these parameters will not be a big issue.

4.3 Data collection

Collecting the data was done dynamically while the tests are running by simulator logging thread.

The Logger thread was writing on two files:

Node state xml file, which contains the node changing state, node movement, receiving packet and dropping packets.

The traffic state csv file, which contains traffic proprieties like overhead, end to end delay and throughput.

4.4 Measuring Network path-bits

Since path-bits are used to represent the multipaths, and one of our research targets is to find the optimal path-bits which is a challenge that depends on the distribution of the nodes and the total number of the nodes, since the distribution of MANET is most likely a normal distribution with can be represented as a mesh topology. Our research and experiment will focus on the relation between the number of nodes and the path bits.

Our experiment result was found in which the path-bits can be categorized in three categories:

1- Path-bits are less than the network need, which negatively affects the purpose of multipath concept, and increase the control overload since any path loos will force the node to recalculate the route one more time.

2- Path-bits are more than the network need; is this case we will find redundant paths or paths that are always down.

3- Path0bits is close to the network need, in this case we have a good number of paths that can serve the traffic needs with high efficiency.

In our experiment, we have to set the number of nodes to a fixed number and change the number of LMC which will affect the path-bits and allow us to collect two kinds of data.

1- Relation between LMC and control overload.

2- Relation between LMC and used path, and relation between LMC and Normalized load.

Note that we measure the normalized load while fixing received packets at 200 since we need enough packets to establish the traffic using all calculated paths of course we can use and number bigger than possible calculated paths after using these experiments we found the none of them can reach 150 packets we use 200 packet as safe number.

Number of nodes 10:

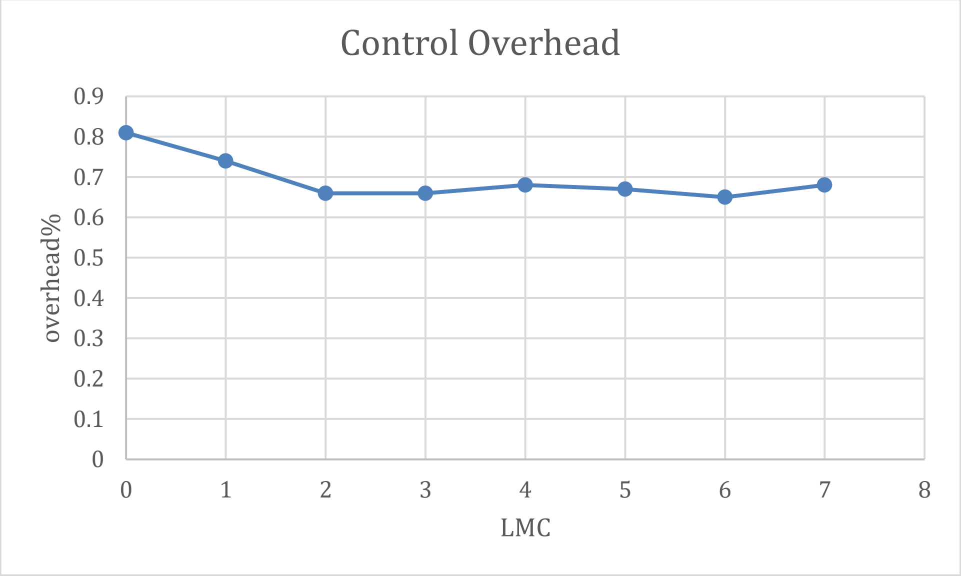

Figure 4.1: Control Overhead vs LMC 10 nodes

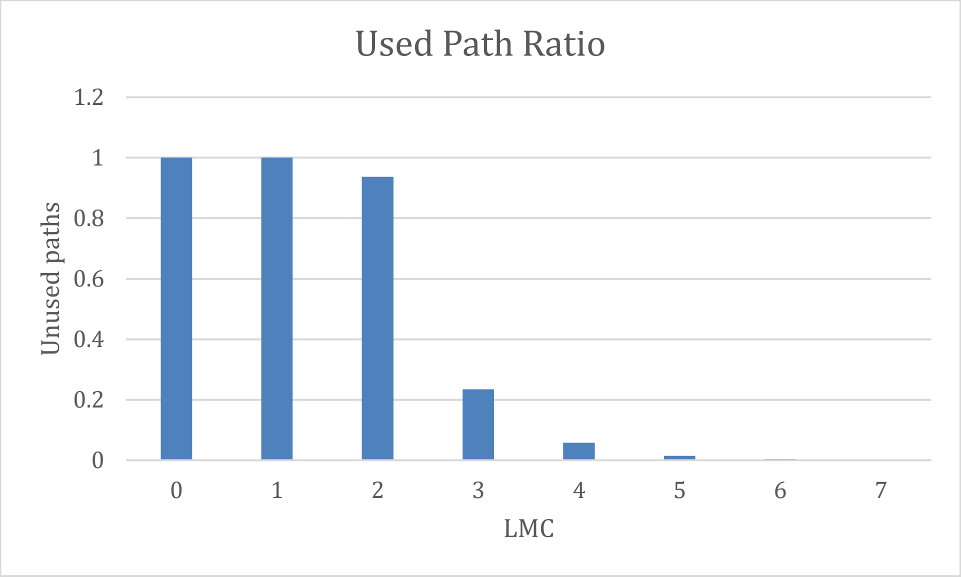

Figure 4.2: Used path ratio vs LMC at 10 nodes

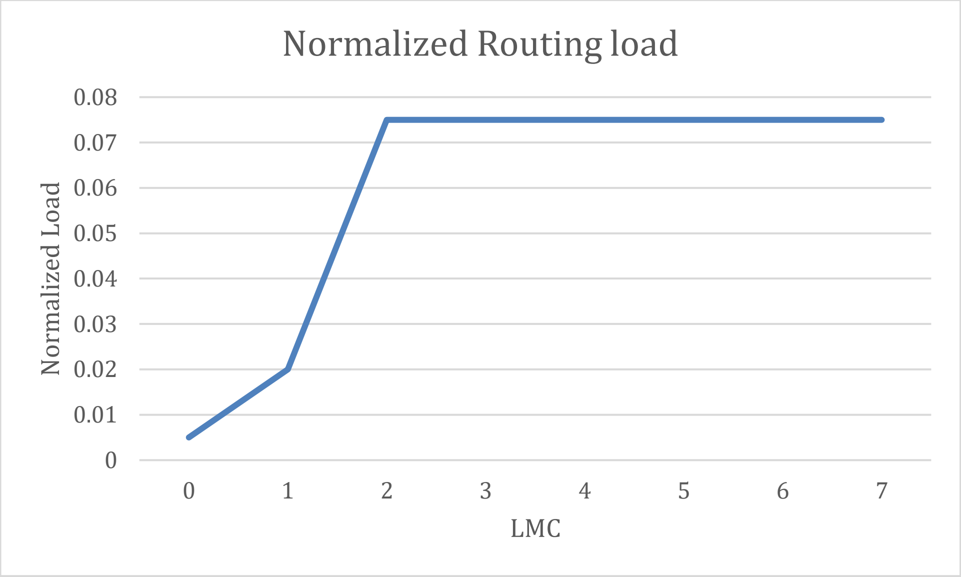

Figure 4.3: Normalized Routing load vs LMC at 10 nodes

At 10 nodes, each node set 15 unique paths, so we can use LMC 2 as optimal LMC.

Number of nodes 25:

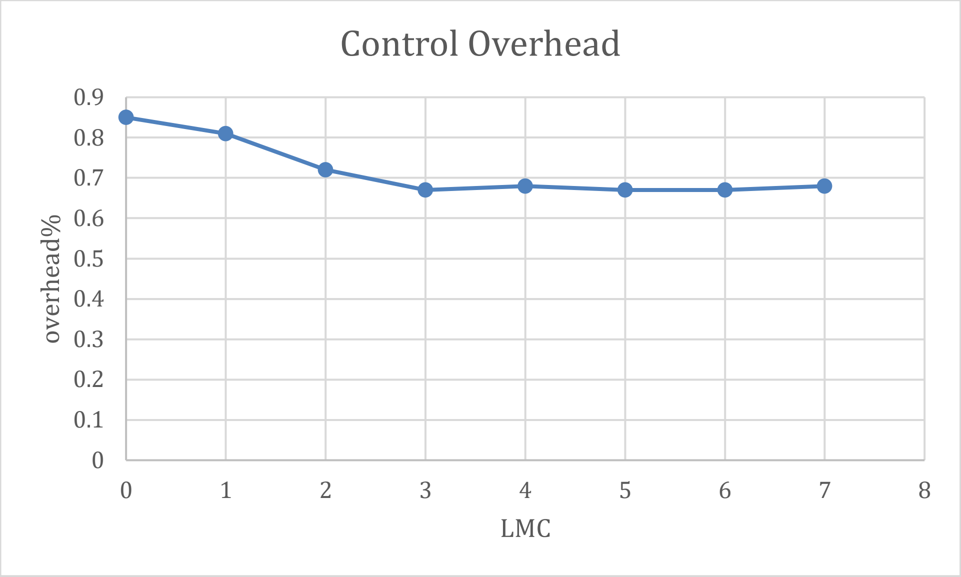

Figure 4.4: Control Overhead vs LMC 25 nodes

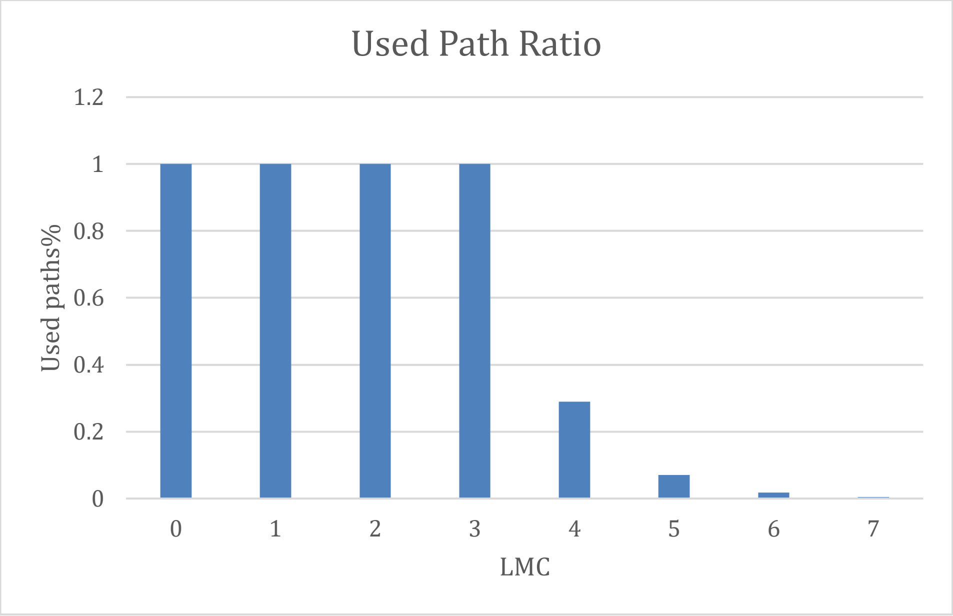

Figure 4.5: Used path ratio vs LMC at 25 nodes

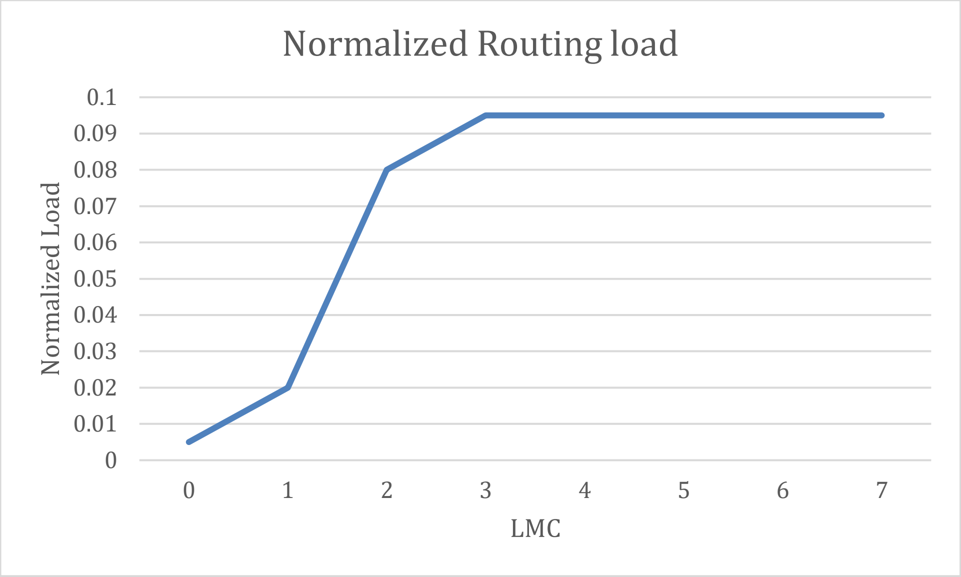

Figure 4.6 Normalized Routing load vs LMC at 25 nodes

At 25 nodes, we figured out that each node can calculate 75 paths but at 19 paths the control over head reaches the optimal result, so the optimal LMC is 3.

Number of nodes is 50:

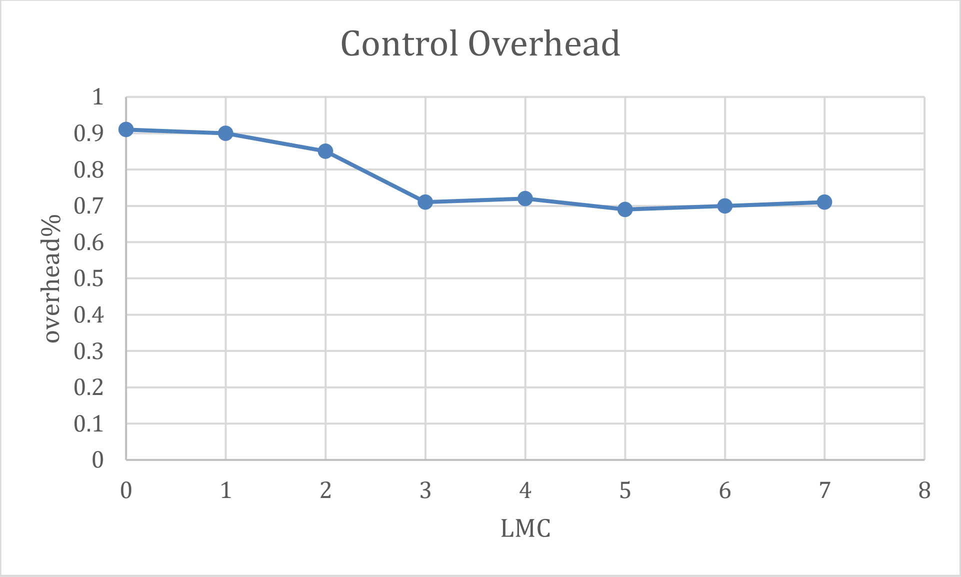

Figure 4.7: Control Overhead vs LMC 50 nodes

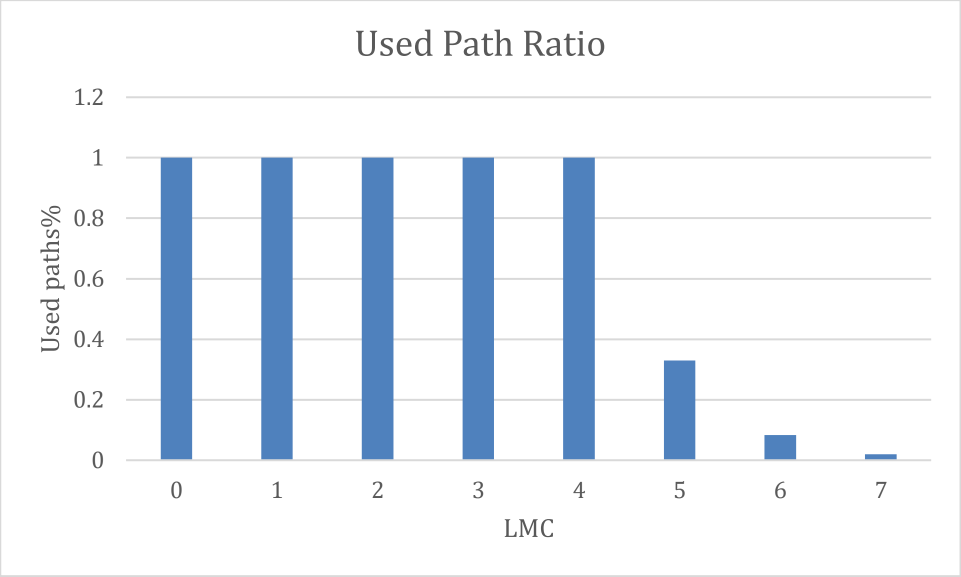

Figure 4.8: Used path ratio vs LMC at 50 nodes

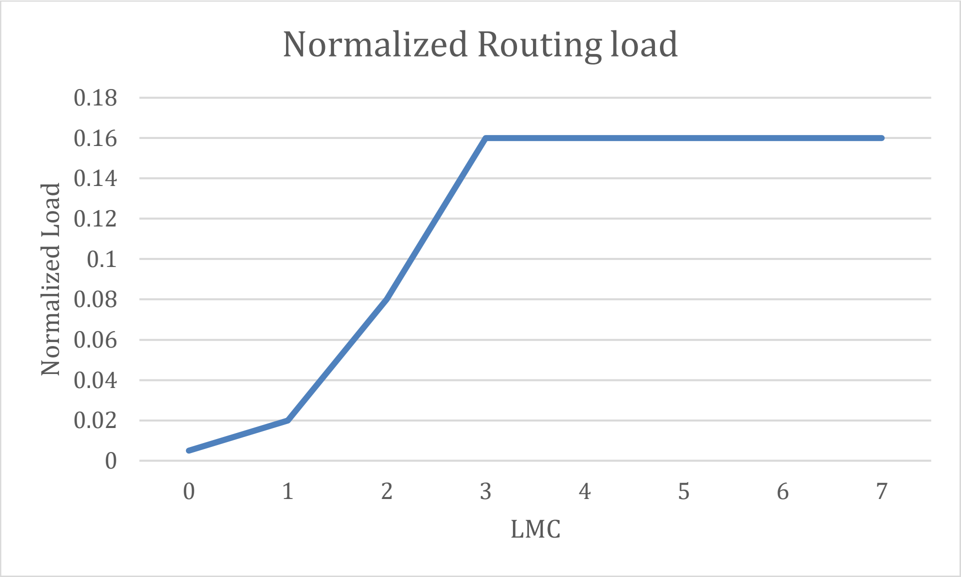

Figure 4.9 Normalized Routing load vs LMC at 50 nodes

At 50 nodes, we figured out that each node can calculate 340 paths but at 22 paths the control over head reaches the optimal result, so the optimal LMC is 3.

Number of nodes 100:

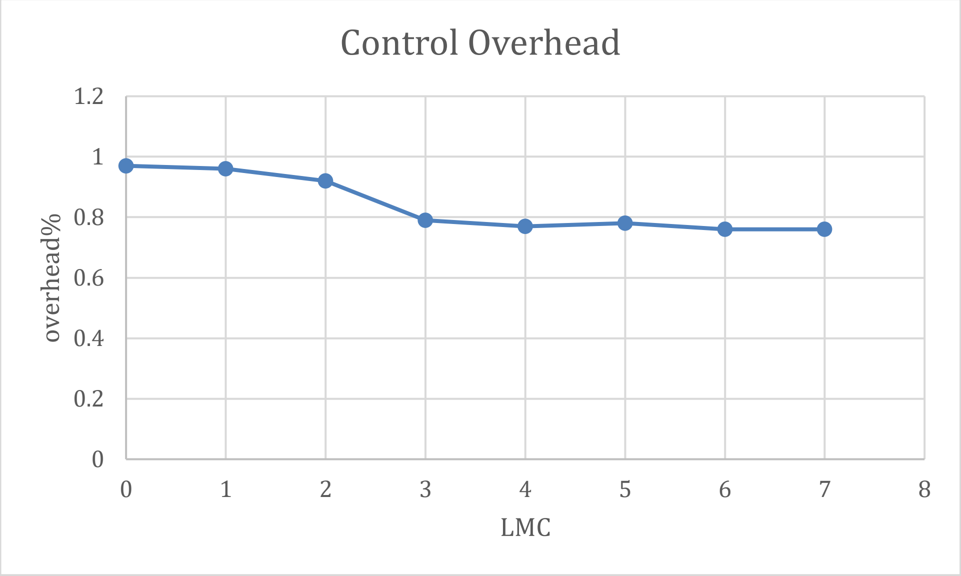

Figure 4.10: Control Overhead vs LMC 100 nodes

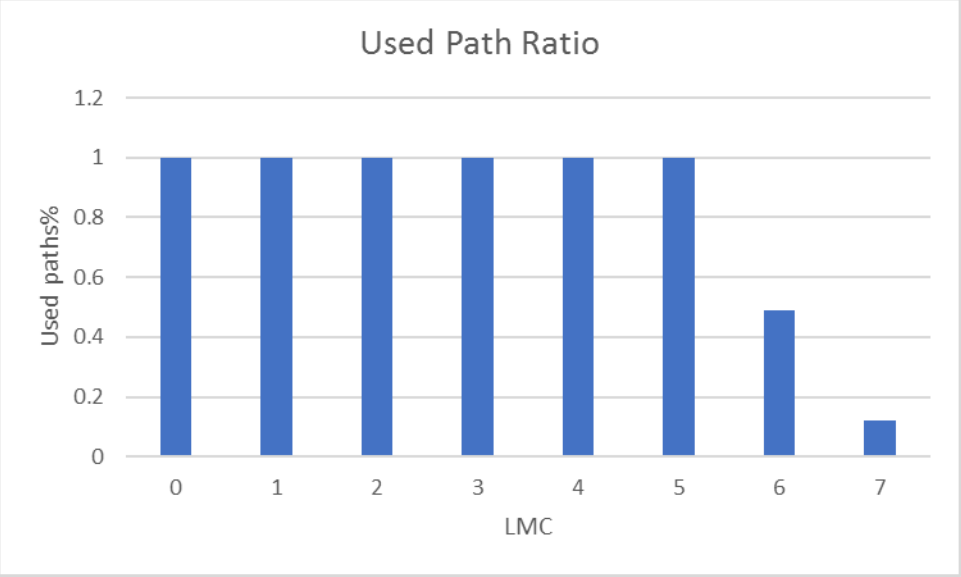

Figure 4.11: Used path ratio vs LMC at 100 nodes

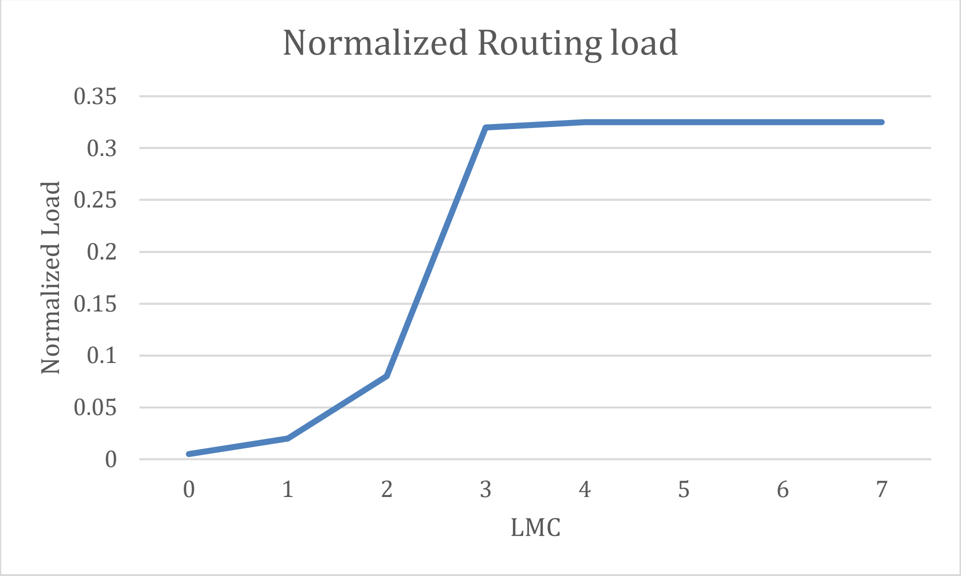

Figure 4.12 Normalized Routing load vs LMC at 100 nodes

At 100 nodes, we figured out that each node can calculate 2025 paths but at 65 paths the control over head reaches the optimal result, so the optimal LMC is 3-4 for a hundred of nodes.

So, we found that calculating the maximum possible paths will not give a unique best performance since we can calculate less number of paths to reach the same exact network performance that can be reached by calculating the full possible paths.

Regarding our simulator capability, we can assume that LMC should be a set in all nodes as following equation: $LMS = (Log_5 \sum Nodes) + 1$

Knowing that, calculating the LMC for individual node is not a focusing aspect of our work study and it may be studied in future.

4.5 Measure Network Performance

In our second research, the main researched part was to focus on the enhancements, we assumed that we made the experiment on QoS and on multipath with QoS networks.

4.5.1 Throughput

Figures 4.13, 4.14 and 4.15 show the throughput changes while changing on the network priorities.

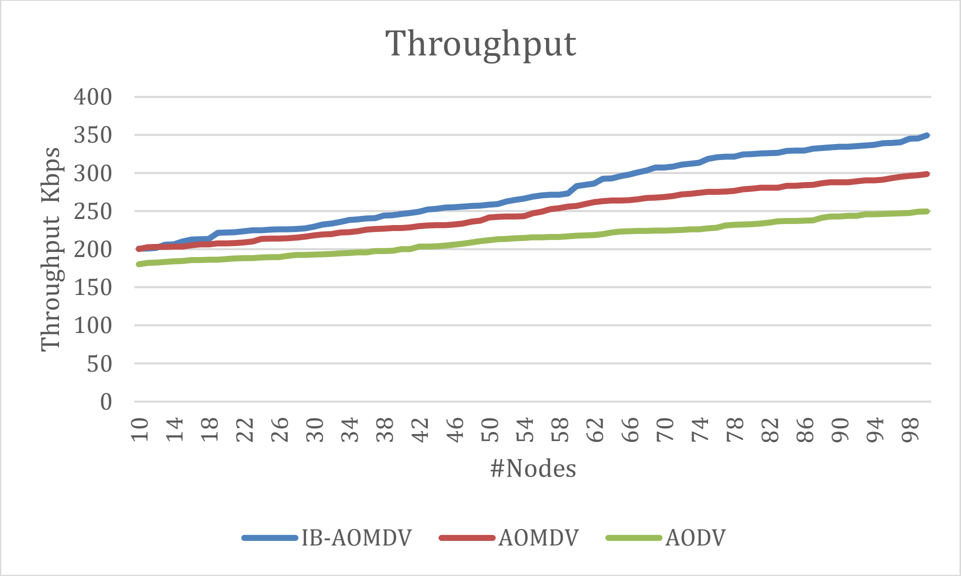

Figure 4.13 Throughput Vs number of nodes

Figure 4.13 shows the relation between throughput and number of nodes, we can conclude that IB-AOMDV throughput is increased as an average of 9% more than AOMDV and 22 times more than AODV. The throughput was increased by a node number increase since nodes can find more variant and independent paths when there are more nodes in the networks. Keep in mind that the increase of the throughput will be a decrease, since the network capability will lead throughput to steady point.

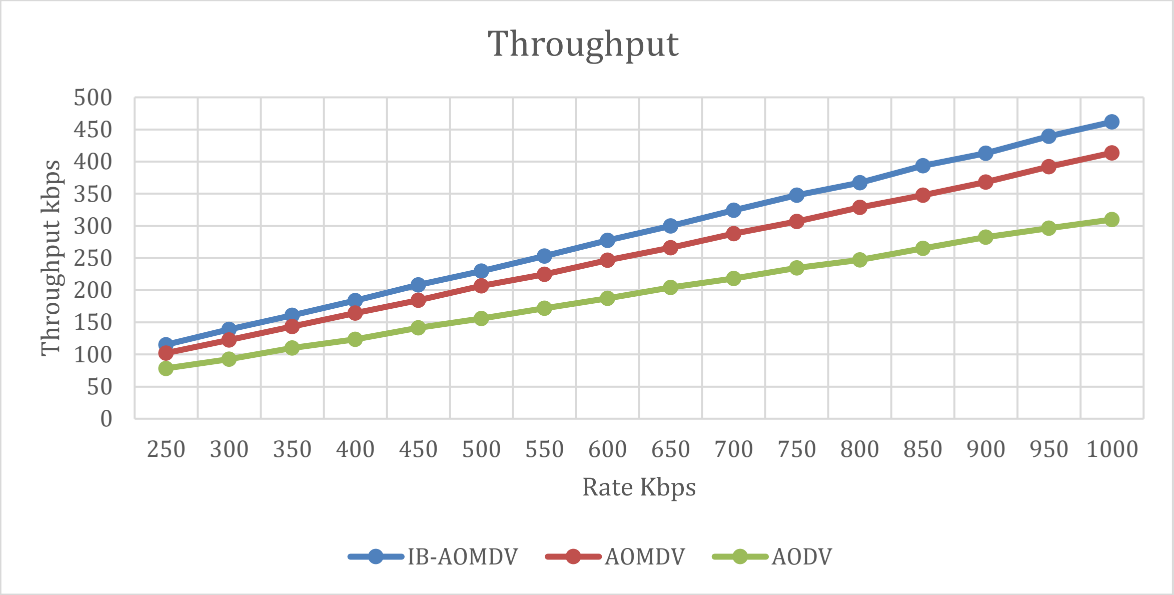

Figure 4.14 Throughput Vs rate

Figure 4.14 shows the relation between the throughput and sending data rate when the number of nodes was 50 nodes and packet size is 1024, as we know the relation between throughput and data rate is linear and throughput at IB-AOMDV is 46.133% of the data rate, AOMDV is 41.057% of data rate and AODV is 31.193% of data rate.

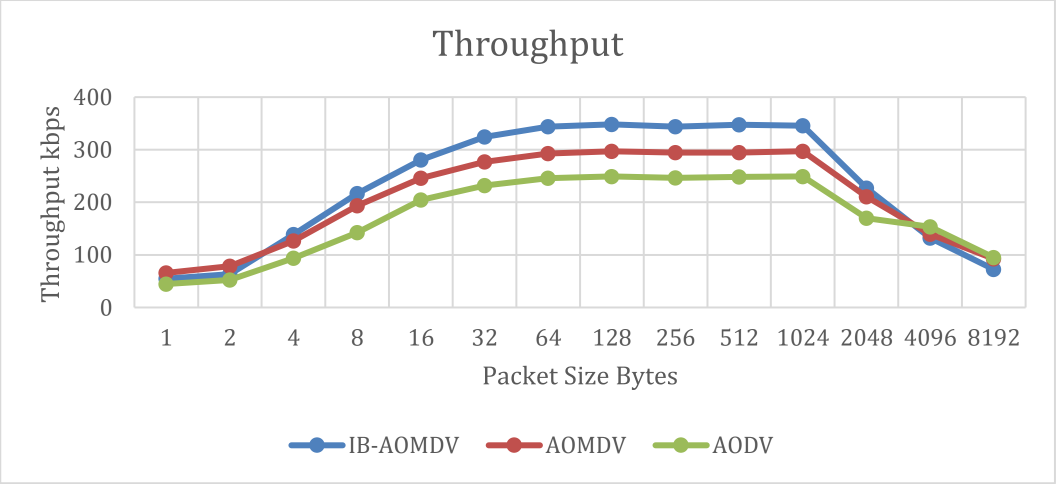

Figure 4.15 Throughput Vs packet size

Figure 4.15 represents the relation between the packet size and the throughput. we have gotten a fix number of nodes ’50 nodes’ and a fixed data rate of 750kbps. We found that at low packet size, throughput suffers in the all three routing algorithms.

It had a positive relation until it reaches the max throughput that can be supplied then after that the relation became a negative relation since dropping a packet is more harmful than dropping smaller packet.

We can figure out the drop occurred since the MTU at node is initiated at 1024 bytes.

4.5.2 Delivery Ratio

Figure 4.16 shows the success packet delivery ratio vs. the number of nodes, and figure 4.17 shows the packet delivery ratio vs. the data rate and figure 4.18 shows the same matric vs packet size. The packet delivery ratio is defined as the ratio of delivered packets to the total number of packets. From these two plots, we extracted the following findings:

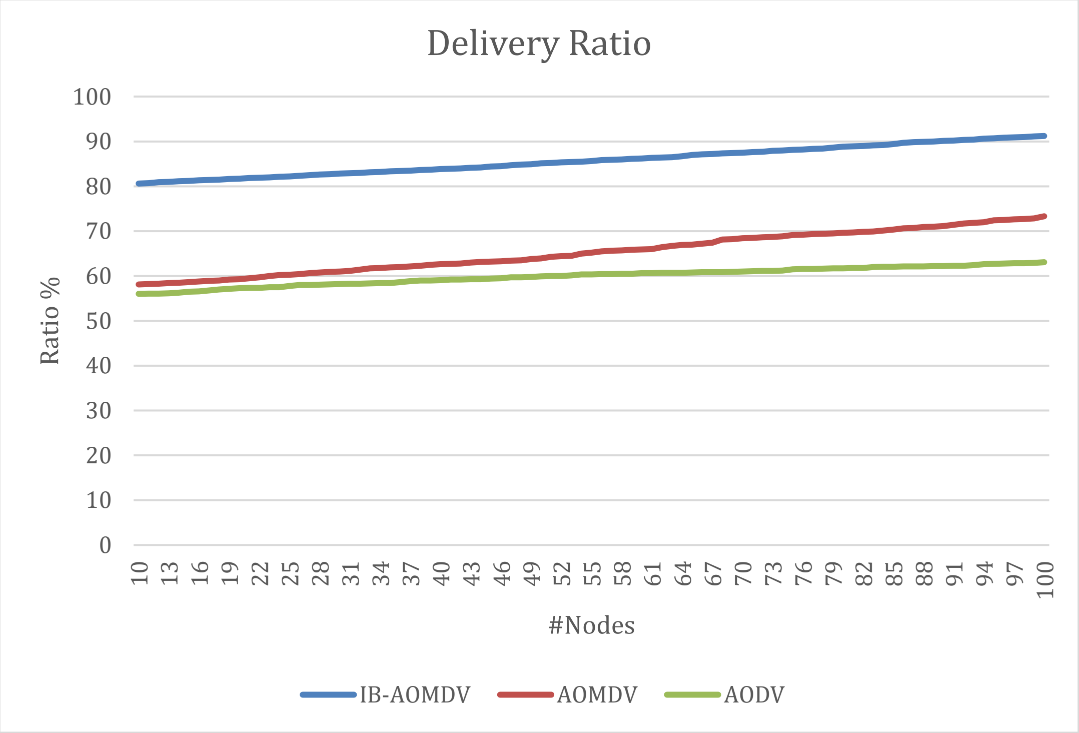

Figure 4.16 Delivery ratio Vs number of nodes

Figure shows, the enhanced version of the AOMDV protocol (IB-AOMDV) outperforms both AODV and AOMDV in terms of the number of packets successfully delivered to their destinations, then comes AOMDV and then AODV in order. In average, IB-AOMDV had achieved about 85.5% packets delivery ratio, 65.5% for AOMDV, and 60% for AODV. This means that the packet delivery ratio has increased by about 24% compared to AOMDV and 30% compared to AODV. As the number of nodes increases, the packet delivery ratio increases as well, and in all three protocols the graph is increasing almost linearly. In IB-AOMDV, when the number of nodes increases, the packet delivery ratio increases linearly, and the same for all protocols, but for AODV the slope of increased ratio becomes smaller after 60 nodes

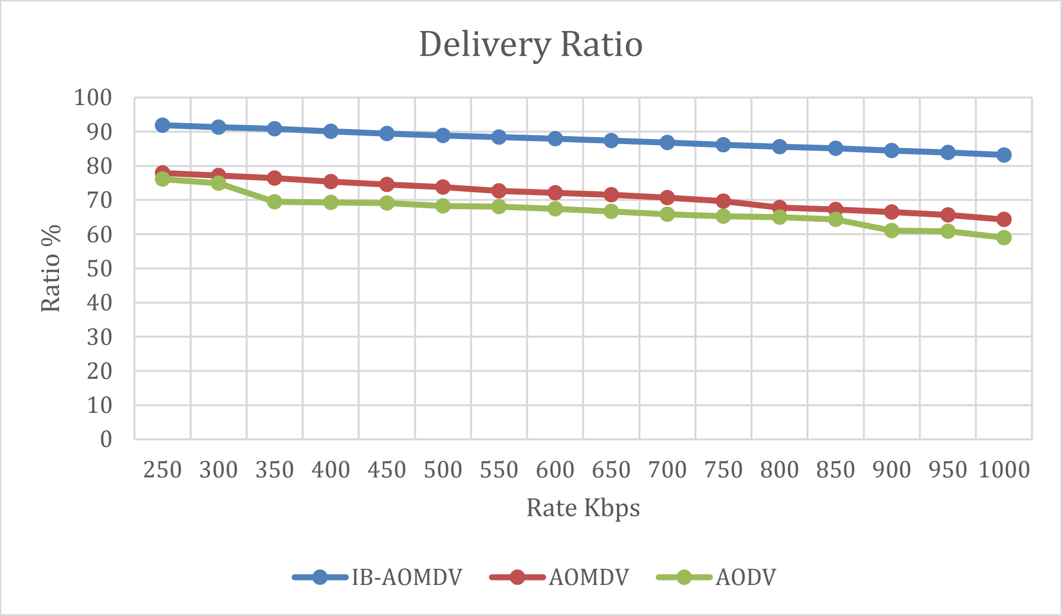

Figure 4.17 Delivery ratio Vs rate

Figure shows the changing of data rate while we have a fixed number of nodes in the topology equal to 50 nodes. IB-AOMDV gives the best results with about 87.6%, then comes AOMDV with about 71.5%, and 70% for the AODV protocol. We can conclude that the packet delivery ratio for IB-AOMDV is enhanced by 18% and 20% when compared to both AOMDV and AODV respectively.

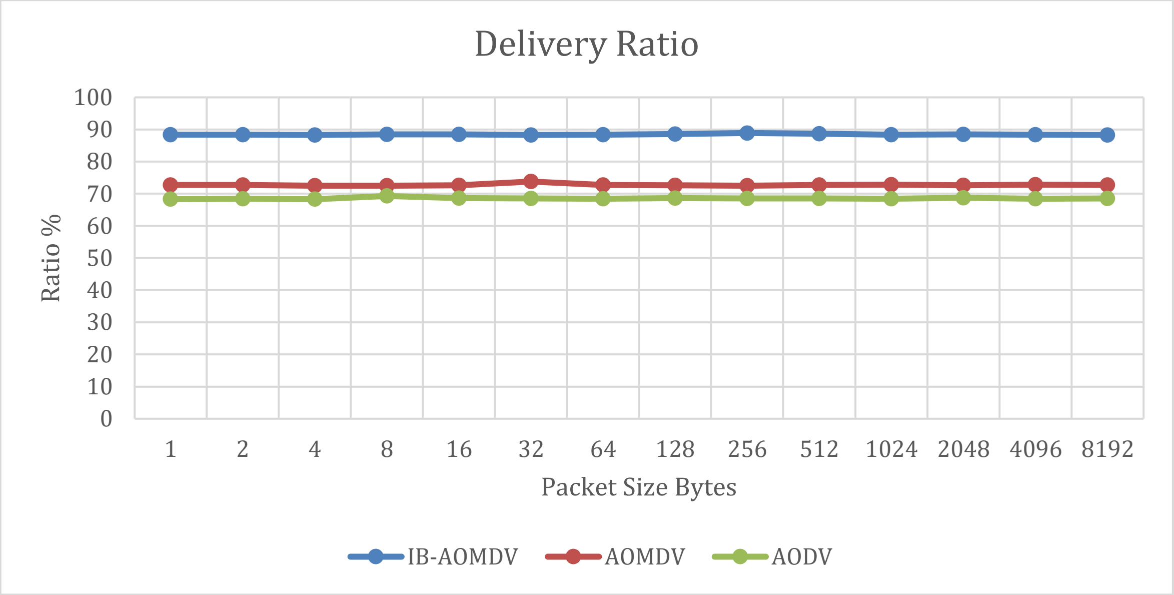

Figure 4.18 Delivery ratio Vs packet size

Figure 4.18 shows the changes of delivery ratio while the packet size is changed while we have 50 nodes in the topology and a fixed data rate of 500Kpbs.

Furthermore, as the data rate increases the performance of the IB-AOMDV is still higher than the others which gives a strong indication of higher scalability and sustainability of the new method.

These results were achieved since we have used VLs, there are 15 independent queues that are filled by packets from several QoS levels. The VLs that contain high important traffic send more packets than VLs with lower priority and find independent paths that are suitable for each QoS rule which increases the delivery between important nodes and keeps less priority traffic low without affecting the important traffic.

4.5.3 Delay:

The following figures, figure 4.19, figure 4.20 and Figure 4.21 represented in this section show the end-to-end delay, which can be defined as the average time of all traffic from the source to the destination. The following figures show the simulation results for both cases; increasing the number of nodes and increasing the data rate. Our main findings in this metric can be summarized as follows

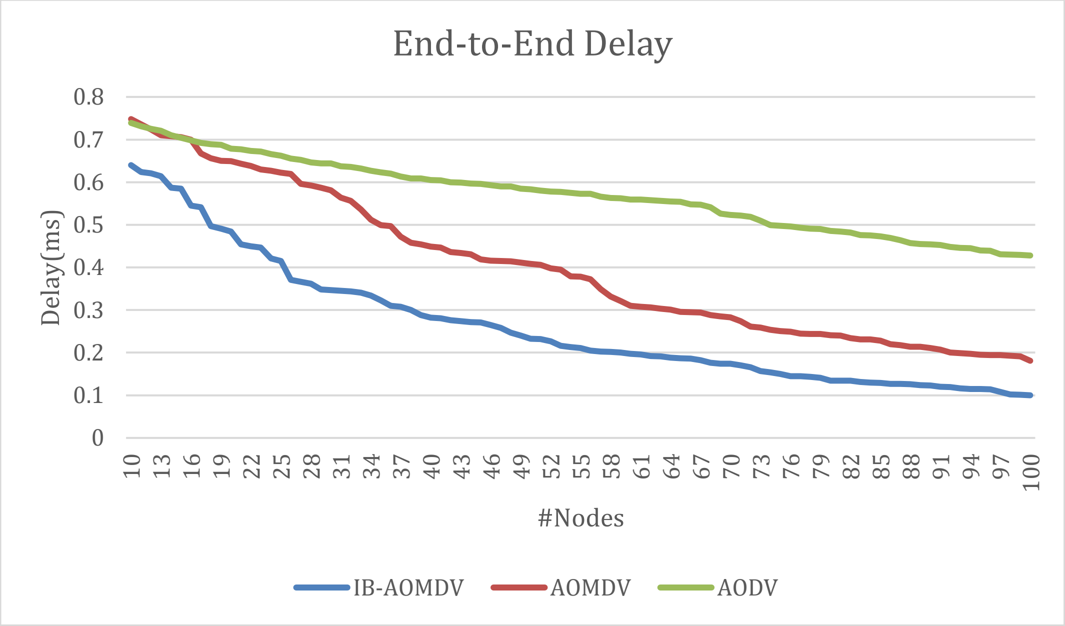

Figure 4.19 Delay Vs number of nodes

Figure 4.19 shows that IB-AOMDV gives the best results among the others in terms of end-to-end delay. On average, IB-AOMDV achieved about 0.26ms delay, AOMDV achieved 0.4ms, and AODV achieved 0.57ms end-to-end delay. This means that we have achieved an enhancement over the other two protocols by 35% and 54.3% respectively

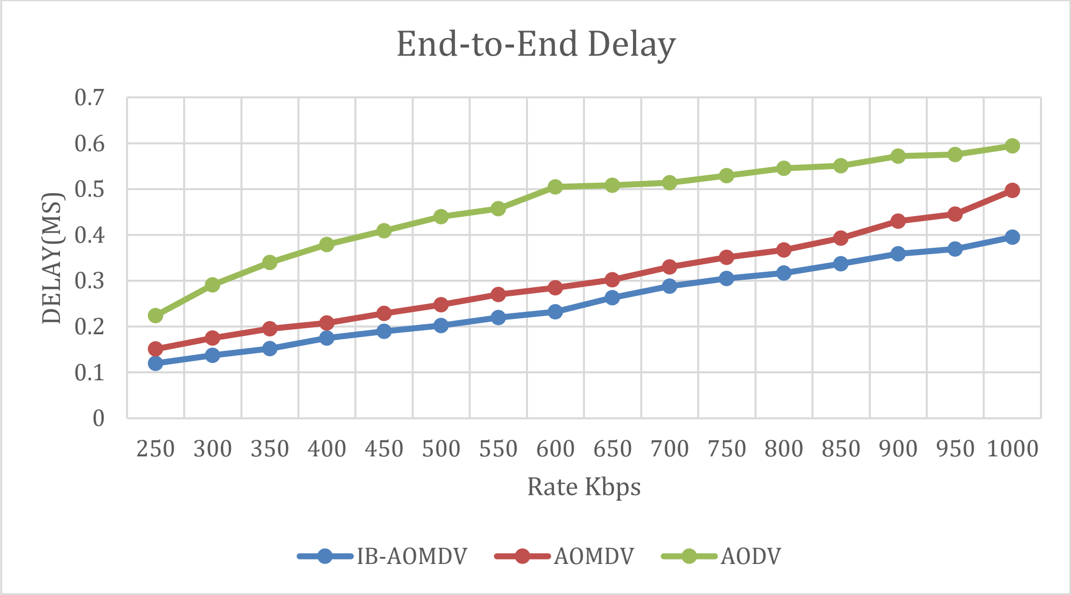

Figure 4.20 delay Vs rate

figure 4.20 shows increasing the data rate – the behavior of the three protocols is still the same. It is obvious that as the data rate increases the end-to-end delay will be increased as well. By taking the average of all values, we obtained 0.25ms, 0.3ms, and 0.46ms for IB-AOMDV, AOMDV, and AODV, respectively. Thus, we achieved a significant enhancement of 16% and 45.6% compared to AOMDV and AODV, respectively.

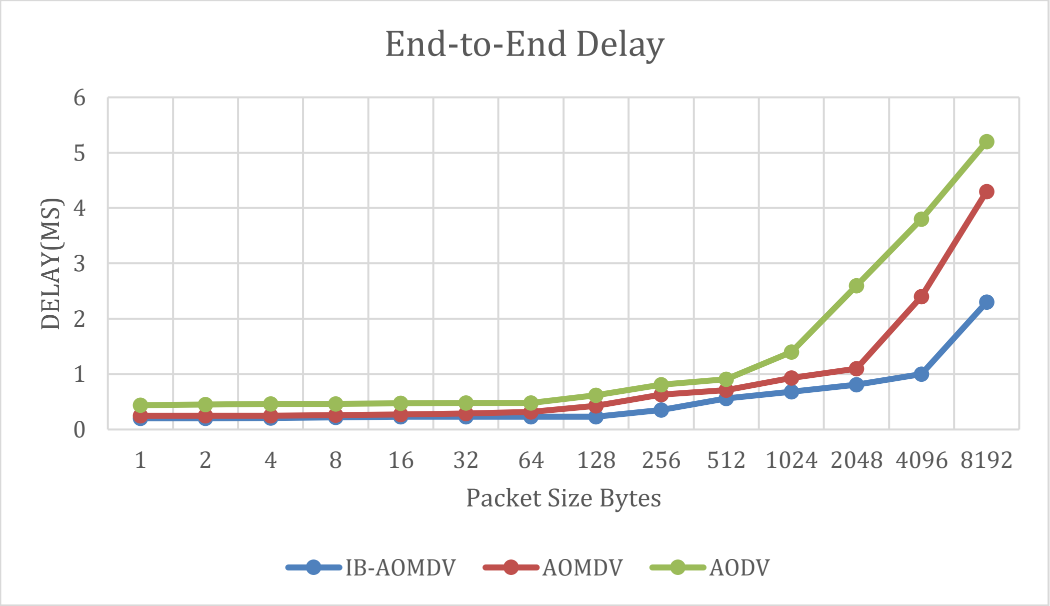

Figure 4.21 delay Vs packet size

Figure 4.21 show the increase of end to end delay while the packet size is being changed at 50 nodes and data rate is 500Kbps, we found out that IB-AOMDV have a less effect of end to end delay of 200% compared to AOMDV and 340% to AODV at packet size 8KBbs. These comes from using virtual lanes queues since packets lines up in 16 queues and packets are not affected by next queue service level.

4.5.4 Control Overhead:

Figure 4.22, Figure 4.23 and figure 4.24 show the simulation results of the third metric considered in this research that is control overhead. It is defined as a ratio of routing packets though VL15 in IB-AOMDV to the total number of packets. In figure 4.22 we set a variant number of nodes and in figure 4.23 we set a variant data rate but in figure 4.24 we set a fixed nodes number ‘50 nodes’ and a fixed data rate of 500Kbs.

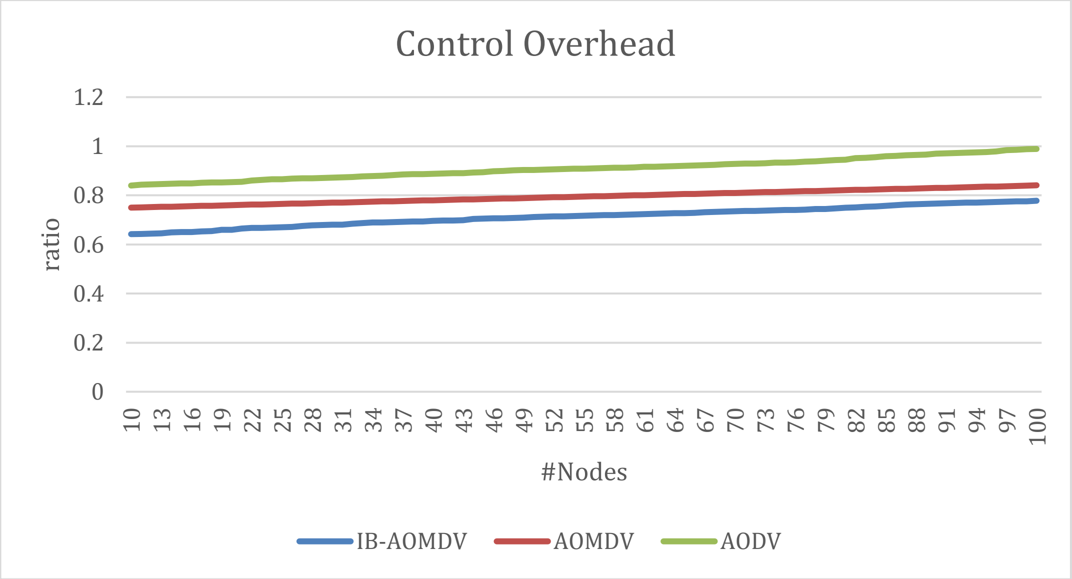

Figure 4.22 Control overhead Vs number of nodes

Figure 4.22 gives the control overhead (%) of the three protocols vs. the number of nodes. As we can see IB-AOMDV outperforms the others and gives the smallest overhead regardless to the number of nodes. In average, 71.4% control overhead ratio for IB-AOMDV, 91.1% for AOMDV, and 79.5% for AODV. This means that the overall performance enhancement of the control overhead that was achieved when considering the InfiniBand mechanism has increased by 21.6% and 10.2% compared to AOMDV and AODV, respectively

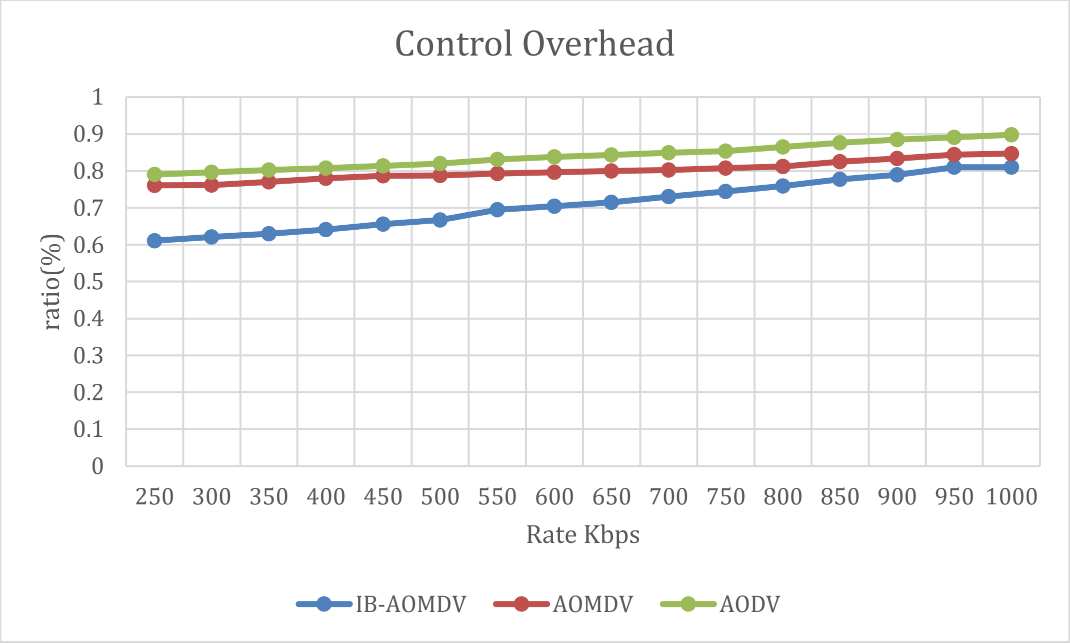

Figure 4.23 Control overhead Vs rate

In figure 4.23 we went further and repeated the same experiments by varying the bit rate to study the behavior of these proposals when the load is increased. The same finding is also valid here. This means that IB-AOMDV comes first, then AOMDV and AODV comes respectively. The average values are 71%, 84%, and 80%, with about 15.5% and 11.3% enhancement.

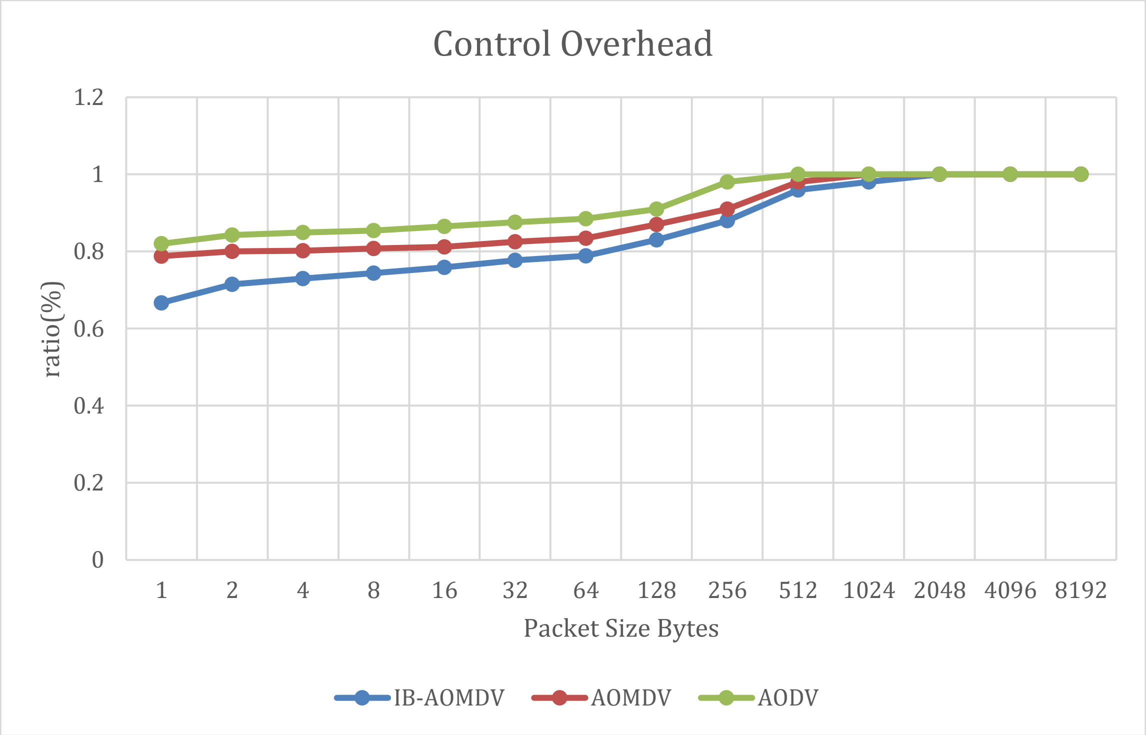

Figure 4.24 Control overhead Vs packet size

Figure 4.24 shows the effect of changing packet size at a network with 50 nodes and all nodes have 500KBbs we found out that there is no advantage of IB-AOMDV since increasing the packet size is negatively affecting all of the routing algorithms and control packets have the most priority over the data packets regardless to the QoS service level that they belong to, however the IB-AOMDV is still giving %16 control overhead less than AOMDV.

Chapter Five – Conclusion and Future Work

5.1 Conclusion

MANET network is a very difficult environment for communication, but it is the most natural existing type since it is dynamic and allows the nodes to join, disjoin and move freely within the network and so the paths are changed, some become invalid and some new others come up to life.

IP protocol was built to manage basic communications but it suffers in the modern network types such as MANET, in the meanwhile other network models were designed to solve issues that IP protocol is unable to solve, we can elicit existing solutions from these existing protocols to come with a sufficient solution for networks problems that IP cannot solve.

Applying the organized mechanisms that InfiniBand provides such as path-bits or multipath and virtual lanes for QoS improves the performance of MANET network and come with a sufficient performance in a chaotic situation such as MANET.

In this work, we suggested a new mechanism based on the InfiniBand architecture to enhance QoS of multipath routing protocols in MANETs. The simulation results of three QoS performance metrics: packet delivery ratio, end-to-end delay, and control overhead have shown that our proposed mechanism outperforms both AOMDV and AODV and gives better performance measures. In average, we have achieved 8% enhancement in Throughput, 26% enhancement in packets delivery, 45% in end-to-end delay, and about 15% reduction in control overhead and this average has been calculated regardless of the nodes number that exist in the network topology and regardless to the data rate used.

This can show that IB architecture with QoS rules might dominate the future architecture solutions for MANET networks, and of course it needs more studies to find out more about the advantages it may have and the disadvantages it may cause and try to solve them in a messy network environment that looks more like a natural unorganized network of nodes.

Finally, we need also to note that IB architecture has no fixed managing nodes, actually every node has a control virtual lane and 15 data virtual lanes so that it has to process the control virtual lane at first and then process the data virtual lanes in a prioritized way so that. That of course has overcome the problem of centralized control and other types of controlling, and also has minimized the packet loss by changing the packet size, and minimized the important packets delay time.

5.2 Future Work

Our future work is split into two parts, first one is to try to apply our solution with more advanced routing algorithms at MANET which use TARA strategy ‘traffic aware routing algorithm’ to make a compete full comparison, second is that since IP MANET network is still suffering from disconnection issues because there is no sufficient soft handover, this issue is considered a big topic at telecommunication and HPC networks, in future we will study the soft handover mechanisms on networks model and come up with sufficient improvements at routing failure and soft handover.

Appendix

Dispatch packet via VL:

```
int
IBMSDispatcher::routeMadToDestByLid(
  madItem &item)
{
  IBMSNode *pCurNode = NULL;
  IBMSNode *pRemNode = item.pFromNode;
  IBPort   *pRemIBPort; /* stores the incoming remote port */
  uint16_t lid = item.madMsg.addr.dlid;
  uint8_t   prevPortNum = 0, index;
  int outPort = 0;
  int hops = 0, vl = 0, vl_sl, sl, isVl15;
  ib_slvl_table_t * slVLTbl;

  isVl15 = (item.madMsg.header.mgmt_class == IB_MCLASS_SUBN_LID ||
      item.madMsg.header.mgmt_class == IB_MCLASS_SUBN_DIR);

  IBMSNode * startNode = item.pFromNode;

  sl = item.madMsg.addr.sl;
  if(item.madMsg.header.mgmt_class == IB_MCLASS_SUBN_LID ||
      item.madMsg.header.mgmt_class == IB_MCLASS_SUBN_DIR) {
    vl = 15;
  } else {
    slVLTbl = startNode->getSL2VLTable(0, 0);
    if (!slVLTbl) {
      return 1;
    } else {
      index = sl/2;
      if (sl % 2)
        vl = (slVLTbl->raw_vl_by_sl[index]) & 0xf;
      else
        vl = (slVLTbl->raw_vl_by_sl[index]) >> 4 & 0xf;
    }
  }

  /* we will stop when we are done or stuck */
  while (pRemNode && (pCurNode != pRemNode))
  {
    /* take the step */
    pCurNode = pRemNode;
    if(hops == MAX_NUM_OF_HOPS) {
      break;
    }
    if(vl > 15) {
      break;
    }
    /* this sim node function is handling both HCA and SW under lock ... */
    if (pCurNode->getIBNode()->type != IB_SW_NODE)
    {
      /* HCA node - we are either done or get out from the client port num */
      if (hops == 0)
```

```
   {
    // catch cases where the lid is our own lid - use the port info for that
    if (cl_ntoh16(pCurNode->nodePortsInfo[item.fromPort].base_lid) == lid)
    {
     pRemNode = pCurNode;
     pRemIBPort = pCurNode->getIBNode()->getPort(item.fromPort);
    } else {
     if (pCurNode->getRemoteNodeByOutPort(
        item.fromPort, &pRemNode, &pRemIBPort, vl)) {
      // Got to dead-end routing to lid
      return 1;
     }
     if (pRemIBPort) {
      prevPortNum = pRemIBPort->num;
     }
    }
   } else {
    /* we mark the fact we are done */
    pRemNode = pCurNode;
   }
  } else {
   /* check adaptive roting */
   if (ib_ar_is_enabled(&pCurNode->AdaptiveRoutingInfo)) {
    if (pCurNode->getRemoteNodeByLidAR(lid, &pRemNode, &pRemIBPort, prevPortNum, sl,
vl, &outPort)) {
              // Got to dead-end routing to lid
     return 1;
    }
   } else if (pCurNode->getRemoteNodeByLid(lid, &pRemNode, &pRemIBPort, vl, &outPort)) {
    // Got to dead-end routing to lid
    return 1;
   }
   /* if the remote port identical to cur node that target is this switch */
   if (pCurNode == pRemNode) {
    int res = pRemNode->processMad(prevPortNum, item.madMsg);
    return(res);
   }
   if (pRemIBPort) {
    slVLTbl = pCurNode->getSL2VLTable(prevPortNum, outPort);
    if (!slVLTbl) {
     /* if not VL15 then look for VL in the SL2VL Table*/
    } else if (vl != 15) {
     /*check if to use vl2vl*/
     if (pCurNode->extSwitchInfo.sl2vl_act == 1)
      vl_sl = vl;
     else
      vl_sl = sl;
     index = vl_sl/2;
     if (vl_sl % 2)
      vl = (slVLTbl->raw_vl_by_sl[index]) & 0xf;
     else
      vl = (slVLTbl->raw_vl_by_sl[index]) >> 4 & 0xf;
    }
    prevPortNum = pRemIBPort->num;
   }
  }
  hops++;
}
```

QoS file example:

```
port-group
        port-group # using port IPv6
            name: Storage
            # "use" is just a description that is used for logging
            #  Other than that, it is just a comment
            use: SRP Targets
            port-IPv6: 0x10000000000001, 0x10000000000005-0x1000000000FFFA
            port-IPv6: 0x1000000000FFFF
        end-port-group

    end-port-groups

    qos-setup
        # This section of the policy file describes how to set up SL2VL and VL
        # Arbitration tables on various nodes in the fabric.
             qos_max_vls 12
             qos_high_limit 8
             qos_vlarb_high 0:128,1:64,2:64,3:32,4:32,5:16
             qos_vlarb_low 6:256,7:196,8:128,9:96,10:64,11:56,12:48
             qos_sl2vl 0,1,2,3,4,5,6,7,8,9,10,11,12,15,15,15.
    end-qos-setup

    qos-levels
        # Having a QoS Level named "DEFAULT" is a must - it is applied to
        # PR/MPR requests that didn't match any of the matching rules.
        qos-level
            name: DEFAULT
            use: default QoS Level
            sl: 0
        end-qos-level

        # the whole set: SL, MTU-Limit, Rate-Limit, PKey, Packet Lifetime
        qos-level
            name: WholeSet
            sl: 1
            mtu-limit: 4
            rate-limit: 5
            pkey: 0x1234
            packet-life: 8
        end-qos-level

    end-qos-levels

    qos-match-rules

        # matching by single criteria: QoS class
        qos-match-rule
            use: by QoS class
            qos-class: 7-9,11
            # Name of qos-level to apply to the matching PR/MPR
            qos-level-name: WholeSet
        end-qos-match-rule

        # show matching by destination group and service id
        qos-match-rule
            use: Storage targets
            destination: Storage
            service-id: 0x10000000000001, 0x10000000000008-0x10000000000FFF
            qos-level-name: WholeSet
        end-qos-match-rule

        qos-match-rule
            use: match by all parameters
            qos-class: 7-9,11
            source: Servers
            destination: Storage
            service-id: 0x0000000000010000-0x000000000001FFFF
            pkey: 0x0F00-0x0FFF
            qos-level-name: WholeSet
        end-qos-match-rule

    end-qos-match-rules
```

References

[1] Yuh-Shyan Chen, Shin-Jr Jan and Ming-Chin Chuang, "A shoelace-based QoS routing protocol for mobile ad Hoc networks using directional antenna," TENCON 2007 - 2007 IEEE Region 10 Conference, Taipei, 2007, pp. 1-4. doi: 10.1109/TENCON.2007.4428806.

[2] P. P. Pham and S. Perreault, "Performance analysis of reactive shortest path and multipath routing mechanism with load balance," IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428), 2003, pp. 251-259 vol.1.
doi: 10.1109/INFCOM.2003.1208677.

[3] Yuh-Shyan Chen, Shin-Jr Jan and Ming-Chin Chuang, "A shoelace-based QoS routing protocol for mobile ad Hoc networks using directional antenna," TENCON 2007 - 2007 IEEE Region 10 Conference, Taipei, 2007, pp. 1-4.
doi: 10.1109/TENCON.2007.4428806.

[4] Infinibandta, http://www.infinibandta.org/, available online, [Last Accessed; 15/2/2017].

[5] OpenFabric, http://linuxsoft.cern.ch/cern/slc57/i386/yum/updates/repoview/ibsim.html, available online, [Last Accessed; 15/2/2017].

[6] F. J. Alfaro, J. L. Sanchez and J. Duato, "QoS in InfiniBand subnetworks," in IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 9, pp. 810-823, Sept. 2004.
doi: 10.1109/TPDS.2004.46.

[7] Pelissier, Joe. "Providing quality of service over Infiniband architecture fabrics." In Proceedings of the 8th Symposium on Hot Interconnects, pp. 127-132. 2000.

[8] A. Vidwans, A. K. Shrivastava and M. Manoria, "QoS Enhancement of AOMDV Routing Protocol Using Queue Length Improvement," 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, 2014, pp. 275-278. doi: 10.1109/CSNT.2014.60.

[9] Lou, Wenjing, Wei Liu, and Yanchao Zhang. "Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks." In Combinatorial optimization in communication networks, pp. 117-146. Springer US, 2006.

[10] Devi, P. Rama, and D. Srinivasa Rao. "QOS enhanced hybrid multipath routing protocol for mobile adhoc networks." International Journal of Distributed and Parallel Systems 3, no. 6 (2012): 89.

[11] Ardagna, Danilo, Giuliano Casale, Michele Ciavotta, Juan F. Pérez, and Weikun Wang. "Quality-of-service in cloud computing: modeling techniques and their applications." Journal of Internet Services and Applications 5, no. 1 (2014): 11.

[12] Yeng-Zhong Lee, J. Chen, Xiaoyan Hong, Kaixin Xu, T. Breyer and M. Gerla, "Experimental evaluation of LANMAR, a scalable ad-hoc routing protocol," IEEE Wireless Communications and Networking Conference, 2005, 2005, pp. 2032-2037 Vol. 4.

doi: 10.1109/WCNC.2005.1424831.

[13] Venkatasubramanian, S., and N. P. Gopalan. "Multi-path qos routing protocol for load balancing in manet." International Journal of Networking & Parallel Computing 1, no. 3 (2013).

[14] Qin, Fujian. "QoS topology control with energy efficiency for MANET." Journal of Convergence Information Technology 6, no. 6 (2011): 300-307.

[15] Pradeep, B. S., and S. Soumya. "A new method for load balancing and QOS in on demand protocols—in the MANET's perspective." International Journal of Advanced Networking and Applications 1, no. 4 (2010): 275-281.

[16] J. Alfaro, L. Sa'nchez, and Duato "QoS in InfiniBand Subnetworks" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 15, NO. 9, SEPTEMBER 2004

[17] J. Pelissier." Providing Quality of Service over InfiniBand™ Architecture Fabrics" 5200 N. E. Elam Young Parkway Hillsboro, OR 97124-6497 2001

[18] Ryan E. Grant, Rashti and Afsahi. "An Analysis of QoS Provisioning for Sockets Direct Protocol vs. IPoIB over Modern InfiniBand Networks" onference: Parallel Processing - Workshops, 2008.

[19] Archlinux, https://wiki.archlinux.org/index.php/InfiniBand, available online, [Last Accessed, 15/2/2017].

[20] A. Vishnu, A. R. Mamidala, S. Narravula and D. K. Panda, "Automatic Path Migration over InfiniBand: Early Experiences," 2007 IEEE International Parallel and Distributed Processing Symposium, Long Beach, CA, 2007, pp. 1-8.
doi: 10.1109/IPDPS.2007.370626.

[21] Richard Draves, Jitendra Padhye, and Brian Zill. 2004. Routing in multi-radio, multi-hop wireless mesh networks. In Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom '04). ACM, New York, NY, USA, 114-128. DOI=http://dx.doi.org/10.1145/1023720.1023732.

[22] Grun, Paul. "Introduction to infiniband for end users." White paper, InfiniBand Trade Association (2010).

[23] OpenFabric, https://sourcecodebrowser.com/ibutils/1.2-OFED-1.4.2/class_i_b_mgt_sim.html, available online, [Last Accessed; 15/2/2017].

[24] M. M. Zeru, "Exploring InfiniBand Congestion Control in Mesh Topologies," PhD Thesis, Department of Informatics, University of Oslo, Oslo, 2016.