Sudan University of Science and Technology

Collage of Post Graduate

Performance Evaluation of Buffer Management Techniques in Delay Tolerant Network

تقييم أداء تقنيات إدارة الذاكرة المؤقتة في شبكات ذات تحمل التأخير

A Research Submitted in Partial fulfillment for the requirements of the Degree of M.Sc. In Electronic Engineering (Computer and Network)

Prepared by:
Ahmed Mohammed Osman Ahmed

Supervised by:
Dr. Fath Elrahmann Ismael Khalifa

Dec 2018

الآية

بسم الله الرحمن الرحيم

(لاَيُكَلِّفُ اللّهُ نَفْساً إِلاَّ وُسْعَهَا لَهَا مَاكَسَبَتْ وَعَلَيْهَا مَااكْتَسَبَتْ رَبَّنَا لاَتُؤَاخِذْنَا إِن نَّسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلاَ تَحْمِلْ عَلَيْنَا إِصْراً كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِن قَبْلِنَا رَبَّنَا وَلاَتُحَمِّلْنَا مَالاَطَاقَةَ لَنَا بِهِ وَاعْفُ عَنَّا وَاغْفِرْ لَنَا وَارْحَمْنَا أَنتَ مَوْلاَنَا فَانصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ)

سورة البقـرة - الآية 286

صدق الله العظيم

DEDICATION

I dedicate this project report to Allah Almighty for his infinite mercy that He has granted to me through my studies and also dedicate this work to my parents for their motivation and moral support throughout my day in study May Allah Almighty bless them and sustain their help to reap the fruit of their efforts .

AKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to everyone who supported me in this project.

And to my parents who have always give me the strength and wisdom to be sincere in my work, for setting high moral standards, supporting me through their hard work unselfish love and affection.

I would like also to express my gratitude to my supervisor **Dr.Fath Elrahman Ismael khalifa** who strongly supported me from the very beginning and provided me with valuable advice. Great thanks for him for the endless support and encouragement. Without his guidance and encouragement, I would not be able to do this.

ABSTRACT

Trying to enhance the network quality of service which becomes an important issue in networks world. Buffer management techniques one of the way to increase the performance of the network precisely in the Delay Tolerant Network (DTN). The main focus was on SHLI and LEPR Techniques to get the maximum potential of the network resources. The challenge is how to differentiate between the packets which delivered to nodes to define which packets have higher priority than others. Then a Differentiated Service Model (DiffServ) mechanism is proposed to solve this obstacle. This work will be done through simulation using MATLAB. A mathematical model was used to evaluate the performance of the network in term of QoS parameters such as (Delay time, Bandwidth Utilization, Throughput and Data rate). After implementing and run both algorithm over three type of services (File transmission, Video and Voice) , results showed that percentage difference of delay time when apply both algorithm SHLI and LEPR for all services is 20%. For Throughput SHLI have higher value than LEPR about 18%.Regarding to Bandwidth Utilization, SHLI is most usage of resource than LEPR compared to LPER due to the increased packet size clustering. Then, data rate for SHLI is greater than the LPER by 23%. Also showed which techniques are better to implement for specific type of service or application to get high utilization of the resource which leading to high performance.

المستخلص

في محاولة لتحسين جودة الشبكة في خدمة محددة والتي أصبحت عامل مهم في عالم الشبكات. تقنيات إدارة المخزن المؤقت هي إحدى الطرق لزيادة على وجه التحديد في شبكات ذات تحمل التأخير . كان التركيز الرئيسي على نوعين من التقنيات : تقنية إسقاط الأقل زمن حياة و تقنية إسقاط الأقل احتمالية وصول للحصول على أقصى إمكانات موارد الشبكة. يتمثل التحدي في كيفية التمييز بين الحزم الواصلة للعقدة لتحديد أي حزمة لها أولوية أعلى من غيرها. ثم تُقترح آليات نموذج الخدمة المتمايزة لحل هذه العقبة. وسيتم هذا العمل من خلال محاكاة باستخدام برنامج الماتلاب. تم استخدام معادلات رياضية لتقييم أداء الشبكة من حيث معلمات جودة الخدمة مثل (وقت التأخير ، استخدام عرض النطاق الترددي ، معدل النقل ، معدل البيانات) . بعد تنفيذ الخوارزمية وتشغيلها على ثلاثة أنواع من الخدمات (نقل الملفات , فيديو و صوت) , أظهرت النتائج أن نسبة الاختلاف في زمن التأخير بالنسبة للتقنيتين تقنية إسقاط الأقل زمن حياة وتقنية إسقاط الأقل احتمالية وصول لجميع الخدمات هي ٢٠%.أما معدل النقل تقنية إسقاط الأقل زمن حياة أعلى من التقنية الأخرى تقنية إسقاط الأقل احتمالية وصول بنسبة ١٨% . وفيما يتعلق باستخدام النطاق الترددي ، فإن تقنية إسقاط الأقل زمن حياة هي الأفضل مقارنة مع تقنية إسقاط الأقل زمن وصول نظرًا لزيادة حجم حزم البيانات . ثم معدل البيانات لتقنية إسقاط الأقل زمن حياة أكثر من تقنية إسقاط الأقل زمن وصول بنسبة ٢٣%. أيضا توضح النتائج أي الخوارزميات هي الأفضل في نوع محدد من الخدمات أو التطبيقات للحصول على أعلى استفادة من الموارد مما يؤدي إلى الأداء العالي.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

BER	Bit Error rate
BU	Bandwidth Utilization
BW	Bandwidth
DR	Data rate
MbPs	Mega Bit Per Second
SNR	Signal to Noise Ratio
TD	Transmission Delay
THP	Throughput

ABBREVIATIONS

AQM	Active Queue Management
DiffServ	Differentiated Service
DTN	Delay Tolerant Networking
IntServ	Integrated Service
IP	Internet Protocol
KPI	Key Performance Indicator
LEPR	Least Probability
MV	Meets and Visits
PREP	PRioritized EPidemic
QoS	Quality of Services
RED	Random Early Detection
RED	Random Early Detection
RTT	Round Trip Time
SHLI	Shortest Life time
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time To Live

CHATPTER ONE
INTRODUCTION

1.1 Preface

Delay Tolerant Networking (DTN) is a technology which supports data transfer in challenging environments where a fully connected end to end path may never exist between a source and destination. There is intensive needed to deal with disconnections of path which directly impacts routing and forwarding of data. Also the main issue is how to deal with limited resources like buffer, bandwidth and power, because most of DTN suppose that the resources have infinite capacity [1] . Consequently an efficient buffer management policy is required under resource constrained DTNs.

Large amount of research has been performed in developing efficient routing algorithms for DTNs. However, it is observed that flood-based routing protocols perform poorly when resources like buffer and bandwidth are limited.

DTNs operate with the principle of store, carry and forward. In order to cope with long disconnection, messages must be buffered for long period of time. It implies that intermediate nodes require enough buffer space to store all messages that are waiting for future communication opportunities. Moreover to achieve high delivery probability, messages are replicated to each and every node they encounter. The combination of long term storage and extensive message replication performed often by many DTN routing protocols imposes a high storage overhead on wireless nodes. In addition, bundles which are application-level data units can often be large. In this context, it is evident that buffers will run out of capacity at certain point of time.

In this research, an adaptive buffer management policy with prioritization is proposed which takes care of both: which messages are to be transmitted when a new contact arises and which message will be dropped when node is overflow. The proposed policy does selective dropping and scheduling. It considers the life time as well as the priority of the messages in making such decisions. The overall performance has been enhanced by using Shortest Life time (SHLI) and Least Probability
(LPER) techniques [2].

1.2 Problem Statement

Due to large number of packets on DTN network and because of massive number of massage buffering and forwarding operation specifically on nodes, each methods of transmitted data has its own criteria and characteristics which affect on performance of the network. As a result of that, efficient utilization and management of each buffer needed to insure that the network is always used its maximum potential and enhanced the QoS of the network in certain key performance indicator (KPI) such as Transmission delay, Throughput, Bandwidth Utilization and Data Rate.

1.3 Proposed Solution

A lot of buffer management techniques are used to prevent the buffer overflow. In this research two methods, which are SHLI technique and LEPR, are evaluated to determine their efficiency for multiple services.

1.4 Aim and Objectives

The aim of this research is to compare between different mechanisms that will effects on overall performance, the detailed objectives of this research include:

- To make the capacity of network full utilized.
- To improve the network throughput and capacity.
- Minimize the latency (delay).
- To Enhance the Data Rate of the services.

1.5 Methodology

The methodology used in the research process has been to start with a literature survey related to same studies, which is DTN (Delay Tolerant Network) and will consider the structure and information of this network, then will go through the key performance parameters (metrics) of overall network and how to compare the performance and goals of thesis after applying the buffer management techniques which is SHLI and LPER for File transfer, voice and video services. After that the proposed solutions are further simulated over a MATLAB tool and will mention the calculation used to compare between the metrics on system similar to Ad Hoc network system model to find out which algorithm is better to used in specific kind of service (File transfer, voice and video) and the simulation will be created and test the result of above will be recorded and used in writing this thesis. More details in chapter 3 and chapter 4.

1.6 Thesis Outlines

This thesis is divided into five chapters. Chapter one is an introduction explains briefly the purpose of this Thesis. Chapter two is a literature review that gives a brief review of DTN – Delay Tolerant Network, also provides over view of different technologies that are used to enhance the performance of such systems. Chapter three discusses the mechanism used to Queuing Policies and Forwarding Strategies and it covers how it can be deployed. Chapter four presents a detailed discussion of the simulation results. Chapter five provides the conclusion and recommendations.

CHATPTER TWO

LITRATURE REVIEW

2.1 Introduction

In the Internet of Things era, every one of over a trillion everyday items will include at least some ability to store and process information; additionally, and more importantly, sharing that information over the global Internet with the other trillion items. The technological goal is to integrate the Internet and the web with everyday objects (such as doors, chairs, electric appliances, cars, etc.) and ultimately interconnect the digital and physical domains. Clearly, the types of objects to be connected with the Internet, e.g., in terms of usage, size and numbers, are extremely diverse, thus having different computation and communication requirements. For this reason, a large number of computing architectures and networking paradigms have been proposed, and different networking standards have been developed [3].

In most cases, the operational and performance characteristics of the newly-introduced technologies make conventional (Internet-like) networking approaches either unworkable or impractical. Concepts of occasionally-connected networks have become a very common approach for a very diverse range of real-world applications; real-world cases that suffer from frequent partitions and that rely on more than one divergent set of protocols or protocol families. The dominant approach of networking in such cases is to provide the nodes with significant memory capabilities, so that messages can be stored for long periods of time. Whenever communication is established, the stored messages are opportunistically forwarded to the connecting node [4].

However, even though standard Internet protocols perform admirably over a varied range of wired and wireless networks, they make some implicit assumptions about the underlying network characteristics : Firstly, there is at least one end-to-end path between the two communicating nodes. Secondly, the maximum round-trip time (RTT) is relatively low. Lastly, the link error rates are sufficiently small [5].

Over the past few years, a number of new network technologies have found increasing application in research as well as in military and civilian deployments. Examples of these include :

1. Terrestrial Mobile Networks: These may include, for example, a limited-power RF transceiver equipped bus, which travels around the city providing opportunistic communication capability to nodes that happen to be within its transmission radius. Such networks may suffer from frequent disconnection, due to node mobility and RF interference [6].

2. Exotic Media Networks: These networks include near-Earth satellite communications, free-space optical communication and underwater acoustic links. Link delay may be high in such scenarios because of lossy links, as well as large RTTs[2].

3. Military Ad-hoc Networks: These frequently consist of several mobile nodes deployed on the battlefield. Seamless communication may not always be possible in such networks because of hostile elements such as intentional jamming or simply, node failures [7].

4. Sensor Networks: These networks are characterized by a high volume of nodes with only limited power and processing capability. Again, these nodes are susceptible to failure due to power outages, or low-duty cycle functioning. As illustrated above, these networks quash several assumptions implicit in the design of the Internet with attributes such as significant link delay, large RTTs, and the possibility of the existence of no single end-to-end path between end-nodes for long durations. Due to these characteristics, such networks are broadly classified as challenged networks [8].

2.1.1 The Delay-Tolerant Network Architecture

A Delay-Tolerant Network (DTN) is envisioned as an overlay over the diverse set of networks described in section 1, including, of course, the Internet. It is divided into regions - networks that are homogeneous in terms of link delay, link connectivity, data-rate asymmetry, quality of service (QoS), reliability mechanisms, etc. Each network may have its own protocol stack, optimized for its characteristics. The DTN as a whole may never be fully-connected, in the sense that, there may not exist a continuous end-to-end path between two nodes, lying in two different regions, at any given instant of time. That is, network partitions are implicitly assumed. However, whenever an opportunity exists to send user data from a node in a given network, to another node present somewhere else in the DTN, a contact is said to have been made [9]. Thus, even though a fully connected end-to-end path may never be available, data is routed sequentially over intermittent links that may be restored over time. This is referred to in the literature as space-time routing [10] .

Toward this end, the DTN architecture proposes the introduction of another layer just above the transport layer, called as the bundle layer in order to ferry data across the DTN . Figure 2.1 is a depiction of a traditional Ethernet-based network being incorporated into a DTN so that it can communicate with a challenged network via a DTN gateway. The challenged network may use transport and network protocols different from TCP/IP. The DTN gateway communicates with the Internet using TCP/IP and with the challenged network using the specific protocols [11].

All nodes in the DTN implement the bundle layer. The bundle layer abstracts data into bundles and routes them from source to destination, hop by hop. This behavior may be reminiscent of the way IP switches packets in the Internet. However, IP, being an asynchronous protocol, only promises best-effort packet switching. Packets, whose next hop is not reachable, would eventually get discarded. Also, since no end-to-end path is ever available, TCP sessions simply timeout. DTNs overcome this issue by terminating TCP and other delay-sensitive transport layer protocols at the Bundle layer so that, in reality, the TCP connection only extends to the DTN gateway and not to the destination node as can be seen in Figure 2.1.

Instead of packet-switching, DTNs use the concept of store-and-forward message- (or bundle-) switching. This is because, at any given instant, there may not be any route to the next hop. In this case, the node must buffer the message in persistent storage, until a contact becomes available.

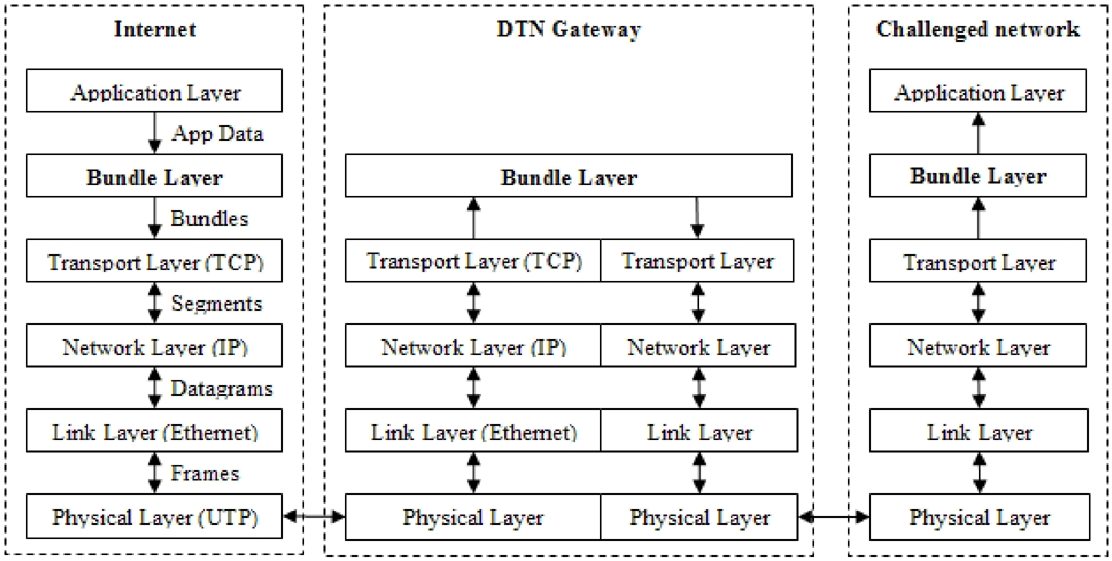

Figure 2.1: Interworking of network protocols in a Delay-Tolerant Network [12]

Once the next hop stores the bundle in persistent storage, it is said to have taken custody of the bundle, and the upstream node can delete its own copy of the bundle. Instead of waiting for the next hop to become available, the DTN gateways may themselves be mobile. This extension of the message switching concept is referred to as store-carry-forward routing [12].

A draft formulation of the DTN architecture has been published as an IETF RFC4838. In [DBF+04], a reference implementation has been proposed and evaluated against traditional Internet style data transfer methodologies, with a special focus on a DTN bundle forwarding system. In this system, the bundle router module gets constantly updated with state information based on which it makes routing decisions and passes them to the bundle forwarder module. The bundle forwarder then executes these decisions. This separation of the policy from function has the advantage that the implementation of the actual routing methodology is largely isolated from the rest of the system [13].

2.1.2 Random Early Detection (RED) mechanism for Congestion Control

Congestion occurs on a network when a device, such as a router, is receiving more packets than it can handle. “The problem with end to end congestion control schemes is that the presence of congestion is detected through the effects of congestion, e.g., packet loss, increased round trip time (RTT), changes in the throughput gradient, etc., rather than the congestion itself e.g. overflowing queues “The gateway can reliably distinguish between propagation delay and persistent queuing delay. Only the gateway has a unified view of the queuing behavior over time; the perspective of individual connections is limited by the packet arrival patterns for those connections. In addition, a gateway is shared by many active connections with a wide range of roundtrip times, tolerances of delay, throughput requirements, etc.; decisions about the duration and magnitude of transient congestion to be allowed at the gateway are best made by the gateway itself.”

A new mechanism called Random Early Detection (RED) was proposed by Sally Floyd [14] . RED is an Active Queue Management (AQM) mechanism that is implemented at the gateway in order to ‘avoid’ congestion rather than ‘respond’ to a situation that may not even be congestion related.

The RED gateway is an AQM congestion avoidance technique that takes advantage of TCP’s
congestion control mechanism to try to keep the queue for connections as low as possible When the average queue size reaches a defined threshold, RED notifies connections of congestion randomly by either dropping the

packets arriving at the gateway or by marking it with a bit but the focus in this paper is notification by dropping of packets. RED particularly relevant for avoiding global synchronization in networks where new or restarted transmissions go through the slow-start phase before reaching the congestion threshold.

2.1.3 Logical Partitioning of the Buffer

One of the way to achieve better performance of DTN, routing schemes in DTNs have to also effectively utilize the limited encounter opportunity.
Message prioritization has emerged to overcome this issue in a node's buffer. This type of buffer management assume that every node has a fixed and limited sized buffer regarding to bandwidth and duration, and nodes do not have any information regarding future network connectivity. At the beginning all massage take the same priority, but practically, the operation of DTN is carried out in the following three stages: Node Discovery, Message Transfer and Buffer Management.
Nodes have to discover each other before the transfer of the messages can take place, then nodes start exchanging messages with each other. the amount of data to be transmitted is limited, then Buffer Management take a place, when new messages are received in the node's buffer must sort the messages according to the buffer management scheme which is logical division and sorting.

The logical division and sorting of the buffer is shown in Figure 2.2. The Messages which have a lowest hop-count are scheduled to be transmitted first. This is because, low hop-count depicts that the message hasn't travelled far from the source and is still far from the destination. Therefore, such message must be prioritized for transmission to create more copies so as to achieve higher delivery rate and lower delivery delay. The message which has higher hop-counts and low TTL are the first to be dropped. This is because, these messages have low probability to reach the destination from the current node because of low TTL and since their hop count is high, it is safe to assume that they have been sufficiently spread into the network that one of the copies will reach the destination if the current copy is dropped. [15]

This scheme is based on the idea that the message will eventually find its destination through transitive exchanges between nodes, if it is spread in the connected portion of the network.

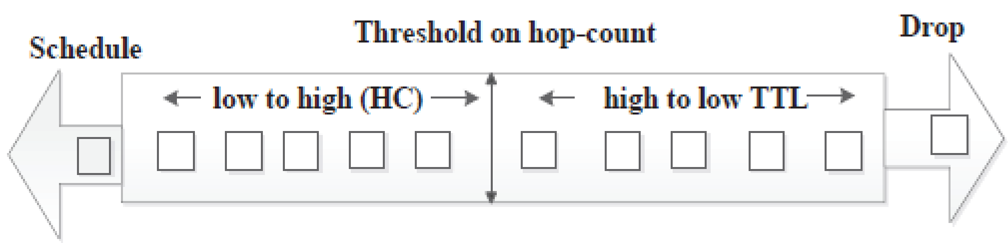

Figure 2.2: Logical division of Buffer space [15]

2.1.4 Epidemic Routing Protocol

One of the ways to get optimal buffer management that takes into account all information that are relevant for encounter-based (or store carry-and forward) message delivery. This policy uses global information about the network either to maximize the average delivery rate or to minimize the average delivery delay. Also there is an algorithm uses to estimate information about the global state of the network after statistical learning, this estimation used to choose the optimal algorithm in practice [16].

The nodes continuously replicate and transmit messages. When a node comes within transmission range of another node, it checks whether its new neighbor has a copy of the message that needs to be transmitted. If it does not, then a replica of the message is forwarded to that node. This is done using Summary Vectors. Each host has a buffer of messages that it originated as well as messages that it is buffering on behalf of other nodes. It maintains a hash table that indexes this list and keys that uniquely identifies each message. Each host maintains a bit vector called summary vector that indicates which entries in the hash table are set. When two hosts come in communication range of one another, the hosts exchange their summary vectors to determine which messages stored remotely have not been seen by the local host. Each host then requests a copy of the messages that it has not seen yet [17]. The receiving host has complete autonomy
to accept or reject a message as shown in figure 2.3.

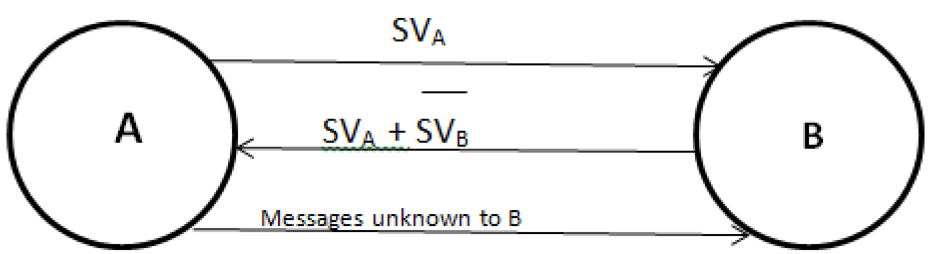

Figure 2.3: Epidemic Routing Protocol when two hosts, A and B, come into transmission range of one another [17].

2.2 Related Works

The following information describes Buffer management techniques that are used to enhancement the DTN. The authors in [18] introduced a drop and forward policy for epidemic routing protocols. PRioritized EPidemic (PREP) scheme prioritizes bundles based on source-destination cost and bundle expiry time. The cost is the average outage time of links on a path, and this information is flooded throughout a DTN and is used by Dijkstra algorithm to compute the minimum source-destination cost. In his drop policy, a node with a full buffer first selects bundles that have a hop-count value greater than a threshold. Accordingly, selected bundles are sorted based on their cost to their intended destination and the bundle with the maximum cost is dropped first.

In terms of transmission priority, if a bundle incurs a lower cost of delivery through an encountered node, the bundle with the longest remaining lifetime will be forwarded first. The main limitation of PREP is that it requires the link cost to be flooded. However, due to large delays and topological changes, the computed path cost may become out of date quickly.

The author in [18] propose Meets and Visits (MV), a scheme that learns the frequency of meetings between nodes and how often they visit a certain region. This information is used to rank each bundle according to the likelihood of delivering a bundle through a specific path. However, many bundles with the same destination may exist in a node's buffer. Hence, in his case, all of them have the same priority to be forwarded whereas their different TTL values can affect bundle delivery.

The author in [19] present an analysis of buffer-constrained Epidemic routing, and evaluate some of the simple drop policies previously described. The authors conclude that Drop First outperforms Drop Last in terms of both delivery delay and delivery ratio. Additionally, giving priority to source messages improves the delivery ratio further, but makes messages spread slower, increasing their delay. The author in [20] evaluate a somewhat more extensive set of combinations of existing buffer management policies and routing protocols for DTNs.

They show that Probabilistic routing [21] together with the right buffer management policy can result in performance improvements in terms of message delivery, overhead, and end-to-end delay. Specifically, in the context of Epidemic routing, the authors found that Drop First (with priority to source messages) gives the highest delivery ratio.

The author in [22] proposed the highly scalable cluster-based hierarchical trust management protocol for Wireless Sensor Network. The utility of hierarchical trust management protocol was demonstrated by applying the trust-based geographic routing and intrusion detection approaches. The existence of optimal trust threshold value effectively minimized the false positives and false negatives. The maintenance of high-quality WSN dependent on the delay requirement.

The overall goal of this research to compare between different mechanisms used to make the proper buffer management for the network nodes and to enhance the performance of it. This accomplishes by using Queuing Policies and Forwarding Strategies such as SHLI and LPER mechanisms simulated over a MATLAB tool. After evaluation the results shows which policy can be used to get the maximum network utilization for specific type of service to get the high QoS leading to best customer experience.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 27-34.

[4] A. V. Vasilakos, T. Spyropoulos, and Y. Zhang, *Delay tolerant networks: Protocols and applications*: CRC press, 2016.

[5] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks," in *International Conference on Research in Networking*, 2005, pp. 1180-1192.
[6] S. Krishnamurthy, M. Ahmed, and G. D. Kondylis, "Method and apparatus for determining position and trajectory of gateways to optimize performance in hybrid non-terrestrial-terrestrial multi-hop mobile networks," ed: Google Patents, 2004.
[7] B. Shubashin and A. S. Thanamani, "An Opportunistic On Routing Protocols and Persisting Challenges in Delay-Tolerant Networking," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 2, pp. 5373-5379, 2014.
[8] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks,* vol. 3, pp. 257-279, 2005.
[9] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*, 2008, pp. 260-268.
[10] Z. Zhang, "Routing protocols in intermittently connected mobile ad hoc networks and delay-tolerant networks," *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks,* p. 219, 2009.
[11] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected areas in communications,* vol. 26, 2008.
[12] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott*, et al.*, "Delay-tolerant networking architecture," 2070-1721, 2007.
[13] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies*, et al.*, "Information-centric networking: baseline scenarios," 2070-1721, 2015.
[14] S. Floyd, "Metrics for the evaluation of congestion control mechanisms," 2070-1721, 2008.
[15] C. Sobin, "An Efficient Buffer Management Policy for DTN," *Procedia Computer Science,* vol. 93, pp. 309-314, 2016.
[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking (ToN),* vol. 16, pp. 77-90, 2008.
[17] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," 2000.
[18] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, 2007, pp. 62-66.
[19] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks,* vol. 51, pp. 2867-2891, 2007.
[20] A. Lindgren and K. Phanse, "Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks," in *International*

Conference on Communication System Software and Middleware: 08/01/2006-12/01/2006, 2006, pp. 1-10.

[21] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *ACM International Symposium on Mobilde Ad Hoc Networking and Computing, MobiHoc 2003: 01/06/2003-03/06/2003*, 2003.

[22] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE transactions on network and service management,* vol. 9, pp. 169-183, 2012.

CHATPTER THREE
Buffer Management Model and Simulation

This chapter presents the methodology of the project including the mathematical model, computer model and the network simulation scenario.

3.1 QoS Implementation

Each network must have a QoS mechanism for real-time traffic that operates at each switch and router to prioritize real-time traffic. A number of different mechanisms exist in modern networks, such as IntServ (RSVP), DiffServ, IEEE 802.11p/q, and IP Precedence. To provide the maximum benefit to any application, QoS must work from end-to-end.

There are two methodologies available in most networks for implementing QoS at layer 3

- Integrated Service Model (IntServ)
- Differentiated Service Model (DiffServ)

3.1.1 Integrated Service (IntServ) model

This model assumes that each node in the network is QoS aware. It provide support for two broad class of services, real-time applications with strict bandwidth and latency requirements can be processed by guaranteed service while the traditional applications which require performance equivalent to lightly loaded network are provided controlled load service **[9]**. To negotiate the QoS with an IntServ capable network signaling and admission control is used.

IntServ has problems of scalability and is not yet widely accepted in deployment across the internet. In this project currently Differentiated Service model has been considered for implementation.

3.1.2 Differentiated Service (DiffServ) Model

Differentiated Service (DiffServ) as defined in RFC-2475 [3] was developed as an alternative approach that would have better scaling properties. Rather than specifying resources for each real-time stream, DiffServ allocates resources for a class of traffic. All traffic allocated to that class is treated with the same policy, such as being queued in a high priority queue.

DiffServ is provided by the router and defines a set of service classes with corresponding forwarding rules. A packet coming to the router with a Type of Service (ToS) field may get better services than other, provided by some classes depending upon that ToS content [10] .

3.2 Queuing Policies and Forwarding Strategies

Nodes may have to buffer messages for a long time and in case of congestion decide which messages to drop from its queue. They also have to decide which messages to forward to another node that is encountered. We use the following queue management policies, that define which message should be dropped if the buffer is full when a new message has to be accommodated.

1 - SHLI – This policy Evict shortest message’s life time first In the DTN architecture, Each node stores and forwards packets destined for other nodes, and each message existing in the network keeps a bunch of information about its source, its destination, the nodes it traversed as well as the Time to Live (TTL) value based on this information, each node requests packets when it is no longer useful and should be deleted to overcome the buffer congestion and to accept new packets. If this policy is used, the message with the shortest remaining life time is the first to be dropped, and this action will lead to increase the QoS.

2 - LEPR – This policy Evict least message's probable to deliver first. DTN protocol assumes that each node has a buffering queue. DTN networks have so many challenges, so to cope with them buffered nodes are used. It decides which message should be dropped in case of buffer is full. It is based on priority basis. When buffer gets full, the node is least likely to deliver a message for which it has a low P-value, drop the message for which the node has the lowest P-value [11] .Then the new message can accommodate in a queue. This mechanism helps to manage the buffer Due to storage limitation and to avoid congestion on nodes.

3.3 Computer model for SHLI Buffer Management

In figure 3.1 the SHLI buffer management technique flowchart was presented, and it starts by comparing packets arrival time to be stored in stack in order to start the time slot to drop the packets whenever the slot time is ended.

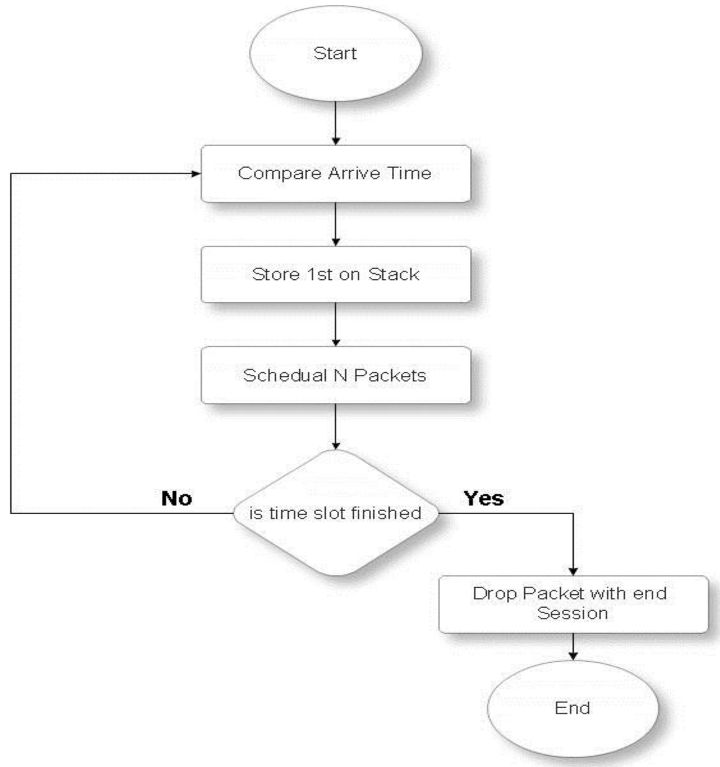

Figure 3.1: Computer Model for SHLI Buffer

the system starts by comparing the arrive time of each packet then store the packets as a first in first out with a time stamp or time slot when its ends a packet is dropped.

3.4 Computer Model for LPER Buffer Management

In figure 3.2 a flowchart that represents the buffer management behavior among received packets. Each packet in the system received has a prediction to the rank of delivery, so the system drops the packets whenever the ranking exceeds the threshold value.

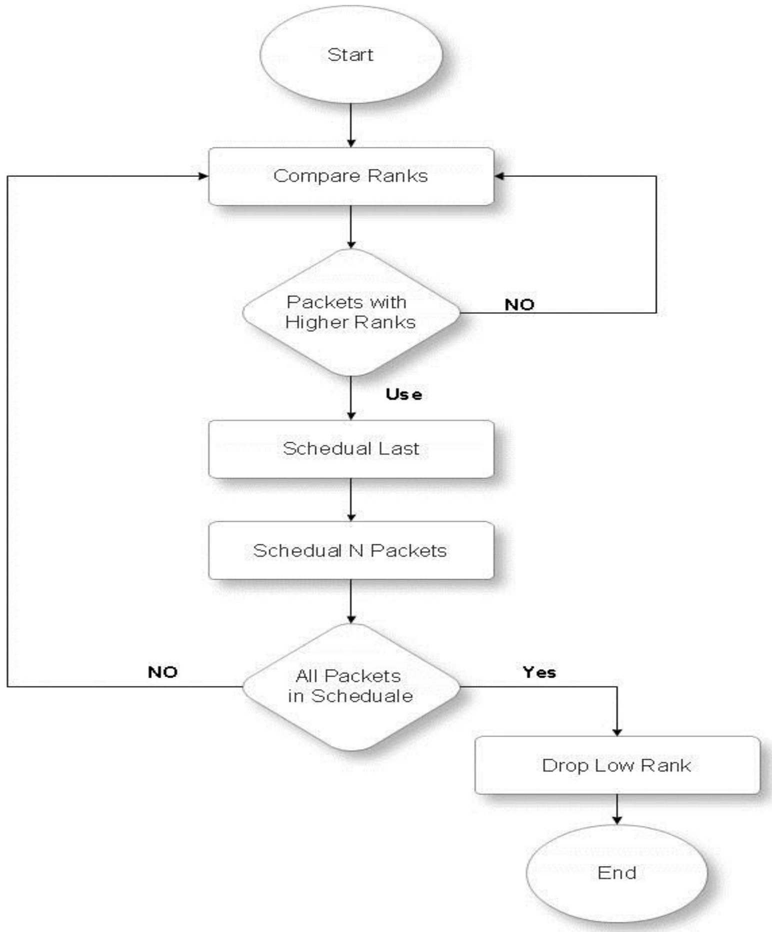

Figure 3.2: Computer Model for LPER Buffer

3.5 Mathematical Model

The usage of QoS parameters to evaluate the performance of the system, using some of the main parameters such as delay time, throughput, data rate and bandwidth utilization

3.5.1 Throughput

The number of data packets sent over the total simulation period refers throughput. This parameter is used to evaluate the number of successful packets received while running the simulation over the time of the simulation to calculate the success throughout of the system. The mathematical formulation for throughput is expressed by equation 3.1 :

$$\text{Thp} = \frac{\text{Number of packets send (bits)}}{\text{Time Period (sec)}} \qquad (3.1)$$

In this section, the percentage of throughput is investigated corresponding to the number of nodes variation.

3.5.2 Transmission Delay

The packet transmission duration is the value that the packets can be delivered to the delay tolerant node. The Transmission Delay can be calculated by equation 3.2 .

$$\text{Dt} = \frac{\text{N}}{\text{R}} \qquad (3.2)$$

Where Dt is the transmission delay in seconds, N is the number of bits and R means the rate of transmission (say in bits per second) 50MbPs

3.5.3 Data Rate

The system data rate is a mixture between the internal system configuration such as the bandwidth and the external noise such as AWGN, the Shannon equation used to calculate the maximum link capacity according to a specific noise variation and available bandwidth as equation 3.3 which expressed system data rate.

$$\text{DR} = \text{BW} * \log2(1 + \text{N}) \qquad (3.3)$$

Where

BW refers to Bandwidth and N means Noise Factor represented in db and can be expressed as SNR or SINR values.

3.5.4 Bandwidth Utilization

For one connection say requires Bandwidth equals to 1500 (Kbps or 1.5Mbps). The efficiency of the bandwidth and distribution of the resources on the services refer to Utilization of the Bandwidth as shown on 3.4 equation.

$$\mathrm{BU} = \frac{\text{Available bandwidth}}{\text{Required bandwidth}} \qquad (3.4)$$

3.6 Simulation Scenario

The delay tolerant network is a discontinued communication network with a high delay time, in this simulation work three scenarios where used to evaluate the network in medium, high and low. Therefore, three traffics where used to generate a heavy, medium and low loads into the network, these service are voice recorded notes, video and image in binary file format to transfer. This scenario was impaired from the NOAA weather satellite which offers a transmission through the day to earth weather round the world in three types of services, binary images, and tones with multiple frequencies and video transmission; moreover the duration is about 10 to 25 seconds.

CHATPTER FOUR
LITRATURE REVIEW

In this chapter the results and discussion was included, the chapter represent the output results of the proposed work including a quality of service parameters comparison for two major buffer management techniques SHLI and LEPR and includes service such as voice service , video service and data service.

4.1 Simulation Parameters

Table 4.1 shows the general simulation parameters used in the different scenarios of the simulation. According to the NOAA parameters [23] the frequency used in 1.5 GHz, with a maximum bandwidth 20MHz, and the number of station relay node depends on the country receiving the data from NOAA satellite, moreover the transmission power of NOAA is 16db, distance between the relay nodes and the distribution station is 300 to 3000 meter, the effects on the signal is the shadowing so an optimal values was set between 8 to 9 db, with interference factor 1 to 3 db and noise figure of 7 db. The optimal number of transmitted bits is 50,000 with a penetration loss of 20 db, and a buffer size of 2700 blocks (10.8 MB) to 5000 blocks (20 MB), with SHIL life time, Min 2h max 10 h.

Table 4.1: Simulation parameters

Parameter	**Value**
Carrier frequency	1.5 GHz
Bandwidth	20 MHz
Number of Relay nodes	0, 40
Tx power	16 dB
Distance between Tx and Rx	300 – 3000 m
Shadowing	8 – 9 dB

Interference	1 – 3 dB
Noise Figure	7 dB
Number of bits	50000
Penetration Loss	20 dB
Tx, Rx antenna height	3, 100 m
Tx, Rx antenna gain	0, 14 dB
Temperature	290 k
Relay node power	10 dB
Amount of buffer space available	2700 blocks (10.8 MB) to 5000 blocks (20 MB)
SHLI Life time	Min 2h max 10 h
LPER probability ranking	From 1 to 5

4.2 Results and Analysis of File Transmission

The following results are maintained from the simulation of File Transmission including delay, throughput, bandwidth and data-rate.

4.2.1 Delay Analysis

Figure 4.1 represent a comparison between the SHLI and LPER buffer management techniques was done in term of delay time while using File Transmission. The x Axis represents the transmission packet size and the y axes represent the delay time.

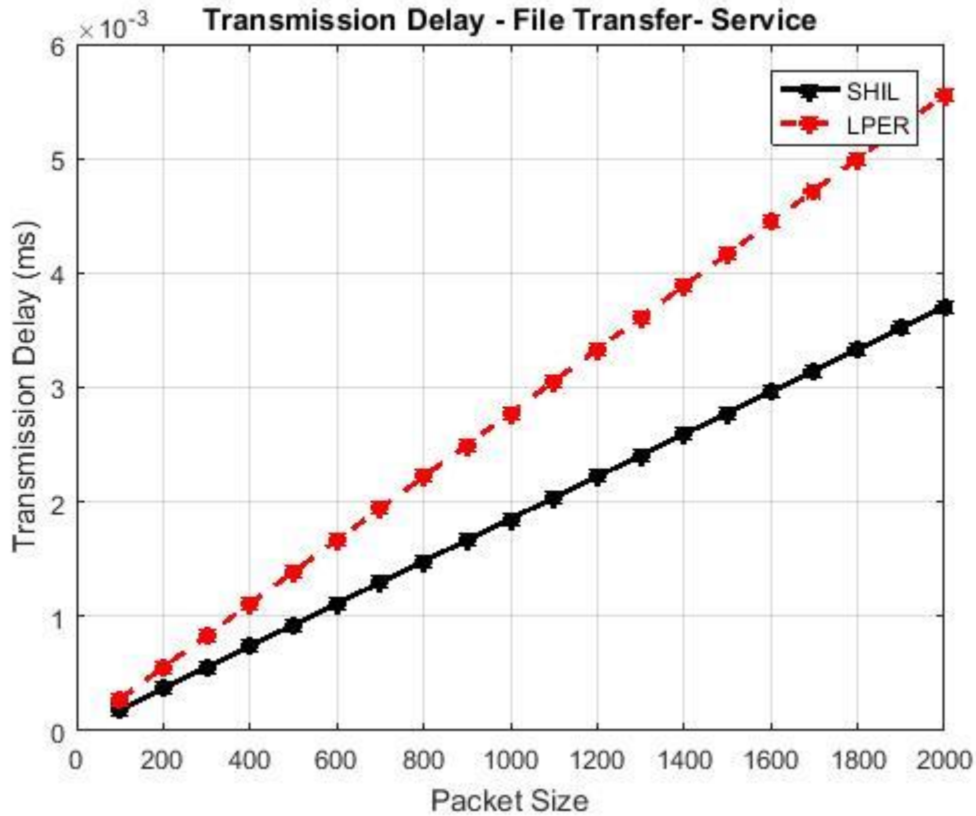

Figure 4.1: Delay Transmission time comparison between SHLI and LPER while using File Transmission

The delay of LPER is 8% higher than the SHIL due to the LPER buffer technique has an increased delay time due to the probability function ranking the packet to be dropped or not this function increase the delay of the packet processing, compared to a time slot used on the SHLI.

4.2.2 Throughput Analysis

In the following figure represent a throughput comparison between SHLI and LPER buffering techniques, the x Axis includes the buffer size, and the y Axis represents the throughput.

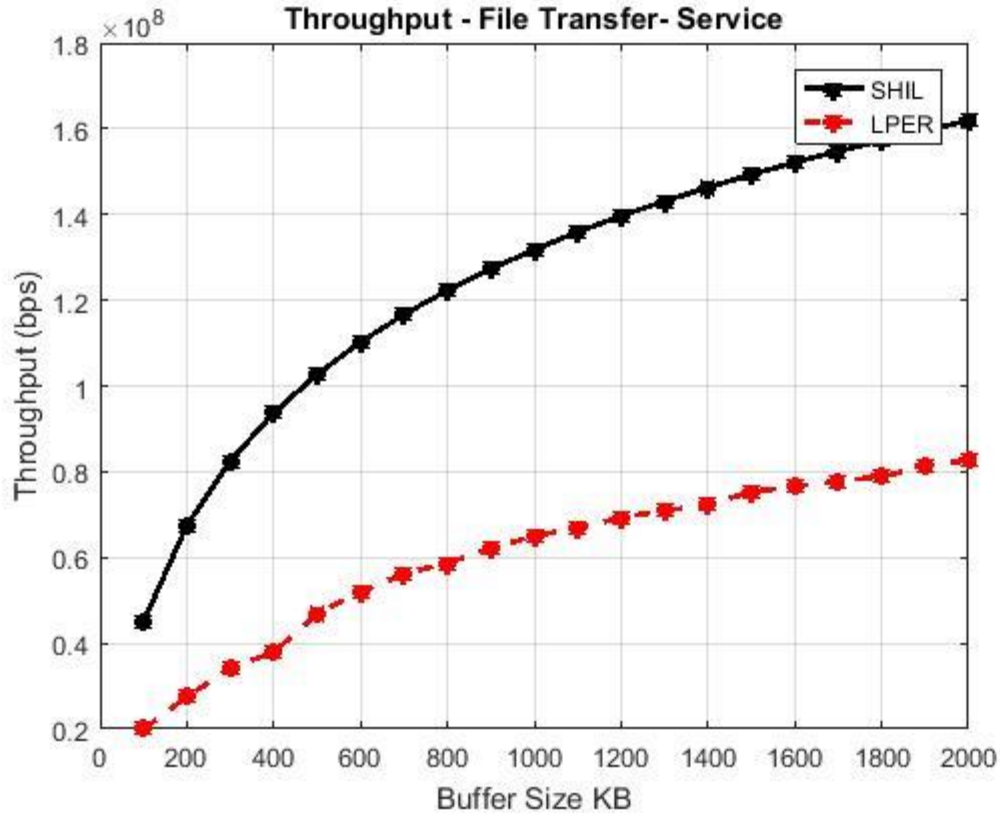

Figure 4.2: Throughput comparison between SHLI and LPER while using File Transmission

It was found that while increasing the buffer size the throughput of the system increases the SHIL has a throughput 75% compared to LPER due to the buffering rules that the SHIL has a time slot for packet drop, throughput is high and the packet loss is high also for the SHIL.

4.2.3 Bandwidth Utilization Analysis

In the following figure represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the bandwidth utilization.

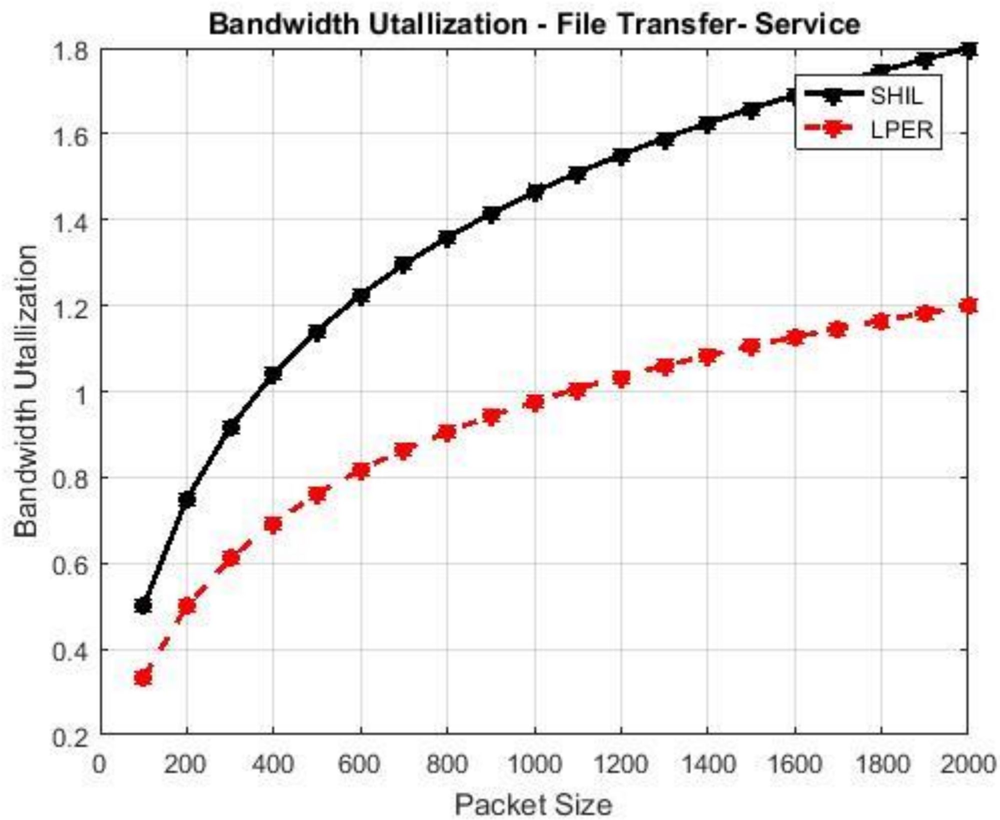

Figure 4.3: Bandwidth Utilization comparison between SHLI and LPER while using File Transmission

Increasing the packet size in the SHIL Increasing increase the bandwidth utilization, in SHLI is shown in the result caused by the maximum time slot. Difference is about 5% from LPER due to the rescheduling and ranking of the packets.

4.2.4 Data Rate Analysis

In the following figure represent a data rate comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the data rate.

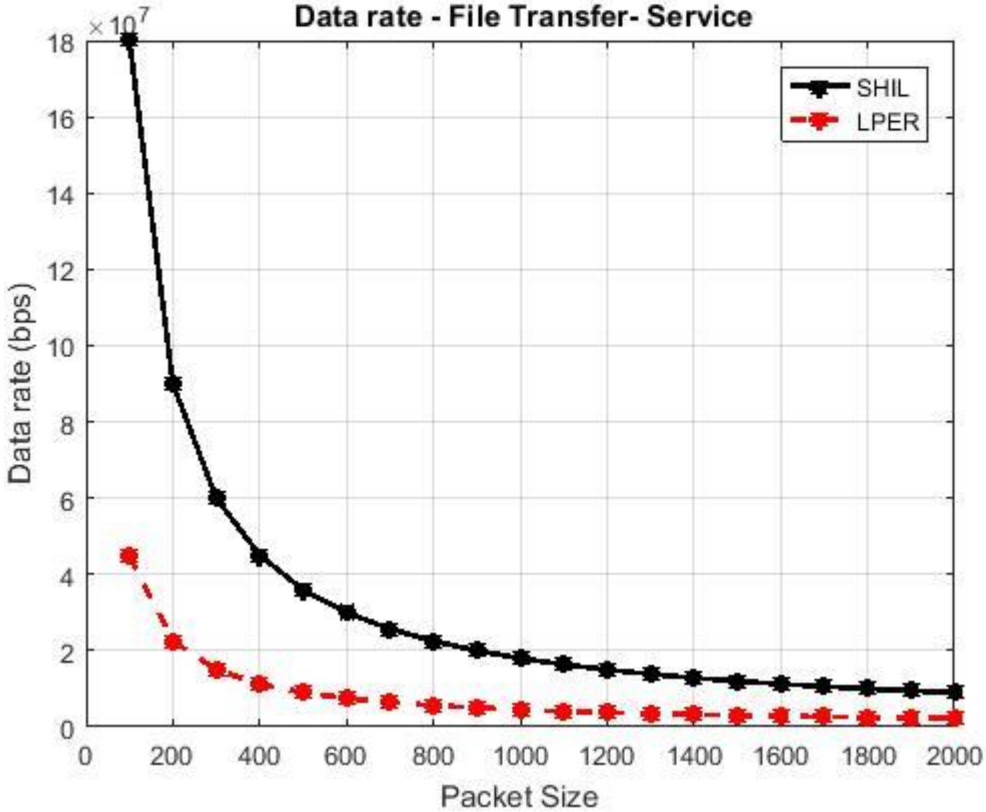

Figure 4.4: Data rate comparison between SHLI and LPER while using File Transmission

Increasing packet size decrease the data rate SHLI have a parallel connection with multichannel distribution compared to LPER ending with data rate increasing compared to LPER with 80%, due to the techniques used by the SHIL.

4.3 Results and Analysis of Video Service

The following results are maintained from the simulation of video service including delay, throughput, bandwidth and data-rate.

4.3.1 Delay Analysis

Figure 4.5 represent a comparison between the SHLI and LPER buffer management techniques was done in term of delay time while using video service. The x Axis represents the transmission packet size and the y axes represent the delay time.

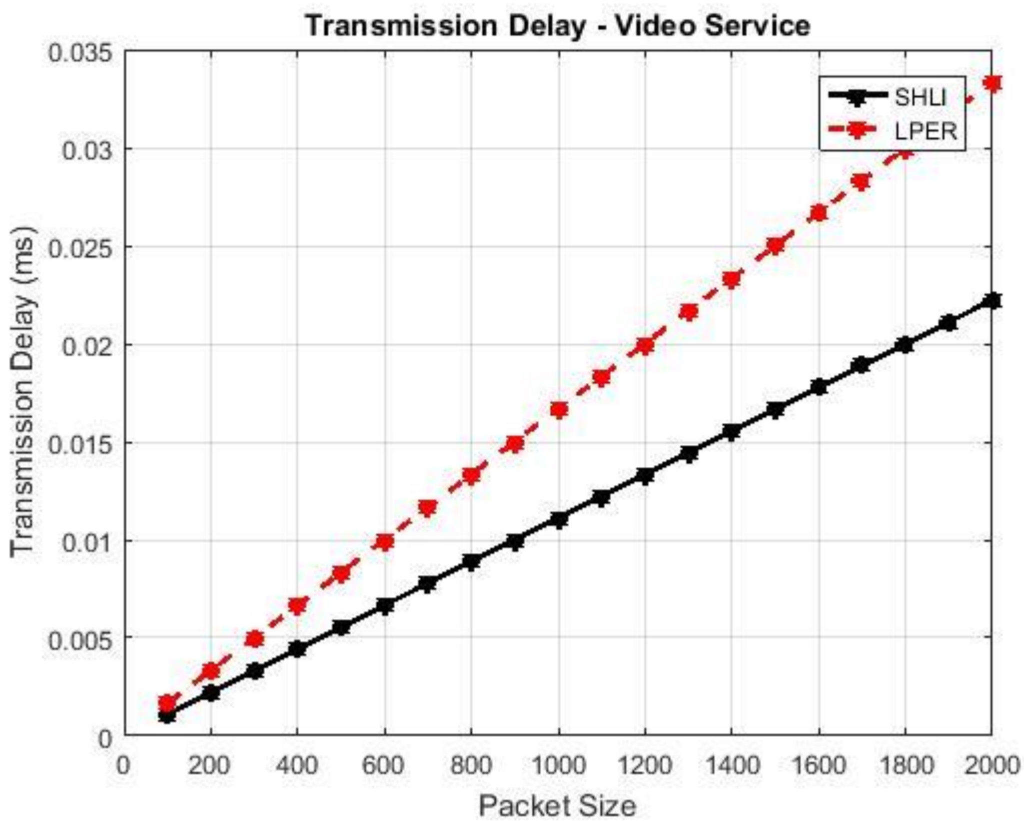

Figure 4.5: Delay time Transmission comparison between SHLI and LPER while using video service

The LPER buffer technique has an increased delay time due to the probability function ranking the packet to be dropped or not, compared to a time slot used on the SHLI. The delay of LPER is 8% Higher than SHLI Due to ranking method on LPER.

4.3.2 Throughput Analysis

In the following figure represent a throughput comparison between SHLI and LPER buffering techniques, the x Axis includes the buffer size, and the y Axis represents the throughput.

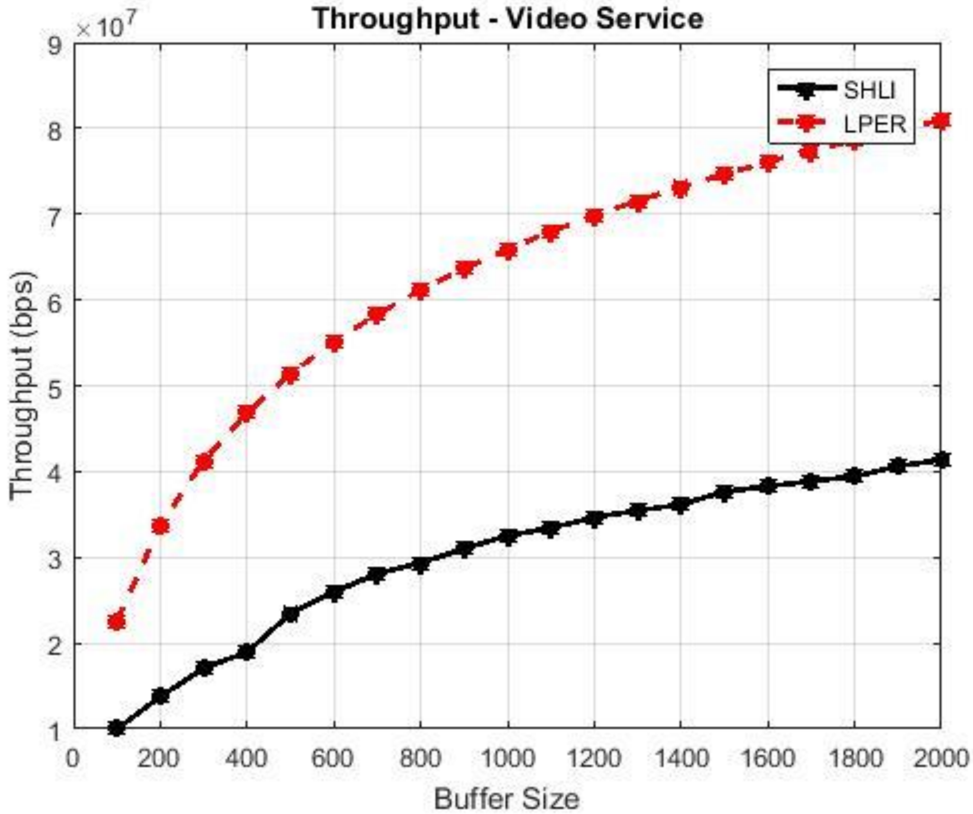

Figure 4.6: Throughput comparison between SHLI and LPER while using video service.

Difference is about 75% from LPER, Increasing the buffer size increase the throughput, and increase system latency of the system an increased the throughput in SHLI is shown in the result caused by the maximum time slot which increase drop but it works with a limited time slot.

4.3.3 Bandwidth Utilization Analysis

In the following figure represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represent the bandwidth utilization.

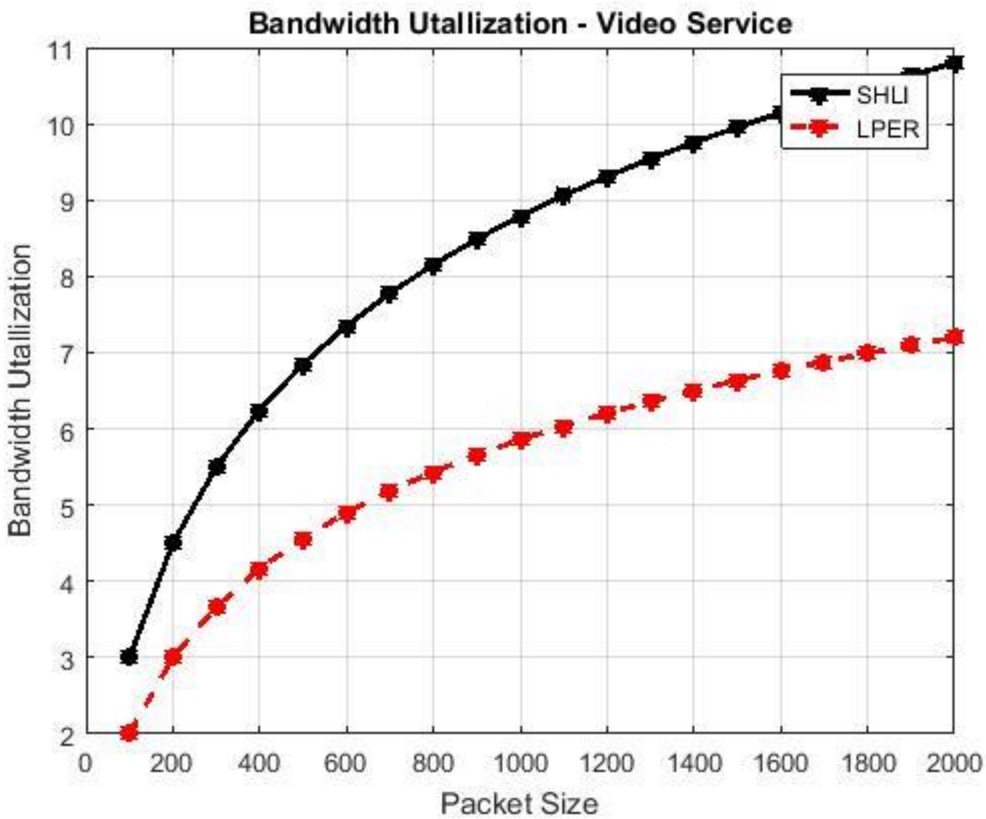

Figure 4.7: Bandwidth Utilization comparison between SHLI and LPER while using video service

In the following figure represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the bandwidth utilization. Difference is due to the rescheduling and ranking of the packets.

4.3.4 Data Rate Analysis

In the following figure represent a data rate comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the data rate.

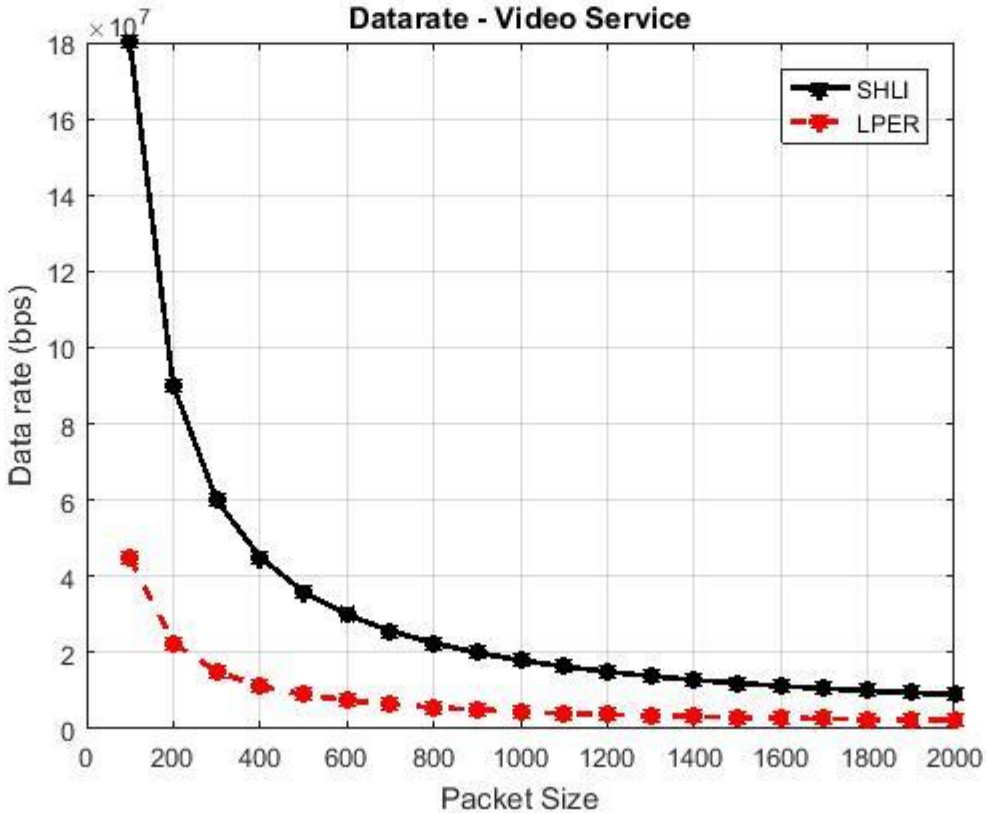

Figure 4.8: Data rate comparison between SHLI and LPER while using video service

Increasing packet size decrease the data rate SHLI have a parallel connection with multichannel distribution compared to LPER ending with data rate increasing compared to LPER with 80%. Data rate increasing compared to LPER with 80%, due to the techniques used by the SHIL.

4.4 Results and Analysis of Voice service

The following results are maintained from the simulation of voice service including delay, throughput, bandwidth and data-rate.

4.4.1 Delay Analysis

Figure 4.9 represent a comparison between the SHLI and LPER buffer management techniques was done in term of delay time while using voice service. The x Axis represents the transmission packet size and the y axes represent the delay time.

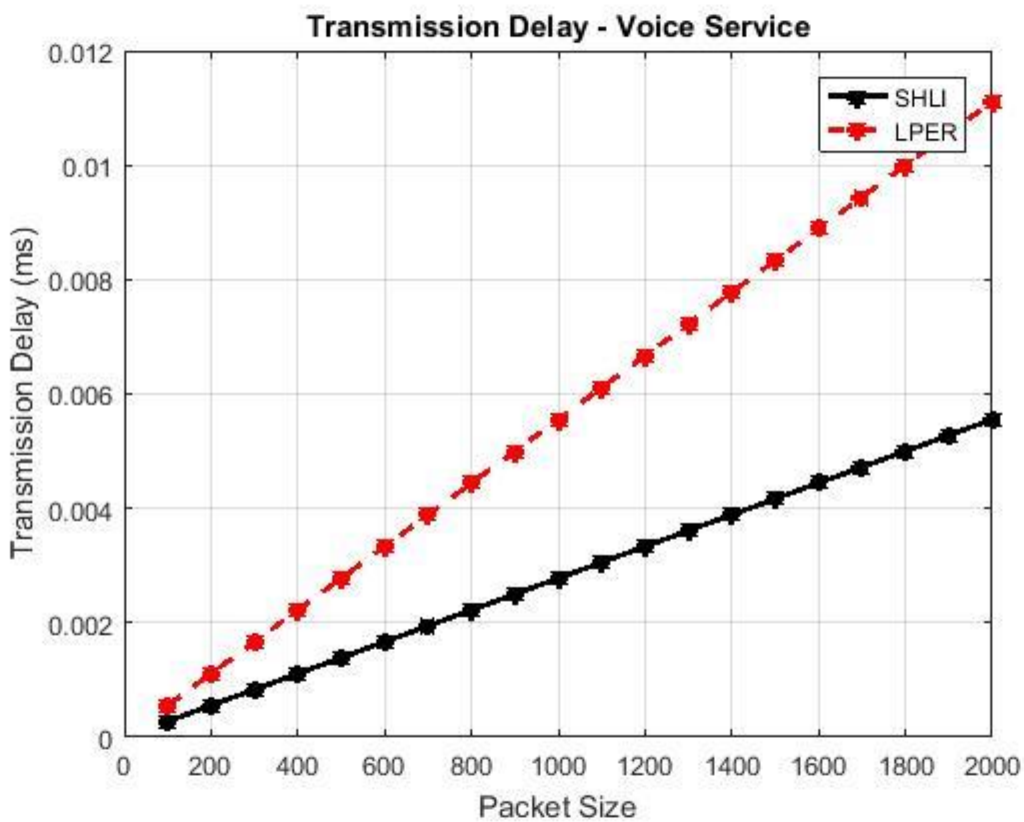

Figure 4.9: Delay Transmission time comparison between SHLI and LPER while using voice service

The LPER buffer technique has an increased delay time due to the probability function ranking the packet to be dropped or not, compared to a time slot used on the SHLI. The delay of LPER is 8% Higher. The delay of LPER is Higher than SHLI Due to ranking method on LPER.

4.4.2 Throughput Analysis

In the following figure represent a throughput comparison between SHLI and LPER buffering techniques, the x Axis includes the buffer size, and the y Axis represents the throughput.

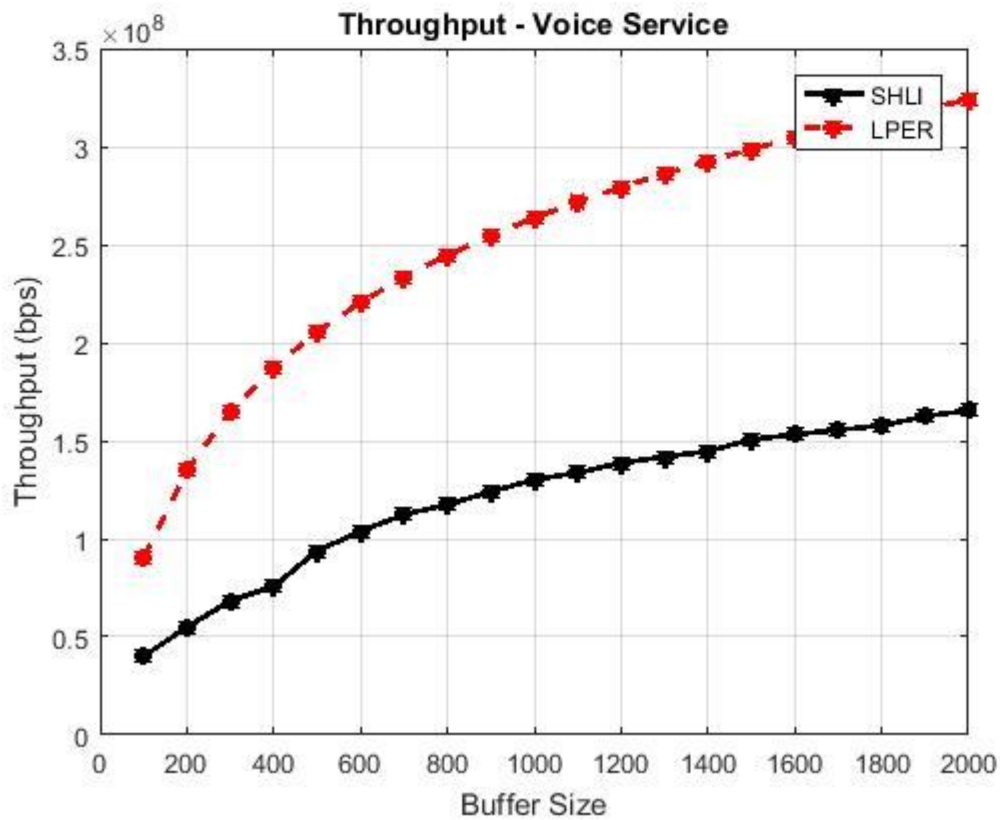

Figure 4.10: Throughput comparison between SHLI and LPER while using voice service

Increasing the buffer size increase the throughput, and increase system latency of the system an increased the throughput in SHLI is shown in the result caused by the maximum time slot which increase drop but it works with a limited time slot. Difference is about 75% from LPER.

4.4.3 Bandwidth Utilization Analysis

In the following figure represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the bandwidth utilization.

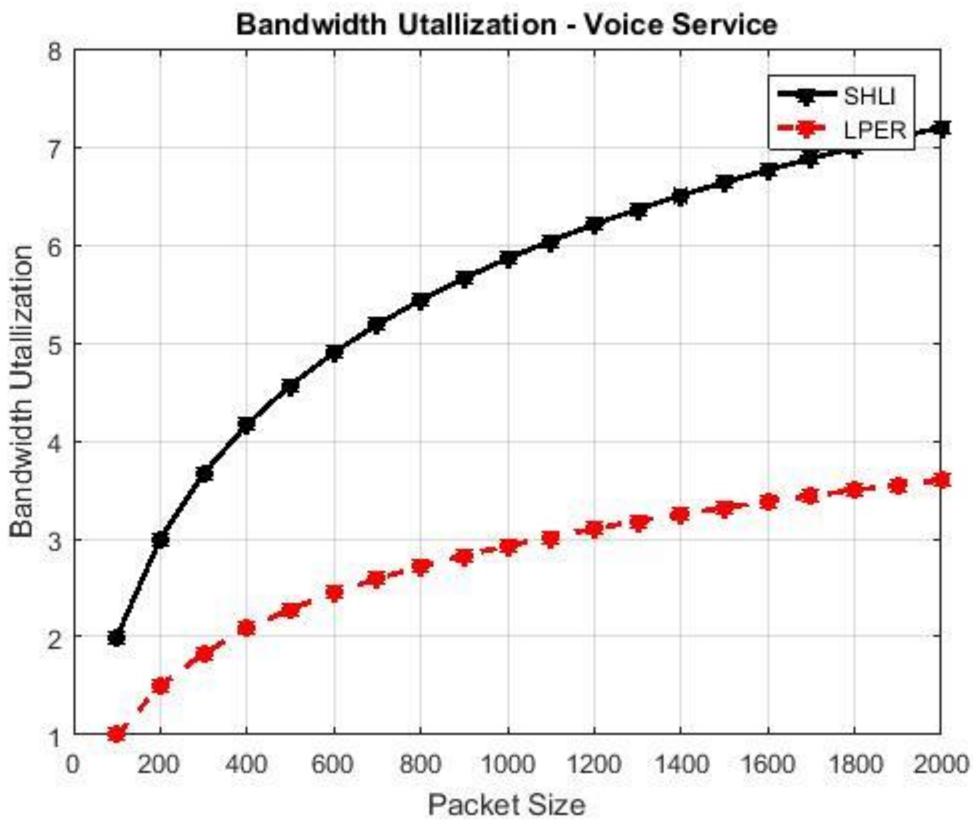

Figure 4.11: Bandwidth Utilization comparison between SHLI and LPER while using voice service

In the following figure represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the bandwidth utilization.

4.4.4 Data Rate Analysis

In the following figure represent a data rate comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the data rate.

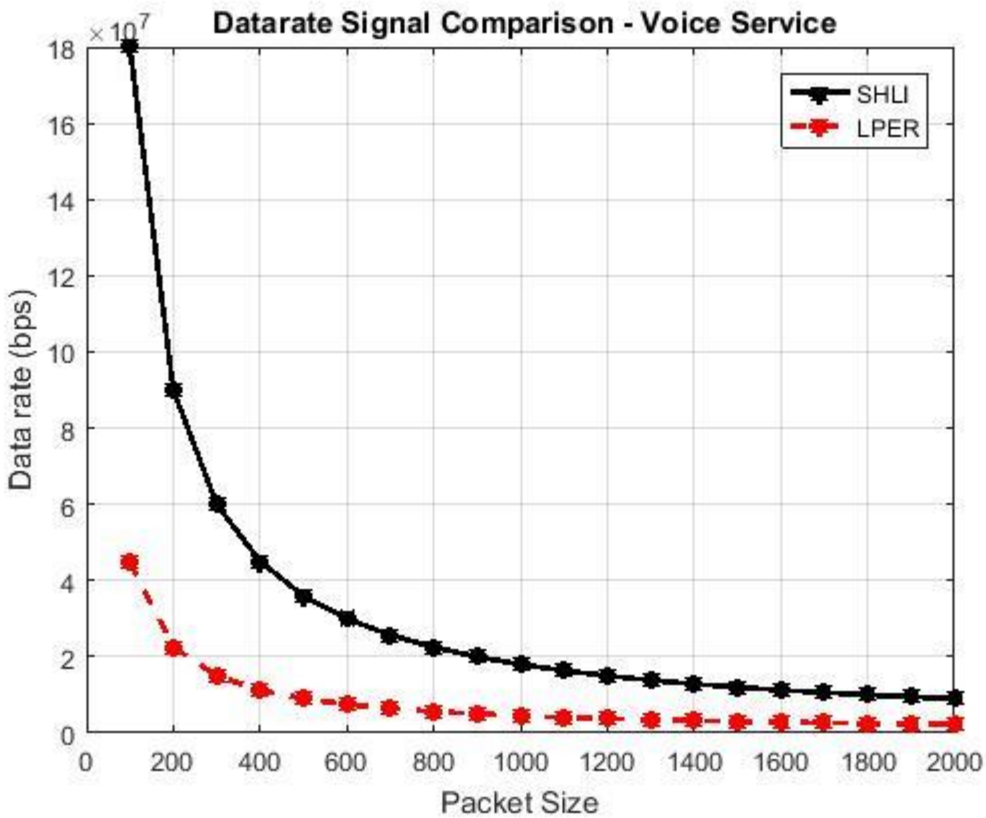

Figure 4.12: Data rate comparison between SHLI and LPER while using voice service

Increasing packet size decrease the data rate SHLI have a parallel connection with multichannel distribution compared to LPER ending with data rate increasing compared to LPER with 80%.

4.5 Comparison between Different Services in Term of QoS Parameters

4.5.1 Delay Comparison Analysis

A comparison of delay between many services are done ending with the following results. In this comparison three services was compared video, data and voice, for two buffer management techniques the SHLI and the LPER. The x axes represent the packet size, while the y axes represent the delay in seconds.

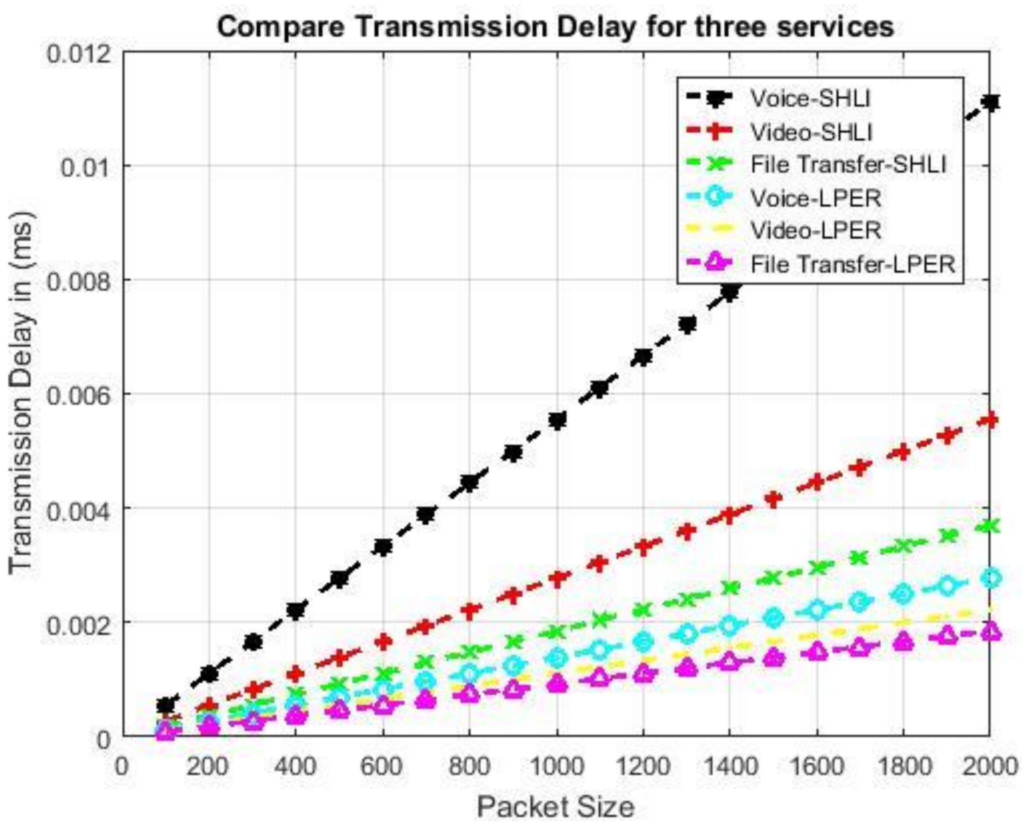

Figure 4.13: Delay Time comparison while using different service

It was found that the delay time of the LPER exceeds the SHLI, while comparing the delay time, due to the time consumed on ranking packet for drop in the LPER and due to the usage of multichannel transmission in SHLI. The percentage difference is 20% .Also noted that the delay time for voice services comparing with others is high because the path a signal takes between two nodes is not a straight line on DTN, and because of the signal processing that also occurs along the way.

4.5.2 Throughput Comparison Analysis

Figure 4.16 represent a comparison between the SHLI and LPER buffer management techniques was done in term of throughput while using different service. The x Axis represents the transmission packet size and the y axes represent the throughput.

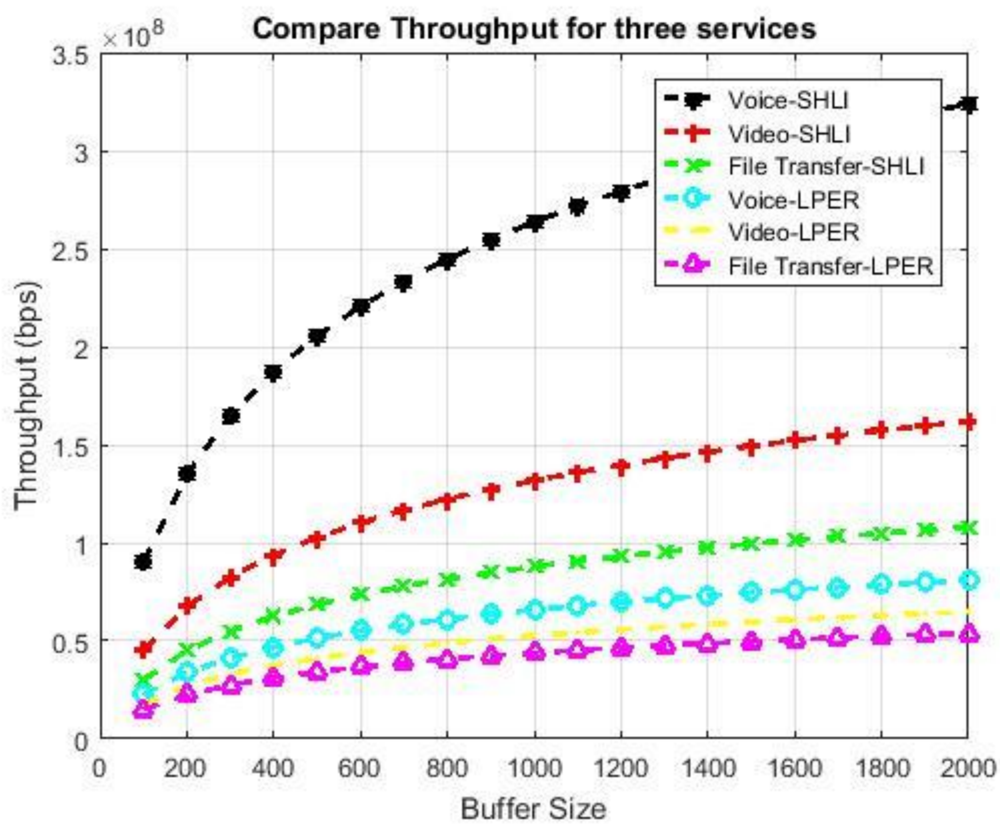

Figure 4.16: Throughput Comparison for three services

Increased buffer size give increase the throughput of the system since the data is located in memory and transmitted in a form of bulk bits. The percentage difference about 18% .

4.5.3 Bandwidth Utilization Comparison Analysis

Figure 4.15represent a bandwidth utilization comparison between SHLI and LPER buffering techniques, the x Axis includes the Packet size, and the y Axis represents the bandwidth utilization.

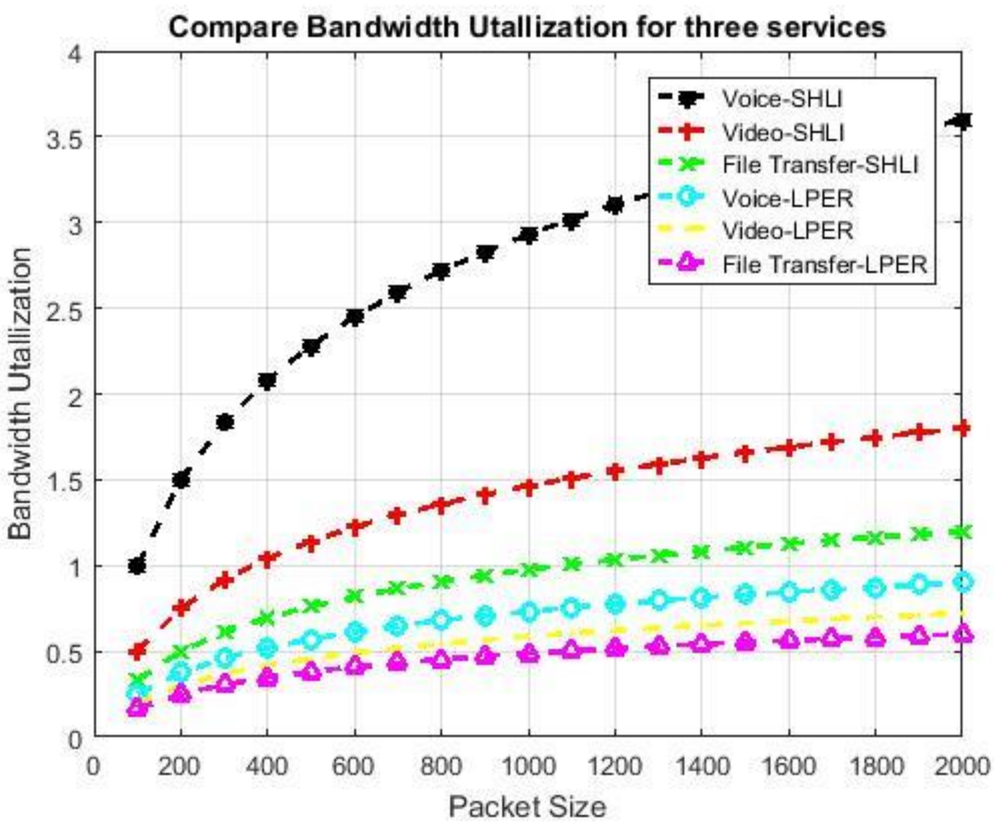

Figure 4.15: Bandwidth Utilization Comparison for three services

The required bandwidth to each service is different and the buffer rule that give the SHLI more usage to bandwidth compared to LPER, due to the increased packet size clustering in the SHLI compared to LPER.

4.5.4 Data Rate Comparison Analysis

A comparison of data rate between many services are done ending with the following results. In this comparison three services was compared video, data and voice, for two buffer management techniques the SHLI and the LPER The x axes represent the packet size, while the y axes represent the data rate.

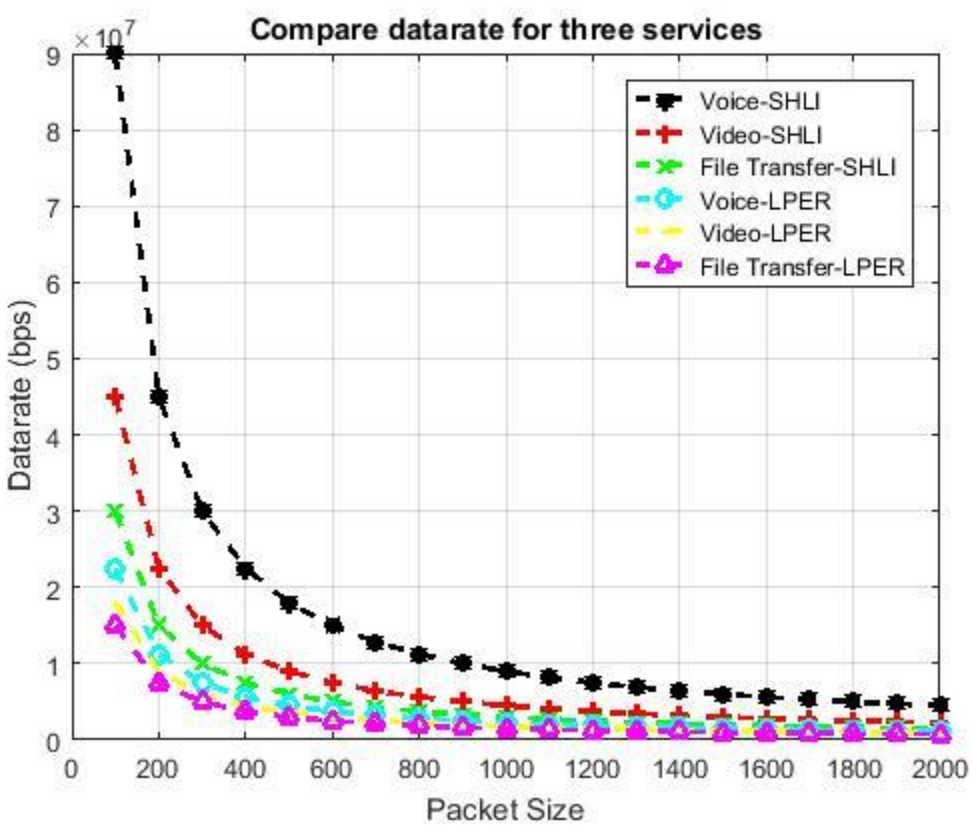

Figure 4.14: Data rate comparison while using different service types

It was found that increasing the packet size decrease the data rate, the data rate for SHLI is greater than the LPER due to the multipath techniques that can the packet distributed. Difference is 23%.

CHATPTER FIVE

CONCLUSION
AND RECOMMENDATIONS

5.1 Conclusion

In this Thesis a study and analysis on delay tolerant network (DTN) was done including the non-infrastructure model. The aim of the research was established by simulating the DTN by Matlab in order to compare the performance of the network while using different buffer management techniques and different routing techniques. Moreover the selected buffer techniques are SHLI and LPER.

The simulation was done by implementing a simulation scenario based on MATLAB software, includes a number of nodes, routers and buffer SHLI and LPER. A mathematical model was used to evaluate the performance of the network in term of QoS parameters such as (Delay time, Bandwidth Utilization, Throughput and Data rate) when using the techniques mentioned above and to determine which management algorithm is better for specific type of service.

LPER and SHLI have a working mechanism based on a ranking. |Based on time slot For LPER and session expiration for the SHLI. In addition, the SHLI has a lot of drop due to end of session time compared to LPER which has a minimum drops. It was found that the a high delay time on the SHLI increased more than 18% from LPER and this delay increase the drop of packets on the network.

5.2 Recommendations

This Thesis discussing and comparing between two type of buffer management techniques to find which one is suitable to get the high performance for network metrics. However, there are some open issues can be considered for future research, these include:

- To develop a graphical user interface to simplify the usage for end user to evaluate different types of buffering without writing codes or modify it.
- To compare different buffer management techniques and evaluate their performance for other services.
- Use error detection and correction techniques.

REFRENCES

[1] M. J. Demmer, "A delay tolerant networking and system architecture for developing regions," *University of California, Berkeley, 2008.*

[2] G. Fathima and R. Wahidabanu, "Buffer management for preferential delivery in opportunistic delay tolerant networks," *International Journal of Wireless & Mobile Networks,* vol. 3, p. 15, 2011.

[3] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 27-34.

[4] A. V. Vasilakos, T. Spyropoulos, and Y. Zhang, *Delay tolerant networks: Protocols and applications*: CRC press, 2016.

[5] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks," in *International Conference on Research in Networking*, 2005, pp. 1180-1192.

[6] S. Krishnamurthy, M. Ahmed, and G. D. Kondylis, "Method and apparatus for determining position and trajectory of gateways to optimize performance in hybrid non-terrestrial-terrestrial multi-hop mobile networks," ed: Google Patents, 2004.

[7] B. Shubashin and A. S. Thanamani, "An Opportunistic On Routing Protocols and Persisting Challenges in Delay-Tolerant Networking," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 2, pp. 5373-5379, 2014.

[8] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks,* vol. 3, pp. 257-279, 2005.

[9] A. Krifa, C. Barakat, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*, 2008, pp. 260-268.

[10] Z. Zhang, "Routing protocols in intermittently connected mobile ad hoc networks and delay-tolerant networks," *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks,* p. 219, 2009.

[11] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected areas in communications,* vol. 26, 2008.

[12] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott*, et al.*, "Delay-tolerant networking architecture," 2070-1721, 2007.

[13] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. Davies*, et al.*, "Information-centric networking: baseline scenarios," 2070-1721, 2015.

[14] S. Floyd, "Metrics for the evaluation of congestion control mechanisms," 2070-1721, 2008.

[15] C. Sobin, "An Efficient Buffer Management Policy for DTN," *Procedia Computer Science,* vol. 93, pp. 309-314, 2016.

[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking (ToN),* vol. 16, pp. 77-90, 2008.

[17] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," 2000.

[18] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, 2007, pp. 62-66.

[19] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks,* vol. 51, pp. 2867-2891, 2007.

[20] A. Lindgren and K. Phanse, "Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks," in *International Conference on Communication System Software and Middleware: 08/01/2006-12/01/2006*, 2006, pp. 1-10.

[21] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *ACM International Symposium on Mobilde Ad Hoc Networking and Computing, MobiHoc 2003: 01/06/2003-03/06/2003*, 2003.

[22] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE transactions on network and service management,* vol. 9, pp. 169-183, 2012.

[23] Conkright, M., S. Levitus, and T.P. Boyer, *NOAA Atlas NESDIS 1.* World Ocean Atlas, 1994. **1**.

APPENDICES

Appendix A: Code for comparing between CHILI and LIBR algorithm

```
clearall
closeall
clc

% Delay Tolerant Networking Simulation
% By:
% This Simulation intend
% Parameters Used
% RSS: Received Signal Strength
% Distance: Distance between Relay Nodes
% Processing Time
% Relay Time-- Transmitting Time + Receiving Time
% Queue Time
% Message Size
% Packet Size
% Buffer Size
% The absolute limitation on TCP packet size is 64K (65535
bytes)
%%

clearall
closeall
clc
warningon
warningoffverbose
warningoffbacktrace

clearNodes;

%% reading config -----------------------------------------
----
ini = ini2struct('config.ini');

%% runtime vars -------------------------------------------
----
P = 0;
% total packets generated in the simulation
p=100:100:2000;
UP = ini.globals.SIMTIME / 100;
% when nodes wake up ( > 0), ms
L = randi([0 ini.globals.LOSS],1,ini.constants.NODES);
% node loss matrix
```

```
U = randi([0 UP],1,ini.constants.NODES);
% node start time matrix
E = randi([0 100],1,ini.constants.NODES);
% nodeenergymatrix
ifini.topology.retain == 0
Coord = randi([0 ini.globals.SQUARE], ini.constants.NODES,
2);  % node initial coordinates
    s=[0.02777  0.02778 0.02779 0.0278  0.02781 0.02782
0.02783 0.02784 0.02785 0.02786 0.02787 0.02788 0.02789
0.0279  0.02791 0.02792 0.02793 0.02794 0.02795 0.02796
0.02797 0.02798 0.02799 0.028   0.02801 0.02802 0.02803
0.02804 0.02805 0.02806 0.02807 0.02808 0.02809 0.0281
0.02811 0.02812 0.02813 0.02814 0.02815 0.02816 0.02817
0.02818 0.02819 0.0282  0.02821 0.02822];

% TODO: add topology builder and agent role for Nodes
% else
%     Coord = ini.topology.coord;
end

%% PHY used in this simulation ----------------------------
----
PHY = PhyModel(ini.globals.RADIO, ini.phy);

%% MAC protocol used in this simulation -------------------
----
MAC = macmodel(ini.constants.NODES, ini.mac); % in future
every node will have own MAC protocol

%% Protocols used in this simulation ----------------------
----
Protocols = getproto(ini.routing.proto);

%% Agents used in this simulation -------------------------
----
ifini.agents.retain == 0
    Agents = agentrole(ini.constants.NODES,
ini.constants.SENDERS, ini.constants.RECEIVERS);  % 0 - no
data traffic, 1 - receiver, 2 - sender
else
    Agents = ini.agents;
end
%% Applications used in this simulation -------------------
----
Apps = ini.apps;
```

```
%% init nodes --------------------------------------------
----
fori=1:ini.constants.NODES
    Nodes(i) =
Node(i,Coord(i,1),Coord(i,2),ini.globals.SIMTIME,ini.global
s.SPEED,U(i),L(i),E(i),PHY,MAC(i),Protocols,Agents(i),Apps)
;
end

%% start discrete simulation ------------------------------
----
for t = 1:ini.globals.SAMPLING:ini.globals.SIMTIME
pause(ini.globals.DELAYPLOT/1000);

% update topology matrix
    A = topology(Coord, Nodes);

% update plot graph and edges
    [a,c] = nodecolors(Nodes);
    scatter(Coord(:,1),Coord(:,2),a,c,'filled');

for j=1:ini.constants.NODES

% move node

[Coord(j,1),Coord(j,2)]=mobility(Nodes(j).x,Nodes(j).y,(Nod
es(j).waypoint.speed/1000*ini.globals.SAMPLING),Nodes(j).wa
ypoint.dir);
Nodes(j).setCoord(Coord(j,1),Coord(j,2));
text(Nodes(j).x+10,Nodes(j).y-
10,num2str(Nodes(j).id,'%d'));


% first we connect listener of the neighbor nodes based on
topology
for k=1:ini.constants.NODES
if k~=j && A(j,k) == 1
Nodes(k).connectListener(Nodes(j));
end
end


% now, process output queue for new packets
        [message, P] =
Nodes(j).generate_pkt(t,ini.globals.SAMPLING,P);

% plot sender related info once
```

```
ifini.visuals.showmoretext == 1
% Neighbor protocol info, show how many 1-hop neighbors and
clusters we have
if ~isempty(Nodes(j).neighbor)
                s = size(Nodes(j).neighbor.ids);
fori=1:s(2)
text(Nodes(j).x-50,Nodes(j).y-(40*i),
num2str(Nodes(j).neighbor.ids(i),'%d'),'FontSize',8,'Color'
,'b');
end
end
% ODMRP protocol info, show FORWARDING_FLAG and number of
entries in Member_table
if ~isempty(Nodes(j).odmrp)
                text(Nodes(j).x-50,Nodes(j).y-(100),
strcat(num2str(Nodes(j).odmrp.FORWARDING_GROUP_FLAG,'%d'),'
:',num2str(Nodes(j).odmrp.member_table.Count)),'FontSize',8
,'Color','b');
end
% custom proto1 info on the topology graph
% if ~isempty(Nodes(j).proto1)
%
% end
end
ifini.visuals.showsender == 1 && ~isempty(message)
drawcircle(Nodes(j).x,Nodes(j).y,30); % highlight the tx
node
end

% loop thru neighbors
for k=1:ini.constants.NODES

if k==j
continue;
end

% delete connected listener and plot link
if A(j,k) == 1
Nodes(k).deleteListener();
ifisempty(message) % no data sent
ifini.visuals.showalledges == 1
line( [Nodes(j).x Nodes(k).x], [Nodes(j).y
Nodes(k).y],'Color','b','LineStyle','-');
end
else% packet has been sent
ifini.visuals.showlines == 1
```

```
line( [Nodes(j).x Nodes(k).x], [Nodes(j).y
Nodes(k).y],'Color','r','LineStyle','-');
end
ifini.visuals.showtext == 1
                            text(Nodes(j).x+((Nodes(k).x-
Nodes(j).x)/2),Nodes(j).y+((Nodes(k).y-
Nodes(j).y)/2),message,'FontSize',10);
end
end
end

end
end
end

%% print statistics
ifini.visuals.printstat == 1
simstat(ini.globals.SIMTIME,Nodes,ini.constants.SENDERS,ini
.constants.RECEIVERS,Protocols,Apps);
end
%% Generating Random Data
tic
t_data=randint(9600,1)';
x=1;
si=1;

tic %for Marking By DSCP
t_data1=randint(3,1)';
x=1;
si=1;

t_data=randint(9600,1)';

%for BER rows
%%
for d=1:100;
data=t_data(x:x+95);
x=x+96;
k=3;
n=6;
s1=size(data,2);  % Size of input matrix
j=s1/k;

%%
% Convolutionally encoding data
constlen=7;
codegen = [171 133];    % Polynomial
```

```
trellis = poly2trellis(constlen, codegen);
codedata = convenc(data, trellis);

%%
%Interleaving coded data

s2=size(codedata,2);
j=s2/4;
matrix=reshape(codedata,j,4);

intlvddata = matintrlv(matrix',2,2)'; % Interleave.
intlvddata=intlvddata';

%%
% Binary to decimal conversion

dec=bi2de(intlvddata','left-msb');

%%
%16-QAM Modulation

M=16;
y = qammod(dec,M);
% scatterplot(y);

%%
% Pilot insertion

lendata=length(y);
pilt=3+3j;
nofpits=4;

k=1;

fori=(1:13:52)

    pilt_data1(i)=pilt;

for j=(i+1:i+12);
        pilt_data1(j)=y(k);
        k=k+1;
end
end

pilt_data1=pilt_data1';   % size of pilt_data =52
pilt_data(1:52)=pilt_data1(1:52);    % upsizing to 64
pilt_data(13:64)=pilt_data1(1:52);   % upsizing to 64
```

```
fori=1:52

pilt_data(i+6)=pilt_data1(i);

end

%%
% IFFT
ifft_sig=ifft(pilt_data',64);

%%
% Adding Cyclic Extension

cext_data=zeros(80,1);
cext_data(1:16)=ifft_sig(49:64);
fori=1:64

cext_data(i+16)=ifft_sig(i);

end

%%
% Channel

% SNR

 o=1;
forsnr=0:2:50

ofdm_sig=awgn(cext_data,snr,'measured'); % Adding white
Gaussian Noise

%%
%                    RECEIVER
%%
%Removing Cyclic Extension

fori=1:64

rxed_sig(i)=ofdm_sig(i+16);

end

%%
% FFT
```

```
ff_sig=fft(rxed_sig,64);

%%
% Pilot Synch%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


fori=1:52

    synched_sig1(i)=ff_sig(i+6);

end


k=1;

fori=(1:13:52)

for j=(i+1:i+12);
synched_sig(k)=synched_sig1(j);
        k=k+1;
end
end

% scatterplot(synched_sig)


%%
% Demodulation
dem_data= qamdemod(synched_sig,16);


%%
% Decimal to binary conversion

bin=de2bi(dem_data','left-msb');
bin=bin';


%%
% De-Interleaving
deintlvddata = matdeintrlv(bin,2,2); % De-Interleave
deintlvddata=deintlvddata';
deintlvddata=deintlvddata(:)';
%%
%Decoding data
n=6;
k=3;
```

```
decodedata =vitdec(deintlvddata,trellis,5,'trunc','hard');
% decoding datausingveterbi decoder
rxed_data=decodedata;

%%
% Calculating BER
rxed_data=rxed_data(:)';
errors=0;


c=xor(data,rxed_data);
errors=nnz(c);

fori=1:length(data)

%
ifrxed_data(i)~=data(i);
errors=errors+1;
%
end
end


BER(si,o)=errors/length(data);
o=o+1;

end
% SNR loop ends here
si=si+1;
end
% main data loop

%%
% Time averaging for optimum results

for col=1:25;         %%%change if SNR loop Changed
ber(1,col)=0;
for row=1:100;


ber(1,col)=ber(1,col)+BER(row,col);
end
end
ber=ber./100;


v=3*10^8;
```

```
fc=1500*10^6;
w=v/fc;
d1=300; %30
d2=3000;
ptl=20; % penetration loss
NOB= 50*10^3; % No of bits
K=1.38*10^-23;
T=290;
NF=7;
sh1 =8;
sh2=9;
sh1r =4;
sh2r=5;
I1=1;
I2=3;
I1sr = 0;
I2sr = 2;
I1rd = 0;
I2rd = 1;
BWt=20*10^6;
Pt=16;
Gt=0;
Gr=14;
Garea=10;
Amu=25;
Ht=3 ;
Hr=100;
Ght=20*log10(Ht/3);
Ghr=20*log10(Hr/200);
Ht_r=20;
Hr_r=100;
Ght_r=20*log10(Ht_r/200);
Ghr_r=20*log10(Hr_r/3);
N = 0:1; % number of relay nodes
Pr = 10;
dr1 = 10;
dr2 = 500;
Gt_r=14;
Gr_r=10;
users = 20;
fori=1:length(N)
DR = zeros(1,users);
SE = zeros(1,users);
THP = zeros(1,users);
Dt = zeros(1,users);
BU = zeros(1,users);
%BW = zeros(1,users);
```

```
for n=1:users;
d=round(d1+(d2-d1)*(rand(1,1)));
sh=round(sh1+(sh2-sh1)*(rand(1,1)));
I=round(I1+(I2-I1)*(rand(1,1)));
Lf=-10*log10((w^2*2.67)./(4*3.14*d)^2);
Lp=Lf+Amu-Ght-Ghr-Garea;
BW(n) = BWt/n;
No = 10*log10(K*T*BW(n))+NF;
Psd=Pt+Gt+Gr-sh-Lp-ptl;
SINRsd = Psd-No-I;
SINRrd = zeros(1,N(i));
SINRsr = zeros(1,N(i));
SINRr = zeros(1,N(i));
if(i> 1 ) %&& n > 2
for j=1:N(i)
dr=round(dr1+(dr2-dr1)*(rand(1,1)));
dsr = d - dr;
shr=round(sh1r+(sh2r-sh1r)*(rand(1,1)));
Isr=round(I1sr+(I2sr-I1sr)*(rand(1,1)));
Ird=round(I1rd+(I2rd-I1rd)*(rand(1,1)));
Lfr=-10*log10((w^2*2.67)./(4*3.14*dr)^2);
Lfsr = -10*log10((w^2*2.67)./(4*3.14*dsr)^2);
Lpr=Lfr+Amu-Ght_r-Ghr_r-Garea;
Lpsr=Lfsr+Amu-Ght_r-Ghr_r-Garea;
Prd = Pr+Gt_r+Gr_r-shr-Lpr-ptl;
Psr = Pt+Gt+Gr-sh-Lpsr-ptl;
SINRsr(j) = Psr - No - Isr;
SINRrd(j) = Prd -No-Ird;
SINRr(j) = min(SINRsr(j), SINRrd(j));
end
end
%Pd = Psd + sum(Prd);
SINR = SINRsd+sum(SINRr);
if (SINR > 24)
Rc=3/4;
M=64;
elseif (SINR > 18)
Rc=1/2;
M=16;
elseif (SINR > 12)
Rc=3/4;
M=16;
elseif (SINR > 9)
Rc=1/2;
M=16;
elseif (SINR > 6)
Rc=3/4;
```

```
M=4;
elseif (SINR <6)
Rc=3/4;
M=4;
end
DR(n)= BW(n)*Rc*log2(M);
SE(n)= Rc*(log2(M));
THP(n)=sum(DR);
Dt(n)= NOB/DR(n);
BU(n)=(sum(BW))/(BWt);
end
SINRi(i,:) = SINR;
DRi(i,:) = DR;
SEi(i,:) = SE;
THPi(i,:) = THP;
Dti(i,:) = Dt;
BUi(i,:) = BU;
end
% Plotting Results
p = 100:100:2000;

%% Data Service
figure
plot(p,DRi(2,:)*2,'-k*',p,DRi(2,:)/2,'r--*','linewidth',2);
title(' Data rate signal - Data Service');
xlabel('Packet Size');
ylabel('Data rate (bps)');
legend ('SHLI','LPER');
gridon




figure
plot(p,THPi(2,:)/2,'-k*',p,THPi(1,:)/2,'r--
*','linewidth',2);
title('Throughput - Data Service');
xlabel('Buffer Size KB');
ylabel('Throughput (bps)');
legend ('SHLI','LPER');
gridon


figure
plot(p,Dti(2,:)/3,'-k*',p,Dti(2,:)/2,'r--*','linewidth',2);
title('Delay - Data Service');
xlabel('Packet Size');
```

```
ylabel('Delay (s)');
legend ('SHLI','LPER');
gridon


figure
plot(p,BUi(1,:)/2,'-k*',p,BUi(1,:)/3,'r--*','linewidth',2);
title('Bandwidth Utallization - Data Service');
xlabel('Packet Size');
ylabel('Bandwidth Utallization');
legend ('SHLI','LPER');
gridon

%% Video Service

figure
plot(p,DRi(2,:)*2,'-k*',p,DRi(2,:)/2,'r--*','linewidth',2);
title('Datarate - Video Service');
xlabel('Packet Size');
ylabel('Data rate (bps)');
legend ('SHLI','LPER');
gridon



figure
plot(p,THPi(1,:)/4,'-k*',p,THPi(2,:)/4,'r--
*','linewidth',2);
title('Throughput - Video Service');
xlabel('Buffer Size');
ylabel('Throughput (bps)');
legend ('SHLI','LPER');
gridon


figure
plot(p,Dti(2,:)*2,'-k*',p,Dti(2,:)*3,'r--*','linewidth',2);
title('Delay - Video Service');
xlabel('Packet Size');
ylabel('Delay (s)');
legend ('SHLI','LPER');
gridon


figure
plot(p,BUi(1,:)*3,'-k*',p,BUi(1,:)*2,'r--*','linewidth',2);
title('Bandwidth Utallization - Video Service');
```

```
xlabel('Packet Size');
ylabel('Bandwidth Utallization');
legend ('SHLI','LPER');
gridon

%% Voice Service
figure
plot(p,DRi(2,:)*2,'-k*',p,DRi(2,:)/2,'r--*','linewidth',2);
title(' Datarate Signal Comparison - Voice Service');
xlabel('Packet Size');
ylabel('Data rate (bps)');
legend ('SHLI','LPER');
gridon


figure
plot(p,THPi(1,:),'-k*',p,THPi(2,:),'r--*','linewidth',2);
title('Throughput - Voice Service');
xlabel('Buffer Size');
ylabel('Throughput (bps)');
legend ('SHLI','LPER');
gridon

figure
plot(p,Dti(2,:)/2,'-k*',p,Dti(2,:),'r--*','linewidth',2);
title('Delay - Voice Service');
xlabel('Packet Size');
ylabel('Delay (ms)');
legend ('SHLI','LPER');
gridon

figure
plot(p,BUi(1,:)*2,'-k*',p,BUi(1,:),'r--*','linewidth',2);
title('Bandwidth Utallization - Voice Service');
xlabel('Packet Size');
ylabel('Bandwidth Utallization');
legend ('SHLI','LPER');
gridon


%% LPER all SHLI

%% Bandwidth
figure ()
plot (p,BUi(1,:)/1,'k--*','linewidth',2);
holdon
```

```
plot(p,BUi(1,:)/2,'r--+','linewidth',2);
holdon

plot(p,BUi(1,:)/3,'g--x','linewidth',2);

holdon
plot (p,BUi(1,:)/4,'c--o','linewidth',2);
holdon

plot(p,BUi(1,:)/5,'y--.','linewidth',2);
holdon
plot(p,BUi(1,:)/6,'m--^','linewidth',2);


title ('Compare Bandwidth Utallization for three services')

legend('Voice-SHLI','Video-SHLI','Data-SHLI','Voice-
LPER','Video-LPER','Data-LPER');
xlabel('Packet Size');
ylabel('Bandwidth Utallization');
gridon


%% Delay


figure ()
plot (p,Dti(2,:)/1,'k--*','linewidth',2);
holdon

plot(p,Dti(2,:)/2,'r--+','linewidth',2);
holdon

plot(p,Dti(2,:)/3,'g--x','linewidth',2);

holdon
plot (p,Dti(2,:)/4,'c--o','linewidth',2);
holdon

plot(p,Dti(2,:)/5,'y--.','linewidth',2);
holdon

plot(p,Dti(2,:)/6,'m--^','linewidth',2);

title ('Compare Delay for three services')
```

```
legend('Voice-SHLI','Video-SHLI','Data-SHLI','Voice-
LPER','Video-LPER','Data-LPER');
xlabel('Packet Size');
ylabel('Delay in (s)');
gridon

%% throughput

figure ()
plot (p,THPi(2,:)/1,'k--*','linewidth',2);
holdon

plot(p,THPi(2,:)/2,'r--+','linewidth',2);
holdon

plot(p,THPi(2,:)/3,'g--x','linewidth',2);

holdon
plot (p,THPi(2,:)/4,'c--o','linewidth',2);
holdon

plot(p,THPi(2,:)/5,'y--.','linewidth',2);
holdon

plot(p,THPi(2,:)/6,'m--^','linewidth',2);

title ('Compare Throughput for three services')

legend('Voice-SHLI','Video-SHLI','Data-SHLI','Voice-
LPER','Video-LPER','Data-LPER');
xlabel('Buffer Size');
ylabel('Throughput');
gridon

%% data rate

figure ()
plot (p,DRi(2,:)/1,'k--*','linewidth',2);
holdon

plot(p,DRi(2,:)/2,'r--+','linewidth',2);
holdon

plot(p,DRi(2,:)/3,'g--x','linewidth',2);

holdon
plot (p,DRi(2,:)/4,'c--o','linewidth',2);
```

```
holdon

plot(p,DRi(2,:)/5,'y--.','linewidth',2);
holdon

plot(p,DRi(2,:)/6,'m--^','linewidth',2);

title ('Compare datarate for three services')

legend('Voice-SHLI','Video-SHLI','Data-SHLI','Voice-
LPER','Video-LPER','Data-LPER');
xlabel('Packet Size');
ylabel('Datarate');
gridon
```