PROVIDING CLUSTER-BASED QUALITY OF SERVICE IN MOBILE AD HOC NETWORKS

By

Khaled Mohammed Ahmed Hushaidan

Supervisor

Dr. Wesam Abdel Rahman Al Mobaideen

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science.

Faculty of Graduate Studies

The University of Jordan

May, 2007

COMMITEE DECISION

This Thesis (Providing Cluster-Based Quality of Service in Mobile Ad Hoc Networks) was successfully defended and approved on May 15, 2007

Examination Committee	**Signature**
Dr. Wesam A. Al Mobaideen, Chairman Assist. Prof. of Wireless Computer Networks	
Dr. Imad Khaled Salah, Member Assist. Prof. of Complex Systems and Networks	
Dr. Saleh Al-Sharaeh, Member Assoc. Prof of Computer Networks	
Dr. Emad Qaddoura, Member Assist. Prof of Wireless Computer Networks (Applied Science University)	Emad

تعتمد كلية الدراسات العليا
هذه النسخة من الرسالة
التوقيع ...... التاريخ ٢٤/٥/٢٠٠٧

DEDICATION

I dedicate this work to my father (*may Allah bless his soul*), to my beloved mother, and to all my family.

ACKNOWLEDGMENT

I sincerely thank my supervisor, Dr. Wesam Al Mobaideen, for his encouragement, guidance, and advice. I would also like to thank all my professors for everything I’ve learnt from them. Special thanks to the discussion committee for their insightful comments.

I am very grateful to the Yemeni Ministry of Higher Education and Scientific Research, for financially supporting me throughout my study.

Finally, I would like to thank my family, friends, colleagues, and everyone who encouraged me and made this work possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgment
ACQR	Adaptive Cluster-based Routing with QoS support
AIFS	Arbitrated IFS
AP	Access Point
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
BE	Best Effort
CDMA	Code Division Multiple Access
CEDAR	Core-Extraction Distributed Ad-hoc Routing
CH	Clusterhead
CBR	Constant Bit Rate
CD	Collision Detection
CFP	Contention Free Period
CP	Contention Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
CW	Contention Window
DARPA	US Defense Advanced Research Project Agency
DCF	Distributed Coordination Function
DG	Distributed Gateway
DiffServ	Differentiated Services
DIFS	DCF IFS
DSCP	Differentiated Services Code Point
DSSS	Direct Sequence Spread Spectrum
ECN	Explicit Congestion Notification
EDCA	Enhanced Distributed Channel Access
EDCF	Enhanced DCF
GLOMOSIM	GLobal MObile information system SIMulator
GSM	Global System for Mobile communications

GW	Gateway

IEEE	Institute of Electrical and Electronics Engineers
FHSS	Frequency Hopping Spread Spectrum
FIFO	First-In First-Out
FQMM	Flexible QoS Model for Mobile ad hoc networks
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HQMM	Hybrid QoS Model for Mobile ad hoc networks
IA	IntrA-cluster
IE	IntEr-cluster
IETF	Internet Engineering Task Force
IFS	Inter-Frame Spacing
IntServ	Integrated Services
IR	InfraRed
ISDN	Integrated Services Digital Network
LAN	Local Area Network
LCA	Link-Cluster Architecture
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork(s)
MPLS	Multi-Protocol Label Switching
NAV	Network Allocation Vector
NTDR	Near-Term Digital Radio
OFDM	Orthogonal Frequency Division Multiplexing
PARSEC	PARallel Simulation Environment for Complex systems
PCF	Point Coordination Function
PHB	Per Hop Behavior
PHY	PHYsical Layer
PIFS	PCF IFS
PSTN	Public Switched Telephone Network
QAP	QoS supported Access Point
QoS	Quality of Service
QSTA	QoS supported STAtion

RED	Random Early Detection
RSVP	Resource ReSeVvation setup Protocol
RT	Real Time
RTS	Request To Send
SDM	Space Division Multiplexing
SIFS	Short IFS
Src	Source
STA	Station
TCP	Transmission Control Protocol
TXOP	Transmission Opportunity
UDP	User Datagram Protocol
WCA	Weighted Clustering Algorithm
WLAN	Wireless LAN
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network

PROVIDING CLUSTER-BASED QUALITY OF SERVICE IN MOBILE AD HOC NETWORKS

By
Khaled Mohammed Ahmed Hushaidan

Supervisor
Dr. Wesam Al Mobaideen

ABSTRACT

Mobile Ad hoc Networks (MANET) are wireless networks that can be easily deployed when and where needed, without the need to a fixed infrastructure or centralized administration. Ad hoc networks are important for their promising applications, such as in emergency situations, disaster recovery, battlefield communications, audio/video conferencing, etc.

A MANET consists of a collection of wireless devices (nodes) that communicate with each other using shared wireless medium. Each node in the network is assumed to be capable of forwarding packets (i.e. acting as a router), in addition to its role in sending and receiving data. Nodes maintain a specific Medium Access Control (MAC) function to contend in accessing the shared wireless link. In clustered MANET, nodes are grouped together into clusters to make a hierarchical control environment and facilitate routing.

Providing Quality of Service (QoS) in MANET is considered a challenging issue due to many constraints including: the network infrastructure-less nature, dynamic topology, low communication bandwidth, and the limited capabilities of wireless devices. QoS provisioning in MANET is desirable to provide better service and to improve the overall network performance.

In this thesis, we propose a new approach for supporting QoS in clustered MANET. The proposed Cluster-Based QoS (CBQoS) provides MANET with *inter-cluster/intra-cluster* service differentiation, and aims to improve the overall performance of clustered MANET, by increasing the overall network throughput and decreasing the overall delay encountered by MANET's applications.

The proposed approach has been evaluated under various network parameters, using GLOMSIM network simulator. The simulation results showed that CBQoS achieves significant improvement in MANET's performance and QoS support, especially for Real-Time applications. The results showed an improvement in the overall network throughput (+1.6%), with an impressive improvement in the overall network delay (-13.25%). This improvement allows for better service differentiation between different applications over the network.

INTRODUCTION

1.1.Mobile Ad Hoc Networks

The Mobile Ad hoc Network (MANET) is a wireless network that can be easily deployed when and where needed, without the need to a fixed infrastructure or centralized administration [Perkins, 2001][Murthy et al., 2004]. The importance of wireless MANET comes from their wide potential useful applications, such as in emergency situations, rescue operations, disaster recovery, battlefield communications, interactive information sharing, conferencing, and other multimedia applications. Ad hoc networks do not require existing infrastructure, so they are easily and rapidly deployed to provide cheap temporary communications.

A MANET consists of a set of wireless devices (nodes) that have limited resources, battery power, and transmission range. In addition to their general role of sending/receiving data, MANET nodes act also as routers. They cooperate together in forwarding messages to enable communications between nodes that are not in reach of each other. Nodes in the network are mobile; i.e. they can move freely. Nodes mobility is the reason behind the *dynamic topology* of MANET. This dynamicity complicates the network, especially from routing point-of-view, since it can lead to route changes and link breaks. These distinguished characteristics of MANET have raised many challenging issues for researchers [Murthy et al., 2004].

For scalability and performance purposes, nodes in MANET are usually grouped together in clusters. Each cluster contains a *clusterhead* node (CH) that serves in routing and management for other nodes in the cluster [Gerla et al., 1995]. The clusterhead is elected according to the clustering algorithm's criteria, which may be based on: node ID, power capability, geographical location, connectivity, etc. The first election happens when the cluster is formed; re-election of cluster-head occurs as a result of mobility or power constraints. Clustering provides MANET with a virtual hierarchy which helps in routing and forwarding data packets through the network.

1.2.Quality of Service

The subject of Quality of Services has been studied for the Internet since the early 1990's, to satisfy the huge number of Internet users and the different requirements of different Internet applications. A great research has been conducted on supporting Quality of Service (QoS) for the Internet and introduced QoS models for the Internet, such as Integrated Service (IntServ) and Differentiated Services (DiffServ); in addition to new QoS capable network technologies, such as Asynchronous Transfer Mode (ATM) and Multi Protocol Label Switching (MPLS) [Wang, 2001]. Supporting QoS in different networks is desirable to improve the performance of communications and satisfy the needs of the applications.

1.3.Problem Formulation

MANET QoS is considered a challenging issue due to many constraints including: the network infrastructure-less nature, dynamic topology, low communication bandwidth, and the limited capabilities of wireless devices [Murthy et al., 2004]. These constraints make it not straightforward to adopt traditional Internet QoS models for MANET, since these models were proposed for relatively high speed stationary networks. QoS provisioning in MANET is very important to provide better service, and to improve the overall network performance. In addition, the on-demand nature of MANET makes it suitable for real time applications (e.g. voice and video communications). These applications have stringent QoS requirements in terms of throughput and delay.

For these issues, a lot of work has focused on the study of QoS support for MANET and led to developing new QoS solutions, on different levels. On the Routing level, some existing routing protocols were changed and adapted for MANET; in addition to new protocols that were specially proposed for this kind of wireless networks [Barua et al., 2002]. On the MAC level, Medium Access Control protocols, such as IEEE 802.11, have been amended to support MANET with QoS [IEEE, 2005]. Special QoS Signaling systems were also proposed. In addition to cross layer and general QoS Models that allow for better service provisioning in MANET. However, the subject of supporting QoS for MANET is still a challenging and interesting research area to cope with the network dynamics and the limited resources [Wu et al., 2001].

In this thesis, we propose the CBQoS, a new solution for supporting QoS in MANET. This solution provides MANET with cluster-based service differentiation and aims to improve the overall performance of clustered MANET.

1.4. Thesis Contribution

The goal of this thesis is to study the subject of supporting QoS in MANET. This thesis also proposes a new QoS approach (CBQoS) that aims to improve the overall performance of the network, allowing for better service provisioning. The proposed approach targets clustered MANET, which adopts the cluster-based hierarchical structure, with a novel service differentiation approach that differentiates between inter-cluster communications and intra-cluster communications. We evaluate the overall performance of clustered MANET with the CBQoS to investigate the possibility of improving the performance of MANET.

The CBQoS has been implemented using the Global Mobile Information System Simulator (GLOMOSIM) [Bajaj et al., 1999]. GLOMOSIM is a sequential and parallel simulator for wireless networks. It is based on libraries that have been developed using the PARSEC (Parallel Simulation Environment for Complex Systems) [Bagrodia et al., 1998] simulation language.

The main performance metrics of QoS which we used to evaluate the CBQoS are *throughput* and *delay*. These two metrics are the main critical requirements for such dynamic environment. The simulation results show that CBQoS achieves significant improvement in MANET's performance and QoS support. The results showed that the overall network throughput has been increased and the overall end-to-end delay has been decreased using CBQoS. Increasing the throughput and decreasing the delay in the network allows for better service differentiation between different applications over the network.

1.5.Thesis Outline

The thesis is organized as follows: it begins with an introduction that introduces and clarifies different aspects related to the subject of the thesis. The Introduction introduces the subject of study, problem definition, and thesis contribution. Chapter 2 gives background knowledge on wireless networking, mobile ad hoc networks, and clustering. Chapter 3 introduces QoS concepts and models. Chapter 4, reviews the previous work conducted in supporting QoS in mobile ad hoc networks from different QoS aspects; QoS

Routing, QoS Medium Access Control, QoS Models, and QoS Signalling Systems.

Chapter 5 introduces the Cluster-Based QoS approach for Mobile Ad Hoc Network (CBQoS); stating the idea of the CBQoS, its justifications, architecture, design choices, and implementation. Performance evaluation of the CBQoS is presented in Chapter 6, where simulation results are shown and discussed. Chapter 7 draws conclusions and findings with recommendations for future work.

1.6. Wireless Networking

Wireless networks (vs. wired networks) are networks that use the *wireless radio* instead of wire *cables* as medium of communication. Wireless networks are being extensively deployed instead of wired networks for many reasons. The first reason is the ease of installation; the complications of installing wire cables and the costs of changes in the cabling plan are eliminated with the use of wireless networks. Wireless network also support easier mobility of the network devices. Another reason is the widespread use of handheld devices (PDAs, Pocket PCs, and smart phones) and portable computers, which has raised the need for easy and portable communication technologies among these devices, and between them and fixed devices. [Halsall, 2005]

Several wireless networking technologies have emerged and become widely used in different environments and applications. Wireless networks technologies are classified based on their range of coverage into: Wireless Local Area Network (WLAN), Wireless Personal Area Network (WPAN), Wireless Metropolitan Area Network (WMAN), and Cellular Networks [Tanenbaum, 2003].

Wireless Local Area Networks (WLAN) are being extensively used to replace the wired LAN for Internet and services accesses, and peer-to-peer communications in campuses, airports, shopping malls, stock market, commercial companies, and other environments [Halsall, 2005]. The IEEE 802.11 [IEEE, 1999] is the most widespread standard for wireless LANs and will be discussed subsequently in this section. The European counterpart standard for wireless LAN is the HIPERLAN [ETSI, 2007].

The Wireless Personal Area Network (WPAN) covers a very small area (several meters) such as a room or an office to serve in connecting personal devices; such as computers, printers, Personal Digital Assistants (PDAs), and smart Phones; with each other [Murthy et al., 2004]. The de-facto WPAN standard is Bluetooth. The IEEE 802.15 is a WPAN standard that has been derived based on the Bluetooth [IEEE802, 2007].

Wireless Metropolitan Area Network (WMAN) is used to enable broadband Internet access via antennae. The IEEE 802.16, also known as WiMAX, is the standard for Wireless MAN [IEEE802, 2007]. The European Telecommunications Standards Institute (ETSI) protocol for WMAN is known as HIPERMAN [ETSI, 2007].

Cellular Networks have been used for voice communications, but recently they have been improved to support multimedia communications. The Europeans developed the Global System for Mobile communications (GSM) for cellular communications, whereas the Americans developed the CDMA (Code Division Multiple Access) [Tanenbaum, 2003].

1.7.IEEE 802.11 WLAN Standard

The Institute of Electrical and Electronics Engineers (IEEE) firstly released the 802.11 standard in 1997. This standard is commercially known as Wi-Fi standard. It specifies the Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications [IEEE, 1999]. Several modulation techniques of the 802.11 standard have been released, including 802.11a, 802.11b, and 802.11g. Several service enhancement amendments have been, also, released by IEEE 802.11 working group, including IEEE 802.11e QoS enhancement, which will be discussed in chapter 4.

1.7.1. IEEE 802.11 PHY

Different physical layer specifications are used, namely: Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), and Infrared (IR) [IEEE. 1999]. The original 802.11 standard (802.11Legacy) operates on the ISM (Industrial, Scientific and Medical) radio frequency band, with maximum data rate of 2Mbps transmitted via IR, FHSS or DSSS.

The IEEE 802.11 working group has released several standards of the 802.11 standards family, including 802.11a, b, g, and n. The 802.11a, released in 1999, operates in the 5GHz ISM frequency with maximum data rate of 54Mbps using the Orthogonal Frequency Division Multiplexing (OFDM). The 802.11b was released also in 1999 with a direct extension of the DSSS modulation used in 802.11Legacy on the same 2.4GHz frequency and maximum data rate of 11Mbps. In 2003, 802.11g was released. 802.11g operates on the 2.4GHz frequency band using the OFDM modulation increasing the maximum data rate to 54Mbps.

The latest standard of 802.11 standards family is 802.11n, which is expected to be released in 2007 [Broadcom, 2007]. IEEE 802.11n aims to enable emerging media-rich applications supporting higher rates and increased reliability. This standard could offer up to 600Mbps data rate. It uses better implementation of OFDM, in addition to the Multi-Input Multi-Output (MIMO) components which exploits the *multipath* property of radio waves, using Space-Division Multiplexing (SDM), by splitting data streams into multiple *spatial streams* using multiple antennas. The *mulipath* is the property of radio waves to bounce off walls and other objects reaching the receiving object multiple times; If not controlled it degrades the wireless communications performance. The use of MIMO enhances the performance with higher data rates. [Broadcom, 2007]

1.7.2. IEEE 802.11 MAC

IEEE 802.11 MAC sub-layer [IEEE, 1999] defines two coordination functions: the mandatory Distributed Coordination Function (DCF), and the optional Point Coordination Function (PCF). DCF provides distributed channel access based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), while PCF provides contention-free centralized channel access control through *polling*.

CSMA/CA

IEEE 802.11 [IEEE, 1999] medium sharing mechanism uses MACAW, which is an improved mechanism of the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA or shortly MACA). The *Carries Sense* (CS) is performed either physically or virtually; the Physical layer provides a sensing mechanism, called Clear Channel Assessment (CCA), to determine whether or not a channel is idle. CCA senses by detecting bits in the air or checking the Received Signal Strength (RSS) of the carrier against a threshold. Virtual carrier sensing is used at the MAC layer; when a node hears a frame that is not directed to it, it reads the *Duration* field in the frame header, and sets it *Network Allocation Vector* (NAV) accordingly, as will be clarified subsequently in this section.

Multiple Access mechanisms (MA) are different in DCF and PCF and are discussed later on. The *Collision Avoidance* (CA) mechanism is used for 802.11 instead of the 802.3 Collision Detection (CD), because collision detection in the high error-rate wireless medium would drastically reduce the throughput [Murthy et al., 2004].

Collision avoidance is achieved as follows [Peterson et al., 2001]: Suppose a data *sender* A is about to transmit data to a data *receiver* B. Before A transmits the actual data, it transmits a *Request to Send* (RTS) control frame to B. The RTS frame is a small control frame that indicates its sender, receiver, and how long the sender wants to hold the medium (data length). When B receives the RTS, it senses the medium to be idle for a short time period known as the *Short Inter-Frame Spacing* (SIFS). If the medium is idle the receiver replies with a *Clear to Send* (CTS) control frame, which echoes the length field transmitted by RTS. When any other node C in the network listens to the CTS frame, it sets a local variable called *Network Allocation Vector* (NAV), which indicates how long the medium is reserved to avoid collision during this period. Other nodes, which do not hear the CTS frame, are free to transmit, because they are far away from the receiver, and may not cause collision. The sender, then, sends the data frame (after waiting for a SIFS time period after receiving CTS) to the receiver.

MACAW, which is used for 802.11 standards, is an extension to the original MACA with the addition of an *Acknowledgement* (ACK) control frame sent from the receiver to the sender after successfully receiving a data frame [Murthy et al., 2004].

Distributed Coordination Function (DCF)

The basic and mandatory function of the 802.11 MAC is DCF. The period of time during which the DCF occurs is called the *contention period* (CP). Nodes contend to access the medium in a distributed manner. For a node to use the medium, it must sense the medium for a time period of DIFS (DCF Inter-Frame Spacing). If the medium was sensed idle for DIFS, nodes trying to use the medium enter in a contention to access the medium. The contention is done using a Contention Windows (CW), which starts at a predetermined *CWmin*, and grows exponentially up to *CWmax*, using the *exponential back-off* technique. Each node randomly picks a number between 0 and its current CW; if two nodes pick the same smallest number, the *binary back-off* procedures occurs until only one node picks the smallest number among others. The node which picks the smallest number uses the medium, while other nodes freeze their back-off process and wait for an interval of DIFS plus the current declared NAV, if heard.

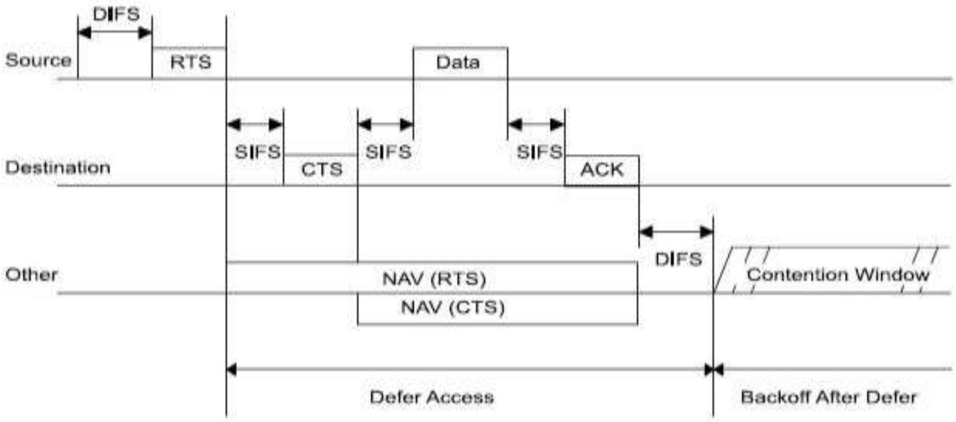


Figure 0.1: DCF access method. Src: [IEEE, 1999]

Point Coordination Function (PCF)

The optional PCF function occurs during a period known as the *Contention Free Period*

(CFP), where the wireless nodes rely on a centralized *Access Point* (AP) to control the medium access. The AP uses the PCF to efficiently distribute the time and throughput among the mobile stations. All frames in this period are transmitted via the AP. The AP uses *Polling* to control accesses to the medium as follows: AP periodically broadcasts a *Beacon* frame, which is a management frame that contains information about the data rate and modulation scheme, in addition to invitation for new mobile nodes to register with the AP. To gain priority over other nodes to access the medium, the AP (if it has data to transmit) waits for a period called *PCF Inter-Frame Spacing* (PIFS), which is shorter than the DIFS used by other nodes. The AP implements a *Point Coordinator* (PC), which splits time into *Super Frame* periods. Each supper frame consists of a CFP, and CP, consequently. The PC plays the role of a polling master, and determines which node has the right to transmit at any time.

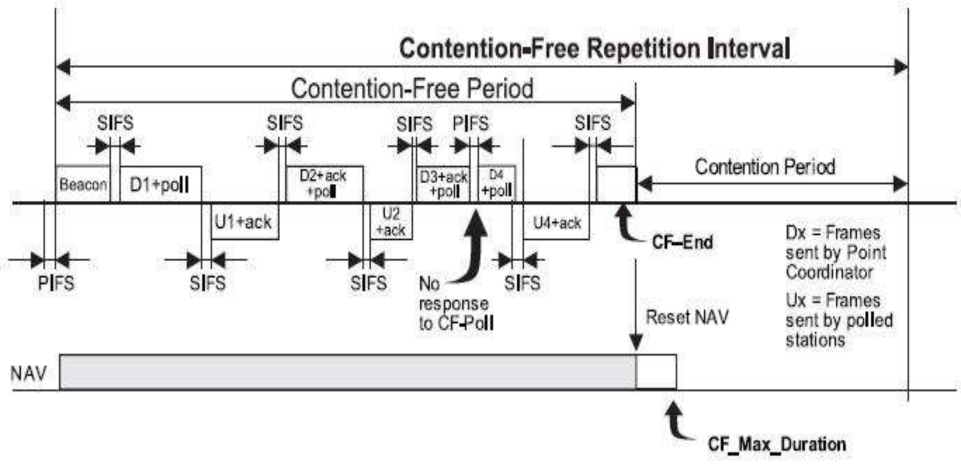
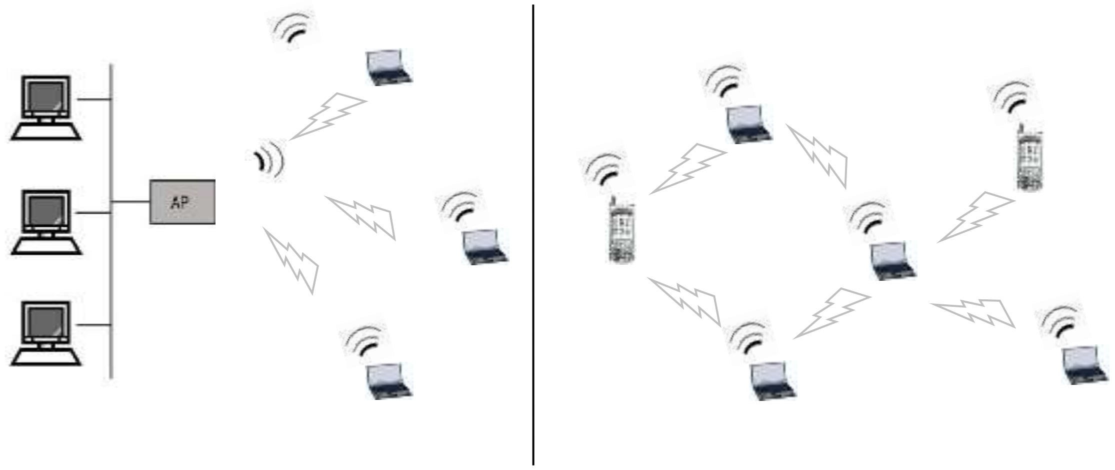


Figure 0.2: PCF access method. Src: [IEEE, 1999]

Infrastructure vs. Ad Hoc Wireless Networks

A simple infrastructure wireless LAN consists of a number of mobile *stations* (STA), and a fixed AP that controls accesses to the wireless medium. In this case, any two nodes (stations) can not communicate directly; i.e. all transmissions pass through the AP. In addition to its role in controlling the medium access, the AP also acts as a bridge to other networks (wireless or wired). The main application for this type of wireless networks is to replace the wired LAN [Halsall, 2005].

Ad Hoc Network, on the other hand, are infrastructure-less wireless networks that do not use AP. Nodes in ad hoc networks contend for the medium access in a distributed manner, using their own MAC functions. This type of wireless networks is discussed thoroughly in the next section. Figure 2.3 shows the two operational modes that are supported by IEEE 802.11 standard.



a) Infrastructure Mode

b) Ad Hoc Mode

Figure 0.3: Operational Modes of Wireless Networks

1.8.Mobile Ad Hoc Networks (MANET)

A Mobile Ad Hoc Network (MANET), also known as mobile multi-hop radio network, is a wireless network that can be easily deployed on-demand and does not require a pre-installed infrastructure [Perkins, 2001][Murthy et al., 2004]. It is composed of a set of wireless devices (nodes) that can communicate with each other via the electromagnetic radio medium. MANET has been receiving a lot of research works in the latest years. The importance of wireless MANET is due to the ease of deployment when and where needed, without requiring fixed infrastructure or centralized administration. In addition, MANET is suitable for temporal dynamic communications environment promising for many potential applications. The transmission range of wireless nodes is limited to short distances (for power saving). Hence, MANET nodes cooperate with each other in forwarding messages between nodes that are not in reach to each other. Thus, each node acts as a router (gateway for other nodes), in addition to its role in sending and receiving data. This multi-hop dynamic environment leads to distinguished characteristics of the network, making it different from other network topologies with new applications and additional requirements. Figure 2.3b shows one possible scenario of a mobile ad hoc network.

1.8.1. Applications of MANET

Ad hoc networks are very suitable for temporal and occasional group communication applications. The reason is that they are easy and fast to deploy in dynamic environments, in locations where communications networks are not available, or in situation where it is hard or expensive to use the fixed infrastructures. Examples of applications that can make use of MANET are: military applications (e.g. communication among individuals, tanks, ships or aircrafts), emergency and rescue

operations, collaborative and distributed computing, interactive information sharing, and other multimedia applications (e.g. audio, video conferencing) [Perkins, 2001] [Murthy et al., 2004].

1.8.2. Characteristics of MANET

Mobile Ad hoc networks have several characteristics that differ from those in wired environments. These characteristics raise new issues [Murthy et al., 2004] [Perkins, 2001].

- MANET is a multi-hop network that requires each node to accomplish two-fold functionality: the sending/receiving function, and the function of forwarding messages for other nodes.

- The wireless communications medium is shared, variable, unpredictable, and have low bandwidth. For instance, IEEE 802.11 has a low bandwidth (of 2Mbps, 11Mbps, and 54Mbps), compared to capacities of 100Mbps, 1000Mbps, or higher for wired networks. In addition, the already scarce wireless medium is shared between several nodes.

- Nodes mobility implies dynamic changing topology, in which links may break and routes may change frequently.

- Wireless devices are usually limited in battery life, processing power, memory, and transmission range.

1.9. Clustering

Mobile networks are dynamic and have unpredictable *self-organizing* topology that can be dynamically built and maintained by mutual cooperation of nodes, in response to network changes. Designing a dynamic control algorithm for mobile networks depends on the size of the network, and the expected dynamicity and changes in the network [Steenstrup, 2001]. The *clustering* is suitable for large dynamic networks, like MANET, for scalability and performance improvement.

Clustering can be seen as a graph partitioning problem [Chatterjee et al, 2002]. Peer-to-peer networks, e.g. MANET, are represented by undirected graph $G = (V, E)$, where V represents the set of nodes, and E represents the set of links. The clustering algorithm finds the dominant set S (the set of clusterheads) of the graph G, where each vertex v in S has a neighbourhood of nodes within its transmission range $N(v)$; and every vertex in G belongs to S or has a neighbour in S.

1.9.1. Cluster Members

In clustered networks, nodes are grouped together to form a virtual network of interconnected nodes (*clusters*). Member nodes in each cluster are of three types, namely: clusterhead nodes, gateway nodes, and ordinary nodes [Gerla et al., 1995]. The *Clusterhead (CH)* is a node that is elected to control the cluster in transmission scheduling and resource allocation. The *Gateway-node* is a cluster member that routes packets between its cluster and other clusters in the network. Other nodes in the cluster are referred to as *Ordinary-nodes*.

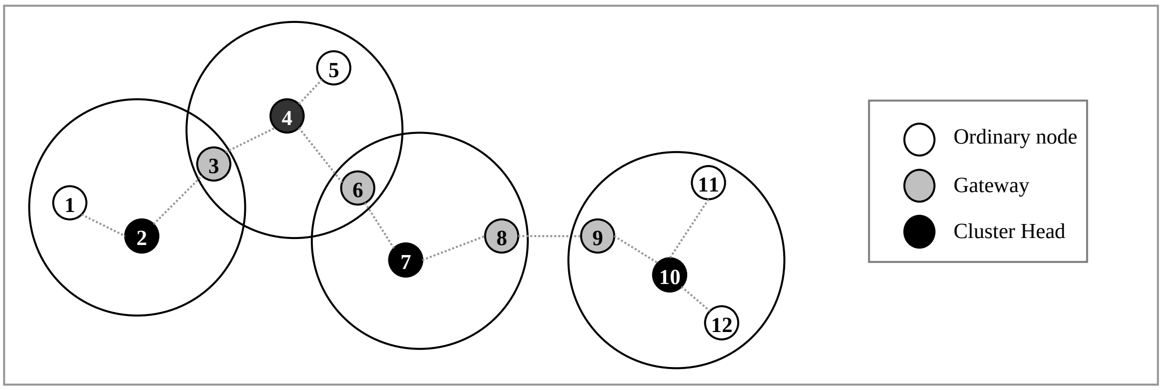
The CH, which acts as a controller that serves in controlling nodes in the cluster and serves in communications with other clusters, is elected according to the clustering algorithm's criteria which may be based on node ID, power capability, geographical location, connectivity, etc. The first election happens when the cluster is formed; re-election of cluster-head occurs as a result of mobility or power constraints. [Gerla et al., 1995] [Chatterjee, 2002]

To improve the network performance and eliminate the single point-of-failure introduced by assigning a single CH to each cluster, the *Distributed Clusterhead Architecture* was proposed in [Qaddoura et al., 2006]. This architecture distributes the load among multiple clusterheads in the same cluster improving the throughput and the routing reliability.

In Figure 2.4, an example of clustered mobile ad hoc network is shown. Assuming node 1 wants to send to node 8, it sends packets to its clusterhead, CH 2. Then, CH 2 forwards the packet to the neighbouring cluster through the gateway node 3, which forwards it to CH 4, and so on until reaching the destination.

1.9.2. Cluster Communications

Communications between nodes in clustered networks can be inter-cluster or intra-cluster [Steenstrup, 2001]. In *Intra-cluster communication (IA)*, the source node and destination node are in the same cluster; packets are usually sent from source to CH which forwards the packet to the destination. In *Inter-cluster communication (IE)*, the source and destination nodes are in different clusters; packets are sent from the source node to its CH which forwards the packet to the corresponding node's cluster. IE Packets may pass through multiple hops and clusters until reaching the corresponding CH which delivers packets to destination.



.Figure 0.4: Example of Clustered MANET

In addition to the clusterhead which plays the main role in cluster communications and management, the gateways play critical role in IE communications. A gateway node can be shared between two neighbouring clusters; hence, it follows to two clusters and has two clusterhead nodes. This arrangement is known as *overlapped clustering* [Steenstrup, 2001]. Another clustering approach, the *disjoint clustering*, limits each node belong to exactly one cluster. For IE communications, two gateway nodes in two adjacent clusters can communicate directly, making a bridge between their clusters. This pair of nodes is called the *distributed gateway (DG)* [Gerla et al., 1995]. The disjoint clustering overcomes the overlapping clustering in building clusters with higher connectivity.

In Figure 2.4, the three clusters headed by clusterheads 2, 4, and 7 are overlapped; while the clusters headed by CH 7 and 10 are disjoint, with the pair nodes (8, 9) as the *DG* between them. The clustering function can be seen as a process of building a graph containing all nodes in the network and aims to increase the connectivity in this graph.

1.9.3. Clustering Algorithms

Several *Clustering Algorithms* have been proposed for dynamic networks such as ad hoc networks. In [Baker et al., 1981], the well known LCA clustering algorithm was proposed. The distributed clustering algorithm for wireless networks was proposed in [Gerla et al., 1995]. Lin and Gerla [1997] proposed an adaptive clustering for mobile wireless networks. Another well known clustering algorithm, the weighted clustering algorithm for mobile ad hoc networks (WCA), was introduced in [Chatterjee et al., 2002].

The clustering algorithms are assessed based on their *quality* and *speed*; a good clustering algorithm should maintain high *conductance* (connectivity) of clusters; the speed of the clustering algorithm is considered in dynamic environments, where the clustering function happens frequently [Kannan et al., 2004]. In mobile environments, a good clustering algorithm should also be *stable* to nodes motion; i.e. the clustering configuration should not be drastically changed when the topology changes. [Gerla et al., 1995]

Some clustering algorithms place all member nodes within one hop to their clusterhead and hence within two hops of each other. Other algorithms like the one proposed in [Chatterjee et al, 2002] allows member nodes to be multiple hops far from their clusterhead and employ intermediate nodes in forwarding IA communications.

LCA Algorithm

The Link-Cluster Architecture (LCA) is a cluster-based architecture that was proposed for multiple access broadcast environment to reduce interference [Baker et al., 1981][Gerla et al., 1995]. Nodes are grouped into interconnected clusters. The IA transmission within a cluster can be scheduled in a contention-free manner. The IE transmission in adjoining clusters can be isolated by using different spreading codes in each through *Spread-Spectrum Multiple Access* [Gerla et al., 1995]. Each cluster contains three types of nodes: a single clusterhead node (CH) that controls transmission and manages cluster resources, one or more gateway (GW) nodes that facilitate inter-cluster transmissions, and a number of ordinary nodes.

The LCA can be used for building both overlapping and disjoint clusters, but it limits member nodes in a cluster to one hop from their clusterhead aiming to provide low-delay IA communications [Baker et al., 1981][Gerla et al., 1995]. The clustering function includes the following steps:

- *Neighbour discovery*: each node discovers its bidirectional connectivity with other nodes, by broadcasting a list of neighbours it can hear, and receiving broadcasts from others.

- *Cluster formation*: clusterheads are elected and clusters are formed. Two election algorithms were proposed by LCA: identifier-based and connectivity based [Gerla et al., 1995]. In *identifier-based* clustering (also known as ID-based clustering), the node with the smallest (or largest) identifier is chosen as the clusterhead. *Connectivity-based* clustering chooses node with the largest number of

- neighbouring nodes. The cluster members are the clusterhead and its one-hop neighbours.

- *Building Gateways:* adjacent clusters agree on gateways between them. In overlapping clusters, a node is chosen as a gateway if it is in range to two clusterheads. In disjoint clusters, if multiple nodes are candidates, ID-based criteria can be used to choose a single gateway-pair between each two adjacent clusters.

The algorithm described by Lin and Gerla [Lin et al., 1997] is a variant of the LCA algorithm. It is an ID-based clustering algorithm that always forms disjoint clusters.

Near-Term Digital Radio Network

The Near-Term Digital Radio Network (NTDR) [Zavgren, 1997], is a clustered architecture designed for large mobile tactical communications. Like the LCA algorithm, each cluster members in NTDR network architecture are one-hop far from their clusterhead. In NTDR, unlike LCA, the inter-cluster communication is restricted to clusterhead nodes. Clusterhead nodes are linked together to form a routing backbone and function as gateways. This architecture copes with the mobility by maintaining the backbone link. The clusterhead communicates on two different radio frequencies, one for inter-cluster communications with other clusterheads and the other for intra-cluster communications.

The clusterhead election in the NTDR is not based on nodes properties like the ID or connectivity. Rather, each node discovers its neighbouring connectivity by receiving periodical *beacons* from clusterheads. A node elects itself as a clusterhead if it does not hear *beacons* in its neighbourhood. To avoid two nodes from attempting to become clusterhead at the same time when election condition occurs, each node waits a short random time. The node retests the condition after this interval; if the election condition remains true (i.e. no clusterhead has been elected yet), it immediately sends a *beacon* assuming itself as a clusterhead.

WCA Algorithm

Weighted Clustering Algorithm (WCA) [Chatterjee et al, 2002] is an on-demand clustering algorithm for ad hoc networks. The clusterhead election takes into consideration the ideal degree of a node (the number of nodes within its transmission range), its transmission power, mobility, and battery power. WCA uses on-demand non-periodic election to make the election procedure as rare as possible, by avoiding re-clustering if the relative distances between nodes and their clusterheads do not change.

The clusterhead election procedure finds the *combined weight* of each node to select clusterhead nodes and their neighbourhood. This procedure consists of the following steps:

1- Find the *degree* of each node v. $\mathrm{d}_{\mathrm{v}} = |\mathrm{N}(\mathrm{v})| = \sum_{v' \in V(v)} \{dist(v,v') < tx_{range}\}$.

2- Compute the degree-difference. $\Delta v = |d_v - \delta|$, where δ is a pre-defined threshold (set in the initialization step) represents the maximum number of nodes a clusterhead can support. This threshold is defined to ensure efficient MAC functioning of the clusterhead limited delay encountered by nodes in the cluster.

3- Computer the sum of distances of each node with its neighbours. The more the distances, the higher the consumed transmission power. $D_v = \sum_{v'=N(v)} \{dist(v,v')\}$

4- Measure the mobility (M_v) of each node by taking the running average of its speed till the current time T. $M_v = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$; Where (X_t,Y_t) and (X_{t-1},Y_{t-1}) are the coordinates of the node at time t and (t-1). The more stable nodes are preferred to be clusterheads.

5- Compute how long the node has been a clusterhead, and consequently how much power (P_v) has been consumed to achieve clusterhead roles during this period.

6- The combined weight (W_v) for each node (v) is computed. $W_v = w_1\Delta_v + w_2 D_v + w_3 M_v + w_4 P_v$; Where w1, w2, w3, and w4 are the weighing factors for each parameter respectively.

7- The node with the smallest weight (W_v) is chosen as a clusterhead, and all its neighbours are not further considered in election.

8- Steps 2-7 are repeated for the remaining nodes, except the nodes eliminated in step 7 (the already elected clusterheads and their neighbours).

The WCA clustering is not invoked periodically; rather, it is *adaptively* invoked based on nodes mobility. The clusterhead election is delayed as long as possible, to reduce computation. WCA achieves *load balancing* through specifying a maximum number of nodes per clusterhead.

QUALITY OF SERVICE

1.10. Overview

Communication networks have been originally built on the Circuit Switching scheme, where a connection between each two parties has its dedicated permanent link that is installed to provide the intended services. Circuit switched networks was designed to provide telephony services, which are very sensitive to delay but require relatively small bandwidth. Most communication networks, such as the Public Switched Telephony Network (PSTN) and the Integrated Services Digital Network (ISDN), are circuit switched [Halsall, 2005].

The need for a robust data transmission network was behind designing the Internet by the US Defence Advanced Research Project Agency (DARPA) in1960’s [Wang, 2001]. The Internet was built on the *Datagram Model* [Peterson et al., 2003], which uses packet switching scheme to overcome the complexity and inflexibility problems of circuit switched networks. The main reason behind developing the packet switching model was *fault tolerance,* because packets are allowed to choose any available link, instead of a single link that may fail [Tanenbaum, 2003].

The Internet and many network technologies use the datagram model to provide several services; like File Transfer, Email, Remote Access, World Wide Web, and even Telephony applications. The variety of Internet applications and the huge number of Internet users were the reasons behind the idea of service differentiation [Wang, 2001]. Although high speed technologies have been developed to solve the problem, the problem remains with the different requirements of different applications. File Transfer applications, for example, require high bandwidth, but they can tolerate long latencies.

On the other hand, multimedia applications like *telephony* and video conferencing, are sensitive to the timing of data and do not tolerate latency. They require special treatment to their packets passing through the network. Network Quality of Service (QoS) is the capability of the network to provide different levels of service [Peterson et al., 2003].

1.11. Network Congestion

Packet switched networks allows packets that belong to different flows to share the same bandwidth and the same buffers, in routers and switches, along the link between the sender and the receiver. Data packets are buffered in queues waiting for their turn in transmission over the link. When too many packets are waiting to be transmitted via the same link (high contention occurs), the buffer overflows and some packets are dropped. In this case, the network is said to be congested [Tanenbaum, 2003][Peterson et al., 2003]. When congestion occurs, application flows that share the congested link suffer higher delay and lower throughput. Some applications use congestion control mechanisms to deal with this problem. However, some applications require stringent requirements of throughput and delay, to be delivered to the users with an acceptable quality. For this purpose, the subject of service differentiation and QoS has been studied aiming to assure better service to satisfy applications requirements.

1.12. Real Time Applications

Real Time (RT) applications, versus *non real-time* (or *elastic*) applications, are applications that are sensitive to the timing of data; i.e. they require the network to deliver their data packets on time [Peterson et al., 2003]. Examples of real-time applications are: voice communications, remote video, multimedia conferencing, visualization, virtual reality, and industrial control applications [Braden et al., 1994]. RT applications are classified bases on their characteristics [Peterson et al., 2003]; Based on their tolerance to occasional loss of data, RT applications are categorized into *tolerant* (e.g. voice) and *intolerant* (e.g. robot control programs). RT applications are also categorized into *adaptive* and *non-adaptive*, according to their adaptability to the amount of bandwidth or delay experienced by data packets. Adaptive RT applications monitor the characteristics of the network and adjust their service accordingly. These applications, for example, can speedup their transmission rate when the network is lightly-loaded, and slowdown (and may use other performance improvement techniques like compression) when the network is highly-loaded.

An important class of RT applications are *playback* applications [Braden et al., 1994][Peterson et al., 2003]. In these applications, packets are transmitted over the network from the source to the destination. The receiver buffers the received packets, and plays them back after a specific delay from the original sending time. The *playback point*, which is the specific delay from the original sending time, is adjusted according to the delay and jitter (variability of the delay) introduced by the network. The playback point of some applications, like video streaming, can have no stringent limits; while there are limits to the playback time for applications like conferencing.

1.13. QoS Requirements

QoS requirements include throughput, delay, jitter, reliability (error rate) [Tanenbaum, 2003]. Different applications require different QoS requirements; so, there have been different QoS models proposed to satisfy different QoS requirements. For example, QoS mechanisms that concentrate on reliability are not mainly concerned with the application’s throughput requirements. The different QoS requirements are briefly defined in [Peterson et al., 2003], [Halsall, 2005], and [Tanenbaum, 2003] as follows.

Throughput: measures the actual amount of data that is delivered from one node to another, over a communication link in one unit of time. It is usually measured in bit per second.

Delay: also called latency; is the delivery time of packet. End-to-end delay refers to how long it takes a message to travel from its source to its destination. The end-to-end delay has three components: the propagation delay (through the transmission medium), the transmission time (the time it takes to transmit a unit of data), and the queuing delay (in intermediate nodes and buffers).

Jitter: is the variation in delay. Buffering techniques, usually provided by the application layer, are necessary to overcome the negative effect of jitter, by adjusting the playback time.

Reliability: measures how an application is tolerant to errors. Some applications; like control systems, email, and file transfer; do not tolerate errors in data; while, others, like telephony and video broadcasting, may use some techniques to tolerate errors.

1.14. QoS Models

The Internet Engineering Task Force (IETF) [IETF, 2007] has worked to develop new technologies for the Internet to support QoS. The basic service model provided by the Internet is the Best Effort service (BE), in which packets of all applications are dealt the same when they contend for network resources [Wang, 2001]. BE Packets are queued for transmission in a first-in first-out (FIFO) Queue. This model provides no guarantee, and represents the lack of QoS, because there is no differentiation between packets. IETF effort on service differentiation models aimed to achieve two goals: performance assurance, and service differentiation [Wang, 2001]. Performance assurance implies improving the network to provide the predictable performance; whereas, Service Differentiation aims to provide multiple levels of services, to meet different application requirements and different customer needs.

The Approaches for supporting QoS can be divided into two broad categories [Peterson et al., 2003]:

- *Fine-grained* approaches: QoS is provided for individual applications or flows. These are also known as *per-flow* QoS approaches.
- *Coarse-grained* approaches: which are also known as *per-class* QoS approaches. They provide QoS to large classes of data.

Two major service differentiation models were developed by the IETF: Integrated Services (IntServ) which is a *fine-grained* QoS approach, and Differentiated Services (DiffServ) which is a *coarse-grained* approach. These two major QoS models are discussed subsequently in this section. Other mechanisms that were developed to enable QoS, like Multi Protocol Label Switching (MPLS) and Traffic Engineering [Wang, 2001], are aimed for specific network technologies and are not discussed here.

1.14.1. Integrated Services (IntServ)

Real time applications, with their stringent QoS requirements, have pushed forward towards supporting the Internet with new types of service that provide some level of assurance to these applications. The first major attempt to support the Internet with QoS is Integrated Services (IntServ) [Braden et al., 1994]. The idea of IntServ was adopted from the Telephony system and Circuit Switched networks in the resource allocation and reservation mechanisms. IntServ aimed to control the shared link to guarantee resources to specific user flows. To assure resource allocation, applications reserve the required resources before they transmit data onto the network. IntServ is called per-flow service, since the resource reservation is made per each individual flow.

IntServ Reference Model

The Integrated Services Reference Model [Braden et al., 1994], as shown in Figure 3.1, includes four main components: the packet classifier, the packet scheduler, admission control module, and the reservation setup protocol.

Admission Control is an algorithm that decides whether to accept a flow and grant it the requested QoS [Braden et al., 1994]. This decision is done at the reservation time based on the administrative policies. When a flow is accepted, its subsequent packets are treated upon the reserved resources.

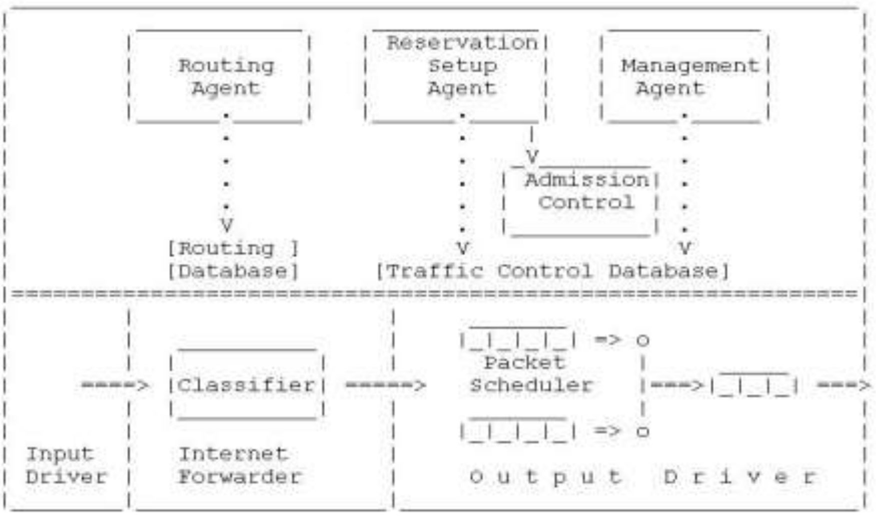


Figure 0.1: IntServ Reference Model. Src: [Braden et al., 1994]

Packets are classified, by a *packet classifier,* according to their flow state. The classifier identifies packets by the content of their existing IP header, or by using another classification number added to each packet. Classes are per-flow, but sometimes many flows are aggregated into few classes, especially in backbone routers.

Packet scheduling is performed by a specific scheduler that queues packets and forwards them according to their reserved resources. The scheduler, also, implements a dropping policy to control flows and ensure that they conform to their QoS specifications.

The *Reservation Setup Agent* module installs a reservation state for each accepted flow in both endpoints of the flow, and in the routers along the flow's path.

RSVP

The Resource Reservation Setup Protocol (RSVP) [Zhang et al., 1993] [Braden et al., 1994] has been developed for reservation setup in Integrated Services. RSVP goes through the path between the communicating parties (sender and receiver), and installs the reservation state in routers to setup reservation.

An application that needs QoS specifies its requirements in a list of parameters called *FlowSpec*. The *FlowSpec* is used to determine the resource quantity. The corresponding flow packets are specified and classified by a *FilterSpec*. Has the reservation been made, the *FlowSpec* is used by the packet scheduler to set the parameters of the flow's class; while *FilterSpec* is used to the incoming packets to decide in which class they will be classified.

RSVP uses receiver-initiated reservation with “Out-of-band” signalling mechanism (that dedicates special messages for reservation). RSVP sender distributes a *PATH* message that carries *FlowSpec*, distributes information about the traffic source, and passes information about the path from the sender to receivers. RSVP receiver learns the FlowSpec of the sender from the PATH message, then sends *RESV* message back towards the sender to request the reservation. The RESV message goes along the reverse path of the PATH message, specifies the resource requirements *RSPEC*, and builds a reservation state for the flow in each router. The sender starts transmitting packets after it receives the RESV message.

IntServ Services Classes

IntServ provides two service classes, in addition to Best Effort service: Guaranteed Service and Controlled Load Service [Braden et al., 1994]. The *guaranteed service*, also called hard QoS, provides guaranteed bandwidth and bounded end-to-end queuing delay. It is intended for applications that have stringent bandwidth and delay requirements (e.g. mission control systems and intolerant playback application).

Guaranteed service reserves network resources for the worst case. This makes it suitable for hard real-time applications that have bounded predictable bandwidth. *Controlled Load (predictive) Service* provides less strict guarantees and lower cost of reservation. It is suitable for applications that require some performance assurance but have no absolute bandwidth or delay bounds. This service model is also referred to as *better-than-best-*

effort service, because its service is in-between the Best Effort service model, and the Guaranteed service model [Wang, 2001].

1.14.2. Differentiated Services (DiffServ)

The Differentiated Services Model (DiffServ) [Blake et al., 1998], also referred to as *Soft QoS*, provides a scalable service differentiation in the Internet. In contrast to the per-flow classification of IntServ, DiffServ (interchangeably abbreviated as DS in this context) achieves scalability by aggregating traffic into a specific number of service classes. Each class is assigned different DSCP (Differentiated Services Code Point). Different service classes are configured to receive different priorities on network resources. Packets are marked and classified to receive specific per-hop priority-based forwarding behaviour (Per Hop Behaviour: PHB) at network boundary nodes, making use of the IP DS field. Unlike IntServ, DiffServ does not require explicit reservation of resources.

The contiguous set of nodes, that provides DiffServ, is called *DS Domain*. Domain nodes that are capable with DiffServ functionalities are called Boundary Nodes; whereas, other nodes in the domain are called Interior Nodes. A Node in the Differentiated Services is classified into 3 categories based on its role in handling traffic: Ingress, Egress, and Interior. A node can play different roles for different traffic at the same time.

1- *Ingress Node*: a boundary node that handles traffic as it enters a DS domain.
2- *Egress Node*: a boundary node that handles traffic as it leaves a DS domain.
3- *Interior Node*: a node that belongs to the DS domain but not a boundary node.

The functions, that DS boundary nodes are supposed to achieve in providing differentiated services, are described below [Blake et al., 1998].

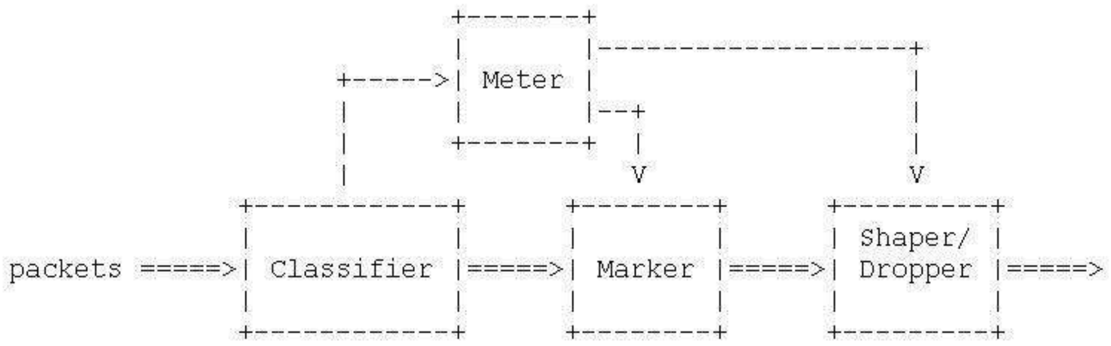


Figure 0.2: Logical View of DiffServ Components. Src: [Blake et al., 1998]

Traffic Classification

Traffic classification identifies packets which may receive a differentiated service. The classifier can be of two types: Behaviour Aggregate (BA) or Multi-Field (MF). BA classifier classifies packets based only on the DSCP in their IP header; whereas, MF classifier classifies them based on a combination of multiple IP header fields.

The properties of a traffic stream, selected by traffic classifier, are maintained in a *Traffic Profile*, to determine whether a packet is in-profile or out-of-profile. The final function of the classifier is steering packets matching particular rules to the appropriate component of the traffic conditioner for further processing.

Traffic Conditioning

Traffic Conditioning ensures that the traffic entering the DS domain conforms to the service provisioning policy. It includes several functions, such as: metering, shaping, policing, and/or re-marking; as shown in Figure 3.2.

Traffic *meters* are used to measure packets against their traffic profile, to determine in-profile and out-of-profile packets, and pass them to the appropriate conditioning function. The *marker* marks a packet with a particular DSCP, which is used to select the PHB of the packet. The marker may re-mark a pre-marked packet with a different DSCP, according to the state of the meter.

Traffic Shaping is the process of delaying packets in a traffic flow, to ensure that this flow conforms to its traffic profile. The shaper has a finite buffer and may discard packets, if the space is not sufficient.

The *dropper* discards some packets in a traffic stream, according to the state of the traffic profile. This function is known as Policing.

Per-Hop Behaviour

The DiffServ Per-Hop Behaviour (PHB) is the forwarding behaviour that is applied to allocate resources to DS behaviour aggregates at DS-compliant nodes [Blake et al., 1998]. PHB is selected for a packet according to its DSCP field. PHB can be simple or complex depending on the constraints on the characteristics of the associated behaviour aggregate. An example of a simple PHB is to guarantee minimal bandwidth to specific behaviour aggregate. A complex PHB would satisfy multiple constraints, like guaranteeing minimal bandwidth, maximal delay, and fair sharing of link capacity. Traffic classes (i.e. behaviour aggregates) differ in their PHB, in their relative resources priority (bandwidth, buffer, etc.), or in their traffic characteristics (delay, loss, etc.)

DiffServ Service Classes

Two service classes of DiffServ were presented: *Premium Service* and *Assured Service* [Nichols et al., 1999]. In *Premium Service* (PS), a specific percentage of the network capacity is allocated for premium flows, which are charged higher than other flows. The rest of the capacity is used for other service classes like Best Effort. However, the whole capacity can be utilized by other flows, when there is no *PS* flows. PS is suitable for commercial RT applications that require guarantees in bandwidth and delay. *Assured Service* (AS) is a moderate service that aims to provide expected throughput to specific traffic better than *BE*, but not as good as *PS* service. This service gives its traffic less dropping probability than that of *BE* traffic.

1.15. MANET QoS Issues

It is more difficult and challenging to provide QoS in MANET than in wired networks, because of many constraints, including: MANET's infrastructure-less nature, dynamic topology, low communication bandwidth, and limited capabilities of wireless devices. Applying classic QoS Models that were proposed for the Internet on MANET raises many issues.

1.15.1. IntServ and MANET

Pure IntServ is not practical in MANET for the following issues [XIAO et al, 2000]:

Scalability: IntServ provides per-flow QoS by establishing an end-to-end connection for each flow with state information reserved in every interior node in a connection. This limits IntServ to lightly loaded and high speed networks, and it is difficult to provide per-flow service in MANET.

Dynamicity: With MANET mobility and topology changes, RSVP Connection maintenance overcomes the connection establishment. RSVP assumes long timescale connection (i.e. fixed networks), which is not expected in MANET. When a node involved in an RSVP connection moves out, it is necessary to either handover state information to a new node that can replace the old, one or to establish a new RSVP connection; both these functions consume high processing overhead of routers. This is undesirable for power-constrained nodes of MANET.

1.15.2. DiffServ and MANET

DiffServ has a potential usage in MANET for the following [XIAO et al, 2000]:

- It is a lightweight service that requires simple node functionality,
- No virtual circuit is established and consequently no connection maintenance is needed with nodes movements.
- It does not burden interior nodes with state information and signalling.
- DiffServ Assured Service (AS) aims to provide expected throughput to specific traffic and can be used in MANET's applications that requires high throughput.

But DiffServ has some features that make it not wise to adopt it as is for MANET. These features are [XIAO et al, 2000]:

- DiffServ was designed for high speed fixed networks while MANET is limited in speed.
- DiffServ Premium Service (PS) is supposed to provide guarantees in bandwidth, delay, and loss rate. This is hard to maintain in the dynamic structure of MANET.

PREVIOUS WORK

A lot of work has been conducted in supporting QoS for the Internet and other wired network technologies, producing QoS models like Integrated Services (IntServ) and Differentiated Services (DiffServ), and QoS enabled technologies like ATM (Asynchronous Transfer Mode) and MPLS (Multi-Protocol Label Switching). However, QoS techniques used in wired networks can not be directly used in mobile ad hoc networks (MANET), because of their bandwidth constraints and dynamic network topology [Wu et al., 2001].

Many researches have focused on studying QoS in MANET and introduced new QoS routing, QoS resource reservation signalling, QoS MAC, and general QoS Models.

1.16. QoS Routing

Several routing algorithms were proposed to provide QoS in MANET, such as CEDAR [Sivakumar et al., 1999], Ticket-Based Probing [Chen et al., 1999], and ACRQ [Barua et al, 2002].

1.16.1. CEDAR

Core-Extraction Distributed Ad-hoc Routing (CEDAR) algorithm presented in [Sivakumar et al., 1999] is a robust and adaptive QoS routing for ad hoc network environment. CEDAR establishes a core network dynamically then propagates the link state of established bandwidth links to the core nodes incrementally. CEDAR was proposed for small to medium size networks that consist of tens to hundreds of nodes. CEDAR has three key components:

- *Core Extraction*: establishes and maintains a self-organizing routing infrastructure to perform route computations.
- *Link State Propagation:* propagates the link-state of established links to the core.
- *Route Computation:* a QoS route computation algorithm executed at the core nodes using locally available state.

CEDAR allows for any well known routing protocol such as DSR, TORA, AODV, ZRP, etc. to be used in the core graph. In addition, CEDAR has its own QoS route computation. The major disadvantage of CEDAR is that it is not scalable for large networks.

1.16.2. Ticket-Based Probing

Ticket-Based Probing [Chen et al., 1999] is a multipath distributed routing scheme for ad hoc wireless networks. Instead of flooding for route discovery, this scheme tries to minimize the overhead by localizing the routing activity in a portion of the network searching for a specific number of paths between the sender and receiver, and choosing the best candidate paths among them. Ticket-Based Probing works as follows: The sender issues a probe message with one or more tickets based on the number of paths needed to satisfy QoS requirements. If an intermediate node receives the probe message that is carrying more than one ticket, the intermediate node splits the probe message and sends each with one ticket to different paths, such that each probe message contains at least one ticket; so, the number of probes at any time is limited by the number of tickets issued. The intermediate node chooses the best candidate path(s) to forward the probe(s). Ticket-based probing can handle different QoS requirements.

Ticket-Based Probing has some advantages, including: the lower overhead route discovery (by localizing the route discovery messages), fault tolerance (by finding multiple paths), and the ability to tolerate imprecise state information by sending multiple tickets to increase the chance of finding a feasible path.

1.16.3. ACRQ

Barua and Chakraborty [Barua et al, 2002] have proposed Adaptive Cluster-based Routing with QoS support (ACRQ). ACQR is a cluster-based route discovery and dynamic route management protocol for ad hoc networks. It deals with the inaccurate information in MANET nodes due to the network dynamics. ACRQ provides cluster-based routing. Each node in the cluster has only one link connecting it to its clusterhead. The clusterhead collects link information from each node to find the maximum delay (d_{max}) and minimum rate (r_{min}), the value of d_{max} and r_{min} provided are qualified and used to compute the probability that the resources are still available.

1.17. QoS Models

Several QoS Models were proposed for mobile ad hoc networks. In this section, we investigate some of theses models.

1.17.1. FQMM

FQMM [Xiao et al., 2000] is a Flexible QoS Model for MANET which considers

MANET characteristics. It is a hybrid QoS provisioning scheme that combines IntServ and DiffServ, taking the advantages of both.

FQMM defines three kinds of nodes: Ingress node, Interior node, and Egress node; as in DiffServ (see section 3.5.2). Ingress nodes perform traffic shaping functions, including: classification, marking, and policing of packets. Interior nodes forward data based on a specific PHB, determined by the DSCP field in IP packets.

FQMM overcomes the scalability problem of IntServ by guaranteeing per-flow service to a small portion of network traffic (which is classified with high priority). It takes the advantages of DiffServ simplicity, lightweight, coarse grain features for low-priority traffic; in addition to its use of a dynamic profile, which is suitable for MANET. However, FQMM authors did not clarify some aspects, such as the ratio of per-flow traffic to the overall network traffic, and the scheduling and classification mechanism.

1.17.2. SWAN

Another service differentiating model for wireless ad hoc networks, SWAN, was proposed in [Ahn et al, 2002]. SWAN is a simple, distributed, and stateless network model that provides service differentiation in MANET. SWAN is stateless, since there is no need to maintain per-flow state information in intermediate nodes.

To regulate best-effort traffic, SWAN performs rate control in every mobile node, in a distributed manner. For soft real-time, SWAN uses feedback-based control mechanisms utilizing the *explicit congestion notification* (ECN). ECN use the last 2 bits of the IP TOS header field (the first 6 bits are used for DSCP): *ECN-Capable Transport* bit that indicates weather ECN is used, and *Congestions Experienced* bit that indicates that congestion has occurred. SWAN's ECN mechanism forces RT flows to re-establish their real-time service when a mobile node observe violation of real-time sessions.

SWAN model is illustrated in Figure 4.1. The Classifier differentiates RT and BE packets, forcing the *shaper* to regulate BE packets, but not RT packets. A Rate controller calculates the rate of BE packets, and feedbacks this rate to the *shaper* to delay the BE packets accordingly. Admission control test is done solely at the source node, based on estimation the availability of local bandwidth; there is no admission control mechanism or state information maintained at intermediate nodes.

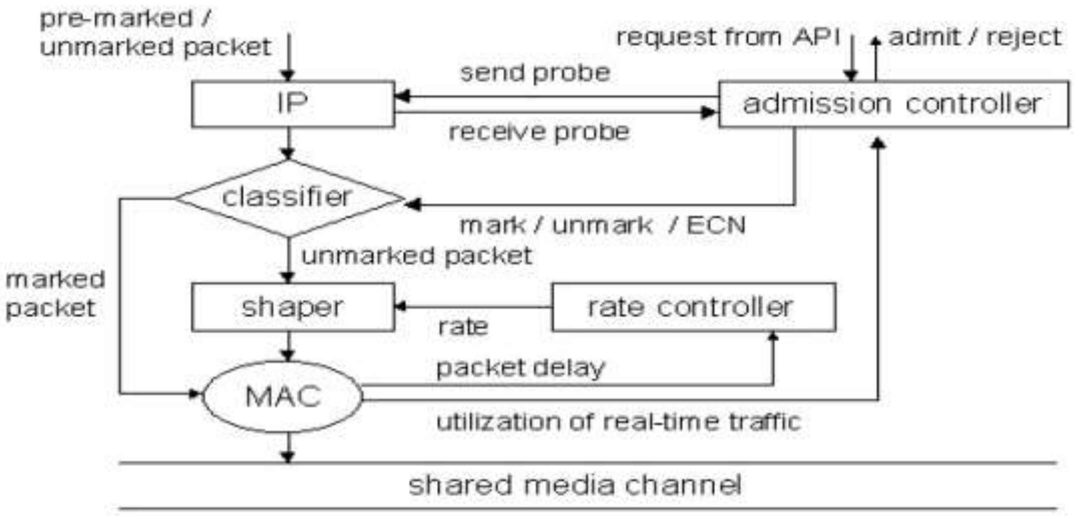


Figure 0.1: SWAN Model. Src: [Ahn et al., 2002]

SWAN advantages appear in its stateless mechanism, which does not bother intermediate nodes with state management, and does not reserve resources. However, SWAN is not suitable for providing hard QoS, because it does not reserve resources for specific flows.

1.17.3. HQMM

The Hybrid QoS Model for MANET (HQMM) [He et al, 2006], combines the responsive per-flow service of INSIGNIA (section 4.3.1), and the flexible per-class granularity of DiffServ, to support QoS in MANET. The idea behind HQMM is similar to that of FQMM, except that the per-flow signaling in HQMM is provided by INSIGNIA, instead of IntServ which is used in FQMM. INSIGNIA was adopted for its lightweight and highly responsiveness to the dynamics of MANET.

As in DiffServ and FQMM, nodes in HQMM are classified into three types, namely: ingress, interior, and egress nodes. Each node can perform various roles at the same time according to its position in each flow. Each node has a traffic conditioner that classifies traffic, marks the packets by setting the DSCP filed or INSIGNIA options, and drops out-of-profile packets. Nodes are also incorporated with a packet-forwarding module that forwards packets according to their priority, and delivers the signaling messages to INSIGNIA module. In addition to a link management module that monitors the channel state, estimates the available bandwidth, reports the instantaneous available bandwidth to the scheduling module, and reports the average available bandwidth to the admission control module of INSIGNIA.

HQMM inherits the advantages of FQMM and overcomes FQMM, by adopting the lightweight INSIGNIA rather than IntServ. However, a decision about the QoS routing and its effect on HQMM performance is still unresolved.

1.18. QoS Signaling

Analogous to the RSVP signaling, which is used in wired networks, some signaling system have been proposed for wireless networks, such as INSIGNIA [Lee et al., 1998] and INORA [Dharmaraju et al., 2002]. INSIGNIA, a good example of ad hoc QoS signaling cited in the literature, is studied in the next section.

1.18.1. INSIGNIA

INSIGNIA [Lee et al., 1998] is a QoS resource reservation signalling that was designed solely for MANET. It was designed as a lightweight and highly responsive to changes in the network, to support fast flow reservation for adaptive real-time applications. Unlike RSVP, which is *out-band* signalling protocol (implements its own control messages of reservation); INSIGNIA is *in-band* signalling protocol that encapsulates some control signals in the IP option of every data packet. The in-band approach allows for fast restore of flow-state in response of topology changes. INSIGNIA framework has the following components: packet forwarding module, routing module, INSIGNIA module, admission control module, packet scheduler, and MAC module; as shown in Figure 4.1. The INSIGNIA Module performs the main signalling operations which include reservation, restoration, adaptation, and state management.

A new IP option called INSIGNIA option is used to establish, restore and adapt resources between the source-destination pairs. INSIGNIA IP Option consists of five fields as follows:

1- *Reservation Mode*: One bit to identify if the reservation has been. This bit is used to decide either to accept, or deny reservation if the reservation has not been made. Otherwise, it is used to indicate that the packets have passed admission control.
2- *Service Type*: One bit indicates the level of service that is either real-time (RT) or best-effort (BE), depending on the reservation mode.
3- *Bandwidth Request*: 16 bits, allows a source to specify its maximum (MAX) and minimum (MIN) bandwidth for adaptive real time service.
4- *Payload Indicator*: allows INSIGNIA to support two layers of payload, namely: Base Load (BL), and Enhancement Load (EL).
5- *Bandwidth Indicator*: A MAX/MIN bit that indicates resource availability in intermediate nodes during flow setup. If a packet is received with bandwidth indicator of MAX, it indicates that all intermediate nodes that the packet has passed are all capable to provide the flow with the maximum bandwidth, indicated by the bandwidth request field. Otherwise, the bit value of MIN indicates that there is at least one node in the path that can support only the minimum bandwidth requirements.

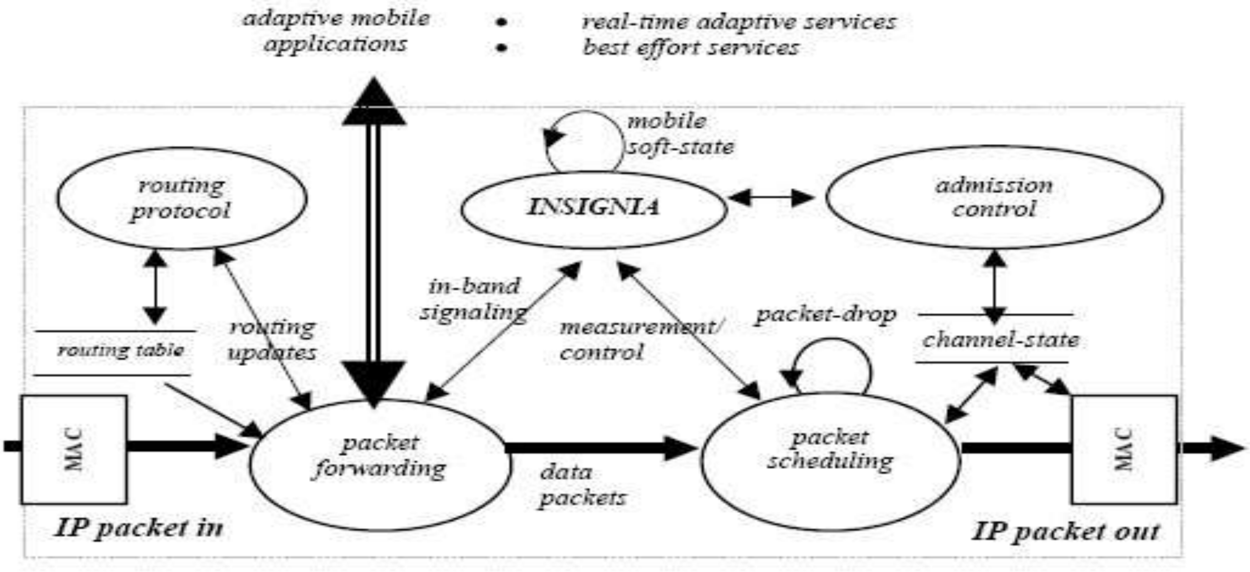


Figure 0.2: INSIGNIA model. Src: [Lee et al., 1998]

INSIGNIA has many advantages. It was the first signalling system solely designed for MANET, considering its dynamics and scarce resources. It provides highly adaptive service which dynamically assigns resources for flows based on the availability of resources.

INSIGNIA has a soft-state reservation and fast restoration to deal with topology changes. However, INSIGNIA does not provide hard-state reservation, so it is only suitable for adaptive RT applications that do not have stringent QoS requirements.

1.19. QoS MAC

QoS MAC aims to provide different opportunities for nodes to access the medium. IEEE 802.11e is the QoS enhancement for the 802.11 standard.

1.19.1. IEEE 802.11e

IEEE 802.11e standard [IEEE, 2005] provides Medium Access Control (MAC) QoS enhancement in wireless networks with a Hybrid Coordination Function (HCF), which combines and enhances the aspects of the contention-based and contention-free access method, to provide QoS access to the wireless medium. An enhanced DCF (EDCF) replaces the legacy DCF. Figure 4.3 depicts the MAC architecture of the IEEE 802.11e.

EDCF is used only during the contention period, while the HCF can be used in both the contention period and the contention free period.

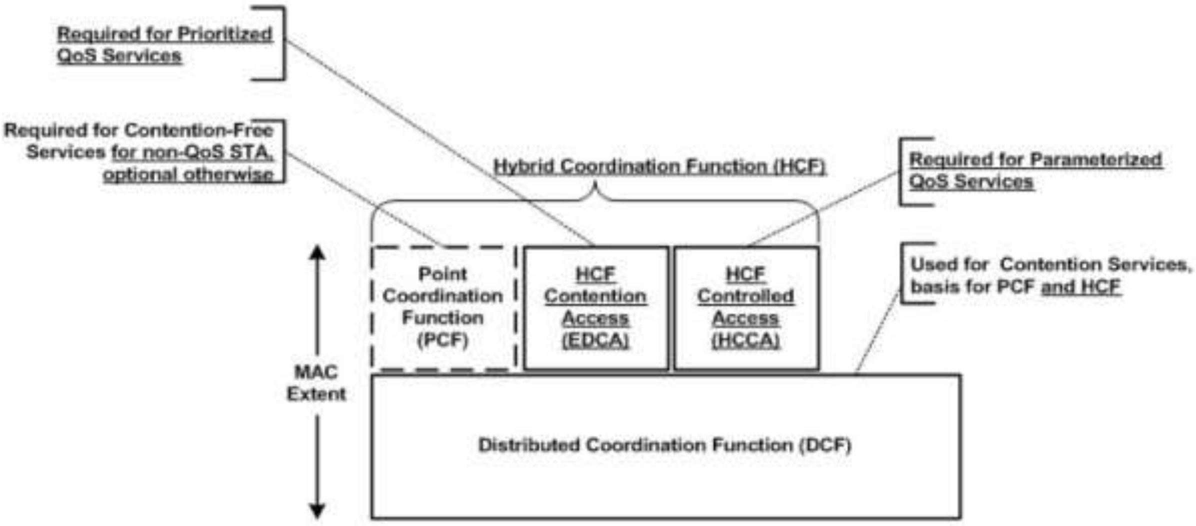


Figure 0.3: IEEE 802.11e MAC Architecture. Src: [IEEE, 2005]

Enhanced Distributed Coordination Function (EDCF)

It is a prioritized CSMA/CA access mechanism that enhances the original 802.11 DCF function. EDCF provides differentiated distributed access to the wireless medium for QoS supported stations (QSTA) with eight *user priorities* (UP); as shown in Figure

4.5. The service differentiation is achieved by using different CW_{min} and CW_{max}, and different Inter-Frame Spacing (IFS), for each traffic category. High priority traffic is given small CW_{min} and CW_{max} and small IFS. *Arbitration Inter-Frame Spacing* (AIFS) is used instead of DIFS during the distributed coordination function. Different AIFS are used for different traffic categories. Therefore, the back-off time differs for different traffic categories. The longest AIFS is used for traffic that does not require QoS.

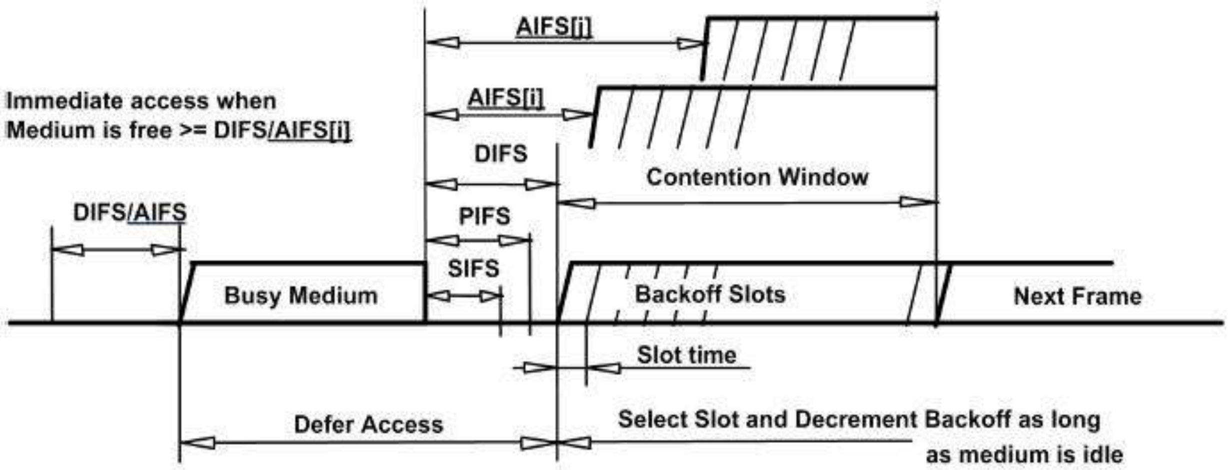


Figure 0.4: Representation of EDCF access method. Src: [IEEE, 2005]

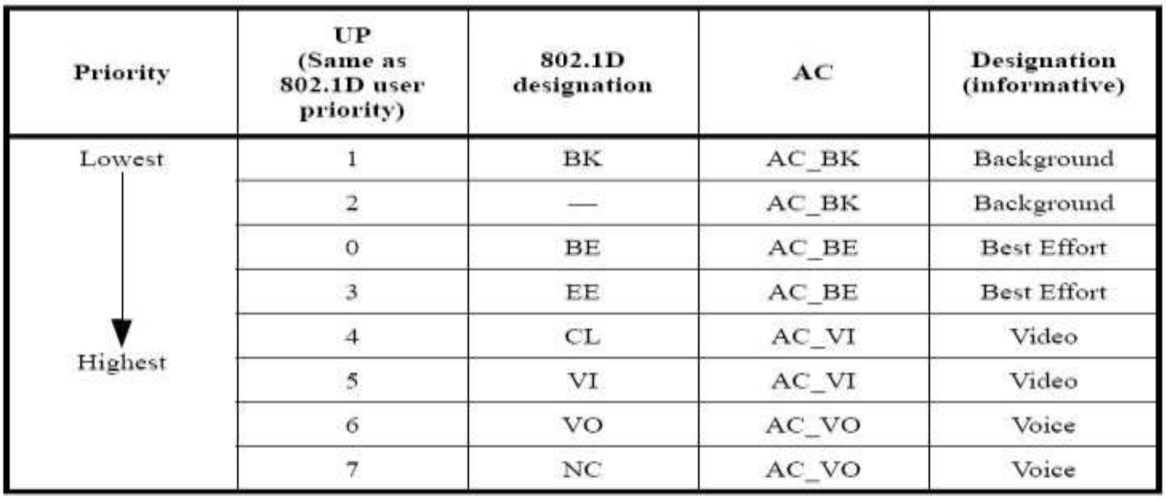
Hybrid Coordination Function (HCF)

The HCF coordination function combines and enhances the aspect of the contention-free and contention-based access methods, to provide QSTA with prioritized and parameterized access to the wireless medium[IEEE, 2005].

The HCF still supports non QoS contention for backward compatibility. In addition, HCF can work during CP and CFP, to meet QoS requirements. It supports two channel access mechanisms: Enhanced Distributed Channel Access (EDCA) for contention-based transfer; and HCF Controlled Channel Access (HCCA) for contention-free transfer. HCF uses a *hybrid controller* (HC) located in the QoS enhanced Access Point (QAP). HCF introduces the *controlled contention* (CC), which is a way for HC to know which stations need to be *polled*. CC occurs during the controlled access phase (CAP), and gives more guaranteed service than EDCF especially under heavy load.

HCF Contention-based Channel Access (EDCA)

EDCA is the HCF contention-based channel access. EDCA provides differentiated, distributed access to the medium using different priorities for different types of data traffic. EDCF provides differentiated distributed access to the wireless medium for QSTA using eight *user priorities* (UP) and four *Access Categories* (AC); as shown in Figure 4.5. EDCA is similar in function as EDCF, and it represents the contention based function supported by the HCF function.



Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
Highest	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

Figure 0.5: User Priorities and Access Categories of 802.11e. Src:[IEEE, 2005]

HCF Controlled Channel Access (HCCA)

This function uses the hybrid controller (HC), which allocates the transfer bandwidth, and is required for the parameterized QoS service. QAP allocates transmission

opportunities (TXOP) to stations contending to access the medium. A transmission opportunity (TXOP) is an interval of time when a QSTA has the right to use the wireless medium. TXOP is either obtained by the QAP, or by successfully contending for the channel in the contention part of the HCF.

Figure 4.6 shows a typical super frame of the IEEE 802.11e MAC when a QAP is used. Note that EDCF is used only during the contention period, while the HCF can be used in both the contention period and the contention free period.

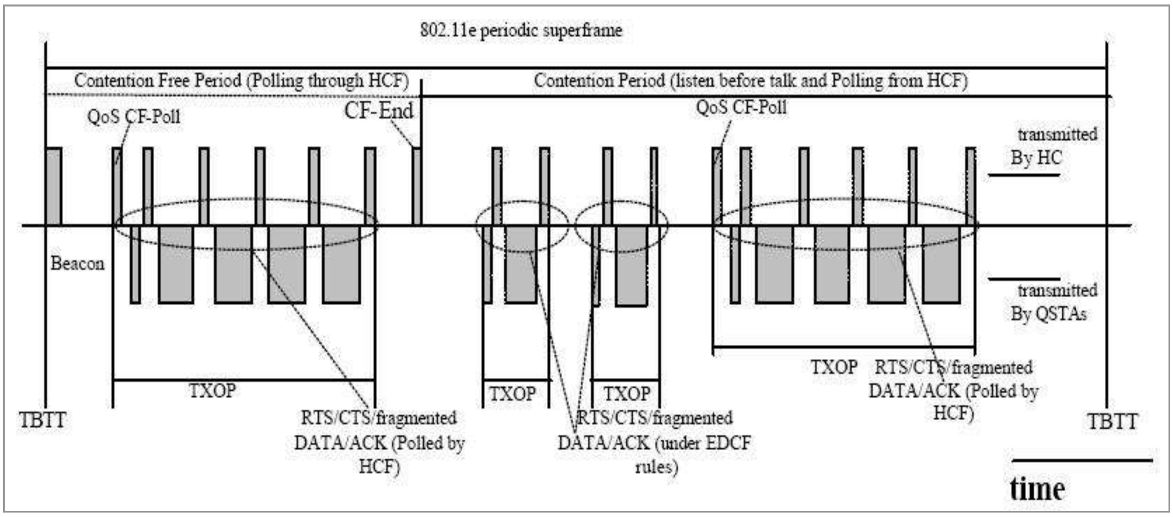


Figure 0.6: A typical frame of IEEE 802.11e MAC. Src: [IEEE, 2005]

A CLUSTER-BASED QoS FOR MOBILE AD HOC NETWORKS

In this chapter, we introduce a Cluster-Based QoS approach (CBQoS) for supporting QoS in mobile ad hoc networks. At first, we briefly review general information on the nature of MANET environment, clustering, and communications. Then, we illustrate the proposed approach by stating its idea, the reasons and justifications behind this idea, assumptions, architecture, and the design choices. In section 4.3, the implementation of the system is demonstrated.

1.20. Overview

A Mobile Ad Hoc Network consists of a collection of wireless devices that communicate with each other using shared wireless medium. The Wireless devices in ad hoc networks are called nodes. Each node in the network is required to be capable of forwarding packets (i.e. acting as a router), in addition to its role in sending and receiving data. Nodes maintain a specific Medium Access Control (MAC) function, to contend in accessing the shared wireless link.

By incorporating the IEEE 802.11 [IEEE, 1999] as the MAC function in MANET, nodes use the Distributed Coordination Function (DCF) function in the Contention Period (CP) to get a chance in accessing the medium. Two nodes can not use the wireless medium for transmission simultaneously, if this would lead to cancelling the transmission signals at any receiver. Data transmission between two nodes flows directly from the sender to the receiver, when they are both in the same transmission range. However, when the receiver is far away from the sender, the communication between these two parties requires cooperation from intermediate nodes, which cooperate to make a bridge between sender and receiver, and help in packets routing and forwarding.

In clustered MANET, nodes are grouped together in clusters making a hierarchical structure that improves routing, forwarding, and load balancing [Steenstrup, 2001]. Communications between nodes in clustered MANET can be classified into, *inter-cluster* and *intra-cluster*. In *Intra-cluster* communication (IA), the source and destination nodes both belong to the same cluster. Packets are usually sent from source node to the *clusterhead* (CH), which forwards the packet to the destination node (in some clustering algorithms, direct communication between *member-nodes* in a cluster is allowed under the control of the cluster-head). However, in *Inter-cluster* communication (IE), the source and destination nodes are in different clusters, and the packets are sent from the source node to its CH which forwards the packet to the corresponding node's cluster. Packets may pass through multiple hops and clusters until reaching the corresponding CH, which delivers packets to the destination.

1.21. The Proposed Approach

We propose a Cluster-Base QoS approach (CBQoS) that provides MANET with *inter-cluster/intra-cluster* service differentiation, and aims to improve the overall performance of clustered MANET, by increasing the overall network throughput and decreasing the overall delay encountered by MANET's applications.

1.21.1. The Basic Idea

Packet-switched networks utilize routers for supporting multi-hop data transmission. Routers receive packets, buffer them in the forwarding queues, and forward them to the next hop according to a specific scheduling mechanism. When QoS is not provided in a specific router, all packets are treated in a First-In-First-Out (FIFO) forwarding

mechanism. In high contention periods, the router's buffers get full, so incoming packets -meant to be forwarded by this router- have no place in the forwarding queues. These packets, therefore, are dropped and must be retransmitted (if their application requires so). MANET nodes function as routers to facilitate transmission between nodes that can not directly reach each other. Intermediate nodes for a specific transmission flow in MANET act as routers for the packets of this flow. Multiple flows may pass through the same intermediate node contending for the same buffer space and transmission bandwidth.

The Idea of CBQoS is as follows: to provide a cluster-based service differentiation, the forwarding module of intermediate nodes (routers) classifies packets into *inter-cluster* packets (IE) and *intra-cluster* packets (IA). The router provides better treatment (higher forwarding priority and lower dropping probability) for *IE* packets than *IA* packets. The goal of the CBQoS is to improve the overall network performance, allowing for better service and broader range of usable applications on MANET.

1.21.2. Justifications

In Figure 5.1, two data flows are depicted. One is the flow between nodes 4 and 8. Node 4 is the source, and node 8 is the destination of the data packets. The other flow is between nodes 11 (source) and 7 (destination). Packets in the first flow take the path 4→2→3→5→6→8; while the other flow has the path 11→6→7.

Considering the flow 4→8 (name it flow A), packets are initiated in the source node (4), forwarded to the clusterhead (CH 2), which forwards them to the gateway node (3) up to node (5) which in turn forwards them to the clusterhead (CH 6). CH 6, finally, delivers packets to their destination (node 8). In the second flow 11→7 (flow B), packets are initiated in the source (node 11), sent to the clusterhead (CH 6) which delivers them to the destination (node 7). CH 6 is a shared router between the two flows; it is considered the bottleneck in congestion situations.

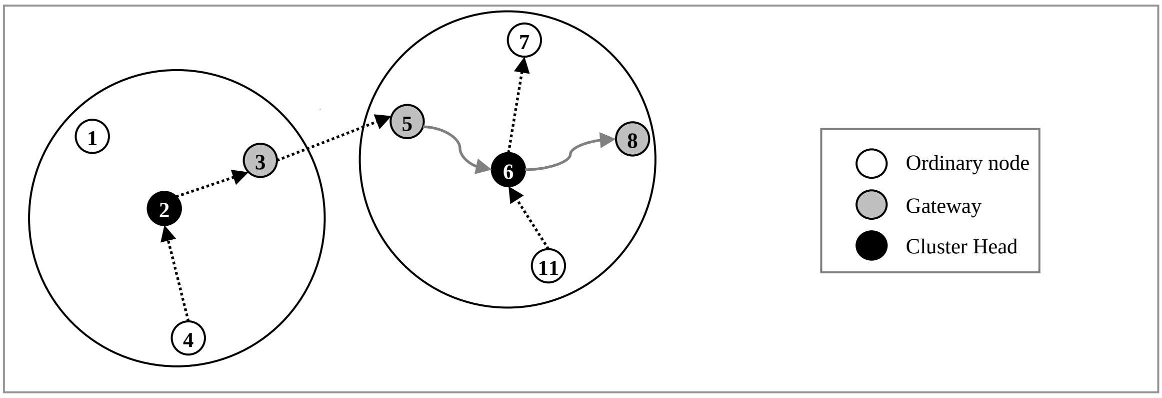


Figure 0.1: A Clustered MANET with two traffic flows: A (4→8) and B (11→7)

Suppose that a congestion case occurs at the node 6. Packets of both flows (A and B) contend for the buffering space and transmission time. When the buffers are full, any incoming packet belonging to either flow is dropped. Hence, packets of both traffic flows have the same forwarding chance, and the same dropping probability. Observe that, at node 6, packets of flow A have travelled 4 hops from their source (these hops are 4→2, 2→3, 3→5, 5→6); while packets of flow B have travelled only one hop (11→6). In the case of dropping a flow A's packet, the network wastes the effort of 4 hops of successful transmission (the cost includes: queuing delay, transmission delay, and wireless channel access at each hop). Moreover, the *retransmission* of this packet requires another 4 hops until reaching this point. On the other hand, dropping a flow B's packet costs the network merely the effort of one hop transmission and one hop of retransmission.

Another aspect of the congestion is that IA flows may *starve* IE packets. The *starvation* phenomenon occurs in the scenario of Figure 5.1, as a result of subsequently dropping flow A's (IE) packets at node 6, which leads the application of flow A to assume that the network is congested, and call its *congestion control* mechanisms either by deferring the transmission of subsequent packets and then try to transmit again through the 4 hops up to node 6 (in TCP based applications), or by using some sort of application layer adaptation (in adaptive real-time applications). During the period when flow A is busy with its congestion control, flow B can recover faster (because the retransmission requires only one hop) and exploit any space in node 6's buffer. If flow B continues transmission, it is likely that IE packets will find the buffer full again. In this situation, it is said that flow B is *starving* flow A. This results in deficiency in the overall network performance.

The current QoS approaches deals with packets equally, if they belong to the same flow (in per-flow QoS models) or to the same class (in per-class QoS models), regardless of how many hops has a packet travelled.

Based on the above observations, we supposed that prioritizing *inter-cluster* (IE) packets over *intra-cluster* (IA) packets will lead to improving the overall performance of the network. This is the idea behind our proposed CBQoS approach. With CBQoS, packets in their originating cluster are dealt with as IA until they leave to another cluster, where they are dealt as IE. In the case of the scenario shown in Figure 5.1, flow A's packets are dealt as IA in the first three hops (4→2, 2→3, and 3→5) and IE in (5→6 and 6→8).

Whereas, flow B's packets are dealt as IA in the whole path (11→6 and 6→7). At node 6, where congestion is more likely, IE packets (flow A's packets) are prioritized over IA packets (flow B's packets).

CBQoS aims to improve the network performance by reducing the probability of discarding or dropping inter-cluster (IE) packets. This is because dropping a packet that has travelled across many clusters results in degrading the performance more than that of dropping an intra-cluster (IA) packet which can be treated with lower cost of buffer-space and transmission bandwidth. It gives higher priority to packets that have travelled longer across network clusters. This approach decreases the delay encountered by inter-cluster traffic (which is usually high), and eliminates intra-cluster packets from starving inter-cluster ones. CBQoS provides nodes in MANET with traffic classification and queue management mechanisms.

Due to the cluster-based architecture of the network, the mobility will not have obvious effect on CBQoS. However, the mobility may add to the advantages of the proposed approach, as can be noticed from different mobility scenarios. One scenario is when a node, that is involved in transmission (as an intermediate node), leaves its own cluster and joins another cluster (*Handoff* occurs). A process of re-clustering and/or route maintenance may occur according to the clustering algorithm. We are interested in the packets that may have been buffered in the node before it leaves its cluster. If this node had no packets buffered in its queues, there is no need to *handover* (transferring state information between the new and old clusterheads) any information relevant to the QoS, because the proposed approach does not install any state information in nodes. If there have been some packets buffered in the leaving node, it might be necessary to *handover* these packets from the new cluster to the old one. Without a cluster-based service differentiation, these packets will be treated equally as other packet. However, CBQoS provides a better service to these packets and treats them as IE packets. This is an advantage of the proposed approach because it tries to shorten the time of the *handover* operation which is considered time-consuming.

Another scenario regarding nodes mobility is when the end points of a transmission move closer towards each other. Suppose that the sender node left its current cluster to the next cluster which was bridging the transmission between the sender and receiver. The sender affiliates itself to the new cluster, and there is no need to a handover operation regarding the QoS. Considering the packets that have been buffered in the old cluster (in the clusterhead and potentially in a gateway node) before the sending node leaves, these packets will arrive at the new cluster and dealt with as IE packets, whereas new packets originated from the sender will be treated as IA in this cluster. This will be advantageous to the transmission especially for TCP based transmissions, since old packets will have the chance to arrive earlier to the receiver.

One final scenario on nodes mobility can be thought of; it is the case when the end points of a transmission move farther from each other. Assuming that the sender is moving backward to another cluster, the routine procedure of handoff occurs and probably re-clustering and/or routing.

Consider the packets that have been buffered in the old cluster and waiting for transmission. These packets are dealt as IA in the old cluster, whereas new originated packets are dealt as IE packets in that cluster. Some of those packets will be forwarded during the handoff process. Another number of those packets will have the chance for transmission, while the new originated packets are waiting for transmission as IA packets in the new cluster. The rest of those packets, if any, will suffer some contention from the new originated packets.

1.21.3. Assumptions

CBQoS assumes clustered MANET. The network should be incorporated with a clustering algorithm that group nodes into clusters and identifies clusterheads, gateways, and ordinary nodes. Each node in the clustered network should be aware of some clustering-related information like: the nodes status (its role in the cluster; i.e. either it is a clusterhead, gateway, or ordinary node), and the cluster-range (cluster-radius).

Each node in the network is assumed to be capable to send/receive packets, forward packets, participate in the clustering processes, and act as clusterhead or gateway if elected for this role. Each node is also assumed to have bidirectional radio channel with the same radio transmission bandwidth for both transmission and receiving. All nodes in the network are assumed to have the same transmission range.

CBQoS assumes a time-sharing mechanism for the medium access control. IEEE 802.11 MAC is used in the implementation of the approach. Finally, the network layer module in every node in MANET must be amended with the CBQoS implementation. In this thesis we concentrate on providing CBQoS on the *Network* layer. However, this approach can also be further investigated for the MAC layer.

1.21.4. Architecture

The components of the CBQoS approach include: *Traffic Classifier* that classifies packets either to inter-cluster (IE) or intra-cluster (IA), *Packet Scheduler* that schedules packets for transmission providing higher priority to IE packets, and *Packet Dropper* that is supposed to use some dropping policy in case of congestion. These components are shown in Figure 5.2.

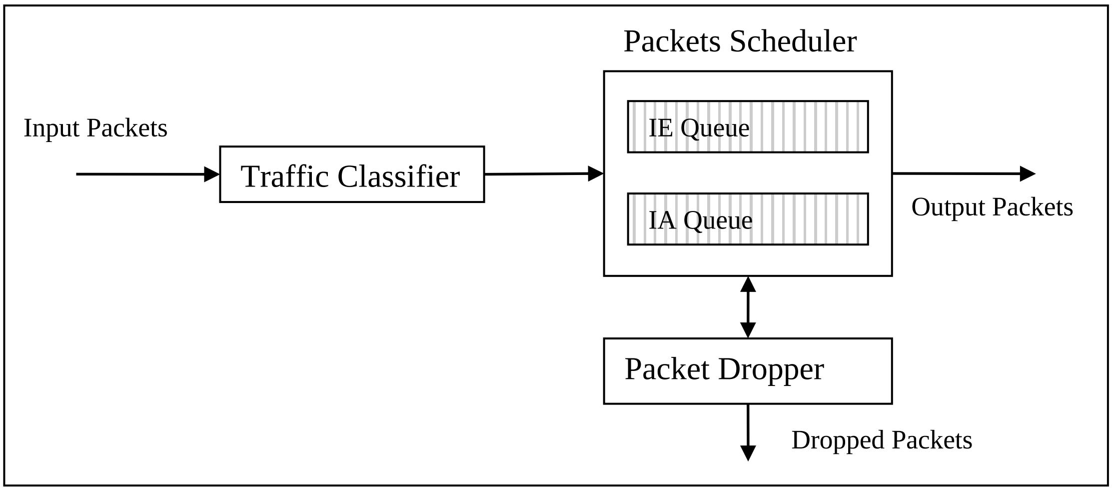


Figure 0.2: Traffic Classification/Scheduling in CBQoS.

Traffic Classification

Traffic is classified into two classes, namely: inter-cluster (IE) traffic and intra-cluster (IE) traffic, where IE traffic gets higher priority. Packets are given IA forwarding behaviour in their initiating clusters and IE forwarding behaviour (higher priority) in other clusters. Traffic classification can be done based on two values: the packet's IP TTL field, and the cluster-range (cluster radius), which is determined by the clustering algorithm and assumed to be known to all nodes in the network.

When a packet is generated, it is classified in its source node as IA packet. Once this packet arrives at another node, its IP TTL field is checked to know how many hops has the packet travelled; and based on the TTL value together with the cluster-range value, the packet is classified as IE or IA and queued accordingly.

Packet Scheduling

After a packet gets classified into either IE or IA, it is queued in the relevant queue. Queue management can be achieved as follows: two queuing priorities (for IE and IA packets) are implemented. IE packets are given higher priority than IA ones.

Different Queuing disciplines [Semeria, 2001]; such as, Priority Queuing, Weighted Fair Queuing, or Class-Based Queuing; can be adopted to implement the CBQoS.

Packet Dropping

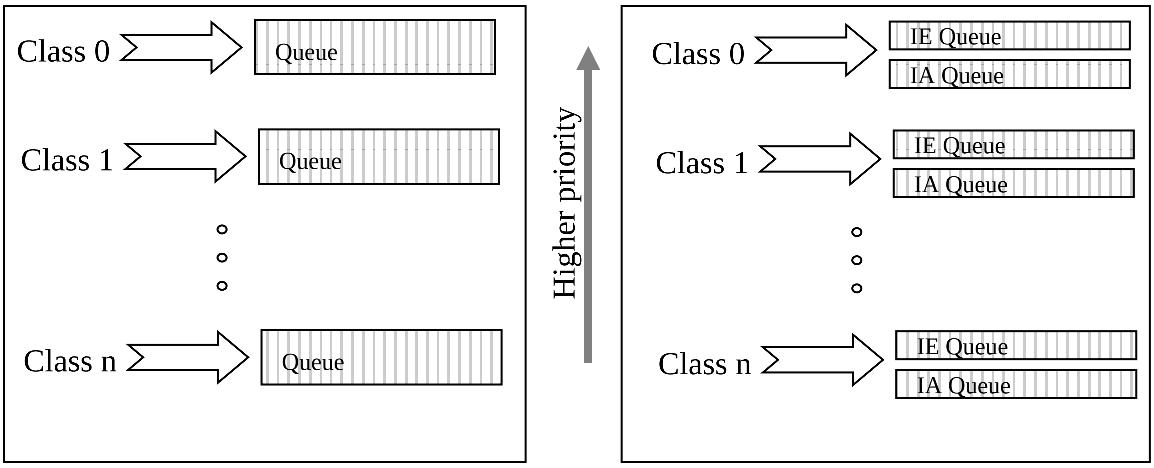
Another aspect of the queue management policy is the *dropping policy* which can be implemented to deal with congestion. The dropping policy can be simple by dropping input packets when the buffers are full. A complex dropping policy may require running statistics and complex measurements. A good dropping policy, for CBQoS, is the one that provides IE packets with dropping probability without drastically starving IA packets.

Random Early Detection (RED) [Floyd et al., 1993] can be used as a dropping policy, with IE packets having less drop probability than IA ones. RED is a congestion control mechanism that monitors different queues; each queue is given a specific average length and dropping probability. When a queue exceeds a specific threshold, RED drops its packets with a certain dropping probability.

1.21.5. Design Choices

CBQoS can be designed as a standalone service provisioning approach, as it can also be applied over classical Differentiated Services (DiffServ) [Blake et al., 1998], without any extra header field in IP packets, and without affecting the concept of DiffServ.

When CBQoS is implemented over DiffServ, packets are firstly classified into forwarding classes using DiffServ, and then CBQoS is applied within each forwarding class as shown in Figure 5.3.



a) Original DiffServ Forwarding Classes

b) DiffServ Forwarding Classes with Cluster-based QoS

Figure 0.3: Implementing CBQoS over Differentiated Services.

1.22. Implementation

CBQoS has been implemented using GLOMOSIM. Packets classification considered two classes of packets, IE and IA. Priority queuing has been adopted for packet scheduling, with IE packets given the higher priority. The RED dropping policy has been used, with higher dropping priority for IA packets. The implementation environment and parameters is illustrated next, with an overview to the GLOMOSIM network simulator, which was used to implement the system. Then, we introduce the QoS metrics that were considered in the proposed approach.

1.22.1. Simulation Environment

The Global Mobile Information System Simulator (GLOMOSIM) [Bajaj et al., 1999] was used to implement the proposed approach. GLOMOSIM is a library-based sequential and parallel simulator for wireless networks. It is designed as a set of library modules, each of which simulates a specific wireless communication protocol in the protocol stack [Zeng et al., 1998].

GLOMOSIM is a scalable simulation environment that uses parallel execution to reduce the simulation time [Bajaj et al., 1999], which is important to this study. GLOMOSIM was built based on the PARSEC (Parallel Simulation Environment for Complex Systems) [Bagrodia et al., 1998] simulation language. GLOMOSIM has a layered approach similar to the OSI model; It provides different models for the Physical layer (PHY), Data Link layer (MAC), Network layer (Routing), Transport layer, and Application layer.

The general environment parameters are as follows:

- *Physical layer*: Two-Ray propagation *pathloss* model, the ISM Radio Frequency band of 2.4GHz, 2MHz bandwidth, and a transmission range of 376 meters for each node.

- *MAC layer*: The IEEE 802.11 medium access control protocol.
- *Network Layer*: The IP was used as the network layer protocol.

Some parameters, such as the simulation seed and the simulation time, have been given different values to check the consistency in the results. Table 5.1 summarizes the simulation environment parameters that were used.

Table 0.1: Summary of Simulation Parameters

Simulation Time	5 and 15 (minutes)
Propagation Pathloss	Two-Ray
Radio Frequency	2.4e9
Radio Bandwidth	2MHz
Transmission Range	376 meters
MAC Protocol	IEEE 802.11
Network Protocol	IP

Each simulation experiment was run considering *two* cases:

- *QoS OFF*: Which is the traditional case, where CBQoS is not use. The simulation was run *ten* times for each simulation experiment, each with different simulation seed. No traffic differentiation was provided.

- *QoS ON*: Where CBQoS is used. The simulation was also run ten times for each simulation experiment, each with different simulation seeds.

1.22.2. Performance Metrics

The main performance metrics in this study are the overall network *throughput* and *delay*, as defined in section 3.4. These metrics reflect the effect of the proposed QoS approach on the network performance. “Bandwidth and latency (Delay) combine to define the performance characteristics of a given link or channel” [Peterson et al., 2001].

A third metric, the *network power*, differentiated from both the *throughput* and *delay,* is also used. The Network Power (NP) is a network performance metric which reflects the overall performance of the network. Network power is computed using the formula [Mankin et al., 1991]:

$NP=\frac{Throughput^{alpah}}{Delay}$, where *alpha* is chosen based on the relative importance of *throughput* versus *delay*. Value of *alpha*=1 is used when *throughput* and *delay* are of equally importance. If *delay* is more important than *throughput*, then *alpha* should be chosen smaller than one. We use the ratio of the overall network *throughput* to the overall network *delay* with *alpha=1* to represent the *Network Power*.

We considered these metrics to evaluate the proposed QoS approach and suggest situations in which it is best suitable. In chapter 6 the simulation results are shown and analysed

RESULTS AND DISCUSSIONS

In this Chapter, we evaluate and analyze the performance of the proposed CBQoS approach on clustered MANET, through simulation. Two simulation scenarios are used; each scenario is experimented under different simulation parameters, different traffic loads, and different types of traffic. The results of the simulation are shown and discussed. The results are organized into two scenarios: scenario 1 in section 6.1, and scenario 2 in section 6.2. Scenario 1 considers two types of network traffic: Real-Time traffic with three different experiments (section 6.1.1), and Non Real-Time traffic (section 6.1.2). Scenario 2 presents the results of three types of network traffic: Real-Time traffic (section 6.2.1), Non Real-Time traffic (section 6.2.2), and Hybrid RT/NRT traffic (section 6.2.3). This chapter concludes by an overall result discussion in section 6.3.

The results in this chapter are shown in figures, and brief tables that show the results in percentage format. The detailed results of each simulation experiment are provided in Appendix A, and can be referenced as needed.

The following terminologies are used through this chapter:

- *IE*: IntEr-cluster traffic.
- *IA*: IntrA-cluster traffic.
- *Overall Throughput*: the summation of throughput for all flows in the network.
- *Overall Delay*: the average of the end-to-end delay encountered by all flows in the network.

- *NP:* Network Power; a network performance metric that measures the throughput to delay ratio, as defined in section 5.3.2.

1.23. Scenario 1

The scenario shown in Figure 6.1 is used in the simulation. This scenario represents a clustered MANET that occupies a terrain of (2500*2500) meters, and consists of 15 nodes grouped in 3 clusters, with nodes 1, 6, and 9 as the three clusterheads (CH).

A disjoint clustering method is used, with the pair (4,5) and (8,10) as the distributed gateways (DG). Each member node in a cluster can communicate with others via its own clusterhead. Gateway nodes provide connections between neighbouring clusters.

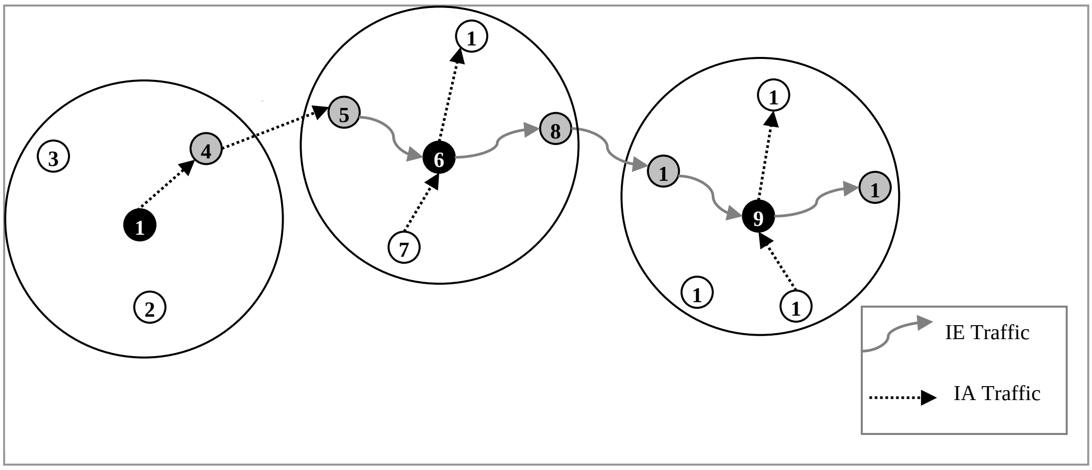


Figure 0.1: Scenario 1

The simulation considers Real-Time traffic (RT), and Non Real-Time traffic (NRT). The specific parameters for each are discussed next.

1.23.1. Real Time Traffic

The network *traffic* used in the simulation is Constant Bit Rate (CBR). CBR uses the UDP (User Datagram Protocol) for the transport layer. UDP is a connection-less transport protocol, and is usually used with real-time multimedia applications that can tolerate loss in data. The CBR *packet size* used in the simulation is 1000 bytes. The IEEE 802.11 MAC supports a maximum packet size of 2312bits. However, large packets are preferred to be fragmented into smaller ones, to confront the high bit error rate of the wireless medium. Even though transmitting large packets reduces the frequency of the MAC Inter-Frame Spacing (IFS) and seems to increase the network capacity, a large packet is more vulnerable to errors due to the high Bit-Error-Rate (BER) wireless medium. On the other hand, using very small packets can reduce the probability of bit errors, and thus reduces retransmissions, but it wastes a lot of time in the *idle* state during the IFS times. For this reason, a moderate size of packets is used in the simulation.

Two types of traffic are simulated:

- *IE traffic:* which is represented by the flow between nodes (1) and (14) in Figure 6.1.
- *IA traffic*: Two IA flows in two different clusters appear in Figure 6.1. Node (7) sends to node (12) in the first, and node (11) sends to node (13) in the second.

Different *traffic loads* on the network are used in the simulation, to study the effect of applying CBQoS in each load. The simulation considers three different traffic loads of IE traffic (200Kbps, 400Kbps, and 640 Kbps), and five different traffic loads of IA traffic (200Kbps, 400Kbps, 640Kbps, 800Kbps, and 1Mbps) as follows:

1- *Experiment 1***:** 200Kbps IE traffic, with (200Kbps, 400Kbps, 640Kbps, 800Kbps, and 1Mbps) IA traffic

2- *Experiment 2:* 400Kbps IE traffic, with (200Kbps, 400Kbps, 640Kbps, 800Kbps, and 1Mbps) IA traffic

3- *Experiment 3:* 640Kbps IE traffic, with (200Kbps, 400Kbps, 640Kbps, 800Kbps, and 1Mbps) IA traffic

The results of different experiments are provided, and discussed briefly as they appear. The overall result analysis is provided in section 6.3.

Experiment 1

This experiment uses 200Kbps of IE traffic, and five different traffic loads of IA traffic (200Kbps, 400Kbps, 640Kbps, 800Kbps, and 1Mbps). The results are shown for Throughput, Delay, and Network Power.

Throughput:

In Figures 6.2, the throughput of IE traffic is shown before and after using CBQoS. The results show that the IE throughput is improved when CBQoS is used.

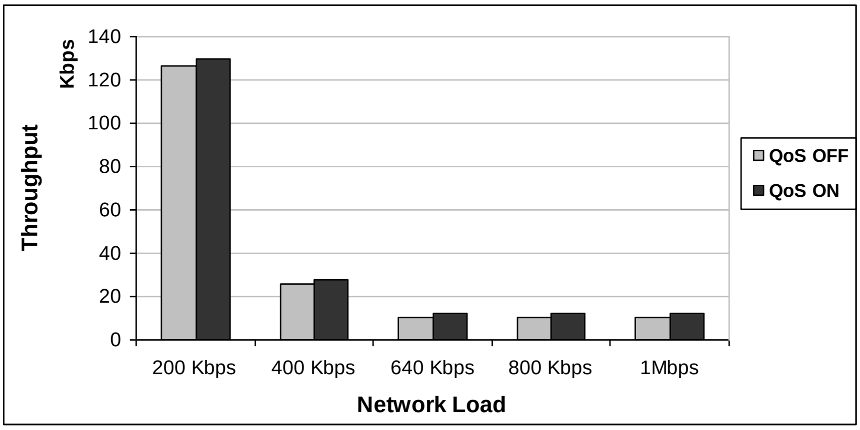


Figure 0.2: Throughput of IE traffic with/without CBQoS (200Kbps IE Traffic)

The throughput of IA traffic is shown in Figures 6.3. It is slightly increased when the CBQoS is ON (with an average of 2%), although IE traffic is getting priority over it. Without QoS, IE packets encounter higher contention and waits longer in the queue occupying a valuable queuing place which can be rather left to IA traffic. However, when CBQoS is ON, IE packets are prioritized for transmission; thus, freeing some space and allowing for IA transmission. This justifies the slight increase in IA throughput.

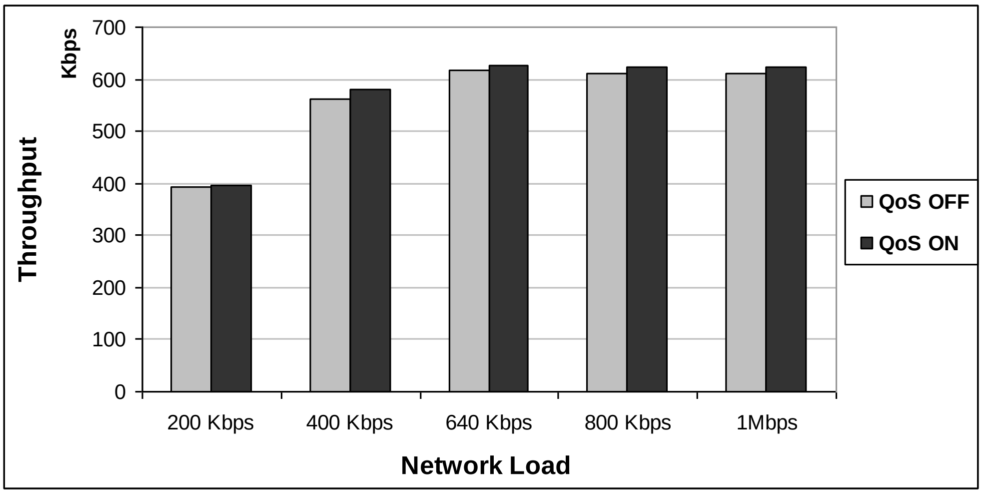


Figure 0.3: Throughput of IA traffic with/without CBQoS (200Kbps IE Traffic)

Figure 6.4, shows the overall network throughput before and after using CBQoS. It is obvious that CBQoS improves the overall network throughput in different network loads.

Table 6.1, which lists the percentage of throughput improvement using CBQoS compared to normal throughput, shows that IE gains higher throughput using CBQoS. This gain is due prioritizing IE traffic over IA traffic, such that IE packets are queued on the front and scheduled for transmission before IA packets.

Table 0.1: Throughput Improvement using the CBQoS (200Kbps IE Traffic)

Network Load	IE Throughput	IA Throughput	Overall Throughput
200 Kbps	+2.35%	+0.68%	+1.09%
400 Kbps	+9.64%	+3.14%	+3.42%
640 Kbps	+17.28%	+1.64%	+1.90%
800 Kbps	+21.06%	+2.19%	+2.50%
1Mbps	+17.53%	+2.25%	+2.50%
Average	**+13.57%**	**+1.98%**	**+2.28%**

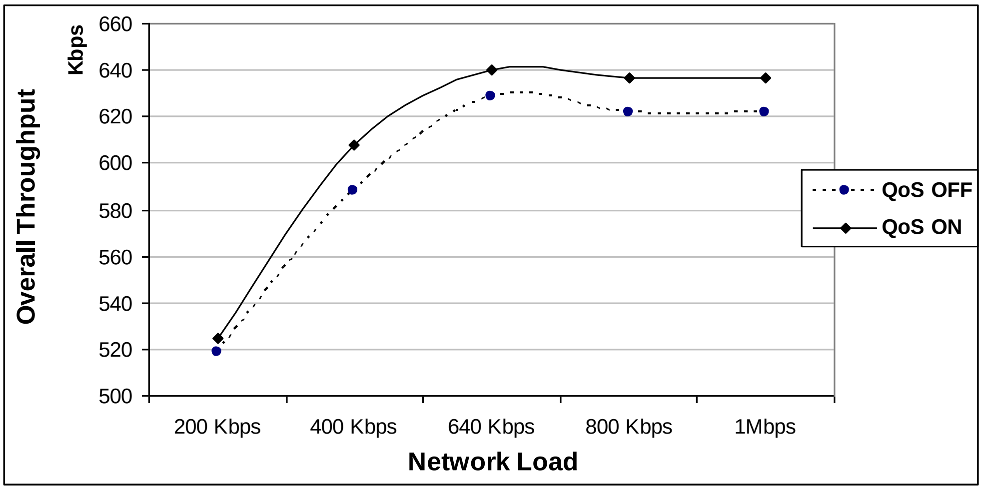


Figure 0.4: Overall Network Throughput with/without CBQoS (200Kbps IE Traffic)

As shown in Table 6.1, the gain in IE throughput starts relatively small (of 2.35%) when the network is lightly loaded (200kbps), and increases gradually with higher network loads.

Delay:

The average end-to-end delay for IE traffic is depicted in Figure 6.5. It shows that IE packets encounter lower delay with CBQoS.

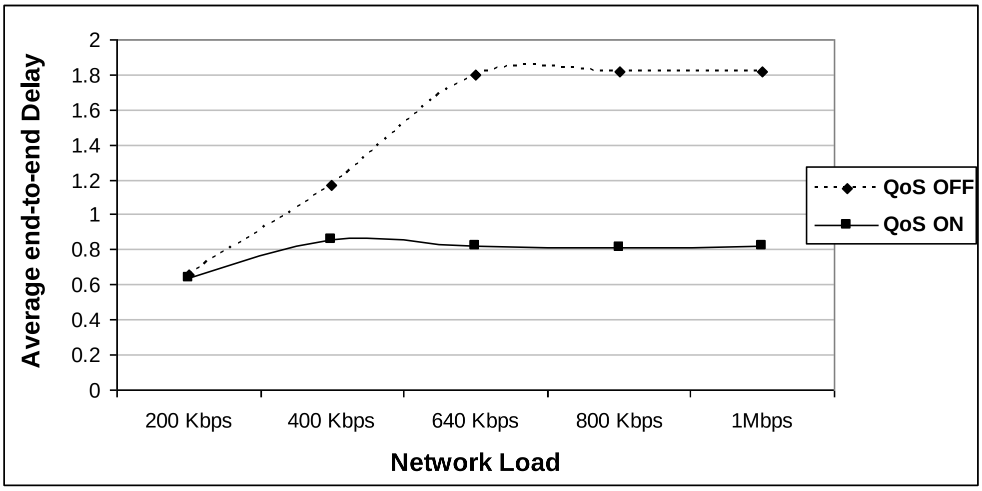


Figure 0.5: End-to-End Delay of IE traffic with/without CBQoS (200Kbps IE Traffic)

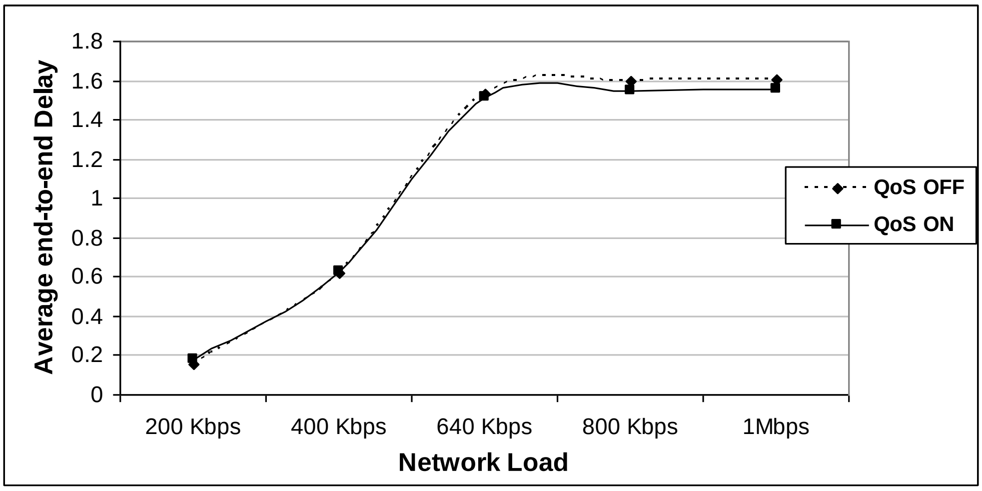


Figure 0.6: End-to-End Delay of IA traffic with/without CBQoS (200Kbps IE Traffic)

Figure 6.6, shows that the average IA end-to-end Delay is almost the same before and after using CBQoS. The overall end-to-end delay is shown in Figure 6.7. A reduction in the overall network delay is noticeable, especially with high IA traffic loads.

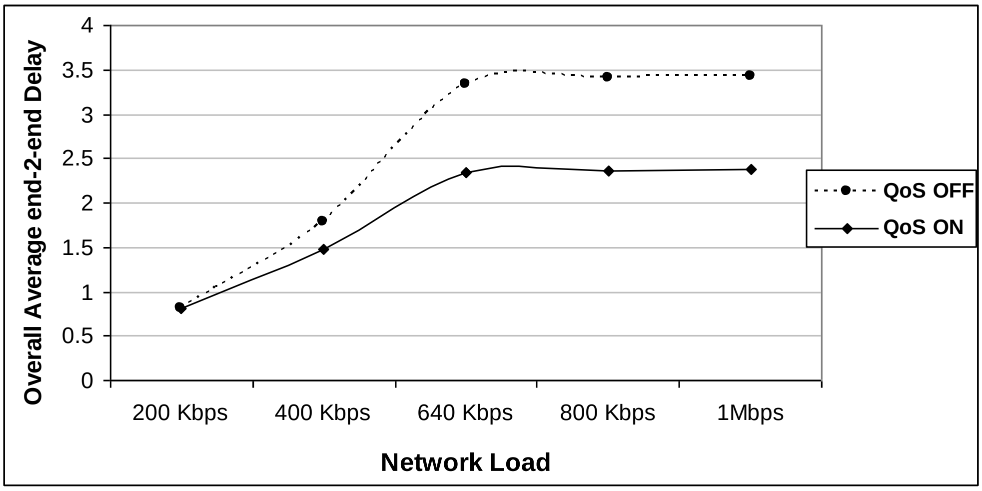


Figure 0.7: Overall End-to-End Delay with/without CBQoS (200Kbps IE Traffic)

Table 0.2: Delay Improvement using the CBQoS (200Kbps IE Traffic)

Network Load	IE Delay	IA Delay	Overall Delay
200 Kbps	-3.41%	+13.50%	-0.16%
400 Kbps	-26.43%	+2.03%	-16.59%
640 Kbps	-54.20%	-0.93%	-29.72%
800 Kbps	-54.95%	-3.21%	-30.71%
1Mbps	-54.79%	-3.23%	-30.58%
Average	**-38.75%**	**+1.63%**	**-21.55%**

Table 6.2 lists the percentage of improvement in the end-to-end delay using CBQoS.

Network Power

We use the ratio of the overall network throughput to the overall network delay to calculate the Network Power. Choosing alpha=1, and assuming that throughput and delay are equally important, the results show that CBQoS empowers the network with an average increase of (+33.13%), as listed in Table 6.3, and Figure 6.8.

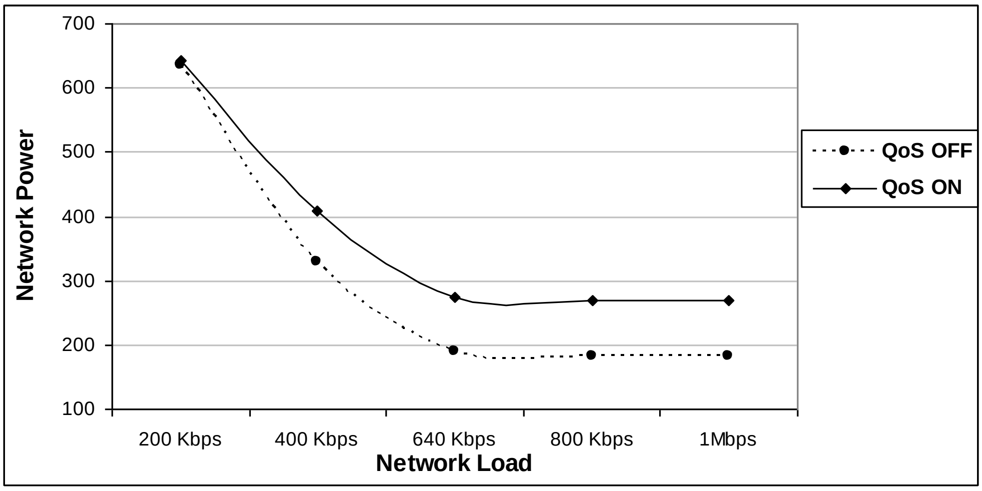


Figure 0.8: Network Power with/without CBQoS (200Kbps IE Traffic)

The results, according to Tables 6.1 and 6.2, show that CBQoS provides higher *network power* when the delay is of higher importance than throughput, because CBQoS contributes to the reduction of delay more than its contribution in increasing the throughput (see the equation of computing the *network power* in section 5.3.2). This conclusion makes it more suitable to apply CBQoS on MANET that are deployed for real time multimedia applications that require low delay, such as voice communications.

Table 6.3, illustrates the results, and shows that CBQoS achieves significant performance improvement in clustered MANET. This gain in performance is important to support QoS provisioning.

Table 0.3: Network Performance using the CBQoS (200Kbps IE Traffic)

Network Load	Network Power
200 Kbps	+1.25%
400 Kbps	+24.00%
640 Kbps	+44.97%
800 Kbps	+47.83%
1Mbps	+47.60%
Average	**+33.13%**

Experiment 2

In this experiment, IE traffic of 400Kbps is used, with different IA traffic loads.

Throughput:

Figure 6.9 shows the IE throughput. With light traffic load (200Kbps), the improvement in IE traffic is small (0.46%). This is because the IE traffic does not suffer high contention even when CBQoS is OFF.

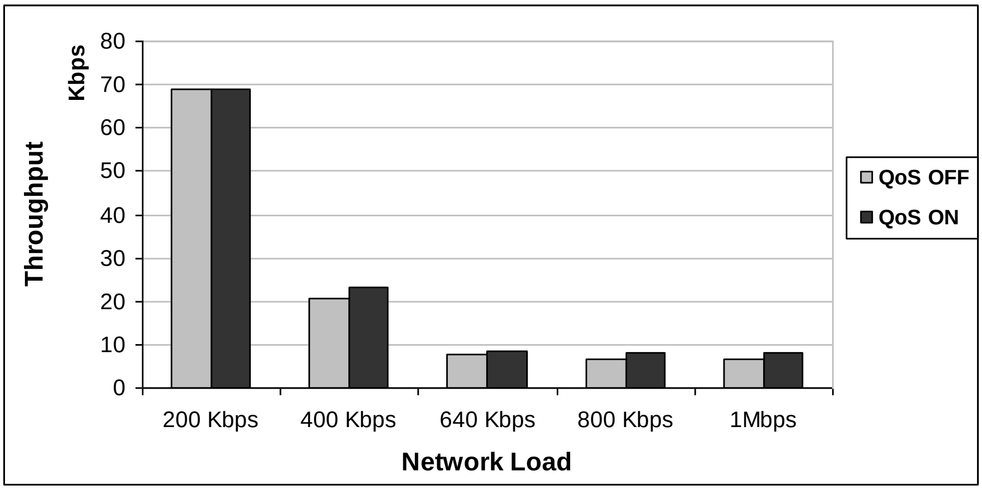


Figure 0.9: Throughput of IE traffic with/without CBQoS (400Kbps IE Traffic)

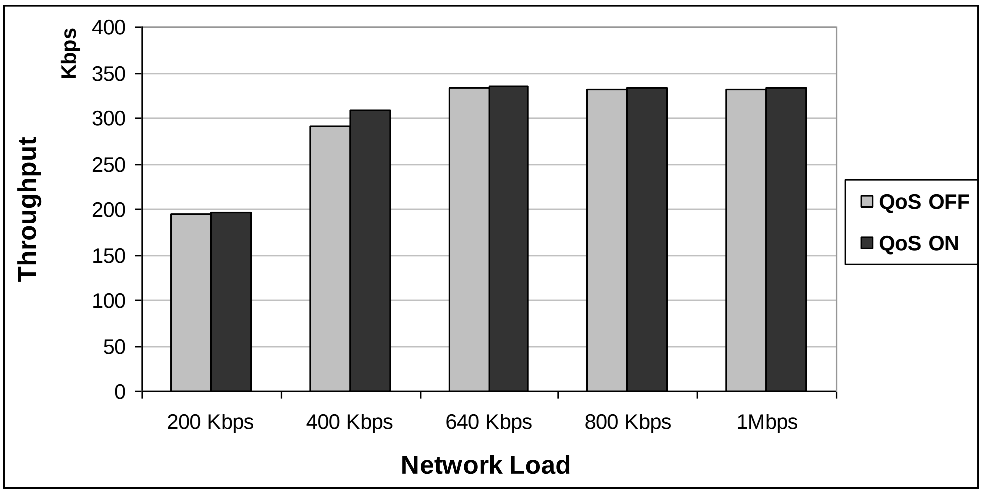


Figure 0.10: Throughput of IA traffic with/without CBQoS (400Kbps IE Traffic)

Figure 6.10 shows the IA throughput, and Figure 6.11 shows the overall network throughput.

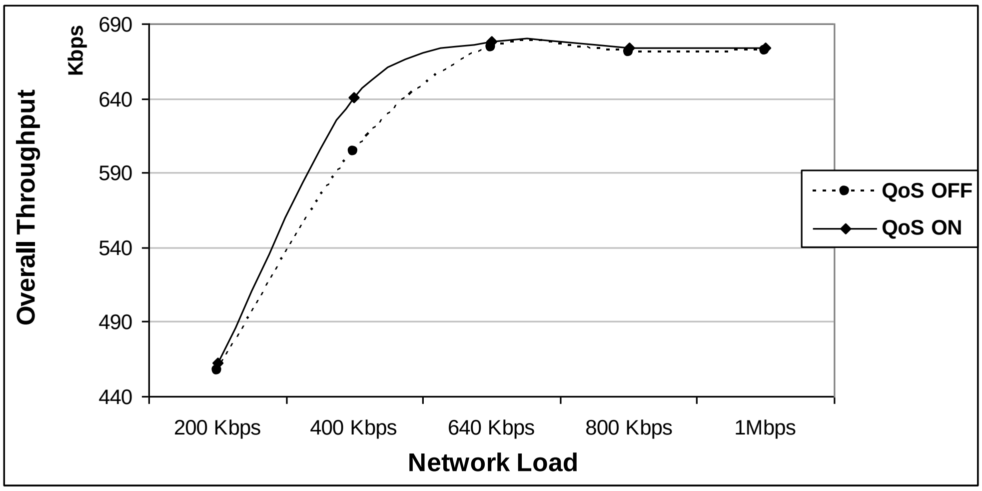


Figure 0.11: Overall Network Throughput with/without CBQoS (400Kbps IE Traffic)

According to Table 6.4, CBQoS improves the IE throughput with an average of (13.4%).

The improvement in IA throughput is insignificant, especially when the network is highly loaded. However, the overall network throughput is improved with 1.7%. The detailed results can be referenced in Appendix A.

Table 0.4: Throughput Improvement using the CBQoS (400Kbps IE Traffic)

Network Load	IE Throughput	IA Throughput	Overall Throughput
200 Kbps	+0.46%	+1.29%	+1.16%
400 Kbps	+12.60%	+5.68%	+5.92%
640 Kbps	+11.76%	+0.46%	+0.59%
800 Kbps	+21.33%	+0.21%	+0.42%
1Mbps	+20.86%	+0.17%	+0.38%
Average	**+13.40%**	**+1.56%**	**+1.69%**

Delay:

The average end-to-end delay for IE traffic, and IA traffic, is depicted in Figure 6.12, and Figure 6.13, respectively.

The CBQoS improvement in IE delay starts small, when then network load is relatively low (with -4%). When the network is higher loaded, CBQoS introduces lower delays. The reason is that the contention increases, and the IE packets are delayed for long time, when CBQoS is OFF.

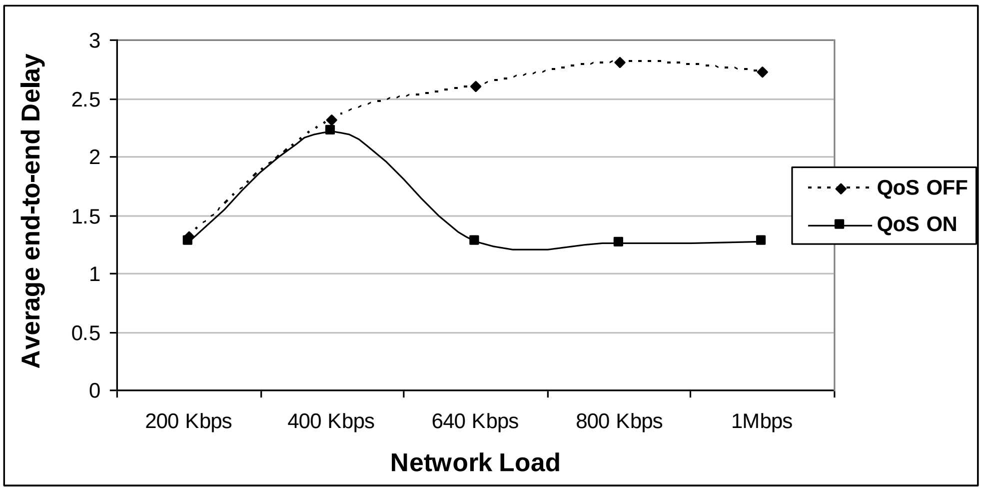


Figure 0.12: End-to-End Delay of IE traffic with/without CBQoS (400Kbps IE Traffic)

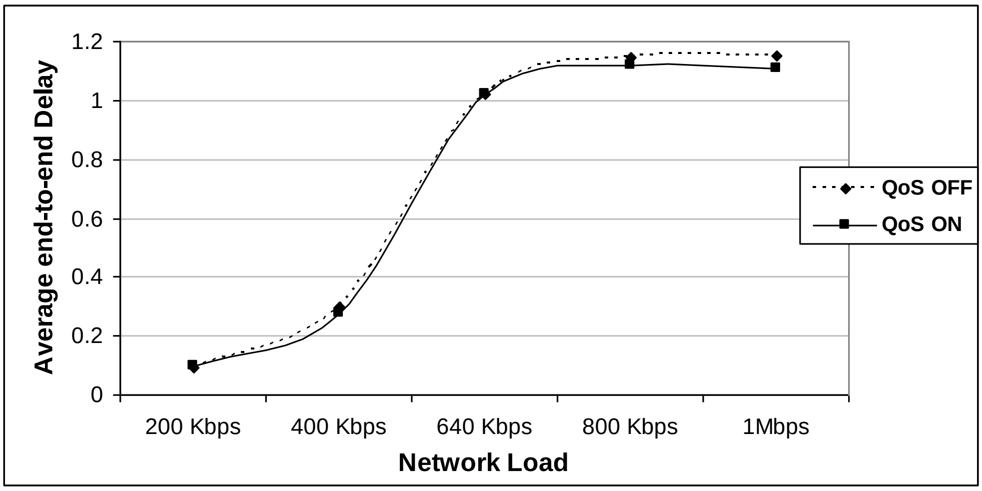


Figure 0.13: End-to-End Delay of IA traffic with/without CBQoS (400Kbps IE Traffic)

The average delay of IA traffic is almost the same before and after applying CBQoS. The high sending rate of IE traffic (400kbps) slightly increases the IA delay when the last is low (200Kbps). However, the overall results, shown in Table 6.5 and Figure 6.14, show that the IA traffic does not negatively affected by CBQoS.

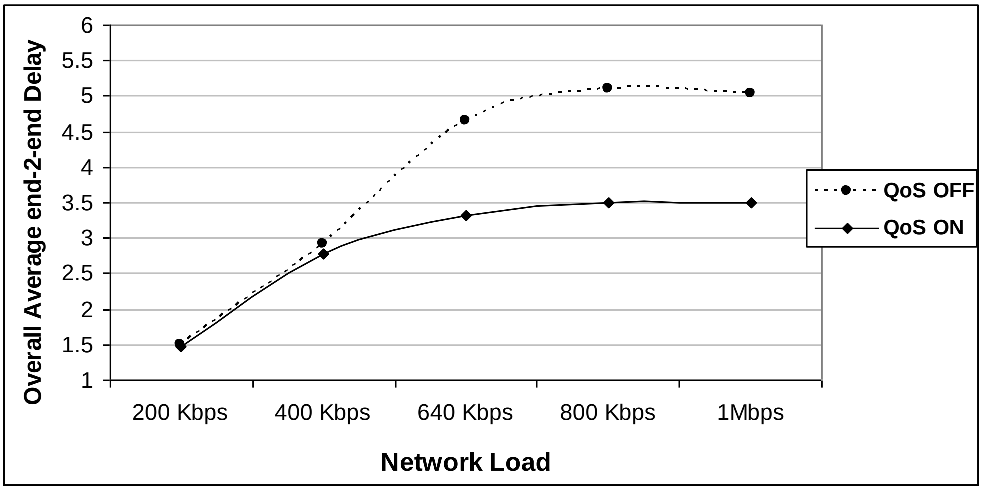


Figure 0.14: Overall End-to-End Delay with/without CBQoS (400Kbps IE Traffic)

The improvement in the overall end-to-end delay is shown in Table 6.5. IE end-to-end delay is surprisingly improved with an overall decrease of 33%. The IA traffic suffered longer delay when its sending rate is small (200Kbps), as expected, due to prioritizing IE traffic, but it recovers with higher sending rates.

Table 0.5: Delay Improvement using the CBQoS (400Kbps IE Traffic)

Network Load	IE Delay	IA Delay	Overall Delay
200 Kbps	-3.50%	+7.96%	-2.12%
400 Kbps	-4.21%	-6.27%	-4.63%
640 Kbps	-51.10%	+0.06%	-28.60%
800 Kbps	-54.79%	-2.52%	-31.25%
1Mbps	-53.08%	-3.57%	-30.45%
Average	**-33.34%**	**-0.87%**	**-19.41%**

Network Power:

The ratio of throughput to delay, network power, is shown in Figure 6.15. This figure shows that CBQoS empowers the network, with better improvement under high traffic loads.

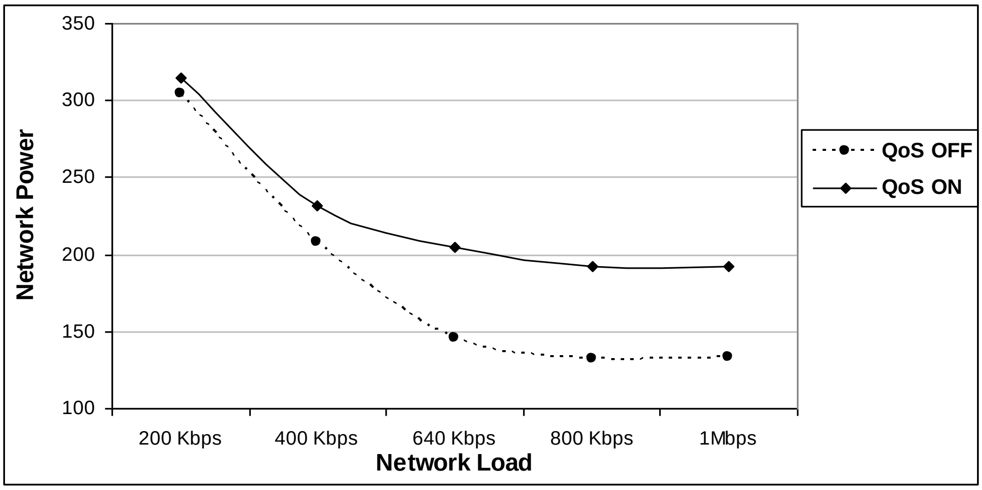


Figure 0.15: Network Power with/without CBQoS (400Kbps IE Traffic)

Table 0.5: Network Performance using the CBQoS (400Kbps IE Traffic)

Network Load	Network Power
200 Kbps	+3.36%
400 Kbps	+11.10%
640 Kbps	+40.73%
800 Kbps	+45.99%
1Mbps	+44.25%
Average	**+29.09%**

Experiment 3

As in the previous two experiments, this experiment use five different IA traffic loads. The sending rate of IE traffic in this experiment is 640Kbps.

Throughput:

Figure 6.16 shows the IE throughput, Figure 6.17 shows the IA throughput, and Figure 6.18 shows the overall network throughput.

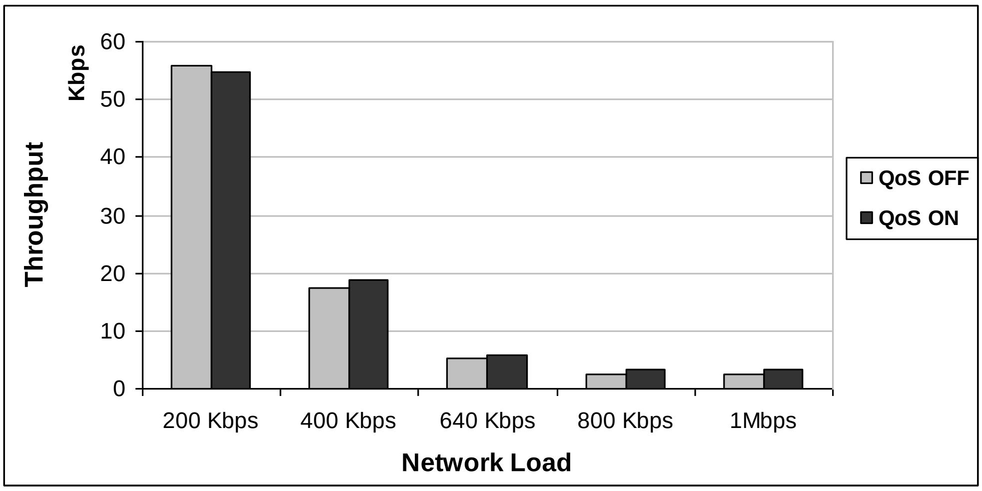


Figure 0.16: Throughput of IE traffic with/without CBQoS (640Kbps IE Traffic)

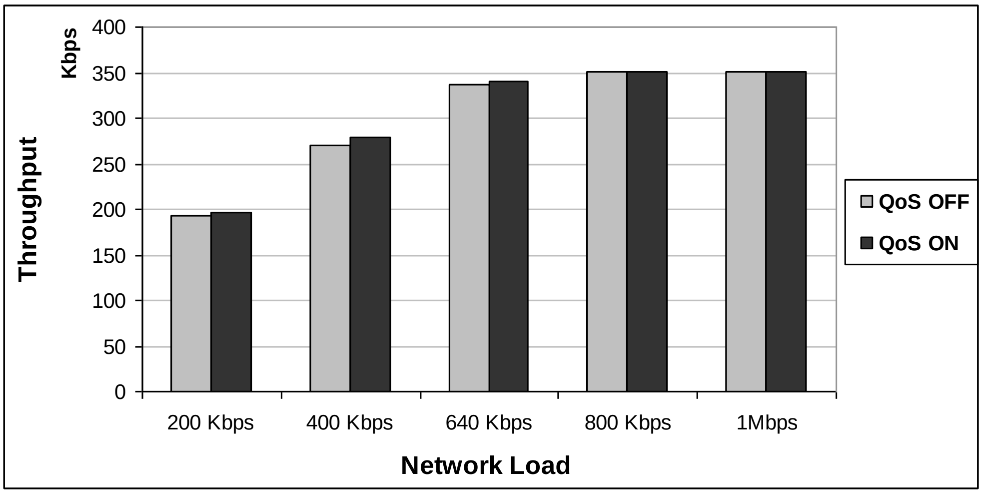


Figure 0.17: Throughput of IA traffic with/without CBQoS (640Kbps IE Traffic)

Table 6.7 summarizes the results. The throughput of IE traffic is improved, with and average of 14.7%. However, when the IA traffic is low (200kbps), the IE traffic is decreased with -2.25%. This result is due to the high ratio of IE packets to the overall number of packets (IE+IA) in some intermediate nodes. In this case, IA packets are of low rate, and are happy with their small share of the queuing space (IA throughput is increased), while IE packets suffer higher contention due to their high sending rate.

However, with higher loads of IA traffic, the IE packets suffer when CBQoS is OFF; whereas, they get higher throughput with CBQoS. The overall results are further discussed in section 6.3, and listed in Table 6.6.

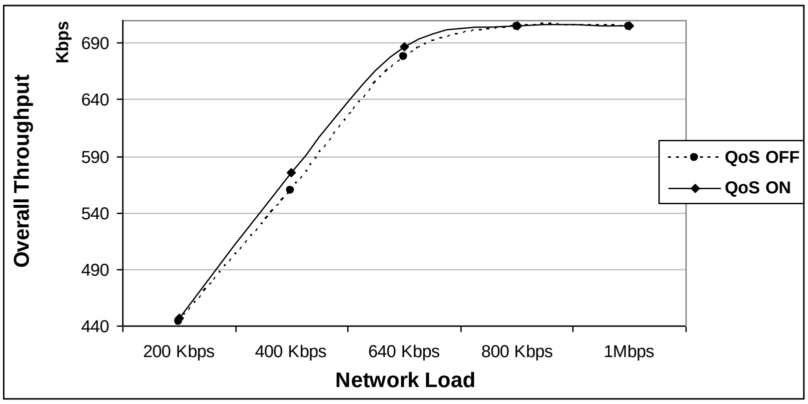


Figure 0.18: Overall Network Throughput with/without CBQoS (640Kbps IE Traffic)

Table 0.6: Throughput Improvement using the CBQoS (640Kbps IE Traffic)

Network Load	IE Throughput	IA Throughput	Overall Throughput
200 Kbps	-2.25%	+1.21%	+0.77%
400 Kbps	+8.59%	+2.87%	+3.05%
640 Kbps	+13.28%	+1.17%	+1.26%
800 Kbps	+30.05%	+0.02%	+0.12%
1Mbps	+23.65%	+0.13%	+0.22%
Average	**+14.67%**	**+1.08%**	**+1.08%**

Delay:

Figures 6.19 and 6.20, show the average end-to-end delay for IE traffic, and IA traffic, respectively. The results are quite similar to the results of experiment 1 (previously discussed in this section). The QoS improvement is more obvious on higher traffic loads.

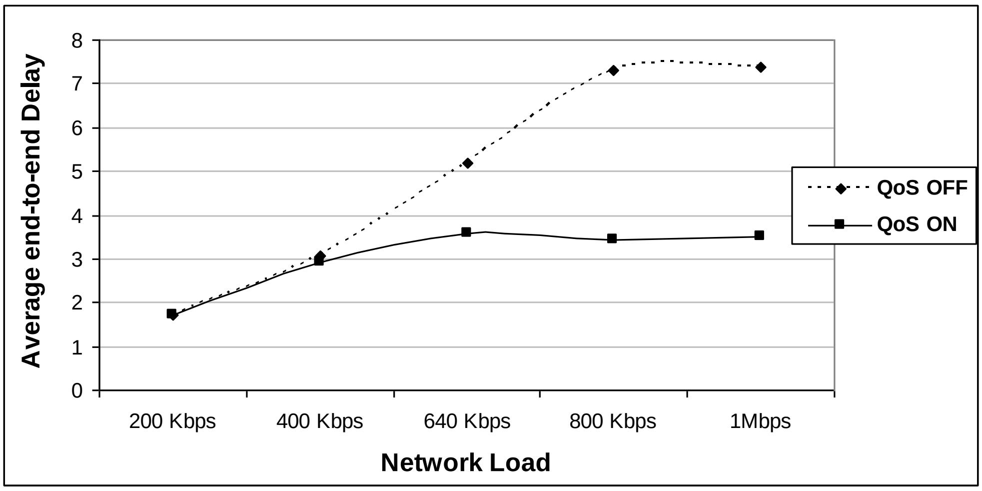


Figure 0.19: End-to-End Delay of IE traffic with/without CBQoS (640Kbps IE Traffic)

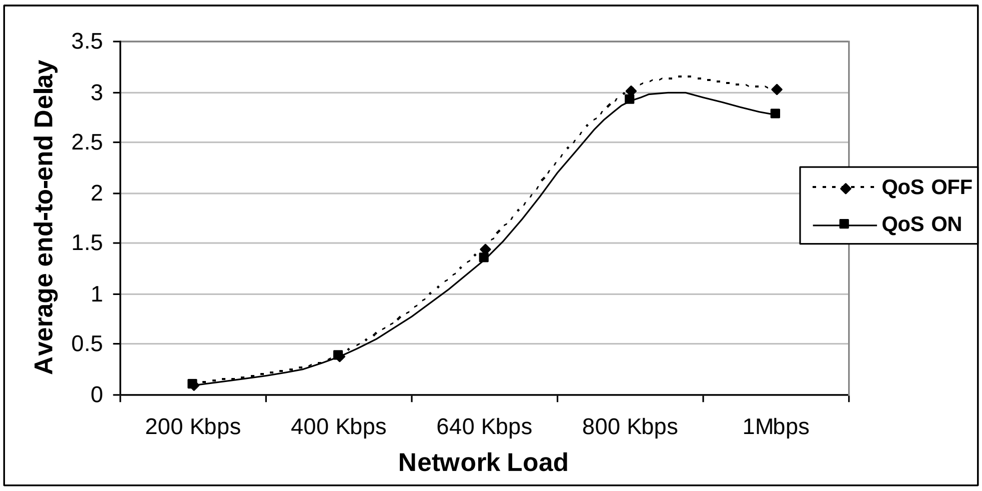


Figure 0.20: End-to-End Delay of IA traffic with/without CBQoS (640Kbps IE Traffic)

Figure 6.21 shows the overall average end-to-end delay in the network. When the network is highly loaded, the delay is increased for different traffic flows. With CBQoS, however, the delay is reduced with an average improvement of -17.5%. Table 6.8 gives details about the delay improvement gained using CBQoS.

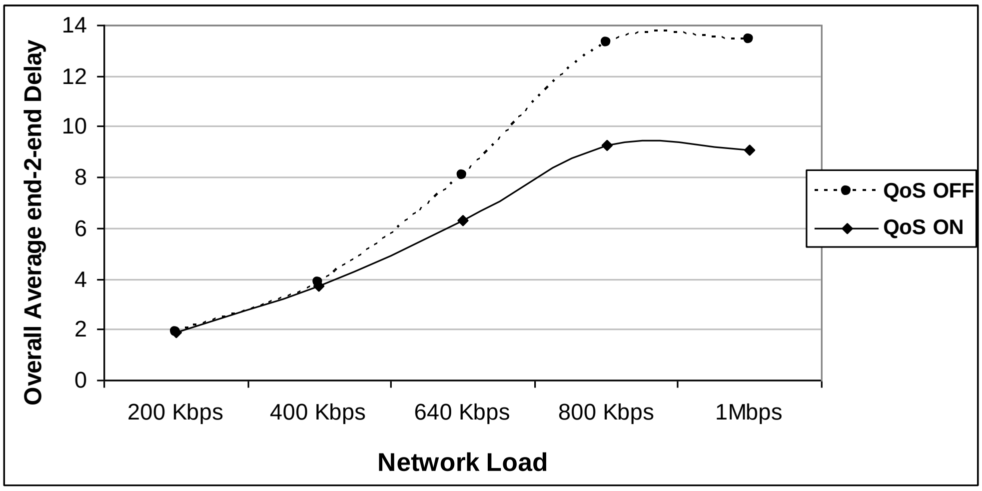


Figure 0.21: Overall End-to-End Delay with/without CBQoS (640Kbps IE Traffic)

Table 0.7: Delay Improvement using the CBQoS (640Kbps IE Traffic)

Network Load	IE Delay	IA Delay	Overall Delay
200 Kbps	-0.23%	+7.62%	+0.55%
400 Kbps	-3.90%	-0.02%	-3.54%
640 Kbps	-31.06%	-0.07%	-22.33%
800 Kbps	-52.99%	-0.03%	-30.45%
1Mbps	-52.35%	-0.08%	-32.44%
Average	**-28.11%**	**1.49%**	**-17.64%**

Network Power:

The network power is improved with the CBQoS. This is obvious in Figure 6.22 and Table 6.9. As the network gets loaded higher, the throughput to delay ratio (NP) increases.

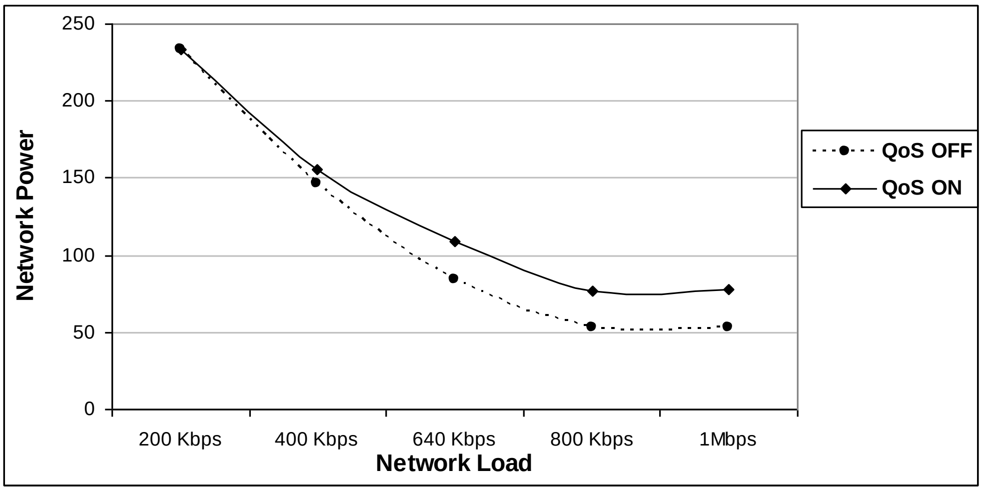


Figure 0.22: Network Power with/without CBQoS (640Kbps IE Traffic)

Table 0.8: Network Performance using the CBQoS (640Kbps IE Traffic)

Network Load	Network Power
200 Kbps	+0.22%
400 Kbps	+6.89%
640 Kbps	+30.25%
800 Kbps	+43.64%
1Mbps	+47.76%
Average	**+25.75%**

1.23.2. Non-Real Time Traffic

In this section, we investigate the effect of CBQoS to Non Real-Time Traffic. We consider simulating FTP traffic, to assure that the CBQoS does not negatively influence Non-Real Time applications.

The GLOMOSIM File Transfer Protocol traffic (FTP/GENERIC) is used in the simulation. Unlike CBR which uses UDP protocol, FTP uses the reliable Transport Control Protocol (TCP) for the transport layer.

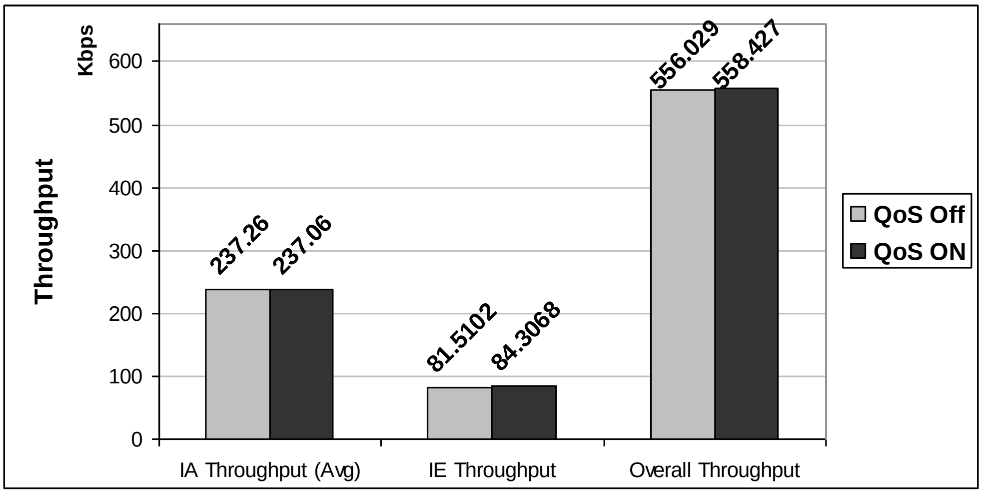


Figure 0.23: Network Throughput of FTP traffic with/without CBQoS (FRP Traffic)

In FTP experiments, we consider *Throughput* as the main QoS metric. FTP applications are tolerant to *delay;* so, delay is not considered in these experiments. In addition, FTP is based on TCP, and the delay encountered by TCP connections does not concisely reflect the actual delay in the network, because it is influenced by the TCP congestion control mechanisms.

The results, shown in Tables 6.10 and Figure 6.23, illustrate the behaviour of FTP traffic when CBQoS is applied. The results show that there is an increase in the IE throughput (of 3.4%), insignificant decrease in the IA throughput (of -0.08%), and a slight increase in the overall network throughput (of 0.43%). This positively shows that the CBQoS does not negatively influence NRT applications.

Table 0.9: Throughput Improvement using the CBQoS (NRT Traffic)

IA Throughput	-0.08%
IE Throughput	+3.43%
Overall Throughput	+0.43%

1.24. Scenario 2

To further evaluate the CBQoS approach with various topologies, we performed the simulation on different network scenarios. The scenario, depicted in Figure 6.24, represents a clustered MANET, consisting of 6 clusters and 30 nodes placed on (2000*2000) terrain. We carry out several experiments with Real-Time traffic (section 6.3.1), Non Real-Time traffic (section 6.3.2), and Hybrid RT/NRT traffic (section 6.3.3). To investigate the performance of CBQoS under different traffic loads and various parameters, the experiments, in this scenario, consider various numbers of traffic flows and connections, in each experiment.

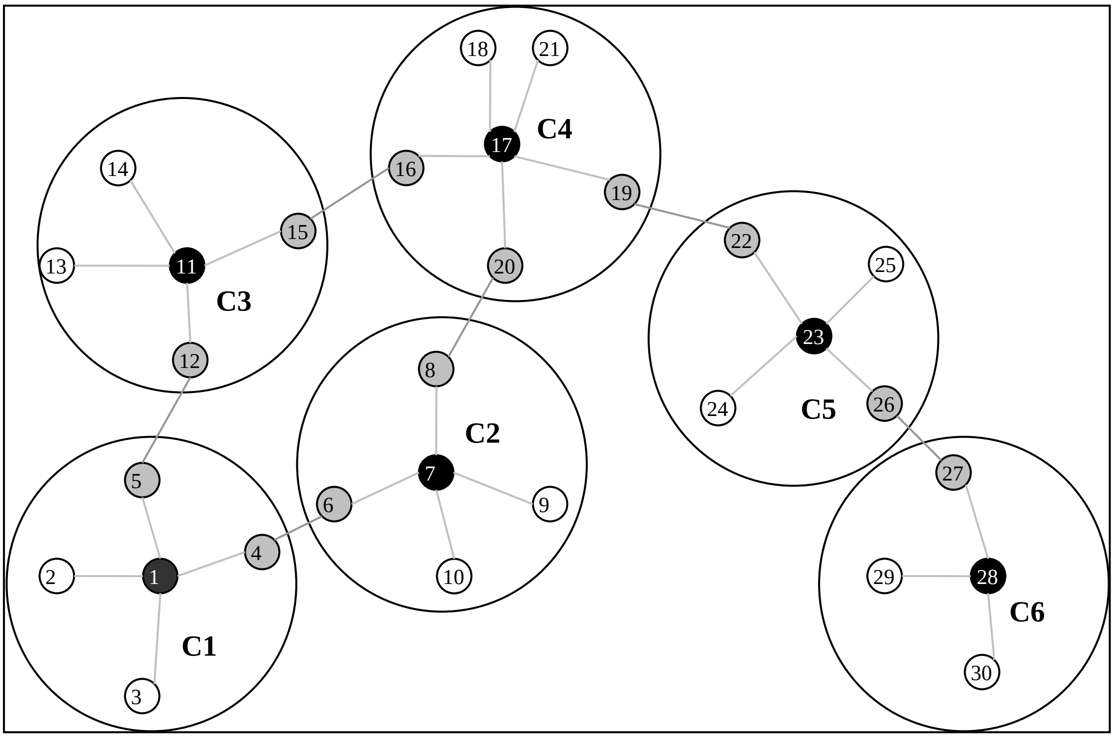


Figure 0.24: Scenario 2

1.24.1. Real Time Traffic

Three experiments are carried out with different numbers of RT traffic flows as follows.

1- *Experiment 1***:** 3 RT flows (1 IE and 2 IA)
2- *Experiment 2:* 5 RT flows (2 IE and 3 IA)
3- *Experiment 3:* 7 RT flows (2 IE and 5 IA)

These different experiments are carried out to study the performance of the network against different number of traffic flows, and different ratios of IE to IA traffic (50%, 66%, and 40%). The results of these experiments are merged and shown in figures, and summarized in tables. Further details about the results can be found in Appendix A.

Throughput:

The results of IE throughput with different number of flows are shown in Figure 6.25. Throughput improvement is achieved by CBQoS. The results show an improvement in the IE throughput, especially with high number of traffic flows.

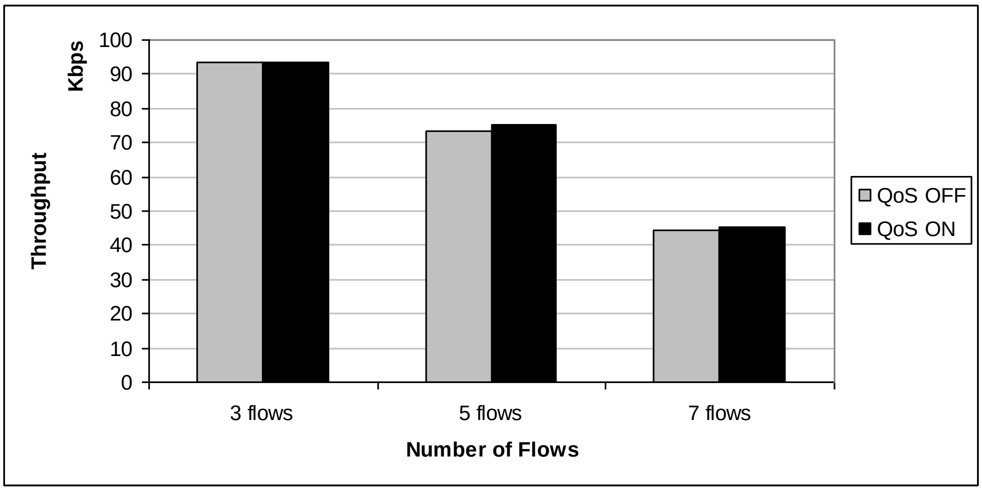


Figure 0.25: Throughput of IE traffic with/without CBQoS (Scen2: RT Traffic)

Figure 6.26 shows that the IA throughput is also improved using CBQoS. The overall throughput is depicted in Figure 6.27, and summarized in Table 6.11. These results show that CBQoS improves the overall network throughput with 1.25%. This improvement is significant, especially when the network is highly loaded.

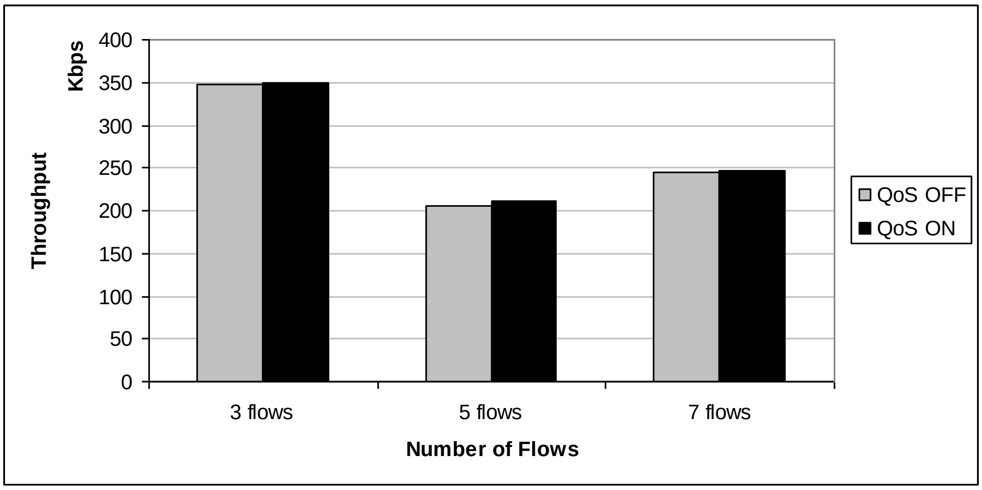


Figure 0.26: Throughput of IA traffic with/without CBQoS (Scen2: RT Traffic)

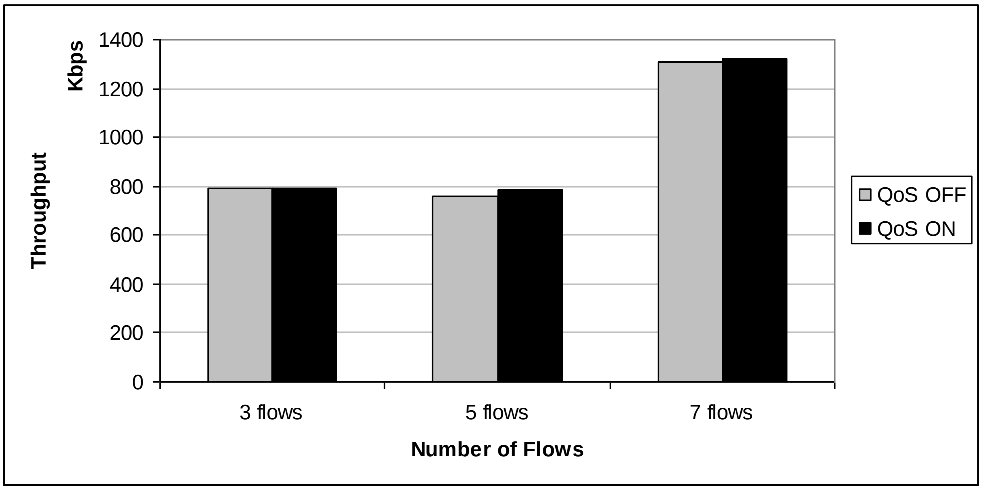


Figure 0.27: Overall Network Throughput with/without CBQoS (Scen2: RT Traffic)

Table 0.10: Throughput Improvement using the CBQoS (Diff. no. of Flows)

Number of Flows	IE Throughput	IA Throughput	Overall Throughput
3 flows	-0.25%	+0.41%	+0.33%
5 flows	+2.51%	+2.77%	+2.72%
7 flows	+1.80%	+0.62%	+0.70%
Average	**+1.36%**	**+1.27%**	**+1.25%**

Delay:

The results of IE delay, with different number of flows, are shown in Figure 6.28. Delay improvement is achieved when CBQoS is used. The average IE delay is decreased as the ratio of IE to IA traffic is increased. The behaviour of IA delay is shown in Figure 6.29. It is in its peak when the ratio of IE to IA traffic is high.

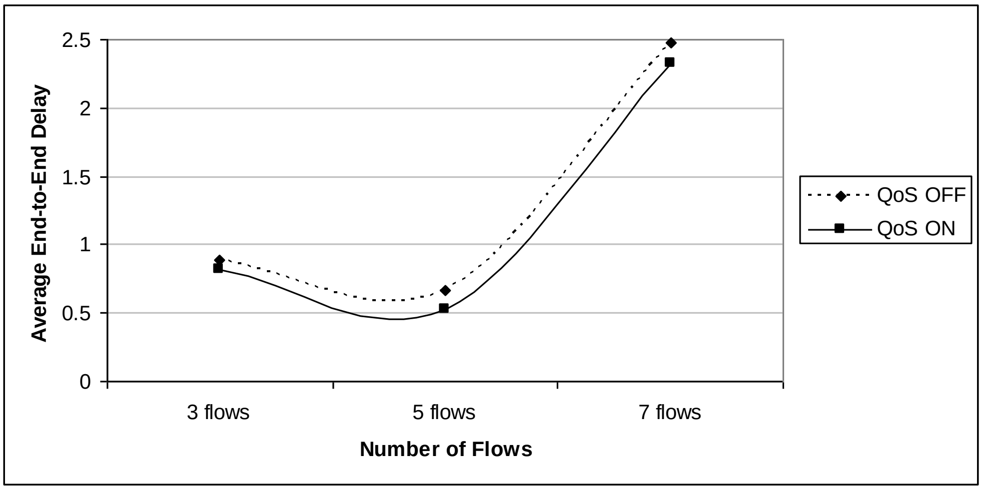


Figure 0.28: End-to-End Delay of IE traffic with/without CBQoS (Scen2: RT Traffic)

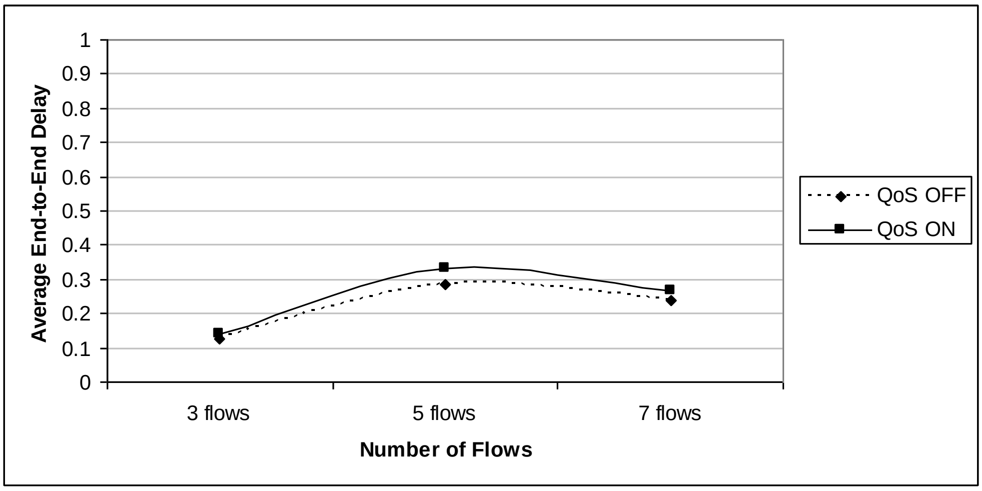


Figure 0.29: End-to-End Delay of IA traffic with/without CBQoS (Scen2: RT Traffic)

The results of IE throughput, IA throughput, and the overall network throughput are summarized in Table 6.12.

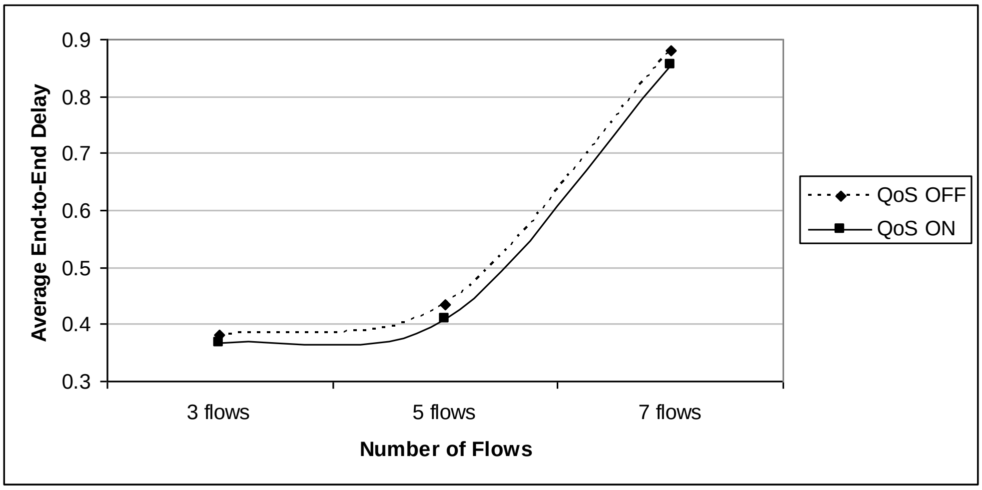


Figure 0.30: Overall End-to-End Delay with/without CBQoS (Scen2: RT Traffic)

Table 0.11: Delay Improvement using the CBQoS (Diff. no. of Flows)

Number of Flows	IE Delay	IA Delay	Overall Delay
3 flows	-7.44%	10.00%	-3.60%
5 flows	-20.31%	15.41%	-6.23%
7 flows	-6.20%	10.88%	-2.88%
Average	**-11.32%**	**12.10%**	**-4.24%**

Network Power:

The ratio of the overall network throughput to the overall network delay is shown in Figure 6.31 and Table 6.13.

Table 0.12: Network Performance using the CBQoS (Diff. no. of Flows)

Number of Flows	Network Power
3 flows	+4.07%
5 flows	+9.55%
7 flows	+3.69%
Average	**+5.77%**

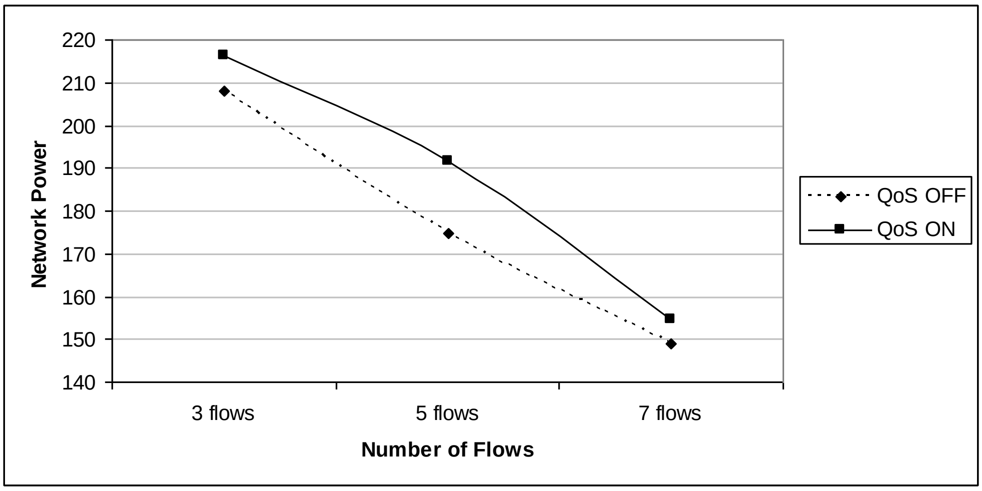


Figure 0.31: Network Power with/without CBQoS (Scen2: RT Traffic)

1.24.2. Non Real Time Traffic

By analogy to the RT traffic (section 6.3.1), NRT traffic is simulated with different numbers of connections (3, 5, and 7 connections). We consider NRT traffic in the simulations, to study the effect of applying CBQoS on this type of traffic. In this context, it should be reminded that most of the QoS models, like IntServ [Braden et al., 1994] and

DiffServ [Blake et al., 1998], were mainly proposed for to support RT traffic and it is on the cost of NRT traffic.

Figure 6.32 and Figure 6.33 show the behaviour of different NRT traffic connections, with/without CBQoS. We note that IE traffic slightly suffers when CBQoS is ON, while the IA throughput and the overall network throughput are increased. The reduction in the IE throughput is because the NRT traffic, used in the simulation, is FTP which is based on the TCP. TCP sender requires the receiver to acknowledge the arrival of packets using specific small packets (ACK). An ACK packet is treated in the cluster of the connection destination as IA packet, while the actual data packets are treated there as IE packets. This placement makes the data packets to content with the ACK of the same connection, delaying the ACK packets, and causing the TCP to call its congestion control mechanism. However, the overall network throughput, in which we are interested, was slightly increased with 0.18%. This result, together with the result drawn in section 6.2.2, emphasizes that CBQoS does not negatively affect the NRT connection oriented traffic.

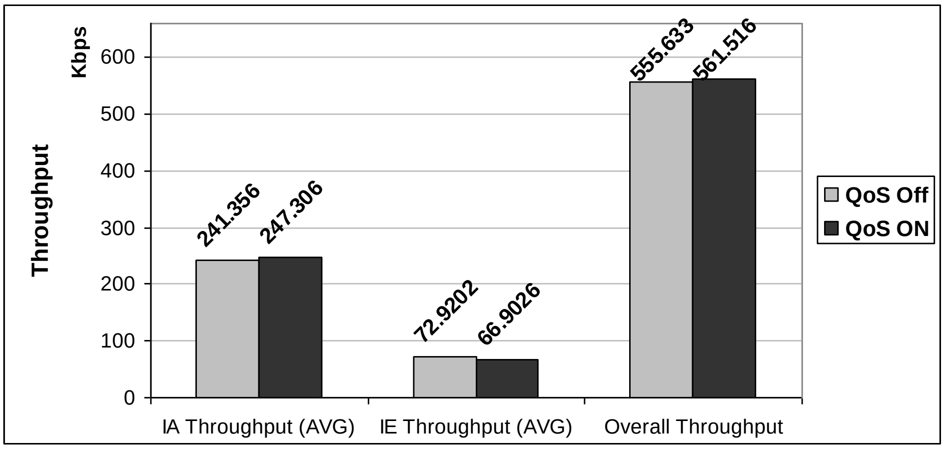


Figure 0.32: Throughput of NRT Traffic with/without CBQoS (Scen2: 3 Connections)

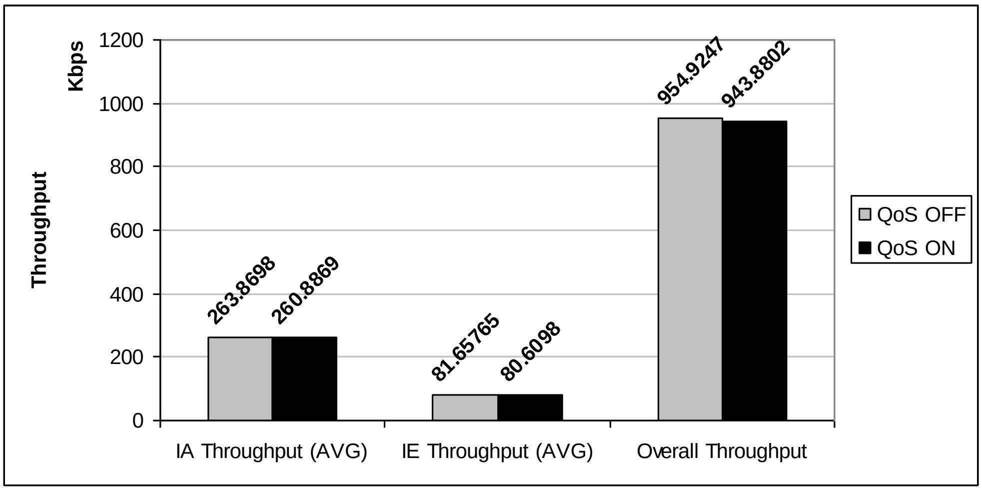


Figure 0.33: Throughput of NRT Traffic with/without CBQoS (Scen2: 5 Connections)

The results of NRT throughput improvement are summarized in Table 6.14. An overall improvement of 0.18% is achieved using CBQoS.

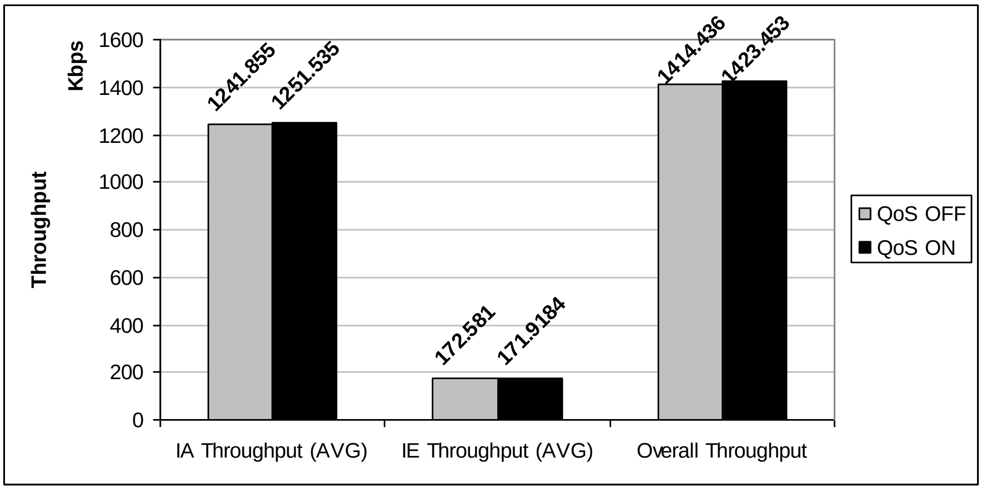


Figure 0.34: Throughput of NRT Traffic with/without CBQoS (Scen2: 7 Connections)

Table 0.13: Throughput using the CBQoS (Diff. No. of NRT Connections)

Number of Connections	IA Throughput	IE Throughput	Overall Throughput
3 flows	2.47	-8.25	1.06
5 flows	-1.13	-1.28	-1.16
7 flows	0.78	-0.38	0.64
Average	**0.70**	**-3.31**	**0.18**

1.24.3. Hybrid RT/NRT Traffic

In this section, we use hybrid traffic of Real-Time and Non Real-Time traffic. Three experiments, with the same numbers of CBR flows (RT) and FTP connections (NRT), are used. For the *worst case* evaluation, each RT flow are accompanied with one NRT connection that takes the reverse path (with the RT receiver and RT sender being the NRT sender and NRT receiver respectively). We apply CBQoS over the GLOMOSIM DiffServ implementation, which supports three queues for three types of traffic: Control traffic, RT traffic, and NRT traffic with a decreasing priority.

Throughput:

The RT traffic gains some improvement in the IE, IA, and the overall network throughput, as appears in Figure 6.35.

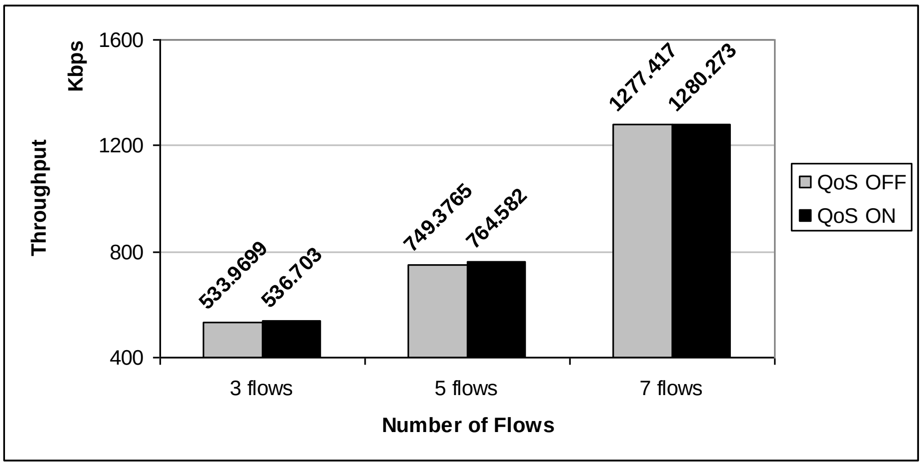


Figure 0.35: Throughput of RT traffic with/without CBQoS (Scen2: Hybrid Traffic)

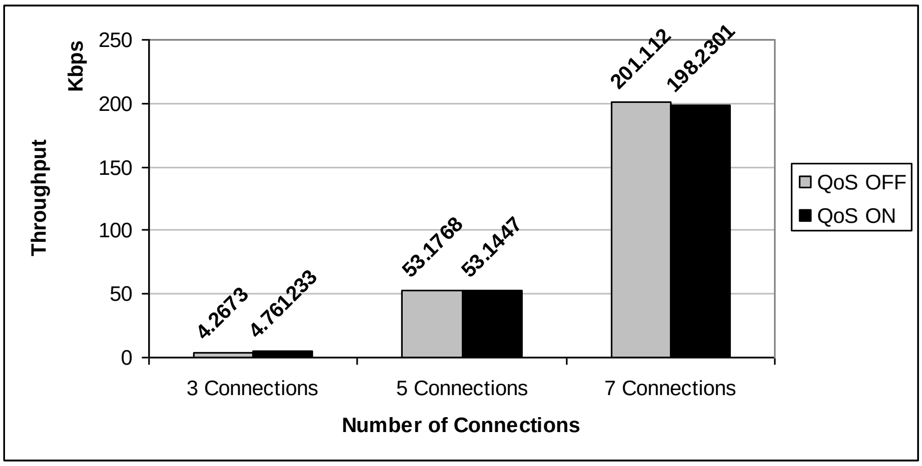


Figure 0.36: Throughput of NRT traffic with/without CBQoS (Scen2: Hybrid Traffic)

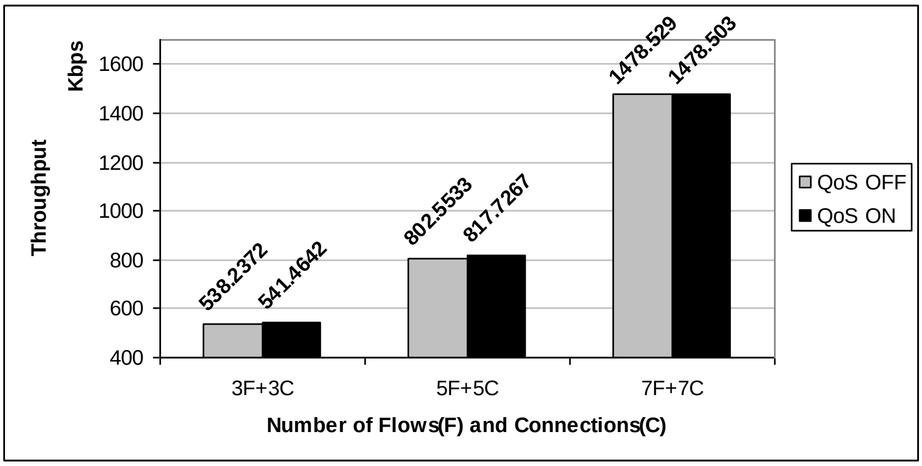


Figure 0.37: Overall Network Throughput with/without CBQoS (Scen2: Hybrid Traffic)

In Figure 6.36, it is obvious that NRT traffic suffers a little decrease in its throughput, especially when the network gets highly loaded. This is predicted as a penalty for the better treatment of RT traffic.

Table 0.14: Throughput using the CBQoS (Hybrid Traffic)

Number of Flows	IA Throughput	IE Throughput	Overall Throughput
3 flows	+0.51%	+11.57%	+0.60%
5 flows	+2.03%	-0.06%	+1.89%
7 flows	+0.22%	-1.43%	+0.00%
Average	**+0.92%**	**+3.36%**	**+0.83%**

Delay:

The RT delay gains improvement with CBQoS, as shown in Figure 6.38 and Table 6.16.

Table 0.15: Delay Improvement using the CBQoS (Hybrid Traffic)

Number of Flows	End-to-end Delay
3 flows	-5.00%
5 flows	-4.36%
7 flows	-0.91%
Average	**-3.42%**

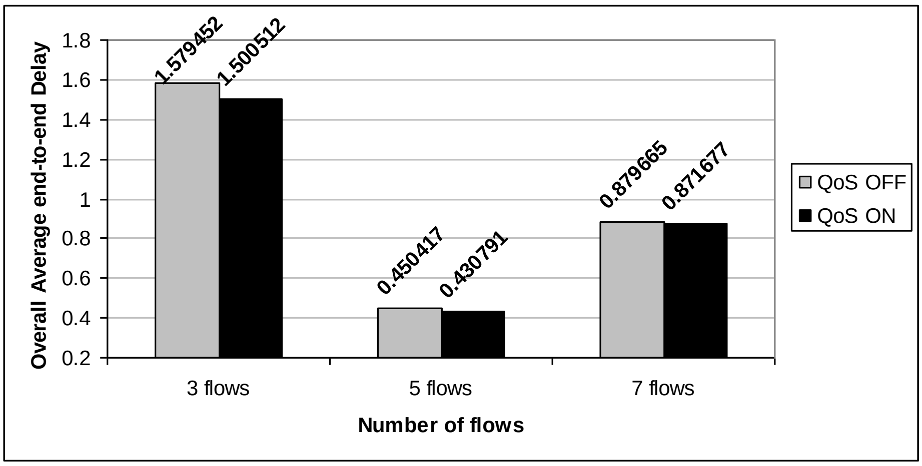


Figure 0.38: End-to-End Delay of RT traffic with/without CBQoS(Scen2: Hybrid Traffic)

1.25. Overall Results Discussion

In this section, an overall result analysis is provided.

Throughput:

The first performance metric, we are using to evaluate the CBQoS approach, is throughput. According to the results, IE traffic did not suffer high contention from other traffic, under lightly loaded traffic. In this case, both IE and IA got high throughput, even without the use of CBQoS; this explains the small increase in IE throughput in lightly loaded environment compared to the high increase of IE throughput in higher network loads. The CBQoS contributes in increasing the throughout of IE traffic, in different network loads, with an average improvement of 9.27%. This improvement is justified as follows:

In the traditional case, without using CBQoS, IE traffic suffers more and more as the network load increases, because the network gets congested and the contention increases. This contention occurs in every hop. IE packets travel across several hops, while IA packets travel at most two hops; therefore, IE packets face the contention multiple times what IA packets face. This, obviously, decreases the throughput of IE traffic. Moreover, after all the queuing in several intermediate nodes, an IE packet gets equal dropping probability as that of IA packets. This waste of queue capacity and wireless medium affects both IA and IE traffics. On the Other hand, Under CBQoS, IE flows get relatively higher throughput, because they are prioritized over IA flows. CBQoS gives better treatment to IE packets, in terms of scheduling priority and dropping probability; i.e. they are given higher forwarding priority, and less dropping priority.

Fortunately, even though CBQoS prioritizes IE over IA, the results showed that IE traffic did not starve IA. Furthermore, an increase of the IA throughput was also noticed when using CBQoS (with an average of 1.36%). The reason for this is that when CBQoS is OFF, IE packets encounter higher contention and waits longer in the queue, occupying a valuable queuing place which can be rather left to IA traffic. However, when CBQoS is ON, IE packets are prioritized for transmission; thus, freeing some space and allowing for IA transmissions. There is another reason that limits the increase in IE traffic, and eliminates it from starving IA; it is the DCF function of IEEE 802.11 MAC, which gives equal priority for all nodes in the network. When IE is prioritized on the network layer and scheduled first, it has equal contention priority on the MAC layer.

As a result of the improvement in IE and IA throughput, the overall network throughput was improved in all experiments with an average of 1.6% when CBQoS was used.

For NRT traffic, the different simulation experiments showed that CBQoS approach did not negatively affect this type of traffic. Moreover, it slightly increased the overall network throughout (of nearly 0.31%). This is an advantage of this approach over many other QoS approaches in which NRT traffic pays the penalty of improving RT traffic.

Delay:

The delay is an important performance metric, especially for real time applications. Significant improvement in the average end-to-end delay for IE traffic was achieved when using CBQoS (of around 17%). This is because; IE is queued and scheduled before IA traffic. Consequently, the intermediate queuing delay of IE packets was reduced. Delay for IE traffic was small when the network load was light, even without QoS support; however, as the network load got higher, IE traffic suffered higher delay when it is not supported with CBQoS. With CBQoS support, IE got less delay, especially in highly loaded environments.

The delay of IA is a little bit higher when using CBQoS in the majority of the simulation experiments (with average increase of 2.19%). This was expected due to prioritizing IE over IA packets. However, the overall delay, encountered by all flows (IE+IA) in the network, is smaller in all experiments and placements when using CBQoS (with 13.25% average improvement).

Network Power (NP):

The result of RT traffic showed that the NP was surprisingly improved with CBQoS (with an average of 23.44%). Furthermore, we noticed that CBQoS contributes on improving the *delay* more than its contribution in the *throughput*. According to the formula of computing the NP (section 5.3.2), the NP would be higher if the delay was considered more important than throughput. This makes this approach more suitable for RT applications.

CONCLUSIONS AND RECOMMENDATIONS

In this thesis, a new approach for supporting QoS in wireless ad hoc networks was proposed and implemented. This chapter concludes the thesis with remarks on the CBQoS approach, and some recommendations for future work.

1.26. Conclusions

- A new cluster-based approach was proposed to support QoS in clustered MANET. This approach, CBQoS, provides a service differentiation between Inter-cluster (IE) and Intra-cluster (IA) communications. Realizing that IE communications suffer lower throughput and higher end-to-end delay, CBQoS gives IE packets higher priority than that of IA. This prioritization reduces the penalty of dropping packets that have travelled across multiple clusters and encountered longer delays. Traffic classification is achieved using existing information and does not require extra fields.

- The proposed approach improves IE communications which are usually starved by IA communications. IA communications are usually one or two hops far, and so they usually get high throughput and low delays. The proposed QoS allows the network nodes, which are too far away from each other, to have better quality of communications. The results showed that IE traffic was improved with higher throughput (+9.27%) and lower end-to-end delay (-17%). The results showed that IA was not drastically affected by CBQoS; the IA throughput was increased (+1.36%), and the IA delay was also increased (+2.19%).

- The proposed cluster-based QoS approach improves the network performance, by increasing the overall network throughput, and reducing the overall average network delay, especially for RT applications. An improvement in the overall network throughput was achieved (+1.6%), with a considerable improvement in the overall network delay (-13.25%). This improvement allows for broader range of services, especially for multimedia, which have vital applications over such occasional ad hoc networks. The overall network performance with Non Real-Time (NRT) traffic was not as impressive as RT traffic. It was improved with (+0.31%). This result indicates that the proposed approach is more suitable to be applied for RT applications.

- A good advantage of the proposed approach is its multiple design choices. Not only can it be designed and implemented as a stand-alone QoS support model, but also it can be integrated with other existing QoS models like Differentiated Services.

1.27. Recommendations for Future Work

- The proposed approach was implemented on the network layer of the OSI model. The Idea can also be implemented on the MAC layer. For example, If IEEE 802.11e MAC protocol is used, the IE traffic can be given lower CW_{max} and CW_{min} and longer TXOP. The performance can be evaluated with this approach solely implemented on the MAC layer. Furthermore, the evaluation is suggested when the MAC layer implementation coexists with the Network layer implementation.

- Coexistence of CBQoS with Other MANET QoS techniques can be investigated, aiming to improve the QoS provisioning in MANET.

REFERENCES

Ahn, G-S., Campbell, Andrew T., Veres A., and Sun, L-H. (2002). **SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks**, IEEE INFOCOM 02

Amis AD, Prakash R, Vuong T, Huynh D. (2000). **Max-Min D-Cluster Formation in Wireless Ad Hoc Networks,** Proceedings of IEEE March 2000, pages 32-41

Bagrodia R, Meyerr R, et al. (1998) **PARSEC: A Parallel Simulation Environment for Complex System**, IEEE Computer, October 98

Bajaj, L., Takai, M., Ahuja, R., Tag K., Bagorodia, R., and Gerla, M. (1999). **GloMoSim:A Scalable Network Simulation Environment**, Technical Report 990027, University of California.

Baker, D. J., and Phremides, A. (1981). **A distributed algorithm for organizing mobile radio telecommunication networks**. In 2nd international conference on distributed computing systems, pages 476-483. Paris, France, April 1981. IEEE

Barua, G, and Chakraborty, I. (2002). **Adaptive Routing For Ad Hoc Wireless Networks Providing QoS Guarantees**, CSE Dept., Indian Institute of Technology, North Guwahati, Guwahati 781031, India.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998). **An Architecture for Differentiated Services**, RFC 2475, IEEE, 1998

Braden R, Clark D, and Shenker S. (1994). **Integrated Services in the Internet Architecture: An Overview**, RFC 1633, Internet Engineering Task Force (IETF), June 1994

Broadcom, (2007), **802.11n: Next Generation Wireless LAN Technology** (White Paper), retrieved February, 15, 2007, from http://www.broadcom.com/docs/WLAN/802_11n-WP100-R.pdf

Chatterjee, M., Das S. K., and Turgut, D. (2002). **WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks**, Journal of Cluster Computing, (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, April 2002, pp. 193-204.

Chen, S. and Nahrstedt, K. (1999). **Distributed Quality-of-Service Routing in Ad Hoc Networks**, IEEE Journal on Selected Areas in Communications, Vol. 17, no. 8, August, 1999.

Dharmaraju, D., Roy-Chowdhury, A., Hovareshti, P., and Baras, J. S., (2002). **INORA-A Unified Signaling and Routing Mechanism for QoS Support in Mobile Ad Hoc Networks**. Proceedings of the International Conference on Parallel Processing Workshops (ICPPW’02), IEEE, 2002.

ETSI, (2007), European Telecommunication Standards Institute, official website, Retrieved 2, 2, 2007 from http://www.etsi.org

Floyd S. and Jacobson V. (1993). **Random Early Detection gateways for Congestion Avoidance.** IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413

Gerla, M., and Tsai, J. T. (1995). **Multicluster, mobile, multimedia radio network**, ACM-Baltzer Journal of Wireless Networks, 1(3):255–265, 1995.

Halsal, F. (2005) **Computer Networking and the Internet**. 5th edition, Pearson Education Ltd. 2005

He Y, Abdel-wahab H. (2006). **HQMM: A Hybrid QoS Model for Mobile Ad-hoc Networks**, Proceedings of the 11th IEEE symposium on Computers and Communications (ISCC'06)

IEEE WG. (1999). **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,** IEEE Standard 802.11

IEEE WG. (2005). **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,** IEEE Standard 802.11e™

IEEE802, (2007), LAN/MAN Standards Committee, official website, Retrieved 2, 2, 2007 from **http://www.ieee802.org**

IETF, (2007), Internet Engineering Task Force, Official website, **http://www.ietf.org**

Kannan R, Vempala S, and Vetta A. (2004) **On Clustering: Good, Bad and Spectral**, Journal of the ACM, Vol. 51, No. 3, May 2004, pp. 497-515

Lee, S. B., and Campbell A.T. (1998). **INSIGNIA: In-band Signalling Support for QoS in Mobile Ad Hoc Networks**, Proc of 5th International Workshop on Mobile Multimedia Communications (MoMuC,98), Berlin, Germany, October 1998.

Lin, C. R., and Gerla, M. (1997). **Adaptive clustering for mobile wireless networks**, IEEE J. Selected Areas Commun., Vol. 15, No. 7, September 1997, pp. 1265-1275.

Mankin A, Ramakrishnan K. (1991). **Gateway Congestion Control Survey**, IETF RFC 1254, August 1991

Murthy, C. S., and Manoj, B. S. (2004). **Ad Hoc Wireless Networks Architectures and protocols**, Prentice Hall, USA

Nichols, K., Jacobson, V., and Zhang, L. (1999). **A Two-bit Differentiated Services Architecture for the Internet**, RFC 2638, IETF, July 1999

Perkins, C. E. (2001). **Ad Hoc Networking**, Addison Wesley, USA.

Peterson, L. Larry, and Davie, S. Bruce. (2003). **Computer Networks A System Approach**, (3rd Ed.). Morgan Kaufmann. USA.

Qaddoura E, AlMobaideen W, and Omari A. (2006). **Distributed Clusterhead Architecture for Mobile Ad Hoc Netoworks**, Journal of Computer Science 2 (7): p. 583-588

Semeria C. (2001). **Supporting Differentiated Service Classes: Queuing Scheduling Disciplines**, (white paper), Juniper Networks Inc. Sunnyvale, USA

Steenstrup, M. (2001) **Cluster-Based Networks**. In C. E. Perkins (Editor), Ad Hoc Networking, (pp. 75-138). Addison-Wesley, USA

Sivakumar R., Sinha P., and Bharghavan V. (1999). **CEDAR: a Core-Extraction Distributed Ad Hoc Routing algorithm**, IEEE Journal on Selected Areas in Communication. Vol. 17, No. 8, August 1999

Tanenbaum, A. S., (2003). **Computer Networks**. 4th edition, Pearson Education, Inc. USA

Wang Z. (2001). **Internet QoS: Architectures and Mechanisms for Quality of Service**, Morgan Kauffmann publishers, USA

Wu, K., and Janelle Harms. (2001). **QoS Support in Mobile Ad Hoc Networks**, Crossing Boundaries – an interdisciplinary journal, VOL 1, No 1 - Fall 2001

Xiao H., Seah, W.K.G., Lo, A., and Chua, K. C. (2000). **A Flexible Quality of Service Model for Mobile Ad-Hoc Networks**, IEEE VTC2000-spring, Tokyo, Japan, May 2000

Zavgren, J. (1997). **NTDR Mobility Management Protocols and Procedures**. In proceedings of the IEEE Military Communications Conference (MILCOM 97), November, 1997.

Zeng, X., Bagorodia, R., and Geral, M. (1998), **GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks**, pads, p. 154, 12th Workshop on Parallel and Distributed Simulation (PADS'98)

Zhang, L., Deering, S., Estrin, D., Shenker, S., and D. Zappala, (1993) **RSVP: A New Resource ReSerVation Protocol**, IEEE Network, September 1993

Appendices

Appendix A: Simulation Results

1 Results of Scenario1

1.1 RT Traffic

1- 200kbps IE, 200kbps IA

Table 1a: QoS OFF, 15 nodes, CBR Traffic, 200 Kbps IE, 200Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	126534	195858.5	518251	0.65981858	0.07820416	0.816226898	634.9349688
2	126788	196052.5	518893	0.66003816	0.07860024	0.817238642	634.9344896
3	127029	195720	518469	0.65884953	0.08006162	0.818972767	633.0723327
4	126681	195946	518573	0.65844899	0.07781679	0.814082575	637.0029478
5	126602	196182	518966	0.66776932	0.07882173	0.825412785	628.7351122
6	126736	196320	519376	0.661412	0.0792946	0.820001189	633.3844474
7	126648	196004	518656	0.65805534	0.07904836	0.816152071	635.4894124
8	126698	195862.5	518423	0.65567371	0.07779032	0.811254359	639.0387851
9	126711	196321	519353	0.6616373	0.07824659	0.818130488	634.8046034
10	126684	196311.5	519307	0.66408934	0.07725073	0.818590795	634.3914483
AVG	**126711.1**	**196057.8**	**518826.7**	**0.66057923**	**0.07851351**	**0.817606257**	**634.5788548**

Table 1b: QoS ON, 15 nodes, CBR Traffic, 200 Kbps IE, 200Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	129831	197703	525237	0.63865418	0.0889695	0.816593186	643.2052202
2	129864	197239	524342	0.63811478	0.0902692	0.818653186	640.4934458
3	129367	197413.5	524194	0.63974441	0.08774912	0.815242644	642.991389
4	129890	197347	524584	0.63687626	0.08961142	0.816099093	642.7944897
5	129571	197240	524051	0.6401376	0.08881949	0.817776572	640.8241786
6	129510	197245.5	524001	0.63736169	0.08936042	0.816082524	642.0931518
7	129972	197196.5	524365	0.63350845	0.08947802	0.812464483	645.4005202
8	129925	197426.5	524778	0.6391865	0.08954549	0.818277478	641.3203517
9	129440	197383	524206	0.6355657	0.09007024	0.815706177	642.640714
10	129474	197676	524826	0.64159671	0.08722408	0.816044864	643.1337579
AVG	**129684.4**	**197387**	**524458.4**	**0.63807463**	**0.0891097**	**0.816294021**	**642.4897219**

2- 200kbps IE, 400kbps IA

Table 2a: QoS OFF, 15 nodes, CBR Traffic, 200 Kbps IE, 400 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	25761	280348.5	586458	1.172852766	0.30936478	1.79158233	327.3408042
2	25397	281133.5	587664	1.150865106	0.30742854	1.76572218	332.8179295
3	24588	279676.5	583941	1.15479324	0.31065482	1.77610288	328.7765637
4	26108	281537.5	589183	1.159917131	0.30913379	1.77818471	331.3395939
5	25681	280343	586367	1.165331478	0.30976734	1.78486615	328.5215537
6	26002	282508	591018	1.187383082	0.30424058	1.79586423	329.0994882
7	25587	281433.5	588454	1.173247701	0.30612037	1.78548844	329.5759225
8	25613	281951.5	589516	1.164780465	0.30549608	1.77577263	331.9771852
9	25629	280677.5	586984	1.166844961	0.30808041	1.78300579	329.2103723
10	24971	280683	586337	1.166026922	0.30965861	1.78534414	328.4167949
AVG	**25533.7**	**281029.25**	**587592.2**	**1.166204285**	**0.30799453**	**1.78219335**	**329.7076208**

Table 2b: QoS ON, 15 nodes, CBR Traffic, 200 Kbps IE, 400 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	27886	289455.5	606797	0.844806848	0.31434624	1.47349934	411.8067688
2	27664	289151.5	605967	0.86169813	0.31724916	1.49619646	405.0049692
3	28197	288974.5	606146	0.84823836	0.31524087	1.47872009	409.9126016
4	28917	289815	608547	0.867584196	0.31396239	1.49550897	406.9163156
5	29059	290499.5	610058	0.86462209	0.31367705	1.49197618	408.892587
6	27104	289831	606766	0.867863696	0.31280464	1.49347297	406.2785278
7	28606	290296.5	609199	0.852689877	0.31433825	1.48136637	411.2412791
8	27747	290685.5	609118	0.858624416	0.31158318	1.48179077	411.0688315
9	27523	290140	607803	0.854741769	0.31435774	1.48345725	409.7206034
10	27246	289751	606748	0.858875772	0.31494742	1.48877061	407.5496902
AVG	**27994.9**	**289860**	**607714.9**	**0.857974515**	**0.31425069**	**1.4864759**	**408.8392174**

3- 200kbps IE, 640kbps IA

Table 3a: QoS OFF, 15 nodes, CBR Traffic, 200 Kbps IE, 640 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	11005	309309	629623	1.75717253	0.7697998	3.2967721	190.9816583
2	10160	309336	628832	1.78790775	0.7603358	3.3085793	190.061033
3	9850	309033.5	627917	1.798546406	0.7719914	3.3425292	187.856847
4	10698	307650.5	625999	1.822672193	0.762866	3.3484043	186.9544258
5	10795	309329	629453	1.845323925	0.7524341	3.3501921	187.885644
6	9427	308925	627277	1.834509279	0.7761934	3.3868962	185.2070358
7	10597	309361.5	629320	1.751469777	0.7579238	3.2673173	192.6106153
8	10428	307343.5	625115	1.852140956	0.7752888	3.4027185	183.7104639
9	10711	309386	629483	1.732676953	0.7567365	3.2461499	193.9168012
10	10906	308777	628460	1.789727221	0.7582408	3.3062088	190.0847871
AVG	**10457.7**	**308845.1**	**628147.9**	**1.797214699**	**0.764181**	**3.3255768**	**188.9269311**

Table 3b: QoS ON, 15 nodes, CBR Traffic, 200 Kbps IE, 640 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	12241	313709.5	639660	0.823140128	0.762674	2.3484882	272.3709665
2	12534	314137.5	640809	0.839774262	0.748163	2.3361003	274.3071392
3	11628	314394.5	640417	0.82392903	0.7588997	2.3417283	273.4804842
4	12625	313611	639847	0.830962363	0.7569207	2.3448038	272.8786999
5	12610	313854	640318	0.817332715	0.7500242	2.3173811	276.3110537
6	11752	312769	637290	0.823248639	0.7773644	2.3779775	267.9966521
7	11868	314135.5	640139	0.816712069	0.7636822	2.3440765	273.0879358
8	12330	313861	640052	0.812477491	0.7572281	2.3269336	275.0624229
9	12462	313444.5	639351	0.824555728	0.7647609	2.3540776	271.5929979
10	12599	315174	642947	0.819575901	0.7313084	2.2821926	281.7233732
AVG	**12264.9**	**313909.05**	**640083**	**0.823170833**	**0.7571026**	**2.3373759**	**273.8811725**

4- 200kbps IE, 800kbps IA

Table 4a: QoS OFF, 15 nodes, CBR Traffic, 200 Kbps IE, 800 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	10747	306325.5	623398	1.799762046	0.7761031	3.3519682	185.97969
2	9209	303818.5	616846	1.921362071	0.839222	3.599806	171.35535
3	9848	303423	616694	1.831678666	0.8246203	3.4809192	177.16412
4	10857	308232.5	627322	1.723297372	0.7735212	3.2703397	191.82166
5	10303	304735.5	619774	1.794430733	0.790375	3.3751807	183.62691
6	10061	305904	621869	1.78552424	0.793957	3.3734382	184.34279
7	9994	305606	621206	1.843220473	0.7979883	3.4391971	180.6253
8	10066	307450	624966	1.842109638	0.782255	3.4066197	183.45635
9	10580	306781	624142	1.745072297	0.7846395	3.3143514	188.31498
10	9594	304248	618090	1.850372331	0.8271225	3.5046174	176.36447
AVG	**10125.9**	**305652.4**	**621430.7**	**1.813682987**	**0.7989804**	**3.4116438**	**182.30516**

Table 4b: QoS ON, 15 nodes, CBR Traffic, 200 Kbps IE, 800 Kbps IA

seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	12570	313168	638906	0.811414073	0.7642335	2.339881	273.05063
2	12001	313067	638135	0.817742944	0.7779563	2.3736554	268.84062
3	11889	311654.5	635198	0.806781724	0.7964011	2.3995838	264.71173
4	12145	311824	635793	0.832554886	0.7728563	2.3782676	267.33451
5	12128	311643	635414	0.823248645	0.7696517	2.3625519	268.95239
6	12091	310810.5	633712	0.801711328	0.7898124	2.3813361	266.11615
7	12624	314162	640948	0.825768491	0.7475435	2.3208555	276.16885
8	12422	313128.5	638679	0.82133813	0.7661395	2.3536172	271.36061
9	12873	313293	639459	0.825841944	0.7593811	2.3446041	272.73645
10	11846	310769	633384	0.804622011	0.7894987	2.3836195	265.72362
AVG	**12258.9**	**312351.95**	**636962.8**	**0.817102418**	**0.7733474**	**2.3637972**	**269.49956**

5- 200kbps IE, 1Mbps IA

Table 5a: QoS OFF, 15 nodes, CBR Traffic, 200 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	9689	306374.5	622438	1.859904882	0.8091089	3.4781226	178.95804
2	10464	305442.5	621349	1.758000655	0.7909121	3.3398248	186.04239
3	10774	305118	621010	1.84651777	0.8209347	3.4883871	178.0221
4	10429	305193	620815	1.817361506	0.8099665	3.4372945	180.61153
5	10905	306711.5	624328	1.727400681	0.7767778	3.2809562	190.28843
6	11088	304579.5	620247	1.791618275	0.8031493	3.3979169	182.53742
7	9852	306152.5	622157	1.912425552	0.8074892	3.5274039	176.37816
8	9569	305432	620433	1.84553162	0.810262	3.4660557	179.0026
9	10650	306259	623168	1.775031618	0.7848291	3.3446899	186.31563
10	10068	303522	617112	1.823256236	0.8225087	3.4682737	177.93059
AVG	**10348.8**	**305478.45**	**621305.7**	**1.81570488**	**0.8035938**	**3.4228925**	**181.60869**

Table 5b: QoS ON, 15 nodes, CBR Traffic, 200 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	12712	313599	639910	0.81768027	0.7622964	2.342273	273.20043
2	11931	312101	636133	0.822141219	0.7905496	2.4032403	264.69804
3	12090	314301	640692	0.820101285	0.7634931	2.3470874	272.97322
4	12109	312402.5	636914	0.822301973	0.7817476	2.3857971	266.96067
5	11934	311209.5	634353	0.830698643	0.7770347	2.3847681	266.00197
6	12261	312268.5	636798	0.818198294	0.7783435	2.3748853	268.13842
7	11699	311291	634281	0.809198557	0.7857657	2.3807299	266.42292
8	11336	310930.5	633197	0.817658483	0.7989364	2.4155312	262.13572
9	13023	312295	637613	0.825714499	0.765201	2.3561165	270.62032
10	12539	313047.5	638634	0.82497978	0.7730207	2.3710211	269.34977
AVG	**12163.4**	**312344.55**	**636852.5**	**0.8208673**	**0.7776388**	**2.376145**	**268.05015**

6- 400kbps IE, 200Kbps IA

Table 6a: QoS OFF, 15 nodes, CBR Traffic, 400 Kbps IE, 200 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	68569	194280	457129	1.329299194	0.088489331	1.506277856	303.4825203
2	67848	194812.5	457473	1.33255938	0.086858698	1.506276775	303.7111158
3	70122	194749	459620	1.309135449	0.09059076	1.490316968	308.4041918
4	69105	193494	456093	1.310179092	0.09667653	1.503532152	303.3476866
5	68622	193241.5	455105	1.322427152	0.09299777	1.508422692	301.7091976
6	68305	193987.5	456280	1.324357813	0.090207504	1.504772821	303.2218509
7	69617	194667.5	458952	1.31611859	0.091364573	1.498847736	306.203218
8	68748	194042.5	456833	1.316403871	0.089148026	1.494699923	305.6352603
9	69182	194488.5	458159	1.326237686	0.092195564	1.510628814	303.2902562
10	67905	194581.5	457068	1.320603678	0.089436765	1.499477207	304.8182379
AVG	**68802.3**	**194234.45**	**457271.2**	**1.320732191**	**0.090796552**	**1.502325294**	**304.3823535**

Table 6b: QoS ON, 15 nodes, CBR Traffic, 400 Kbps IE, 200 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	68595	196837.5	462270	1.285648462	0.019119619	1.478180347	312.7290935
2	68435	197049	462533	1.292655029	0.018921246	1.487709238	310.9028217
3	69668	197163	463994	1.272033964	0.019226733	1.464762103	316.7708934
4	68598	196708	462014	1.27857816	0.01904675	1.473508026	313.5469857
5	70622	196496	463614	1.257973325	0.019070259	1.458381192	317.8963103
6	68866	197031.5	462929	1.279818739	0.019140188	1.472916125	314.2942033
7	70678	196380.5	463439	1.247125266	0.019181025	1.454287476	318.670832
8	67758	196832	461422	1.296268585	0.01883894	1.491540478	309.3593548
9	69742	196088.5	461919	1.262091131	0.019074638	1.455202149	317.4260018
10	68251	196785	461821	1.272732072	0.018747007	1.468913207	314.3963835
AVG	**69121.3**	**196737.1**	**462595.5**	**1.274492473**	**0.09802378**	**1.470540034**	**314.599288**

7- 400kbps IE, 400Kbps IA

Table 7a: QoS OFF, 15 nodes, CBR Traffic, 400 Kbps IE, 400 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	19951	291889	603729	2.259021886	0.296269881	2.851561648	211.7187263
2	21951	290679.5	603310	2.204067703	0.300463701	2.804995104	215.0841544
3	20166	292096	604358	2.318659238	0.296483171	2.91162558	207.5672106
4	19710	293132	605974	2.402454625	0.290720419	2.983895462	203.0815113
5	22750	290864	604478	2.247330997	0.29468785	2.836706697	213.09147
6	18565	290823.5	600212	2.319780951	0.299372949	2.918526848	205.655809
7	21030	290042.5	601115	2.247131196	0.302076517	2.85128423	210.8225457
8	19157	294389.5	607936	2.356045779	0.291150626	2.938347031	206.8972771
9	20910	296637.5	614185	2.407212601	0.289646072	2.986504744	205.6534486
10	21018	289629.5	600277	2.352171944	0.300752964	2.953677872	203.2303542
AVG	**20520.8**	**292018.3**	**604557.4**	**2.311387692**	**0.296162415**	**2.903712522**	**208.2802507**

Table 7b: QoS ON, 15 nodes, CBR Traffic, 400 Kbps IE, 400 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	22858	309370.5	641599	2.26263715	0.279892303	2.822421756	227.3221565
2	23337	311351.5	646040	2.333717538	0.273314491	2.88034652	224.2924577
3	21581	309608	640797	2.245770934	0.276460456	2.798691846	228.9630425
4	25284	303869.5	633023	2.047667569	0.282609264	2.612886096	242.26965
5	23443	310433	644309	2.21689241	0.275086214	2.767064838	232.84926
6	22617	310857	644331	2.288834873	0.271926103	2.832687078	227.4628232
7	25398	307078.5	639555	2.068995986	0.281972886	2.632941757	242.9051073
8	21155	307307	635769	2.222725243	0.277959274	2.77864379	228.8055066
9	22271	307441	637153	2.218628953	0.280064097	2.778757146	229.2942371
10	23125	308773.5	640672	2.235368546	0.27649845	2.788365445	229.7661525
AVG	**23106.9**	**308608.95**	**640324.8**	**2.21412392**	**0.277578354**	**2.769280627**	**231.3930393**

8- 400kbps IE, 640Kbps IA

Table 8a: QoS OFF, 15 nodes, CBR Traffic, 400 Kbps IE, 640 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	6970	333584	674138	2.702247467	1.019627014	4.741501495	142.1781688
2	8002	333131	674264	2.483876942	0.984575396	4.453027734	151.4169775
3	7404	333801.5	675007	2.48092492	1.032836484	4.546597887	148.464196
4	6483	333601	673685	2.811172876	1.036705556	4.884583988	137.9206503
5	7097	332148.5	671394	2.774570516	1.086236529	4.947043574	135.7162091
6	7709	334137.5	675984	2.714788674	1.016130283	4.74704924	142.4008823
7	8432	333501	675434	2.441011548	0.998499483	4.438010513	152.1929698
8	7729	333076.5	673882	2.546750309	1.01869402	4.584138348	147.0029805
9	8055	332767.5	673590	2.559602068	1.01723649	4.594075048	146.621462
10	7915	333102	674119	2.53454598	1.016691387	4.567928753	147.5765137
AVG	**7579.6**	**333285.05**	**674149.7**	**2.60494913**	**1.022723264**	**4.650395658**	**145.149101**

Table 8b: QoS ON, 15 nodes, CBR Traffic, 400 Kbps IE, 640 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	8724	335121.5	678967	1.28043217	1.001554742	3.283541654	206.77886
2	8775	335627	680029	1.260788119	0.971034412	3.202856942	212.3195048
3	7985	335629.5	679244	1.286529603	1.018277649	3.323084901	204.4016389
4	8777	335083.5	678944	1.244698068	1.047813905	3.340325878	203.2568153
5	8185	334857	677899	1.283870856	1.008362662	3.300596179	205.3868341
6	8022	334537.5	677097	1.271368608	1.041745255	3.354859117	201.8257627
7	8537	334796	678129	1.315538175	0.99535777	3.306253715	205.1049491
8	8354	334864	678082	1.254837918	1.02379984	3.302437597	205.3277254
9	8908	333449	675806	1.254364953	1.06304188	3.380448712	199.9160637
10	8440	334196.5	676833	1.285840614	1.062661799	3.411164212	198.417009
AVG	**8470.7**	**334816.15**	**678103**	**1.273826908**	**1.023364991**	**3.320556891**	**204.2735163**

9- 400kbps IE, 800Kbps IA

Table 9a: QoS OFF, 15 nodes, CBR Traffic, 400 Kbps IE, 800 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	7231	330947.5	669126	2.873271563	1.151420005	5.176111572	129.271943
2	6457	331915	670287	2.909204556	1.161822903	5.232850362	128.0921398
3	7263	331408	670079	2.674858909	1.111470837	4.897800582	136.812226
4	5765	331962.5	669690	2.93373952	1.193492652	5.320724823	125.8644306
5	6150	333061	672272	2.677941746	1.110644464	4.899230673	137.2199116
6	6234	332502.5	671239	2.880872626	1.196179237	5.273231099	127.2917851
7	7059	330930.5	668920	2.707253151	1.227795	5.16284315	129.5642693
8	7135	331841.5	670818	2.791247988	1.117841989	5.026931966	133.4448138
9	5948	333889	673726	2.66339285	1.103381911	4.870156671	138.337644
10	7204	332046.5	671297	2.905747549	1.105730875	5.117209299	131.1841984
AVG	**6644.6**	**332050.4**	**670745.4**	**2.801753046**	**1.147977987**	**5.09770902**	**131.7083362**

Table 9b: QoS ON, 15 nodes, CBR Traffic, 400 Kbps IE, 800 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	7363	332537.5	672438	1.236731704	1.169314762	3.575361228	188.0755418
2	7870	333210.5	674291	1.274729981	1.098813783	3.472357546	194.1882399
3	7276	334210	675696	1.272352136	1.109450673	3.491253482	193.5396566
4	8112	332081	672274	1.253750245	1.119392004	3.492534253	192.4888781
5	8333	333396.5	675126	1.310036514	1.064511199	3.439058912	196.3112634
6	10236	331635.5	673507	1.268489946	1.039143447	3.346776839	201.2404867
7	7229	332950	673129	1.221173017	1.178508429	3.578189875	188.1199778
8	7981	331588.5	671158	1.282134823	1.147987534	3.578109891	187.5733335
9	8562	334036.5	676635	1.239170956	1.108153196	3.455477347	195.8152035
10	7657	331795	671247	1.308053683	1.155099156	3.618251994	185.5169295
AVG	**8061.9**	**332744.1**	**673550.1**	**1.266662301**	**1.119037418**	**3.504737137**	**192.2869511**

10- 400kbps IE, 1Mbps IA

Table 10a: QoS OFF, 15 nodes, CBR Traffic, 400 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	7444	330999	669442	2.696866348	1.131671726	4.960209799	134.9624365
2	6780	333023	672826	2.7643538	1.143980485	5.05231477	133.1718293
3	7280	332472	672224	2.738766992	1.091288642	4.921344275	136.5935733
4	5928	331970	669868	2.703028939	1.173096411	5.04922176	132.6675737
5	6560	332781	672122	2.700439984	1.117830181	4.936100345	136.1645739
6	6709	332076	670861	2.762929437	1.178070443	5.119070322	131.0513351
7	6966	332680	672326	2.703500704	1.098481472	4.900463648	137.196406
8	5905	332868.5	671642	2.721758855	1.17110862	5.063976095	132.6313528
9	6135	331675.5	669486	2.85170329	1.247821703	5.347346696	125.1996622
10	7481	332320	672121	2.650964178	1.141371208	4.933706594	136.2304359
AVG	**6718.8**	**332286.5**	**671291.8**	**2.729431253**	**1.149472089**	**5.02837543**	**133.5869179**

Table 10b: QoS ON, 15 nodes, CBR Traffic, 400 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	7443	333647.5	674738	1.291764951	1.130268265	3.55230148	189.9439008
2	8059	334079	676217	1.290101434	1.088632634	3.467366702	195.0232145
3	9319	332357	674033	1.32136988	1.048034929	3.417439738	197.2333243
4	7204	331837	670878	1.243228771	1.155268877	3.553766525	188.7794247
5	9650	331115.5	671881	1.289222252	1.097073081	3.483368414	192.8825551
6	8276	332222.5	672721	1.241583016	1.144350542	3.530284099	190.557185
7	7337	333848.5	675034	1.265791366	1.126857071	3.519505507	191.7979667
8	7768	333896	675560	1.286587968	1.101753686	3.49009534	193.5649128
9	8242	331891	672024	1.274796808	1.116538332	3.507873472	191.575895
10	7907	333683.5	675274	1.301173804	1.075665961	3.452505725	195.5895381
AVG	**8120.5**	**332857.75**	**673836**	**1.280562025**	**1.108444338**	**3.4974507**	**192.6947917**

11- 640kbps IE, 200Kbps IA

Table 11a: QoS OFF, 15 nodes, CBR Traffic, 640 Kbps IE, 200 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	57047	193400	443847	1.670753519	0.096303536	1.863360591	238.1970522
2	54860	193391.5	441643	1.737893587	0.096150108	1.930193803	228.807594
3	55350	194211	443772	1.74269406	0.091380352	1.925454763	230.4764612
4	55236	193971	443178	1.723965172	0.092664261	1.909293694	232.1162016
5	54860	193158.5	441177	1.703848673	0.093136976	1.890122624	233.4118403
6	58067	194454	446975	1.688613417	0.09862816	1.885869736	237.0126587
7	56685	193269	443223	1.687308449	0.096241761	1.879791971	235.7830052
8	56174	193454.5	443083	1.718483472	0.096882055	1.912247581	231.7079673
9	54309	193800.5	441910	1.750841035	0.09175186	1.934344754	228.4546222
10	57239	194123	445485	1.693422999	0.09430652	1.882036038	236.7037565
AVG	**55982.7**	**193723.3**	**443429.3**	**1.711782438**	**0.094744559**	**1.901271556**	**233.2671159**

Table 11b: QoS ON, 15 nodes, CBR Traffic, 640 Kbps IE, 400 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	56487	196216	448919	1.685139716	0.102836241	1.890812198	237.4212523
2	53553	196299.5	446152	1.749776461	0.099866127	1.949508715	228.853555
3	52816	196414.5	445645	1.755790078	0.099474056	1.95473819	227.9819376
4	55370	195363	446096	1.690190356	0.10086247	1.891915296	235.7906831
5	54887	195520.5	445928	1.688851273	0.105099618	1.899050508	234.8162927
6	54436	196564.5	447565	1.699432619	0.099283644	1.897999906	235.8087577
7	56896	195815.5	448527	1.67009043	0.10260808	1.875306589	239.1752915
8	53320	196481.5	446283	1.724448718	0.102391767	1.929232251	231.3267362
9	54735	195614	445963	1.705489939	0.106628271	1.918746481	232.4241396
10	54758	196324	447406	1.70876147	0.100631534	1.910024538	234.2409697
AVG	**54725.8**	**196061.3**	**446848.4**	**1.707797106**	**0.101968181**	**1.911733467**	**233.7839616**

12- 640kbps IE, 400Kbps IA

Table 12a: QoS OFF, 15 nodes, CBR Traffic, 640 Kbps IE, 400 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	17657	269729.5	557116	3.025388109	0.390918444	3.807224997	146.3312519
2	17044	271932.5	560909	3.154428482	0.38247599	3.919380461	143.1116488
3	16754	270208.5	557171	3.063896478	0.387281891	3.838460259	145.1548179
4	17364	272349	562062	3.064096335	0.382429759	3.828955853	146.7924995
5	18030	274863	567756	3.080743283	0.373996713	3.828736708	148.288076
6	16674	271634	559942	3.077915665	0.38416025	3.846236164	145.5818042
7	18059	268489.5	555038	2.972367694	0.396387052	3.765141797	147.4148996
8	17898	270018.5	557935	3.034861637	0.391815868	3.818493372	146.1139108
9	16296	270319.5	556935	3.061364263	0.388228231	3.837820725	145.1175133
10	17205	268734	554673	3.034834921	0.398867247	3.832569414	144.7261459
AVG	**17298.1**	**270827.8**	**558953.7**	**3.056989687**	**0.387656144**	**3.832301975**	**145.8632568**

Table 12b: QoS ON, 15 nodes, CBR Traffic, 640 Kbps IE, 400 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	19204	278853.5	576911	2.792151473	0.378892012	3.549935496	162.5130937
2	18324	279362.5	577049	3.059544769	0.374425439	3.808395647	151.5202341
3	18140	281111.5	580363	2.953459557	0.369130995	3.691721546	157.2066021
4	18724	277812.5	574349	2.920647174	0.38226618	3.685179533	155.8537365
5	17817	276984.5	571786	2.927017778	0.382689566	3.69239691	154.8549666
6	18964	277968.5	574901	2.898239899	0.381778061	3.66179602	156.9997337
7	18112	278790.5	575693	3.051435717	0.382987951	3.817411618	150.8071588
8	18387	279705.5	577798	3.075478571	0.376388453	3.828255476	150.9298435
9	20163	277788	575739	2.868884299	0.382368444	3.633621187	158.4477221
10	20006	277605	575216	2.829749882	0.383331858	3.596413597	159.9415597
AVG	**18784.1**	**278598.2**	**575980.5**	**2.937660912**	**0.379425896**	**3.696512703**	**155.9074651**

13- 640kbps IE, 640Kbps IA

Table 13a: QoS OFF, 15 nodes, CBR Traffic, 640 Kbps IE, 640 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	5844	336970.5	679785	4.890713452	1.402885684	7.69648482	88.32408767
2	5789	336103	677995	5.211409711	1.427200358	8.065810427	84.0578893
3	4721	335244.5	675210	5.459308682	1.507925232	8.475159145	79.66930042
4	4340	336494.5	677329	5.513355563	1.461519205	8.436393973	80.28655397
5	5153	336721	678595	4.84605998	1.428559539	7.703179057	88.09285036
6	5837	337217.5	680272	5.114914429	1.393573261	7.902060951	86.08792114
7	4890	336609.5	678109	5.563275636	1.454095946	8.471467527	80.0462255
8	5137	334756	674649	5.454936171	1.510888099	8.476712369	79.58852095
9	4775	336174	677123	4.999609133	1.460539507	7.920688146	85.48790049
10	5815	336440.5	678696	4.971887446	1.436613811	7.845115068	86.51192419
AVG	**5230.1**	**336273.1**	**677776.3**	**5.20254702**	**1.448380064**	**8.099307148**	**83.8153174**

Table 13b: QoS ON, 15 nodes, CBR Traffic, 640 Kbps IE, 640 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	5657	341266	688189	3.469241247	1.330445303	6.130131852	112.2633275
2	5522	339240	684002	3.641903274	1.409164043	6.46023136	105.8788706
3	5361	339597.5	684556	3.80876944	1.368863989	6.546497418	104.5682838
4	6390	339268	684926	3.596179093	1.385426059	6.36703121	107.5738405
5	6430	340068.5	686567	3.602946556	1.345312568	6.293571691	109.0902009
6	5616	340244.5	686105	3.526103138	1.368071313	6.262245763	109.5621325
7	5637	340130.5	685898	3.685753529	1.327636162	6.341025852	108.1683021
8	5082	339157	683396	3.430475872	1.390057246	6.210590363	110.0372042
9	6962	341445.5	689853	3.422090008	1.290338704	6.002767416	114.9224936
10	6589	341608.5	689806	3.685040993	1.302701136	6.290443265	109.6593628
AVG	**5924.6**	**340202.6**	**686329.8**	**3.586850315**	**1.351801652**	**6.290453619**	**109.1724019**

14- 640kbps IE, 800Kbps IA

Table 14a: QoS OFF, 15 nodes, CBR Traffic, 640 Kbps IE, 800 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	2078	350195	702468	6.4607742	3.504165313	13.46910484	52.15402275
2	2320	351888	706096	7.234939	3.212826801	13.66059261	51.68853358
3	2941	350749.5	704440	6.3806662	2.66585094	11.7123681	60.14496762
4	2276	350884.5	704045	8.0892933	2.890229926	13.86975317	50.7611773
5	2849	351355.5	705560	6.6684039	2.604508864	11.87742167	59.40346481
6	2699	350491.5	703682	7.0245608	2.808834504	12.64222985	55.66122501
7	2276	351041	704358	8.2329495	3.352816069	14.9385816	47.15025958
8	2386	350868	704122	8.2535114	3.067239151	14.38798973	48.93817783
9	2223	350536	703295	7.1819874	2.803600578	12.78918855	54.99137002
10	2788	351527.5	705843	7.7011407	3.109310044	13.91976078	50.70798351
AVG	**2483.6**	**350953.65**	**704390.9**	**7.322822651**	**3.001938219**	**13.32669909**	**53.1601182**

Table 14b: QoS ON, 15 nodes, CBR Traffic, 640 Kbps IE, 800 Kbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	3300	350545.5	704391	3.4891119	2.873769563	9.236651046	76.26043211
2	3387	351292	705971	3.4321628	2.556930005	8.546022853	82.60813388
3	4598	350435	705468	3.3749552	2.529754448	8.43446409	83.64111726
4	2776	350888	704552	3.399211	2.871040139	9.141291304	77.0735749
5	2198	352150.5	706499	3.6804306	3.014164686	9.708759969	72.76923132
6	3473	351051.5	705576	3.4237771	2.81160106	9.046979179	77.99023144
7	3742	350951.5	705645	3.6110485	2.741782417	9.094613361	77.5893347
8	2726	351188	705102	3.2929353	3.24732433	9.787584006	72.04045447
9	3776	350670.5	705117	3.4205867	2.921317415	9.26322151	76.12006247
10	2324	350964.5	704253	3.3033333	3.563087132	10.42950757	67.52504809
AVG	**3230**	**351013.7**	**705257.4**	**3.44275525**	**2.913077119**	**9.268909489**	**76.36176206**

15- 640kbps IE, 1Mbps IA

Table 15a: QoS OFF, 15 nodes, CBR Traffic, 640 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	2615	350324	703263	7.277077321	3.047307695	13.37169271	52.5934162
2	2828	350915.5	704659	7.332934309	3.096533627	13.52600156	52.0966227
3	2793	350766.5	704326	7.910885966	2.906628376	13.72414272	51.32021828
4	2469	350811.5	704092	6.91426141	3.080056741	13.07437489	53.85282324
5	2966	350600.5	704167	6.536876224	2.703956055	11.94478833	58.95181902
6	2595	349865	702325	8.517911679	3.347633011	15.2131777	46.16556868
7	2196	351038	704272	7.295142834	2.80626278	12.90766839	54.56229417
8	2088	350840	703768	7.889010671	3.382265607	14.65354189	48.02715995
9	2668	350502.5	703673	6.695720412	2.686331436	12.06838328	58.30714715
10	2938	350005.5	702949	7.399139233	3.141537696	13.68221463	51.37684353
AVG	**2615.6**	**350566.9**	**703749.4**	**7.376896006**	**3.019851302**	**13.41659861**	**52.72539129**

Table 15b: QoS ON, 15 nodes, CBR Traffic, 640 Kbps IE, 1 Mbps IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	2801	351114	705029	3.526660371	3.05953266	9.64572569	73.09237507
2	3389	351354	706097	3.47194899	2.637341909	8.746632808	80.72786585
3	2839	352332	707503	3.599589568	2.82686898	9.253327527	76.4593059
4	3068	350804.5	704677	3.6987269	2.885043421	9.468813741	74.42083235
5	3072	351667	706406	3.548404434	2.743878746	9.036161926	78.17544725
6	3643	350011.5	703666	3.252784324	2.684008055	8.620800434	81.6242071
7	3841	350292.5	704426	3.569104231	2.70084794	8.97080011	78.5243224
8	3815	350548	704911	3.46748322	2.790718503	9.048920225	77.90001265
9	3254	351002	705258	3.37484262	2.647022333	8.668887286	81.35507785
10	2621	351172	704965	3.638500853	2.772399323	9.183299498	76.76598157
AVG	**3234.3**	**351029.75**	**705293.8**	**3.514804551**	**2.774766187**	**9.064336925**	**77.9045428**

1.2 NRT Traffic

Table 16a: QoS OFF, 15 nodes, 3 FTP/GENERIC, 1 IE, 2 IA

Seed	IA Throughput (AVG)	IE Throughput	Overall Throughput
1	237880.5	95202	570963
2	238169	81224	557562
3	238929.5	82121	559980
4	236096.5	69056	541249
5	236708	79661	553077
6	236123.5	74313	546560
7	236330.5	78916	551577
8	237959	82045	557963
9	236302.5	90471	563076
10	238097	82093	558287
AVG	**237259.6**	**81510.2**	**556029.4**

Table 16b: QoS ON, 15 nodes, 3 FTP/GENERIC, 1 IE, 2 IA

Seed	IA Throughput (AVG)	IE Throughput	Overall Throughput
1	236951	87717	561619
2	237825	80713	556363
3	239508	81769	560785
4	235177.5	94891	565246
5	234552	94302	563406
6	235579	69093	540251
7	235709	84251	555669
8	239301	82544	561146
9	237215	82511	556941
10	238785.5	85277	562848
AVG	**237060.3**	**84306.8**	**558427.4**

2 Results of Scenario2

2.1 RT Traffic

1- Three RT flows

Table 17a: QoS OFF, 30 nodes, 3 CBR Traffic: 1 IE, 2 IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	91622	349925	791472	0.912527032	0.122327483	0.38572733	205.1895038
2	93892	348121.5	790135	0.880233234	0.126617766	0.37782292	209.1283916
3	94156	346228.5	786613	0.878527356	0.125046111	0.37620653	209.0907376
4	92555	348690.5	789936	0.917177357	0.124936603	0.38901685	203.0595826
5	94022	346177.5	786377	0.873307828	0.126135358	0.37519285	209.5927479
6	96770	350041.5	796853	0.866087528	0.12606768	0.37274096	213.7819772
7	93596	348865.5	791327	0.885311943	0.127818775	0.3803165	208.0706477
8	93201	348298	789797	0.894813132	0.125638326	0.38202993	206.7369444
9	92955	349135.5	791226	0.881570713	0.125459258	0.37749641	209.5982848
10	92825	347735	788295	0.900778323	0.125439431	0.38388573	205.3462637
AVG	93559.4	348321.85	790203.1	0.889033445	0.125548679	0.3800436	207.9595081

Table 17b: QoS ON, 30 nodes, 3 CBR Traffic: 1 IE, 2 IA

Seed	IE Throughput	IA Throughput (AVG)	Overall Throughput	IE Delay	IA Delay (AVG)	Overall Delay	Network Power
1	92395	350459.5	793314	0.836481499	0.134680967	0.36861448	215.2150957
2	93304	349415.5	792135	0.814780263	0.137581446	0.36331438	218.0301781
3	95783	347415.5	790614	0.787493937	0.142656853	0.35760255	221.0873511
4	93889	349117	792123	0.816699644	0.140564478	0.36594287	216.4608393
5	92399	349133.5	790666	0.832126648	0.13742746	0.36899386	214.2761967
6	93035	351828.5	796692	0.842643245	0.136768791	0.37206028	214.1298206
7	93836	348532.5	790901	0.831935257	0.140353621	0.37088083	213.2493593
8	92365	350494	793353	0.819569464	0.138710384	0.36566341	216.9626429
9	92209	350748.5	793706	0.836787343	0.134868474	0.36884143	215.1889497
10	94077	350415	794907	0.810581835	0.137377482	0.36177893	219.7217492
AVG	93329.2	349755.95	792841.1	0.822909914	0.138098995	0.3663693	216.4322183

2- Five RT Flows

Table 18a: QoS OFF, 30 nodes, 5 CBR Traffic: 2 IE, 3 IA

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput	IE Delay (AVG)	IA Delay (AVG)	Overall Delay	Network Power
1	74160.5	203133	757720	0.658385601	0.289664514	0.437152949	173.3306392
2	73343	203381	756829	0.663587346	0.286980608	0.437623303	172.9407449
3	74862.5	205358.3333	765800	0.647267677	0.284795071	0.429784113	178.1824819
4	73136.5	206419.3333	765531	0.670790633	0.290692006	0.442731457	172.9109121
5	72521.5	205720.6667	762205	0.664821029	0.282257841	0.435283116	175.1055742
6	72617.5	207158.3333	766710	0.682765504	0.281473269	0.441990163	173.4676616
7	72961.5	206435.3333	765229	0.640377241	0.290797326	0.430629292	177.7001736
8	73253	204378	759640	0.659660026	0.283272762	0.433827667	175.1017875
9	74186	203523.3333	758942	0.658414097	0.28643116	0.435224335	174.3794957
10	74469	204208.6667	761564	0.655970827	0.286572944	0.434332097	175.3414046
AVG	**73551.1**	**204971.6**	**762017**	**0.660203998**	**0.28629375**	**0.435857849**	**174.8315423**

Table 18b: QoS ON, 30 nodes, 5 CBR Traffic: 2 IE, 3 IA

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput	IE Delay (AVG)	IA Delay (AVG)	Overall Delay	Network Power
1	76027.5	209570	780765	0.53371488	0.327066095	0.409725609	190.5580182
2	74311	211453	782981	0.52926481	0.324738253	0.406548876	192.5920956
3	75279	213146.6667	789998	0.528040064	0.326084705	0.406866849	194.1662248
4	76033	212509	789593	0.533088625	0.32795414	0.410007934	192.5799318
5	76922.5	206725.6667	774022	0.545399923	0.339530113	0.421878037	183.4705606
6	75111.5	207950.3333	774074	0.516581051	0.338324825	0.409627315	188.9703082
7	75749.5	211217.3333	785151	0.527472126	0.325959515	0.406564559	193.1184069
8	75886	210290.3333	782643	0.519743251	0.333045066	0.40772434	191.9539559
9	74159	211109	781645	0.508782419	0.329871156	0.401435661	194.7123974
10	74474.5	212556.6667	786619	0.519068866	0.331462997	0.406505344	193.507665
AVG	**75395.35**	**210652.8**	**782749.1**	**0.526115601**	**0.330403687**	**0.408688453**	**191.5270899**

3- Seven RT Flows

Table 19a: QoS OFF, 30 nodes, 7 CBR Traffic: 2 IE, 5 IA

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput	IE Delay (AVG)	IA Delay (AVG)	Overall Delay	Network Power
1	44649.5	244136	1309979	2.438012208	0.240951902	0.868683418	150.8005072
2	44406.5	245789.2	1317759	2.513614429	0.239509033	0.889253432	148.1871144
3	45704.5	243003.4	1306426	2.327612759	0.241575598	0.837586215	155.9751075
4	45174	244364.2	1312169	2.394608255	0.240132853	0.855697253	153.3450055
5	44993	243613.4	1308053	2.545473173	0.240553276	0.899101818	145.4844128
6	44307.5	244907.2	1313151	2.505355838	0.236026745	0.884406486	148.4782191
7	44429.5	244835.4	1313036	2.687299836	0.236130164	0.936464356	140.2120637
8	44502	245571.2	1316860	2.523290912	0.238693058	0.891435302	147.7235642
9	43633.5	243924.2	1306888	2.373134764	0.240346327	0.849714452	153.8031979
10	43521	244219.6	1308140	2.504927431	0.23818361	0.885824702	147.6748162
AVG	**44532.1**	**244436.38**	**1311246.1**	**2.48133296**	**0.239210257**	**0.879816743**	**149.0362749**

Table 19b: QoS ON, 30 nodes, 7 CBR Traffic: 2 IE, 5 IA

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput	IE Delay (AVG)	IA Delay (AVG)	Overall Delay	Network Power
1	45519	247725.8	1329667	2.356871983	0.262738561	0.861062395	154.421678
2	44959.5	245910.6	1319472	2.146230664	0.267018765	0.80393645	164.1264057
3	45703	244427.6	1313544	2.454811261	0.266449369	0.891695623	147.308562
4	44946	247561.6	1327700	2.607892162	0.262800223	0.932826492	142.3308635
5	45176.5	244221.8	1311462	2.188370655	0.267291132	0.816170995	160.6847104
6	45184.5	243113.2	1305935	2.033638905	0.270771966	0.774448234	168.6277974
7	45934.5	245907.4	1321406	2.296485684	0.266741907	0.8466687	156.0711999
8	45501	246782.2	1324913	2.31131594	0.260957903	0.846774485	156.4658624
9	45196.5	245103.2	1315909	2.339894069	0.266545727	0.858930967	153.2031153
10	45233	248827.8	1334605	2.538646441	0.261099311	0.911827063	146.3660221
AVG	**45335.35**	**245958.12**	**1320461.3**	**2.327415776**	**0.265241486**	**0.854434141**	**154.5421979**

2.2 NRT Traffic

1- Three FTP Connections

Table 20a: QoS OFF, 30 nodes, 3 FTP/GENERIC

Seed	IA Throughput (AVG)	IE Throughput	Overall Throughput
1	258645.5	62807	580098
2	207244.5	68756	483245
3	214237.5	67906	496381
4	250733.5	69160	570627
5	253744.5	105253	612742
6	246220.5	70411	562852
7	263572.5	72728	599873
8	213514	69379	496407
9	266058	70427	602543
10	239594	72375	551563
AVG	**241356.45**	**72920.2**	**555633.1**

Table 20b: QoS ON, 30 nodes, 3 FTP/GENERIC

Seed	IA Throughput (AVG)	IE Throughput	Overall Throughput
1	256255	68257	580767
2	195338.5	58545	449222
3	274002.5	64370	612375
4	236592	74687	547871
5	223489.5	60076	507055
6	244320	67172	555812
7	276439	69083	621961
8	238440.5	72935	549816
9	293540.5	61341	648422
10	234647	72560	541854
AVG	**247306.45**	**66902.6**	**561515.5**

2- Five FTP Connections

Table 21a: QoS OFF, 30 nodes, 5 FTP/GENERIC

Seed	**IA Throughput (AVG)**	**IE Throughput (AVG)**	**Overall Throughput**
1	242146	85830.5	898099
2	285062.6667	84033	1023254
3	240008.3333	87644	895313
4	222703.3333	76811.5	821733
5	257398	86926.5	946047
6	334566	69730	1143158
7	244994.3333	79898	894779
8	235846.3333	72078.5	851696
9	290869.6667	88585.5	1049780
10	285103.3333	85039	1025388
AVG	**263869.8**	**81657.65**	**954924.7**

Table 21b: QoS ON, 30 nodes, 5 FTP/GENERIC

Seed	**IA Throughput (AVG)**	**IE Throughput (AVG)**	**Overall Throughput**
1	259305.6667	86912	951741
2	262316	78442.5	943833
3	297510.6667	80942.5	1054417
4	250838.6667	81837	916190
5	265675.6667	88826.5	974680
6	249986	72019.5	893997
7	240154.6667	77262	874988
8	259835.3333	85241.5	949989
9	276350.3333	86269.5	1001590
10	246895.6667	68345	877377
AVG	**260886.8667**	**80609.8**	**943880.2**

3- Seven FTP Connections

Table 22a: QoS OFF, 30 nodes, 7 FTP/GENERIC

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput
1	195996	1307222	1503218
2	157100	1246657	1403757
3	169429	1197237	1366666
4	167178	1181116	1348294
5	160207	1222015	1382222
6	174935	1245932	1420867
7	156273	1235194	1391467
8	187849	1310908	1498757
9	195598	1194483	1390081
10	161245	1277784	1439029
AVG	**172581**	**1241854.8**	**1414435.8**

Table 22b: QoS ON, 30 nodes, 7 FTP/GENERIC

Seed	IE Throughput (AVG)	IA Throughput (AVG)	Overall Throughput
1	162962	1159914	1322876
2	156033	1365412	1521445
3	164975	1235032	1400007
4	179705	1218712	1398417
5	158454	1308479	1466933
6	182795	1247581	1430376
7	185523	1254482	1440005
8	163042	1195158	1358200
9	195056	1235718	1430774
10	170639	1294861	1465500
AVG	**171918.4**	**1251534.9**	**1423453.3**

2.3 Hybrid RT and NRT traffic

1- Three RT Flows+ Three NRT Connections

Table 23a: QoS OFF, 30 nodes, 3 CBR flows (1 IE, 2 IA), 3 FTP/GENERIC (1 IE, 2 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	534027	4307	538334	1.580369756
2	534403	4367	538770	1.577785461
3	531098	4103	535201	1.587239498
4	535091	4048	539139	1.533418756
5	534700	4557	539257	1.635743402
6	535013	3940	538953	1.54304691
7	533483	4530	538013	1.577529325
8	533545	4649	538194	1.608871623
9	536324	3529	539853	1.59565826
10	532015	4643	536658	1.554856076
AVG	**533969.9**	**4267.3**	**538237.2**	**1.579451907**

Table 23b: QoS ON, 30 nodes, 3 CBR flows (1 IE, 2 IA), 3 FTP/GENERIC (1 IE, 2 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	537529	3896	541425	1.479262796
2	535133	5158	540291	1.511040497
3	540666	4728	545394	1.569576264
4	535267	4044	539311	1.539604111
5	537565	4027	541592	1.509624279
6	537724	3843	541567	1.517684833
7	535573	4372	539945	1.566708172
8	534719	4390	539109	1.474814078
9	536368	4567	540935	1.407349432
10	536486	4924	541410	1.429457184
AVG	**536703**	**4761.233333**	**541464.2333**	**1.500512165**

2- Five RT Flows+ Five NRT Connections

Table 24a: QoS OFF, 30 nodes, 5 CBR flows (2 IE, 3 IA), 5 FTP/GENERIC (2 IE, 3 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	749564	62890	812454	0.451365754
2	751485	49790	801275	0.447650623
3	744728	54843	799571	0.454011577
4	745183	51603	796786	0.452597222
5	748124	48565	796689	0.45020088
6	753471	51444	804915	0.448346812
7	751086	51117	802203	0.450184764
8	753007	57015	810022	0.445329446
9	748475	48659	797134	0.453624037
10	748642	55842	804484	0.450863637
AVG	749376.5	53176.8	802553.3	0.450417475

Table 24b: QoS ON, 30 nodes, 5 CBR flows (2 IE, 3 IA), 5 FTP/GENERIC (2 IE, 3 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	763441	53032	816473	0.433521306
2	764671	55194	819865	0.431493066
3	769647	54519	824166	0.426978779
4	767254	57128	824382	0.423702279
5	766108	55674	821782	0.43139794
6	763004	49525	812529	0.428163602
7	761488	48837	810325	0.435335453
8	766520	56800	823320	0.432630693
9	756744	49106	805850	0.436359081
10	766943	51632	818575	0.428327176
AVG	764582	53144.7	817726.7	0.430790938

3- Seven RT Flows+ Seven NRT Connections

Table 25a: QoS OFF, 30 nodes, 7 CBR flows (2 IE, 5 IA), 7 FTP/GENERIC (2 IE, 5 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	1275269	178514	1453783	0.902508908
2	1275510	228932	1504442	0.894027284
3	1292869	203801	1496670	0.917561841
4	1268175	178934	1447109	0.860709319
5	1276150	203912	1480062	0.865111038
6	1278701	225343	1504044	0.810240028
7	1285958	198730	1484688	0.931646157
8	1277079	219362	1496441	0.88013755
9	1276014	193332	1469346	0.843002199
10	1268447	180260	1448707	0.89170142
AVG	**1277417.2**	**201112**	**1478529.2**	**0.879664574**

Table 25b: QoS ON, 30 nodes, 7 CBR flows (2 IE, 5 IA), 7 FTP/GENERIC (2 IE, 5 IA)

Seed	RT Throughput	NRT Throughput	Overall Throughput	Overall AVG Delay
1	1289292	193829	1483121	0.835682785
2	1271091	185073	1456164	0.872566387
3	1281402	169760	1451162	0.84225279
4	1278916	193635	1472551	0.878782328
5	1276652	196514	1473166	0.855717955
6	1280505	221498	1502003	0.89392742
7	1286567	190606	1477173	0.965879176
8	1290229	222548	1512777	0.899815095
9	1274621	203260	1477881	0.813828686
10	1273456	205578	1479034	0.858312878
AVG	**1280273.1**	**198230.1**	**1478503.2**	**0.87167655**

توفير جودة الخدمة المعتمدة على التَجَمُّع في الشبكات المتنقلة العشوائية

إعداد

خالد محمد أحمد حشيدان

المشرف

الدكتور وسام عبدالرحمن المبيضين

Arabic Summary

ملخّـــص

الشبكة المتنقلة العشوائية (MANET)، هي عبارة عن مجموعة من الأجهزة اللاسلكية التي تتصل ببعضها البعض نداً لند. ويمكن إعداد هذه الشبكة في أي زمان وأي مكان حسب الحاجة دون الاعتماد على الشبكات الثابتة المعدة مسبقاً والتي لا تتوفر إلا في أماكن محدودة. وتأتي أهمية الشبكات المتنقلة العشوائية من سهولة وسرعة إعدادها، بالإضافة إلى أهمية التطبيقات التي يمكن أن تستخدم فيها مثل العمليات العسكرية وعمليات الإنقاذ، وتطبيقات التواصل في المؤتمرات العلمية وورشات العمل، بالإضافة إلى التطبيقات اليومية للتواصل بين رجال الأعمال أوالطلاب، وغير ذلك من التطبيقات التي تتطلب سرعة وسهولة الاتصال تحت أي ظروف زمنية أو مكانية.

تتشكل هذه الشبكات من مجموعة من الأجهزة اللاسلكية المتنقلة كالحواسيب المحمولة، أو الهواتف النقالة، أو غير ذلك من نقاط الاتصال اللاسلكية المتنقلة. ويتم التواصل مباشرة بين أي جهازين يبعدان عن بعضهما مسافة لاتزيد عن حد معين، وإذا كانت المسافة بينهما أكثر من الحد الأقصى فإن الأجهزة التي تقع بينهما تقوم بصنع جسر للتواصل وتحويل البيانات بينهما.

يعتبر توفير جودة الخدمة (QoS) في الشبكات اللاسلكية المتنقلة العشوائية من القضايا الهامة وذلك لتحسين جودة الاتصال لمختلف التطبيقات. ويعتبر العلماء والباحثون هذا الموضوع من المواضيع الصعبة نظراً لمحدودية السرعة التي يوفرها الوسط اللاسلكي، بالاضافة الى خصائص الأجهزة المكونة لهذه

الشبكات كالتنقل الحر، والطاقة الكهربائية المعتمدة على البطاريات ذات العمرالقصير، والإمكانيات المحدودة التي تقل عن نظيراتها في الشبكات السلكية.

تم في هذه الرسالة اقتراح وتصميم أسلوب جديد (CBQoS) لدعم وتوفيرجودة الخدمة في الشبكات اللاسلكية المتنقلة العشوائية يعتمد على هيكلية التجمع (Clustering). حيث أن هذه الشبكات يمكن أن تقسم إلى عدد من التجمعات (Clusters) وذلك لتسهيل إدارة موارد الشبكة وتسهيل الاتصالات وعملية توصيل البيانات. يتم انتخاب أحد الأجهزة لكل تجمع (رأس التجمع) حسب اعتبارات معينة تحددها الخوارزمية المستخدمة في بناء التجمعات. ويقوم رأس التجمع بإدارة الاتصال الداخلي بين الأجهزة التابعة له والاتصال الخارجي مع التجمعات الأخرى. وقد بُني الأسلوب المقترح لتوفير جودة الخدمة المعتمدة على التجمع (CBQoS) على أساس توفير أفضلية في الخدمة للاتصالات الخارجية مقارنة بالخدمة المقدمة للاتصالات الداخلية بهدف رفع الكفاءة الإجمالية للاتصالات في الشبكات المتنقلة العشوائية.

تم تصميم النظام المقترح وتقييمه عن طريق المحاكاة في بيئات مختلفة من حيث حجم الشبكة ونوع التطبيقات وعدد الاتصالات المتزامنة. وقد بينت النتائج أن النظام المقترح يحسن الكفاءة الإجمالية للشبكة وذلك بزيادة السرعة وتقليل الوقت اللازم لتوصيل البيانات. حيث تعتبر السرعة والمدة اللازمة لتوصيل البيانات من أهم المقاييس لمعرفة كفاءة شبكات الاتصالات.