1-1-2007

QoS support using SWAN model in mobile ad-hoc networks

Ning Zhang
Ryerson University

QoS Support Using SWAN Model in Mobile Ad-hoc Networks

By

Ning Zhang

A project
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of

Master of Engineering

in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2007

Author's Declaration:

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Ning Zhang

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Ning Zhang

Borrow List

Ryerson University requires the signatures of all persons using or photocopying this thesis.

Please sign below, and give address and date.

Abstract

Mobile Ad-Hoc Network (MANET) is a collection of mobile nodes, dynamically forming a temporary network without pre-existing network infrastructure or centralized administration. Due to the bandwidth constraint and dynamic topology of MANETs, supporting Quality of Service (QoS) in MANETs is a challenging task. MANETs have certain unique characteristics that pose several difficulties in provisioning QoS. Most routing protocols for MANETs are designed without explicitly considering QoS of the routes. QoS-aware routing requires to find a route that satisfies the end-to-end QoS requirement. QoS in MANETs is a rapidly growing area of research interest.

In this report, the challenges of QoS support for MANETs are discussed first. Then the current research on QoS support in MANETs is reviewed, followed by extensive discussion and analysis of QoS models and QoS routing. Finally, one of the QoS models - SWAN is studied to provide a qualitative assessment of the applicability of the model.

Acknowledgements

I would like to thank Dr. Anpalagan for supporting me and encouraging me in my thesis and study work, and also for his continued guidance throughout my Masters program. I am extremely fortunate and grateful to have enjoyed the benefits of his brilliance and generosity. He has been a source of wisdom and great mentor to me all the time.

I thank my immediate family, especially my husband, for his immeasurable understanding, emotional support, companionship, balance, depth, and ongoing dedication to our life together. I will always feel utterly blessed.

Table of Contents

1 Introduction 1
1.1 Mobile Ad Hoc Networks Overview 2
1.1.1 MANETs Characteristics, Challenges and Architectural Choices 2
1.1.2 Routing Protocols in MANETs 5
1.2 Quality of Service 7
1.2.1 QoS Fundamentals 7
1.2.2 QoS Provision in MANETs 7
1.3 Report Organization 8
1.4 Contribution 8
2 QoS Models 9
2.1 IntServ 9
2.2 DiffServ 11
2.3 FQMM 13
2.4 INSIGNIA 14
2.5 SWAN 17
2.6 Conclusion 19
3 QoS-Aware Routing in MANETs 20
3.1 Challenges of QoS-aware Routing 20
3.2 QoS Routing Protocols Overview 24
3.3 Related Work 25
3.4 Main QoS-Aware Routing Protocols 28
3.4.1 Ticket-based QoS Routing 28
3.4.2 Bandwidth Constraint QoS Routing 30
3.4.3 Delay Constraint QoS Routing 32
3.4.4 Location Based QoS Routing 34
3.4.5 Hierarchical Routing: CEDAR 35
3.4.6 Application-aware QoS Routing 38
3.5 Comparison of QoS Routing Protocols 39
4 A Simulation Study on the Performance of SWAN Model 41
4.1 Introduction 41
4.2 Simulation Environment 42
4.2.1 Overview 42
4.2.2 Tools 42
4.2.3 Measurements 45
4.2.4 Simulation Parameters 46
4.2.5 Rate and Admission Control 49

4.3 Simulation Results and Analysis..50
4.3.1 Delay Analysis..50
4.3.2 Throughput Analysis..56
4.3.3 Control Packet Overhead..58
4.3.4 SWAN Model Parameters..60
4.3.5 Node Mobility..66
4.4 Conclusions..69
5 Conclusions..71
5.1 Summary..71
5.2 Future Work..71
Appendix:..74
References:..75

List of Tables

Table 1: Comparisons of flat architecture and hierarchical architecture. 5
Table 2: QoS classes and mappings. 12
Table 3: Comparison of IntServ and DiffServ. 12
Table 4: Comparison of in-band and out-of-band signaling. 14
Table 5: Packet delay (in seconds) comparison of SWAN and BE model. 18
Table 6: Comparison of QOS routing protocols. 40

List of Figures

Figure 1: Flat vs. Hierarchical (multi-tier) network. 4
Figure 2: IntServ architecture foundations. 10
Figure 3: DiffServ architecture foundations 11
Figure 4: FQMM architecture. 13
Figure 5: INSIGNIA QoS framework. 15
Figure 6: SWAN model. 17
Figure 7: An example of hidden route problem. 21
Figure 8: Hidden and exposed terminal problem in band-width calculation. 22
Figure 9: General QoS Model in MANETs 25
Figure 10: Cross layer framework. 27
Figure 11: Illustration of ticket-based QoS routing. 29
Figure 12: Three cases in bandwidth calculation 31
Figure 13: Location prediction at a future time using last 2 updates 34
Figure 14: Examples of core broadcast 36
Figure 15: Network topology of application-aware routing 38
Figure 16: Simulation process. 43
Figure 17: A screen shot of the simulated network. 47
Figure 18: End-to-end delay of best effort traffic vs TCP flows. 50
Figure 19: End-to-end delay of voice traffic vs. TCP flows. 51
Figure 20: End-to-End delay of video traffic vs. TCP flows. 52
Figure 21: End-to-end delay of TCP traffic with DSDV protocol. 53
Figure 22: End-to-end delay of real-time traffic with DSDV protocol. 54
Figure 23: The comparison of DSDV and AODV protocol in end-to-end delay. 55
Figure 24: BE traffic throughput vs. TCP flows using DSDV protocol. 56
Figure 25: RT traffic throughput vs. TCP flows using DSDV protocol. 57
Figure 26: Normalized control packets overhead as a function of TCP flows. 58
Figure 27: The impact of increment rate on end-to-end delay for BE traffic. 60
Figure 28: The impact of increment rate on end-to-end delay for RT traffic. 61
Figure 29: The impact of increment rate on throughout for BE traffic. 62
Figure 30: The impact of decrement rate on the end-to-end delay. 63
Figure 31: The impact of decrement rate on throughput. 64
Figure 32: The impact of admission control rate on the average end-to-end delay. 65
Figure 33: Illustration of node mobility in NS-2. 67
Figure 34: Goodput of the real-time traffic as a function of node speed. 68

Figure 35: Goodput vs node mobility (pause time). 69

List of Abbreviations

ACK	Acknowledgement
AIMD	Additive Increase, Multiplicative Decrease
AODV	Ad-hoc On-demand Distance Vector
BE	Best Effort
CBR	Constant BitRate
CEDAR	Core-Extraction Distributed Ad-hoc Routing
CTS	Clear to Send
DiffServ	Differential Service
DoS	Denial of Service
DS	Differentiated Services
DSCP	DiffServ Code Point
DSDV	Destination Sequenced Distance-Vector
ECN	Explicit Congestion Notification
FQMM	Flexible QoS Model for MANETs
GPS	Global Positioning System
IETF	Internet Engineering Task Force
INSIGNIA	In-band Signaling Support for QoS In Mobile Ad-hoc Networks
IntServ	Integrated Service
ISP	Internet Service Provider
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MPR	MultiPoint Relay
NSI	Network Status Information
ODRP	On-Demand Delay-Constraint Routing Protocol
OLSR	Optimized Link State Routing
PHB	Per-Hop Behaviour
QoS	Quality of Service
RSVP	ReSerVation Protocol
RT	Real Time
RTCP	Real-time Transport Control Protocol

RTP	Real-time Transport Protocol
RTS	Request to Send
SLA	Service Level Agreement
SWAN	Service differentiation in Stateless Wireless Ad-hoc Network
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
VC	Virtual Connection
WRR	Weighted Round Robin
WSR	Wireless Routing Protocol
ZRP	Zone Routing Protocol

1 Introduction

Mobile Ad-hoc Networks (MANETs) are zero configured, self organizing, and highly dynamic networks formed by a set of mobile hosts connected through wireless links. These networks can form “on the fly” without requiring a fixed structure. MANETs were initially proposed for military applications such as battlefield communications and disaster recovery. But the evolution of multimedia technology and commercial interest to widely reach civilian applications made Quality of Service (QoS) in MANETs a more and more important issue. These applications are typically delay-sensitive and have high bandwidth requirements. Although much progress has been done to satisfy the QoS provisions in MANETs, there are still many problems due to the nature and constraints of MANETs.

To support the diverse applications, it is necessary for MANETs to have an efficient routing and QoS mechanism. However, there are a number of technical challenges because of the network restrictions such as dynamically and unpredictably varying topology resulting from nodal mobility, multi-hop communications, contentions from channel access and a lack of central coordination. The dynamic nature of MANETs makes difficult to apply traditional QoS management techniques to negotiate quality between users and networks. The pure IP solutions developed for infrastructure-based networks have shown to be inadequate. And the most commonly used routing algorithms are based on the “shortest path” which is always based on old information. It means each router bases its routing decisions on potentially incorrect assumptions about the network without any QoS considerations. Thus providing QoS guarantees in MANETs remains an open issue. Recent research on QoS provisions mainly addresses this critical issue on QoS models and QoS-aware routings. This report will be providing an extensive survey on these two aspects in QoS provisions in MANETs and performance results using SWAN QoS model.

1.1 Mobile Ad Hoc Networks Overview

The Internet Engineering Task Force (IETF), the body responsible for guiding the evolution of the Internet, provides the definition as given below [1]: A mobile ad hoc network (MANETs) is an autonomous system of mobile routers (and associated hosts) connected by wireless links. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a stand-alone fashion, or may be connected to the larger Internet.

1.1.1 MANETs Characteristics, Challenges and Architectural Choices

In this section, the MANETs characteristics are presented first and due to these characteristics and constraints of MANETs, the challenges of MANETs are discussed. Then the two commonly chosen architecture for MANETs: flat and hierarchy are reviewed and compared.

Characteristics:

Dynamic topologies: Nodes are free to move arbitrarily; thus, the network topology which is typically multi-hop may change randomly and rapidly at unpredictable times, and may consist of both bidirectional and unidirectional links.

Bandwidth-constrained: Wireless links will continue to have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications after accounting for the effects of multiple access, fading, noise, and interference conditions is often much less than a radio's maximum transmission rate. And the congestion in the network is another side effect for the already low link capacities. As the mobile network is often simply an extension of the fixed network infrastructure, the demand for bandwidth will continue to increase as multimedia computing and collaborative networking applications rise.

Energy-constrained operation: Some or all of the nodes in a MANETs may rely on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.

Limited physical security: Mobile wireless networks are generally more prone to physical security threats than are fixed cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service (DoS) attacks should be carefully considered. The majority protocols proposed for MANETs assume a trustworthy collaboration among participating devices and hence introduce security threats, some arising from shortcomings in the protocols, and others from the lack of conventional identification and authentication mechanisms. These inherent properties of MANETs make it possible for the malicious nodes to launch various kinds of attacks.

Challenges:

Because MANETs have these certain unique characteristics, it causes difficulties for providing QoS in such networks. The unique characteristics are dynamically varying network topology, lack of precise state information, shared radio channel, limited resource availability, hidden terminal problem, lack of central control and insecure medium [2].

Dynamically varying network topology: In MANETs, nodes are mobile and network topology is changing dynamically. Consequently, the route which is already set up with required QoS could not satisfy QoS anymore if one of the nodes on this established route moves. The information of the loss of QoS will be sent to all the sources to find another possible QoS-aware route, and hence causes delay and makes the process unacceptable.

Lack of precise state information: Due to the dynamic characteristics, information of nodes transmitted to other nodes may change right after this information is transmitted to its neighbors. The information here can be the data rate available at the neighboring node, since available data rate of nodes is affected by the data rate. As a result, this information which is already transmitted may have been out of date and it may lead to a wrong routing decision.

Shared radio channel: Data transmitted on the radio channel can be received by stations which are in the carrier sensing range of the transmitter. This broadcast characteristic will cause interference to other stations when traffic is transmitted over the air interface. Thus, stations have to share channel with neighbors in their carrier sensing range.

Limited resource availability: The resources such as data rate, battery life, and storage space are all very limited in ad hoc networks. The data rate is very limited for wireless links if we compared it with the data rate available in wired network. In addition, the basic characteristics of the wireless channel e.g. fading, noise, and shared data rate between neighbor nodes (neighbor nodes have to keep silent when it senses some node is transmitting) will also degrade the wireless data rate. As a result, it is hard for a wireless network to provide too high data rate which could be provided by the wired network. It also brings problem of cooperation between wireless network and wired network.

Architecture Choices:

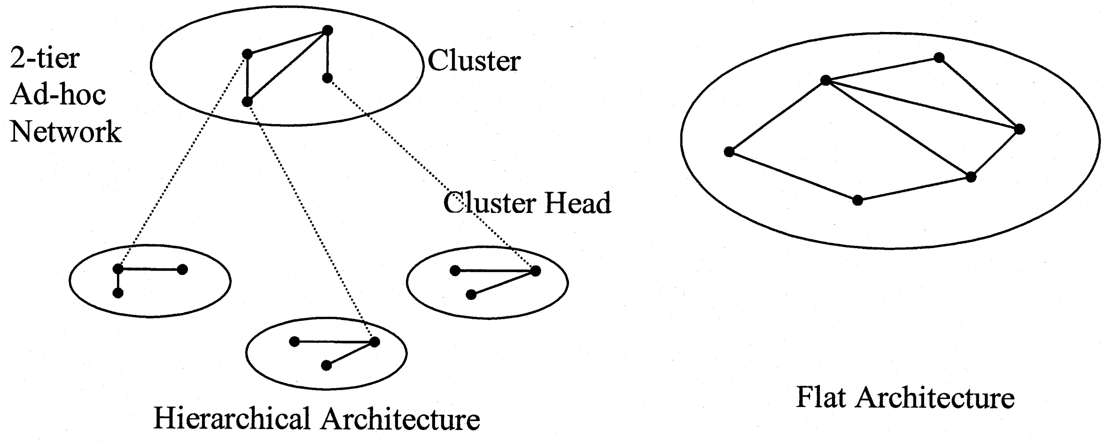


Figure 1: Flat vs. Hierarchical (multi-tier) network.

MANETs could have two different network topologies as depicted in Figure 1: flat and hierarchical architecture. In a flat network design, each node has essentially the same job. A flat network topology is adequate for very small networks and is easy to design, implement and maintain as long as the network stays the same. When the network grows, however, a flat network becomes undesirable and hierarchical network architecture becomes a better choice. In a hierarchical network, the nodes are divided into layers or clusters. Each cluster could have a cluster head which is mainly responsible for the route calculation and communication. Table 1 is the pros and cons of the two different network architectures.

Flat architecture	**Hierarchical architecture**
• Increased reliability/survivability – No single point of failure – Alternative routes in the network • More "optimal routing" • Reduced use of the wireless resources (better coverage) • Better load balancing property (route diversity) • All nodes have one type of equipment	• Easier mobility management procedures (just ask the cluster head) • Better manageability

Table 1: Comparisons of flat architecture and hierarchical architecture.

1.1.2 Routing Protocols in MANETs

Routing protocols in ad hoc networks vary depending on the type of the network. Typically, ad hoc network routing protocols are classified into three major categories based on the routing information updated mechanism. They are proactive (table driven routing protocols), reactive (on-demand routing protocols) and hybrid routing protocols. In addition, protocols can also be classified according to the utilization of specific resources, such as power aware routing protocol and load aware routing protocols and so on.

In proactive routing protocols, routes are calculated independent of intended traffic. All the routes from one station to other stations in the network are calculated and saved in the routing table of each node. Once there is a need of transmission, source node would check from the routing table and the route will be set immediately. Some of the used proactive routing protocols used in ad hoc networks are Optimized Link State

Routing protocol (OLSR) [3] and Destination Sequenced Distance-Vector routing protocol (DSDV) [4].

In table driven routing protocols, to update the table, periodic flood is required which costs too much data rate to transmit the topology information. The main motivation of the designing of on demand routing protocols is to reduce the routing overhead in order to save bandwidth in ad hoc networks. On demand routing protocols execute the path finding process and exchange routing information only when there is a requirement by the station when it wants to initialize a transmission to some destination.

The tradeoffs between proactive and reactive routing strategies are quite complex. Which approach is better depends on many factors, such as the size of the network, the mobility, the data traffic and so on. Proactive routing protocols try to maintain routes to all possible destinations, regardless of whether or not they are needed. Routing information is constantly propagated and maintained. In contrast, reactive routing protocols initiate route discovery on the demand of data traffic. Routes are needed only to those desired destinations. This routing approach can dramatically reduce routing overhead when a network is relatively static and the active traffic is light. However, the source node has to wait until a route to the destination can be discovered, increasing the response time.

A hybrid routing protocol like Zone Routing Protocol (ZRP) [5] is designed to effectively combine the advantages of both proactive and reactive routing protocols. The key concept used in this protocol is to use a proactive routing within a zone in the *r*-hop neighborhood of every node and use a reactive routing for nodes outside this zone. The table driven scope is limited within a zone and when a destination is out of the table driven scope, on demand routing search is initiated. In this situation, control overhead is reduced, compared to both the route request flooding mechanism employed in on demand protocols and periodic flooding of routing information packet in table driven protocol. However, the complexity makes the dynamically adjusted routing strategies hard to implement.

1.2 Quality of Service

1.2.1 QoS Fundamentals

QoS is defined as a set of service requirements that need to be met by the network while transporting a packet stream from a source to its destination. A QoS enabled network shall ensure that its applications and/or their users have their QoS parameters fulfilled, while at the same time ensuring an efficient resource usage and cost efficiency and that the most important traffic still has its QoS parameters fulfilled during network overload.

QoS metric can be: Bandwidth - the rate at which an application's traffic must be carried by the network; Latency - the delay that an application can tolerate in delivering a packet of data; Jitter - the variation in latency; Loss - the percentage of lost data.

1.2.2 QoS Provision in MANETs

QoS provision in ad hoc networks does not depend on any single network layer but on the coordinated efforts from all layers. Many research works are going on for QoS support in MANETs. The major approaches are QoS Model, QoS Resource Reservation Signaling, QoS Routing and QoS Medium Access Control (MAC). These approaches are closely related. A QoS model specifies the architecture in which some kinds of services can be provided in MANETs. QoS signaling is used to reserve and release resources such as buffer space, bandwidth, and to setup, tear down and re-negotiate flows in the networks. QoS routing protocols try to find a path that has a good chance of meeting the QoS requirements. QoS MAC protocol is an essential component in QoS support in MANETs. All the upper-layer QoS components (QoS routing and QoS signaling) are dependent on it and they coordinate with the MAC protocol. Finally, other QoS components such as scheduling and admission control can be borrowed from other network, where scheduling helps in serving multiple connections through one link and admission control decides to serve (or not) requested connections.

1.3 Report Organization

The structure of this project report is organized as follows. Chapter 2 will discuss five major QoS models in MANETs which include IntServ, DiffServ, FQMM, INSIGNIA and SWAN. In Chapter 3, the challenges of QoS-aware routing will be presented. Then an overview and related work that has been done recently will be provided, followed by some methods of classifying QoS-aware routing protocols. A comparison of these routing protocols will be given afterwards. Chapter 4 will be on the simulation study of one of the QoS models, SWAN using the NS-2 simulator. The performance of SWAN model will be evaluated using different ad-hoc protocols (DSDV and AODV), with different parameters (increment rate, decrement rate) under varying conditions (node mobility) and varying traffic (TCP, UDP, different number of connections/streams) to provide a qualitative assessment of the applicability of the model in different scenarios.

1.4 Contribution

In this report, QoS provisioning of mobile ad hoc networks is addressed. The major contributions of the work are as follows:

This report offers a survey of most major solutions to QoS provisions in MANETs. A thorough overview of QoS metrics, QoS models and QoS routings for MANETs is provided.

This report examines in detail how the SWAN mechanisms work with other protocols to provide QoS guarantees. Then it quantitatively shows the performance of SWAN mechanisms interacting with other routing protocols including AODV and DSDV for MANETs. Through the comparison of the SWAN model and best effort model in MANETs, it is verified that SWAN mechanism has better performance than the best-effort model in terms of QoS provisions such as end-to-end delay for real-time traffic and throughput of the network. An explanation for such behaviors is also provided afterwards.

2 QoS Models

A QoS model specifies the architecture which will enable us to offer services that operate better than the current "best effort" model that exists in MANETs. All other QoS components (such as QoS signaling, QoS routing and QoS MAC) must cooperate together to achieve this goal. Therefore, the QoS model should be the first matter to consider for the QoS support in MANETs. This architecture should also take into consideration the challenges of MANETs e.g. dynamic topology and time-varying link capacity. In addition, the potential commercial applications of MANETs require connection to the Internet. Thus QoS model for MANETs should also consider the existing QoS characteristics in the Internet. The rest of this chapter will discuss five representative QoS models namely: IntServ, DiffServ, FQMM, INSIGNIA and SWAN.

2.1 IntServ

The Integrated Service (IntServ/IS) [6] model includes four components: the packet scheduler, the admission control routine, the classifier, and the reservation setup protocol. Another important concept "flow" is defined as distinguishable stream of related datagrams that results from a single user activity and requires the same QoS. It is the finest granularity of packet stream distinguishable by the IntServ. The basic idea of the IntServ model is that the flow-specific states are kept in every IntServ-enabled router. A flow-specific state should include bandwidth requirement, delay bound, and cost of the flow. IntServ architecture allows sources to communicate their QoS requirements to routers and destinations on the data path by means of a signaling protocol such as ReSerVation Protocol—RSVP [7]. IntServ proposes two service classes in addition to best-effort (BE) service. One is guaranteed service; the other is controlled load service. The Guaranteed Service is provided for applications requiring strict delay bound. The Controlled Load Service is for applications requiring reliable and enhanced BE service.

Figure 2 shows how these components work together to provide integrated services. For the purpose of traffic control, each incoming packet must be mapped into some class; all packets in the same class get the same treatment from the packet scheduler. This mapping is performed by the classifier. A classifier must be both general and efficient. The output driver implements the packet scheduler. The basic function of packet scheduling is to reorder the output queue in which packets are ordered by priority, and highest priority packets always leave first. Admission control implements the decision algorithm that a router or host uses to determine whether a new flow can be granted the requested QoS without impacting earlier guarantees.Because every router keeps the flow state information, the quantitative QoS provided by IntServ is for every individual flow. The amount of state on each node scales in proportion to the number of concurrent reservations, which can be potentially large on high-speed links. This model also requires application support for the RSVP signaling protocol.

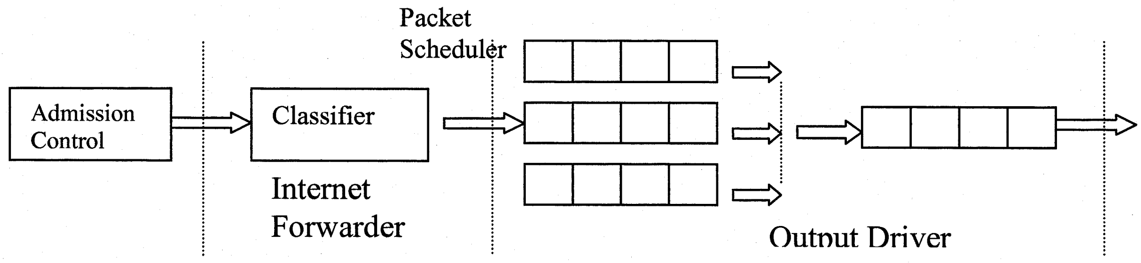


Figure 2: IntServ architecture foundations.

IntServ/RSVP model is not suitable for MANETs due to several factors: 1) Scalability: IntServ/RSVP based on per-flow resource reservation is not appropriate for MANETs because of the frequently changing topology and limited resources in MANETs resulting in more signaling overhead and unaffordable storage and computing process for mobile nodes. 2) Signaling: The RSVP reservation and maintenance process is a network consuming procedure. Thus RSVP signaling packets will grapple with the data packets for resources and more specifically for bandwidth. This happens because RSVP is an out-of-band signaling protocol. 3) Router mechanisms: IntServ imposes high requirement on routers. All routers must have the four basic components: RSVP, admission control routine, classifier, and packet scheduler. Consequently, the processing overheads of routers are high which is undesirable in power-constrained MANETs.

2.2 DiffServ

Differentiated Services (DiffServ/DS) [8] architecture is based on a simple model by implementing complex classification and conditioning functions only at network boundary nodes, and by applying per-hop behaviors to aggregates of traffic which have been appropriately marked using the DS field (the IPv4 header TOS octet or the IPv6 Traffic Class octet) in the headers. Each behavior aggregate is identified by a single DS codepoint (DSCP). Figure 3 is the architecture for differentiated services which is composed of two key components within a differentiated services region: traffic classification and conditioning functions. The traffic classification policy identifies the subset of traffic which may receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates within the DS domain. Traffic conditioning performs metering, shaping, policing and/or re-marking to ensure that the traffic entering the DS domain conforms to the rules specified in the traffic conditioning agreement (TCA). The classification and conditioning of traffic is only done at the edge router whilst within the core of the network, packets are forwarded according to the per-hop behavior (PHB) associated with the DSCP.

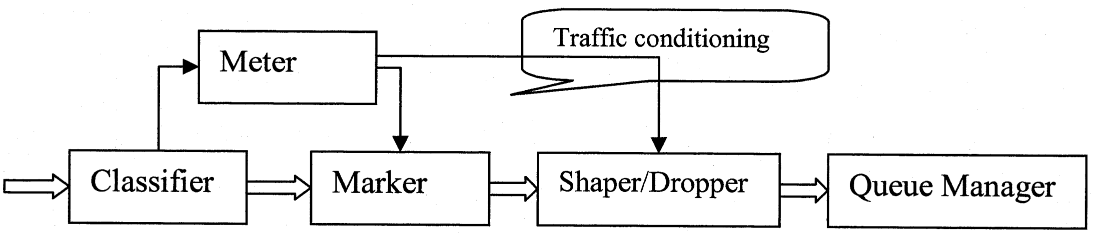


Figure 3: DiffServ architecture foundations

DiffServ on the other hand is a lightweight model for the interior routers since individual state flows are aggregated into a set of flows. This makes routing a lot easier in the core of the network. Thus this model could be a potential model for MANETs. However, there are some drawbacks of this architecture that hinder the DiffServ deployment in MANETs. First, since DiffServ is designed for fixed wire networks, it is easy to identify the boundary routers and the interior routers. But in MANETs, it is ambiguous as to what the boundary routers and interior routers are, and every node should have the functionality as both, hence it is hard to define a DS domain where the flows aggregation is performed. Second, DiffServ is scalable but it does not guarantee services on end-to-end basis. This drawback would again take us back to the IntServ model where several separate flow states are maintained, causing a heavy storage cost in every node. Moreover the concept of the Service Level Agreement (SLA), defined in wire-based QoS models is not more applicable. SLA basically defines the contract between a service provider and a customer that specifies the forwarding service a customer should receive. In a completely ad-hoc topology where there is no concept of service provider and customer, how to make a SLA in MANETs is quite difficult because there is no obvious scheme for mobile nodes to negotiate the traffic rules.

Table 2 shows the three defined QoS classes together with their mappings to IntServ and DiffServ services. The first class has the highest priority and corresponds to applications with real-time traffic such as voice, with their high delay constraints. The corresponding service of this class in DiffServ is referred to "expedited forwarding" and in IntServ to guaranteed service. The second class has less priority and is suitable for applications requiring high throughput such as some transaction-processing applications. The least priority class has no specific constraint and is referred to the best effort in both architectures. Table 3 is a comparison of IntServ and DiffServ architecture.

Priority Class	IntServ	DiffServ
1st class e.g. voice, low delay	Guaranteed	Expedited Forwarding
2nd class e.g. video, high throughput	Controlled Load	Assured Forwarding
3rd class e.g. data, no constraint	Best Effort	Best Effort

Table 2: QoS classes and mappings.

Criteria	IntServ	DiffServ
Granularity	Individual flow	Aggregate of flows
State in routers	Per-flow	Per-aggregate
Classification	Header fields	DS field
Signaling	Required(RSVP)	Not required
Coordination	End-to-end	Per-hop
Scalability	<# of flows	<# of classes

Table 3: Comparison of IntServ and DiffServ.

2.3 FQMM

Flexible QoS Model for MANETs (FQMM) [9] is the first QoS model proposed for MANETs in 2000 by Xiao et al. The model can be viewed as a hybrid of IntServ and DiffServ model. The basic idea of FQMM is that it uses both the per-flow state property of IntServ for highest priority traffic class and the per-class provisioning of DiffServ for other priority classes. This model is based on the assumption that not all packets in the network are actually seeking for highest priority, because this model would then result in a similar model with IntServ where we have per-flow provisioning for all packets. FQMM is for small to medium size MANETs, with less than 50 nodes, using a flat non-hierarchical topology. Simulation results show that FQMM achieves better performance in terms of throughput and service differentiation than the best-effort model.

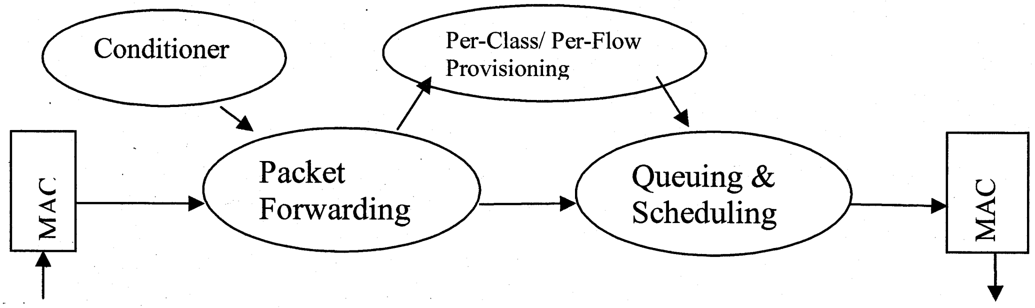


Figure 4: FQMM architecture.

In FQMM, three types of nodes are defined, as in DiffServ: a) ingress, if it is transmitting data, b) interior, if it is forwarding data and c) egress, if it is receiving data. Figure 4 illustrates the FQMM architecture. A traffic conditioner is put at the ingress node where the traffic originates. It polices the traffic according to the traffic profile after a valid route is found. Components of the conditioner include traffic profile, meter, marker and dropper. For FQMM, the absolute traffic profile is not applicable since the effective bandwidth of a wireless link between nodes is time-varying. Thus, the traffic profile is defined as the relative percentage of the effective link capacity, in order to keep the differentiation between classes predictable and consistent under the dynamics of the network. Link bandwidth sharing and buffer allocation are two important aspects of resource management. The former is done by the scheduler which decides the opportunities of flows for link access and the latter holds the valid packets when necessary and drop some packets from the buffer in case of network congestion. Together they achieve the target QoS requirements.

FQMM is the first attempt at proposing a QoS model for MANETs with the following problems: 1) without an explicit control on the number of services with per-flow granularity, the scalability problem still exists, 2) FQMM actually lacks the counterpart to DiffServ's service level agreements, and it remains an open question how the ingress nodes should determine the dynamic parameter for their token bucket metering, 3) the ingress nodes have to take great care in regulating their traffic, since the rate of in-profile traffic must be processable in all network regions, including bottleneck areas where traffic from different sources accumulates.

2.4 INSIGNIA

INSIGNIA [10], in-band signaling support for QoS in MANETs, is a new signaling system for supporting quality of service in mobile ad hoc networks. The term "in-band signaling" refers to the fact that control information is carried along with data in IP packets. In-band signaling is more suitable than explicit out-of-band approaches for supporting end-to-end QoS in highly dynamic environments where network topology, node connectivity and end-to-end QoS are strongly time-varying. Table 4 is the comparison of in-band and out-of-band signaling.

Criteria	In-band signaling	Out-of-band signaling
Control information	Piggybacked into the packet header	Explicit control packets
Scalable	Lightweight and but not scalable	Heavy weight but scalable
Path	Follow the data path	Separate path
Priority	Same as data packet	High priority

Table 4: Comparison of in-band and out-of-band signaling.

INSIGNIA plays a central role in the resources allocation and management between source and destination mobile nodes. The system supports the delivery of adaptive real-time flows and is capable of providing fast reservation, restoration and adaptation services. Reservations are locally cached at each node and managed using soft-state techniques. Based on availability of end-to-end resources, wireless flow management attempts to provide assurances for the minimum (MIN) or maximum (MAX) bandwidth needs depending of resource availability. In addition to supporting adaptive real-time services, the service model also supports IP best-effort packet delivery when the intermediate bottleneck nodes can not satisfy the QoS requirements for the applications.

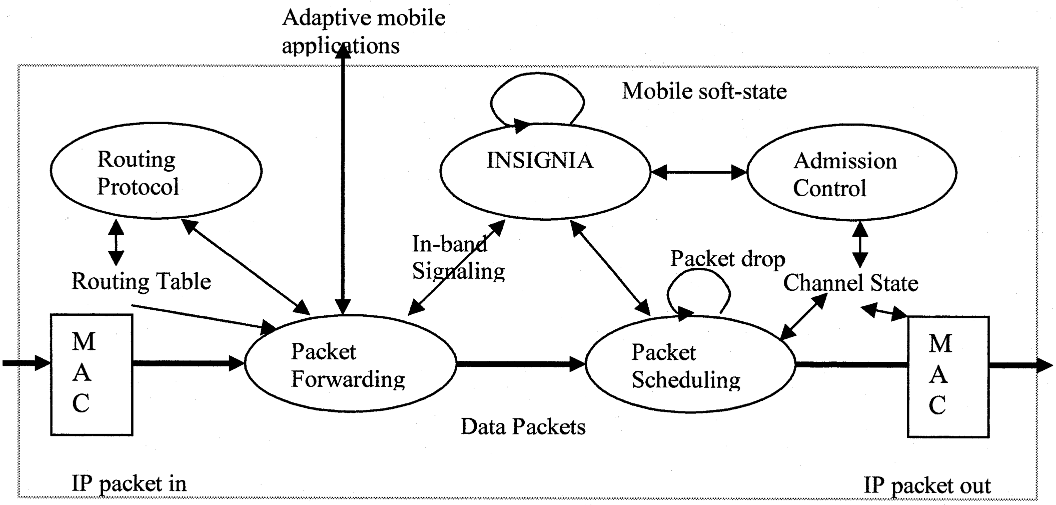


Figure 5: INSIGNIA QoS framework adapted from [7].

Figure 5 shows the position and the role of INSIGNIA in wireless flow management at a mobile host. The packet forwarding module classifies the incoming packets and forwards them to the appropriate modules (routing, INSIGNIA, local applications, and packet scheduling modules). If a received IP packet includes an INSIGNIA option, the control information is forwarded to and processed by the INSIGNIA module. In the meantime, the received packet is delivered to a local application or forwarded to the packet scheduling module according to the destination address in the IP header. Before the packets are sent through the MAC component, a packet scheduling module is used to schedule the output of the flows in order to fairly allocate the resource to different flows. In INSIGNIA, a Weighted Round-Robin (WRR) discipline that takes location dependent channel conditions into account is implemented.

The INSIGNIA module is responsible for establishing, restoring, adapting and tearing down real-time flows. Its operations include flow setup, QoS reporting, soft-state management, flow reservation, restoration and adaptation algorithms which are specifically designed to deliver adaptive real-time service in MANETs. To establish real-time flows, source nodes initiate reservations by setting the appropriate field in the IP option in data messages before forwarding 'reservation request' packets on toward destination nodes. A reservation request packet carried a reservation mode (REQ), service type (RT), a valid payload and a MAX/MIN bandwidth requirement. Reservation packets traverse intermediate nodes executing admission control modules, allocating resources and establishing flow state at all nodes between source-destination pair. The bandwidth indication is set to MAX if all nodes between the source destination pair have successfully allocated resources to meet the bandwidth requirements. Otherwise, the bandwidth indication is set to MIN which indicates the path can only support the minimum bandwidth and the service type is flipped back from RT to BE. QOS reports are periodically sent to source node for the purpose of completing flow establishment and managing adaptation. The QOS reports are forwarded in best effort manner and do not have to travel on the reverse path toward the source. The resources are managed in a soft-state approach which is well suited in dynamic environment. INSIGNIA also adopts a flow restoration method to re-establish reservation as quickly and efficiently as possible. In an ideal case, the restoration of flows can be accomplished within the duration of a few consecutive packets and it is called a fast restoration. Otherwise, the reserved flow may get degraded to best effort service or even the packets have to be dropped if it causes service disruption. These components work together in INSIGNIA to guarantee the timely delivery of the real-time flows.

2.5 SWAN

Service Differentiation in Stateless Wireless Ad hoc Networks (SWAN) [11] is a stateless network QoS model which uses distributed control algorithms with additive increase multiplicative decrease (AIMD) rate control mechanism for TCP traffic and sender-based admission control mechanism for UDP traffic to deliver service differentiation in MANETs. Explicit congestion notification (ECN) is used to dynamically regulate admitted real-time traffic in the face of network dynamics such as mobility and temporary traffic overload. Intermediate nodes do not keep per-flow state information in SWAN wireless networks. As a result, there is no need for signaling or complex control mechanisms to update, refresh, and remove per-flow state information, as is the case with "stateful" mobile ad hoc networks. Changes in topology and network conditions, even node and link failure do not affect the operation of the SWAN control system. This makes the system simple, robust, and scalable. A rate control mechanism uses the MAC delay measurements from packet transmissions as feedback, while a source-based admission control mechanism uses rate measurements from aggregated real-time traffic as feedback.

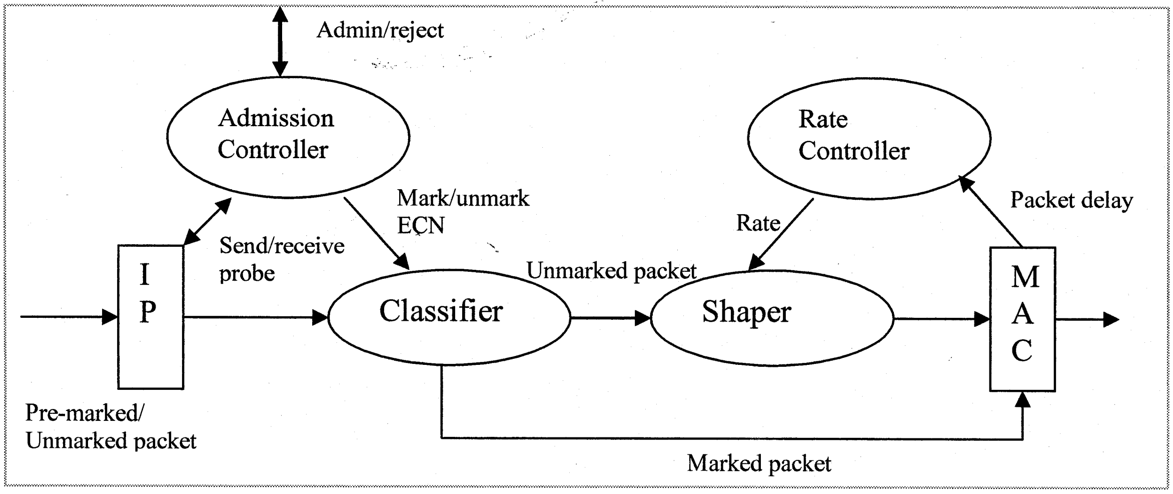


Figure 6: SWAN model adapted from [11].

Figure 6 depicts how the SWAN model uses those control mechanisms to regulate real-time and best-effort traffic. A classifier and a shaper operate between the IP and MAC layers. The classifier is capable of differentiating real-time and best-effort packets, forcing the shaper to process best-effort packets but not real-time packets. The shaper represents a simple leaky bucket traffic shaper. The goal of the shaper is to delay best-effort packets in conformance with the rate calculated by the rate controller. What makes such a stateless approach work is that all nodes independently regulate best-effort traffic and each source node uses admission control for real-time sessions. When a new real-time session is admitted, the packets associated with the admitted flow are marked as RT. The classifier looks at the marking and, if the packet is marked as RT, the packet will bypass the shaper mechanism, remaining unregulated. Here, there is an implicit assumption that a source node regulates its real-time sessions based on its admission control decision.

The following table is obtained from the simulation study later in this report. From the simulation, we can conclude that SWAN guarantees a small delay for the real-time traffic. The average delay of the best-effort traffic is close to 0.3 seconds in the BE model, while it is as large as 1.26 seconds in SWAN. The provision of guaranteed throughput and delay for RT flows is at a cost of large delay deviation of BE traffic.

Statistics	BE Model	SWAN Model
Voice	1.82	0.31
Video	0.81	0.15
TCP	0.31	1.26

Table 5: Packet delay (in seconds) comparison of SWAN and BE model.

What remains unclear is how the amount of bandwidth available for real-time traffic should be chosen in a sensible way: Choosing larger values results in a poor performance of real-time flows and starvation of BE flows, and choosing it too low results in the denial of real-time flows for which the available resource would have sufficed. And also, the model has no flexibility to tolerate channel dynamics such as node mobility. Source nodes, for example, that have been previously admitted flows are unaware of node mobility and the re-routing of flows through new intermediate nodes that may have insufficient resources to support previously admitted traffic. False admission is another example which is a result of multiple source nodes simultaneously initiating admission control at the same instance and sharing common paths and nodes between source-destination pairs. Because intermediate nodes do not maintain state information and admission control is conducted at the edge/source node in a fully decentralized manner, each source node may receive a response to their probe message indicating that resources are available when in fact they are not. Thus though SWAN can be a candidate QoS model, it can not be a complete QoS solution for a highly dynamic network like MANETs.

2.6 Conclusion

In this chapter, five different QoS models: IntServ, DiffServ, FQMM, INSIGNIA and SWAN were discussed in detail. In these QoS models, certain routing protocols, algorithms and implementation are not specified, but the methodology and architecture to provide certain types of services were presented. There are also other architectures that could adopt a hybrid mechanism mentioned above to guarantee the QoS provisions in MANETs. Since achieving QoS in MANETs not only rely on these models, all the components such as QoS routing algorithms, QoS signaling and QoS MAC protocol must work together to ensure this. In the next chapter, different QoS-aware routing mechanisms will be presented and compared.

3 QoS-Aware Routing in MANETs

In any given network, there are two types of flows: best effort flows which requires the data to be reliably delivered to the destination and QoS flows which apart from reliability, requires some additional constraints like available bandwidth, delay, etc. to be satisfied. Reusing best effort routing methods for QoS routing is not feasible since best-effort routing performs these tasks based on a single measure, usually hop-count while QoS routing, however, must take into account multiple QoS measures and requirements. This section takes a look at the different QoS-aware routings in MANETs from different perspectives including its challenges, classifications, algorithms and comparisons.

3.1 Challenges of QoS-aware Routing

Providing QoS is more difficult for MANETs due to at least two reasons. First, unlike wired networks, radios have broadcast nature. Thus, each link's bandwidth will be affected by the transmission/receiving activities of its neighboring links. Second, unlike cellular networks where only one-hop wireless communication is involved, MANETs need to guarantee QoS on a multi-hop wireless path. Further, mobile hosts may join, leave, and rejoin at any time and at any location; existing links may disappear and new links may be formed on-the-fly. All these raise challenges to QoS routing in MANETs.

Routing in general consists of two entities, namely the routing protocol and the routing algorithm. The routing protocol has the task of capturing the state of the network and its available network resources and disseminating this information throughout the network. The routing algorithm uses this information to compute shortest paths.

Dynamic Network Topology: A key challenge in studying protocol behavior lies in how to represent the underlying topology and traffic patterns. The constantly changing and decentralized nature of current networks results in a poor understanding of these characteristics and makes it difficult to define a "typical" configuration. For example, random graphs can result in unrealistically long paths between certain pairs of nodes, "well-known" topologies may show effects that are unique to particular configurations, and regular graphs may hide important effects of heterogeneity and non-uniformity. The performance of QoS routing depends heavily on the underlying network topology. The dynamic nature of MANETs may make the flow stop receiving QoS provisions due to path disconnections. And also new paths must be established because of the disconnections and hence will be causing data loss and delays.

Imprecise state information: In the link-state routing algorithms, the source router selects a path based on the connection traffic parameters and the available resources in the network. The routing protocol distributes topology and load information throughout the network, and a signaling protocol for processing and forwarding connection establishment requests from the source. In MANETs, the Link state changes continuously, hence the QoS routing protocols can impose a significant bandwidth and processing load on the network, since each router must maintain its own view of the available link resources, distribute link-state information to other routers, and compute and establish routes for new connections.

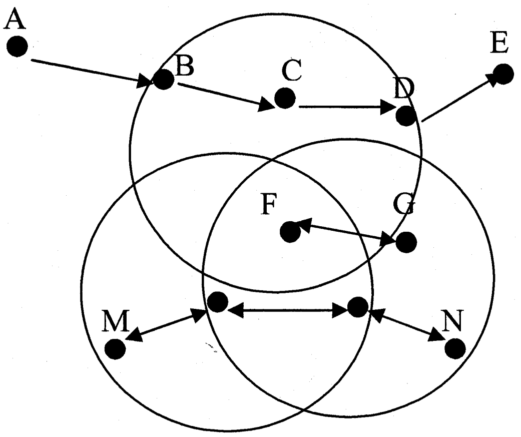


Figure 7: An example of hidden route problem.

Hidden route problem: The hidden route problem arises at the time as the route discovery procedure of a QoS routing protocol is executed. It is because the admission decision in a route discovery procedure considers only the local information, e.g., local capacity of the radio coverage of the node. Considering the example in Figure 7, a route *(A, E)* is currently processing route discovery and there are two routes *(F, G)* and *(M, N)* that have been discovered earlier. For simplicity and convenience, assuming that the capacity is constant, said 11 units, and the bandwidth requirements of routes *(A, E)*, *(F, G)*, and *(M, N)* are 4, 2, and 6 units, respectively. When the route discovery progresses in node *C*, it should consider the capacity of its radio coverage to determine if $C \rightarrow D$ could be established or not. Within the radio coverage of node *C*, node *F* has a flow with bandwidth requirement 2 to node *G*. Hence the available capacity in the radio coverage of node *C* is $11 - 2 = 9$. Since the bandwidth requirement of *(A, E)* is 4, $C \rightarrow D$ can be established on route *(A, E)*. However, the establishment of $C \rightarrow D$ will cause the bandwidth violation for route *(F, G)*. It is because that there are three flows in the radio coverage of node F, the bandwidth for route *(F, G)* remains 11−4−6−2=−1, which is not sufficient apparently.

Error-prone shared medium: Loss in wired networks is typically caused by excessive congestion that causes packets to be dropped at routers in the network. A wireless link, however, typically suffers much more loss due to Error-prone shared medium. One cause of loss in wireless transmission is fading, in which multiple versions of the same signal are received at the destination. If these signals are out-of-phase with each other or Doppler-shifted, they can interfere with each other. Other types of interference may also cause problems in wireless transmissions including electrical noise, or possibly even intentional communication jamming. Propagation delay can also be a tremendous burden to all communication, especially to communication that requires a guarantee on total delay.

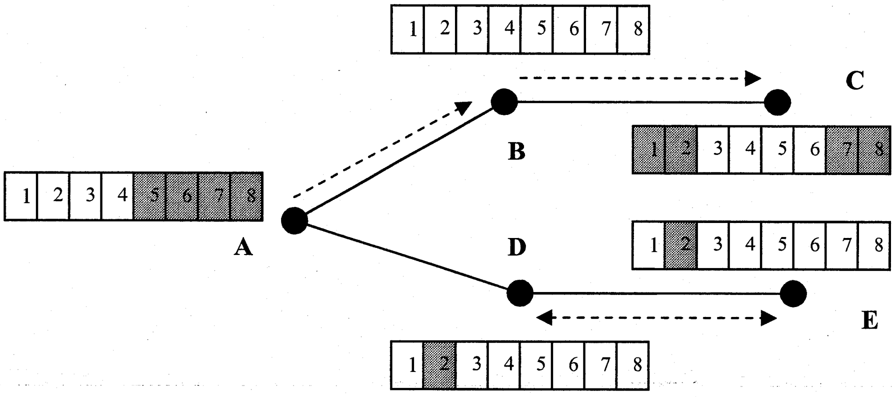


Figure 8: Hidden and exposed terminal problem in band-width calculation.

Hidden and exposed terminal problem: Consider the scenario in Figure 8, where there are four common free time slots between *A* and *B* (1, 2, 3, 4) and four free time slots between *B* and *C* (3, 4, 5, 6), if we reserve slots (1, 2, 3) for *A* to transmit and slots (4, 5, 6) for *B* to transmit, the path bandwidth is only three which is the maximum number. Suppose there is another pair, *D* and *E*, which are currently using slot 2 to communicate. Then two cases will occur. If *D* is a receiver on slot 2, *A* will not be allowed to send on slot 2 because otherwise collision will occur at *D*. This is the hidden-terminal problem. So in the example of Figure 8, the common free time slots between *A* and *B* should be reduced to (1, 3, 4) and the path bandwidth from *A* to *B* has to be downgraded to 2 slots. On the contrary, if *D* is a sender on slot 2, *A* will still be allowed to send on slot 2, because this is an exposed-terminal problem. Then the common free time slots between *A* and *B* (and thus the path bandwidth) remain the same. This simple example shows the complication of the bandwidth reservation problem in QoS routing in MANETs.

Lack of central control: Because of the lack of central controller which can account for and control MANETs' limited resources, nodes must negotiate with each other to manage the resources required for QoS routes. This is further complicated by frequent topology changes. Due to these constraints, QoS routing is more demanding than best-effort routing.

Limited resources availability: In wireless networks, there are additional considerations to be taken into account. The difficulty of satisfying the QoS requirement is aggravated by further constraints on energy reserves and available bandwidth, and signal degradation by noise and limited transceiver resources. Therefore, instead of a traditional layered network control approach, a joint optimization scheme affecting both the link and the routing layer may be necessary.

Insecure medium: Security is a critical issue of ad hoc networks that is still a largely unexplored area [14]. Since nodes use the open, shared radio medium in a potentially insecure environment, they are particularly prone to malicious attacks, such as denial of service (DoS). This characteristic of an ad-hoc network demands a new metrics for routing. Among this is the security-aware routing in which different security attributes is treated as a QoS parameter. Security is often considered to be the major "roadblock" in commercial application of ad hoc network technology.

3.2 QoS Routing Protocols Overview

The routing protocols for MANETs may be broadly classified as table driven protocols and on demand driven protocols. Table driven protocols need to maintain the global routing information about the network in every mobile node for all the possible source-destination connection and acquire to exchange routing information periodically. This kind of protocol has the property of lower latency and higher overhead. On-demand routing protocol creates routes only when the source nodes request. When a node requires a route to a destination, it initiates a route discovery process within the network. On-demand routing protocols are characterized as having higher latency and lower overhead. A majority of existing research about the QoS route in MANETs is based on the two kinds of route protocols. However, the table-driven QoS protocols request globe network state information which is not good for scalability and on-demand QoS protocols need initiates a route discovery based on flooding, which are not fit the dynamic and capability constrain in MANETs.

The QoS measure of a path can either be additive, multiplicative, or min/max. In the case of additive measures (e.g., delay, jitter), the path weight of that measure equals the sum of the QoS weights of the links defining the path. Multiplicative measures can be transformed into additive weights by using the logarithm. The path weight of min(max) QoS measures (e.g., available bandwidth) refers to the minimum(maximum) of the QoS weights along the path. The QoS constraints of an application are expressed in the m-dimensional vector. Constraints on min(max) QoS measures can easily be treated by omitting all links (and possibly disconnected nodes), which do not satisfy the requested QoS constraint. In contrast, constraints on additive QoS measures cause more difficulties.

Figure 9 depicts a general QoS routing model for MANETs. The point *A, B, C... H* represents the mobile nodes in MANETs. The weight of each edge is expressed with a two-tuples, which denote the available bandwidths (Mbps) and the delay (ms) of the relevant link. The pure routing algorithm such as the shortest may not be adequate for satisfying the requirement of QoS routing. For example in figure 9, when node *A* needs to communicate with node *D*, it may choose route *A->B->C->D* or route *A->F->E->D* as its path according to the shortest hop counts rule. But if the QoS requirement for the path's bandwidth is minimum 3 Mb/s, neither of these two routes can satisfy the requirement. Hence a new QoS route has to be chosen to route the packets. An alternative QoS path candidate may be *A->F->E->C->D.* Although it has 4 hop counts, its minimum bandwidth for this path is 3 Mbps which can satisfy the QoS requirement. The delay of the route, however, is 2+6+2+4=14. If the delay requirement for the transmission is less than 10 ms, this path can not be a QoS path too.

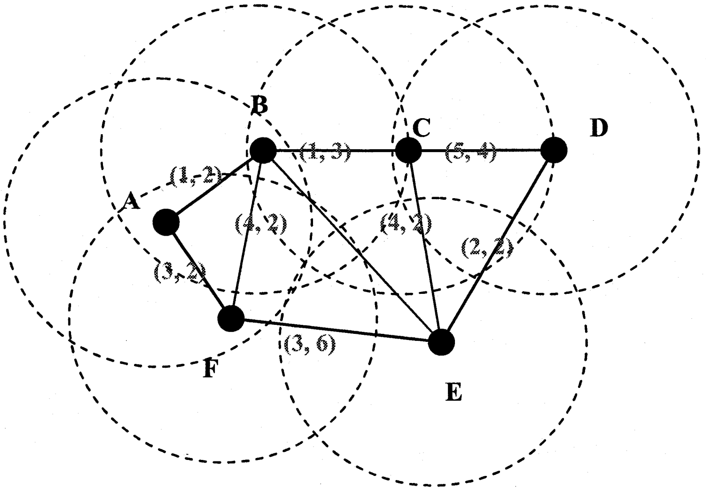


Figure 9: General QoS Model in MANETs

3.3 Related Work

QoS routing has received much attention recently for providing QoS in wireless ad hoc networks and some work has been carried out to address this critical issue. Here, we provide a brief review of existing work addressing the QoS routing issues in wireless ad hoc networks. There have already been several surveys and overviews regarding the QoS routing issues and solutions. Chakrabarti and Mishra [2] summarized the important QoS-related issues in MANETs that were in focus around 2001, and the issues that required further attention. They updated and expanded the article in 2004 [16]. A fairly comprehensive overview of the QoS in networking could be found in [17] [18] [19]. Their conclusions highlighted several significant points:

- Many of the underlying algorithmic problems, such as multi-constraint routing, have been shown to be NP-complete [16].
- QoS and best-effort, routing can only be successfully achieved if the network is combinatorially stable. The dynamic topology, the error-prone channel, the lack of central control and the insecure medium have always been roadblocks for the development of QoS routings [18].
- Different techniques are required for QoS provisioning when the network size becomes very large, since QoS state updates would take a relatively long time to propagate to distant nodes [19].
- The amount of state propagation and topology update information must be kept to a minimum. In particular, every change in available bandwidth should not result in updated state propagation [16].
- QoS routing protocol is designed without considering the situation when multiple QoS routes are being setup simultaneously. If two QoS routes cannot be fully established because they are blocking each other, both will be deleted. Hence how to setup QoS routes when there are multiple competing requests needs further study [15].
- The protocols should be designed to accommodate multiple classes of traffic, in particular, to ensure that lower-class traffic is not starved of network resources in the presence of real-time traffic [19].

Many other metrics, such as security and multicast routing are also taken into consideration. QoS routing solutions existing in early 2004 can be categorized into the following types of approaches: flat (all nodes play an equal role), hierarchical (some nodes are local cluster heads for example), position-based (utilize location information), and power-aware (take battery usage and residual charge into consideration) QoS routing. Other classifications include: QoS provisioning mechanism, interaction between network and MAC layer, routing information update mechanism, etc.

Some cross-layer QoS frameworks have also been proposed which separate QoS metrics at the different layers and the protocol layers interact with other layers through sharing of network status information (NSI) collected at different layers. A general framework for cross-layer solution is shown in Figure10. This framework contains two extra components besides the layered framework of an ad hoc network, QoS requirements component and network status information component. The QoS Requirements component represents the QoS requirement metrics at different layers, which are obtained by mapping the QoS requirements to the corresponding metrics at different layers. This is because the QoS that an application requires depends strictly on the quality of the network at different layers. The NSI component functions as a repository for information that network protocols throughout the protocol stack collect. Each protocol can access the NSI to share its data with other protocols. This avoids duplicating efforts to collect internal state information and helps in using proper network information for adaptation of network protocol functioning.

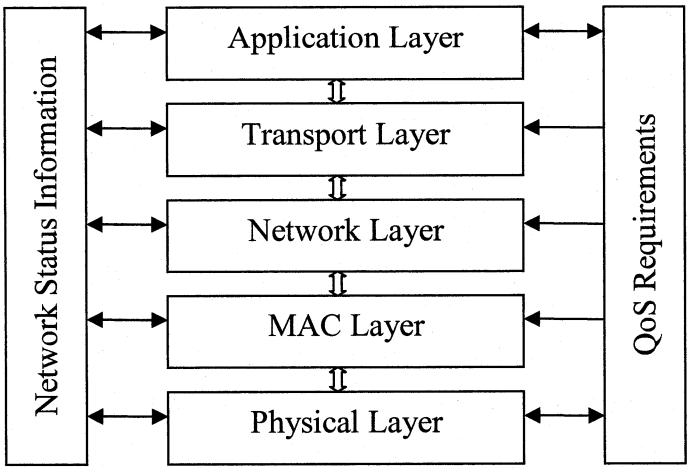


Figure 10: Cross layer framework.

Different layers have different roles in providing QoS:

- The application layer can do the data compression depending on the quality of the underlying networks and the QoS requirements of the applications.
- The transport layer can detect and differentiate the cause of packet losses.
- The Network layer can provide QoS support through the QoS-aware routing.
- The MAC layer can support QoS by enhancing back-off procedure of MAC protocol.

- The Physical layer can perform adaptation of rate, power and coding to meet application's QoS requirements, taking into consideration the current network status.

3.4 Main QoS-Aware Routing Protocols

The problem that concerned the QoS routing protocol designers was that of discovering the paths that satisfy the different QoS requirements such as throughput, delay and jitter in the networks. To find a QoS routing in a MANET is to establish a path that satisfies the QoS requirement given the knowledge of the available channel information at each forwarding node. In this section, some of the main QoS-aware routing protocols in MANETs are presented and the merits and deficiencies of each protocol will be discussed.

3.4.1 Ticket-based QoS Routing

Chen and Nahrstedt [20] proposed a multi-path distributed routing scheme, called ticket-based probing, a QoS routing protocol aimed at reducing the QoS route discovery overhead while providing throughput and delay guarantees. The idea is based on two observations: 1) the QoS routing is done on a per-connection basis to reduce the routing overhead and avoid an exhaustive search; 2) an intelligent routing selection is chosen to guide the search along the best candidate paths. The basic idea of ticket-based probing is that a ticket is the permission to search one path. The source node issues a number of tickets based on the available state information. Probes (routing messages) are sent from the source toward the destination to search for a low-cost path that satisfies the QoS requirement such as delay and bandwidth. At an intermediate node, a probe with more than one ticket is allowed to be split into multiple ones, each searching a different downstream sub-path. The maximum number of probes at any time is bounded by the total number of tickets. See Figure 11 for an example, two probes, *p1* and *p2*, are sent from s. The number in the parentheses following a probe is the number of tickets carried in the probe. At node *j*, *p2* is split into *p3* and *p4*, each of which has one ticket. There are at most three probes at any time. Three paths are searched, and they are *s -> i -> t*, *s -> j -> t*, and *s -> j -> k -> t*.

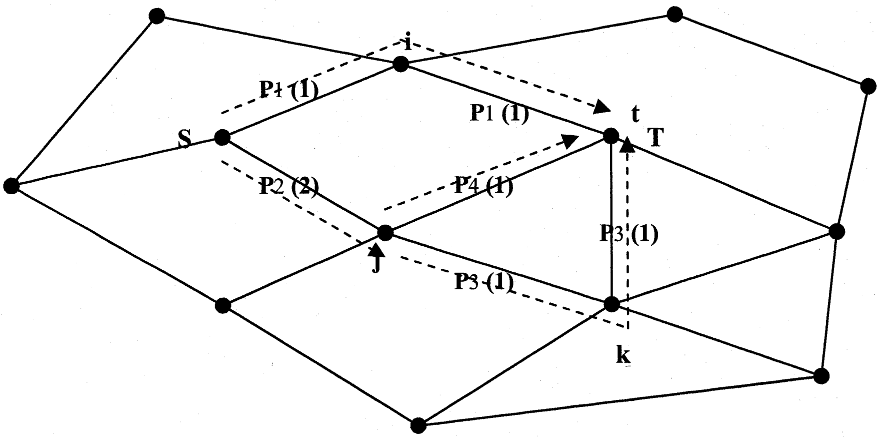


Figure 11: Illustration of ticket-based QoS routing adapted from [20].

Based on the idea of ticket-based probing, two heuristic algorithms are proposed, one for delay constrained QoS routing, and the other for bandwidth constrained QoS routing. In delay-constrained QoS routing, each probe accumulates the delay of the path it has traversed so far. It selects the path with least cost as the primary path and the other paths as the backup paths, which will be used when the primary path is broken due to the mobility of intermediate nodes. The source node issues two types of tickets, yellow tickets and green tickets, and sends them along with probe packets. Yellow tickets prefer paths that satisfy the requirement of a probe in terms of QoS metrics and are used to search for paths that have least delay. Green tickets are to maximize the probability of finding a low cost path which may have larger delays. The number of tickets for both is based on the delay requirements. The maximum number of probes at any time is bounded by the number of tickets. When the delay requirement is smaller, more tickets are issued to increase the chance of finding a feasible path. Bandwidth QoS routing shares the same computational structure as delay constraint routing, but the number of tickets issued is based on the bandwidth requirement.

A number of advantageous properties of the ticket based probing are:

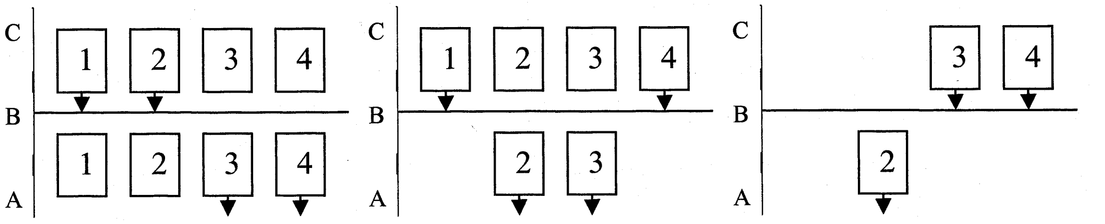
- The routing overhead is controlled by the number of tickets, which allows the dynamic tradeoff between the overhead and the routing performance.
- The scheme is designed to work with imprecise state information. Its path broken detection, rerouting, path redundancy and path repairing mechanism can respond fast to the network dynamics.
- A distributed routing process is used to avoid any centralized path computation that could be very expensive for QoS routing in large networks. The hop-by-hop path selection process is adopted.
- The information at the intermediate nodes, both local and end-to-end states, is collectively used to direct the probes along the low-cost feasible paths toward the destination. This approach not only increases the chance of success but also improves the ability to tolerate the information imprecision because the intermediate nodes may gradually correct a wrong decision made by the source.

The disadvantages of ticket based probing are:

- The proposed heuristic algorithms, which are based on an imprecise state information model, may fail in finding a feasible path in the extreme cases where the topology changes very rapidly.
- In delay-constrained QoS routing, the queuing delay and the processing delay at the intermediate nodes are not taken into consideration while measuring the delay experienced so far by the probe packet. This may cause some data packets to miss their deadlines.
- The routing algorithm records the path in the probe itself which may consume more communication bandwidth and memory space to store the probes when they are waiting in the queue.

3.4.2 Bandwidth Constraint QoS Routing

The link bandwidth is defined here as the common free slots between two adjacent nodes. And the path bandwidth which is also called end-to-end bandwidth is defined as a set of available slots between two nodes. We can use link bandwidth to calculate the path bandwidth. A bandwidth routing protocol usually consists of three components: an end-to-end path bandwidth calculation algorithm to inform the source node of the available bandwidth to any destination; a bandwidth reservation algorithm to reserve sufficient number of free slots for the QoS flow; and a standby routing algorithm to re-establish the QoS flow in case of path breaks.



Equal Case Containing Case Exclusive Case

Figure 12: Three cases in bandwidth calculation adapted from [21].

Figure 12 illustrates the three cases in bandwidth reservation algorithm. We can observe that link *BW(A,B)* = *free_slot(A)* I *free_slot(B)* where *BW(A,B)* stands for the available path bandwidth between *A* and *B*. The definition of *free_slot(X)* is the slots which are not used by any adjacent host of to receive or to send packets. In equal case, the four free slots can only contribute maximum two slots for path bandwidth. Namely, 4/2 =2, otherwise, if one link takes up 3 slots, then there is only 1 slot left for the other link which makes the path bandwidth between them reduced to 1 slot. In containing case, assume *link_BW(A,B)* $\subset$ *link_BW(B,C)*. In this case, *C* should first use slots in *link_BW(B,C)- link_BW(A, B)* (the slots used on link *BC* but not used on link *AB*) to maximize system utilization. Therefore, if *C* uses slots 1, 4, then *B* can use slots 2, 3, so *path_BW(C, A)* = *2*. In exclusive case, If *link_BW(A, B)* I *link_BW(B, C)=0* , no conflict will occur. *C* can choose either 3 or 4 and *B* can choose 2 and the path bandwidth=1. We will find any general case can be regarded as a combination of the previous three cases.

The bandwidth constraint QoS routing protocols [21] [22] [23] mainly contain bandwidth calculation algorithm and slots reservation scheme. The routing can be discovered on-demand so that it neither maintains any routing table nor exchange routing information periodically. When a source node wants to communicate with another node for which it has no routing information, it floods a route request (RREQ) packet to its neighbors. In these protocols, all packets contain following uniform fields: <*packet_type, source_addr, dest_addr, sequence#, route_list, slot_array_list, data, TTL*>. For a source node, in order to send a stream of packets to a destination node, a Virtual Connection (VC), to that node has to be established. The VC establishment process includes route discovery, path bandwidth calculation and bandwidth reservation components. When a node receives a RREQ packet in the route discovery process, it records the status of available slots in the *slot_arrary_list*. When the destination node receives one RREQ packet, it returns a RREP packet by unicasting back to the source following the route recorded in the *route_list*. The destination node selects the path with least cost among them and copies the fields *{route list, slot array list}* from the corresponding RREQ packet to the QoS route reply (RREP) packet and sends the RREP packet to the source along the path recorded in route list. As the RREP traverses back to the source, each node recorded in route list reserves the free slots that have been recorded in the slot array list field. Finally, when the source receives the RREP, the end-to-end bandwidth reservation process gets completed successfully and starts sending data packets in the data phase. The reservations made are soft state in nature in order to avoid resource lock-up.

The disadvantages of these protocols are: 1) when the RREP travels back to the source, the reservation operation may not be successful. This may result from the fact that the slots which we want to reserve are occupied a little earlier by another VC or the route breaks. If this is the case, the route has to be given up and the destination re-starts the reservation process again along the next feasible route which incurs longer delay; 2) once a VC is established, the source can begin sending datagrams in the data phase. At the end of the session, all reserved slots must be released. These free slots will be contended by all new connections. However, if the last packet is lost, we will not know when the reserved slots should be released; 3) the QoS path discovered in this process may satisfy the QoS provisions but not necessarily the shortest path.

3.4.3 Delay Constraint QoS Routing

The On-Demand Delay-Constrained Unicast Routing Protocol (ODRP) is proposed in [24]. The design of ODRP focuses on the operations at the network layer and assumes the capabilities of determining resource availability on neighboring links and the availability of resource reservation functions at nodes. For ODRP to work correctly, each node is required to maintain a distance vector consisting of $|V|-1$ entries where $|V|$ is the number of nodes in the network. The entry for node *v* at node *u* *(u!=v)* contains the following information: the identifier of node *v*, the shortest distance from *u* to *v* (in hop count), and the next hop of *u* along this path to *v*. ODRP utilizes the vectors stored at different nodes to guide route-searching packets to propagate in the promising direction and avoid pure flooding. This vector can be provided by running a proactive wide-area (best effort) distance vector routing protocol in the network.

ODRP employs the following strategies in its route-searching operations: hybrid routing, directional search and link-delay-based scheduling of (control) packet forwarding. The process of discovering a QoS routing includes two phases: 1) probing the feasibility of min-hop routing. The source sends a packet along the min-hop routing to the destination and starts a timer. If the min-hop routing satisfied the delay requirement, this delay constraint routing has been identified; 2) Destination initiated route discovery for delay-constraint path. If the minimum hop path does not satisfy the delay constraint, the destination initiates a directed and limited flood search by broadcasting a RREQ packet. Intermediate nodes only forward the RREQ with the least delay value and ignore any further RREQs. When a copy of the RREQ reaches the source with a path that meets the delay constraint, the route discovery process is complete.

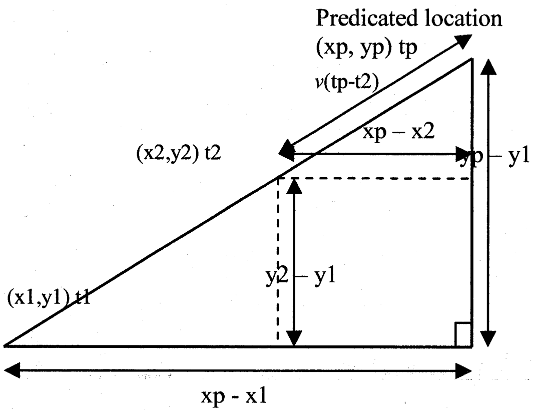
The Advantages of this routing protocol are: 1) the path discovery restricted flooding only when the min-hop routing doesn't satisfy the QoS requirements, which helps to reduce the communication overhead; 2) the route searching process is restricted and limited in a predetermined searching range and each node only forwards RREQ packet once which further limits the communication overhead.

The Disadvantages are: 1) the restricted searching process may lower the probability of finding a feasible path; 2) the on-demand nature of route discovery process leads to higher connection setup time; 3) while the aim of the directed flooding is to avoid global flooding, thereby reducing overhead compared to protocols that are based on that, extra overhead is incurred by the proactive distance-vector protocol which maintains the routing tables.

3.4.4 Location Based QoS Routing

In [25], a predictive location-based QoS routing protocol is proposed. This protocol includes three components: update protocol, predictions (location prediction and delay prediction) and QoS routing.

The update protocol includes two types of updates. Type 1 update is generated periodically at a constant frequency or can vary linearly between a maximum *f(max)* and minimum *f(min)* threshold with the velocity of the node. Consequently, the distance traveled between successive type 1 updates remains constant. Type 2 update is generated when there is a considerable change in the node's velocity or direction of motion. In establishing a connection to a particular destination *B*, source *A* has to first predict the geographic location of the destination *B* as well as the intermediate hops, at the instant when the first packet will reach the respective nodes. Hence, this step involves a location as well as propagation delay prediction. The location prediction is used to determine the geographical location of some node (either an intermediate node or the destination *B*) at a particular instant of time *t* in the future when the packet reaches it. Figure 13 depicts the location prediction algorithm in this protocol using the theory of similarity of triangles. (x_p, y_p) stands for the location of *B* predicted by node *A* at a certain time t_p. The delay from *A* to *B* is predicted as the same delay experienced by a data packet in the lasted update between *A* and *B*.



$$x_p = x_2 + \frac{v(t_p - t_2)(x_2 - x_1)}{[(x_2 - x_1)^2 + (y_2 - y_1)^2]^{1/2}}$$

$$y_p = y_2 + \frac{(y_2 - y_1)(x_p - x_2)}{x_2 - x_1}$$

Figure 13: Location prediction at a future time using last 2 updates adapted from [25].

As a result of the location-resource updates, each node has information about the whole topology of the network. It can thus compute a source route from itself to any other node, using the information it has, and can include this source route in the packet to be routed. The QoS requirements are in the form of a tuple <estimated duration of connection, maximum delay, maximum delay jitter>. The maximum delay QoS requirement can be mapped onto the end-to-end delays observed for the updates from *B* to *A*. Thus, given the resource availability at the nodes and the QoS requirements of the connection, admission control can be performed. To search for a QoS path from *A* to *B*, *A* first runs a location-delay prediction on each node in its proximity list and obtains a list of its neighbors at the current time. It determines which of these neighbors have the resources to satisfy the QoS requirements of the connection. The next step at *A*'s network level is to perform a depth-first search for the destination starting at each of these candidate neighbors to find all candidate routes. From the resulting candidate routes, the geographically shortest one is chosen and the connection is established.

Some of the disadvantages of this protocol include: 1) it relies on accurate location awareness, which limits its usefulness to devices that are capable of being equipped with GPS receivers or such; 2) the update protocol in this paper involves flooding of location and resource information pertaining to a node to all the other nodes in the network. Ordinarily, such a full flooding of the network involves a very large overhead. However, with schemes such as the multipoint relay (MPR) scheme, the overhead associated with flooding can be considerably reduced.

3.4.5 Hierarchical Routing: CEDAR

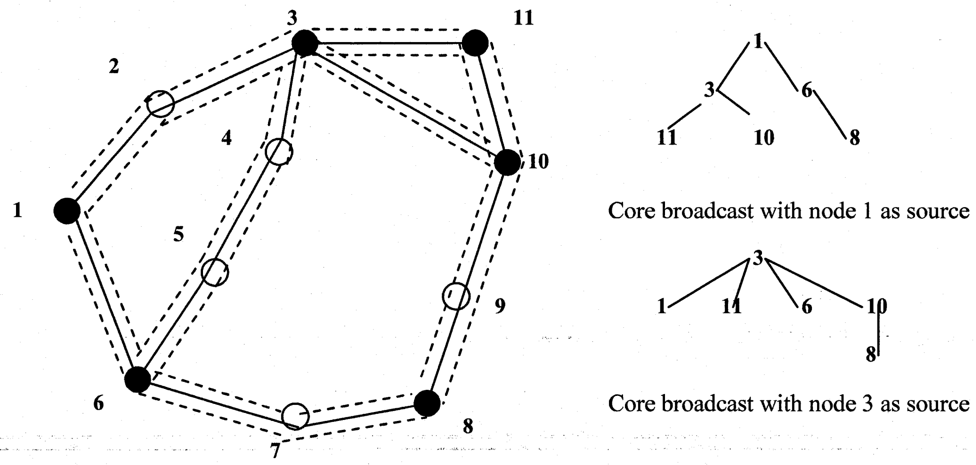
CEDAR [26], a core-extraction distributed ad hoc routing algorithm for quality-of-service (QoS) routing in ad hoc network environments, has three key components: a) the establishment and maintenance of a self organizing routing infrastructure called the core for performing route computations; b) the propagation of the link state of high bandwidth and stable links in the core through increase/decrease waves; and c) a QoS-route computation algorithm that is executed at the core nodes using only locally available state.

Core Extraction

The core structure is used to limit the number of nodes that must participate in the exchange of topology and available bandwidth information. The goal of setting up the core is to proactively create a core set such that every node is either a core node or a neighbor of a core node. As the route computation is done by the core nodes, minimizing the number of core nodes is desirable. Since core computation is local, it makes core computation in CEDAR scalable as the core can be computed in a constant amount of time. When a node is electing a dominator, it gives preference to core nodes already present in its neighborhood (including itself). This provides stability to the core computation algorithm, though it might have implications on the optimality of the number of core nodes. Each core node maintains local topology information and performs route discovery, route maintenance and call admission on behalf of these nodes.

Core broadcast

In order to achieve efficient core broadcast, each node temporarily caches every RTS and CTS packet that it hears on the channel for core broadcast packets only. The purpose of caching RTS/CTS is to use them for the elimination of duplicate packet reception for broadcasts. In the ad hoc network shown in Figure 14, when node 1 is the source of the core broadcast, 10 would not be sending a message to 11, as it would have heard a CTS from 11 when 11 was receiving the message from 3. Similarly, 8 would not be sending on the tunnel to 10, as 9 would have heard the CTS from 10, and hence would send a NACK when 8 sends an RTS to 9.



Core broadcast with node 1 as source

Core broadcast with node 3 as source

Figure 14: Examples of core broadcast adapted from [26].

QoS State Propagation

To propagate state information (available bandwidth) among the core nodes, increase waves and decrease waves are used. These waves are generated when a core node's available bandwidth has changed by a threshold value. A slow-moving increase wave denotes an increase of bandwidth on a link, and a fast-moving decrease wave denotes a decrease of bandwidth on a link. For low-bandwidth links, it makes sense to have as few nodes as possible contending for the link, while for stable high bandwidth links, it makes sense to have as many core nodes as possible know about the link in order to compute good routes. In other words, the maximum distance that the link state can travel (i.e. the time-to-live field) is an increasing function of the available bandwidth of the link. And because every core node that caches information corresponding to a link can potentially use the bandwidth of the link, the number of core nodes that cache the state of a low bandwidth link should be less compared to a stable high bandwidth link to reduce the contention for a low bandwidth link.

QoS routing setup

Briefly, QoS route computation in CEDAR is an on-demand routing algorithm which proceeds as follows: when a source node *s* seeks to establish a connection to a destination node *d*, provides its dominator node *dom(s)* with a *(s, d, b)* tuple, where *b* is the required bandwidth for the connection. If *dom(s)* can compute an admissible available route to using its local state, it responds to immediately. Otherwise, if *dom(s)* already has the dominator of *d* cached and has a core path established to *dom(d)*, it proceeds with the QoS route establishment phase. If *dom(s)* does not know the location of *d*, it first discovers *dom(d)*, simultaneously establishes a core path to *d* , and then initiates the route computation phase. A core path from *s* to *d* results in a path in the core graph from *dom(s)* to *dom(d)*; *dom(s)* then tries to find the shortest-widest-furthest admissible path along the core path. Based on its local information, *dom(s)* picks up the farthest reachable domain until that which it knows is an admissible path. It eventually establishes an admissible route to *d* or the algorithm reports a failure to find an admissible route.

The advantages of this routing protocol includes: route computation does not involve the maintenance of global state and only a few nodes are involved in state propagation and route computation. If the topology stabilizes, then routes will converge to the optimal routes. Disadvantages include: As far as the nature of state maintained at each core node is concerned, at one extreme is the minimalist approach of only storing local topology information at each core node. This approach may result in a poor routing algorithm (i.e., the routing algorithm may fail to compute an admissible route even if such routes exist in the ad hoc network). At the other extreme is the maxima list approach of storing the entire link state of the ad hoc network at each core node. This approach computes optimal routes for stable networks, but incurs a high state management overhead for dynamic networks and potentially computes stale routes based on an out-of-date cached state when the network dynamics are high.

3.4.6 Application-aware QoS Routing

A unique approach to QoS routing is presented in [27]. Instead of using lower layer information, the protocol is based on the aid of the transport layer. It assumes the use of real-time transport protocol (RTP)/ RTP Control Protocol (RTCP) and the real-time streams are delivered in the RTP packets. The delay between two nodes is estimated statistically by examining the difference between timestamps on transmission and receipt of RTP packets between those two nodes. The delay variance is also calculated. Furthermore, each node records the throughput requirement of RTP sessions which are flowing through it. Subtracting the total of these throughput values from the raw channel capacity gives an estimate for the total remaining capacity at that node.

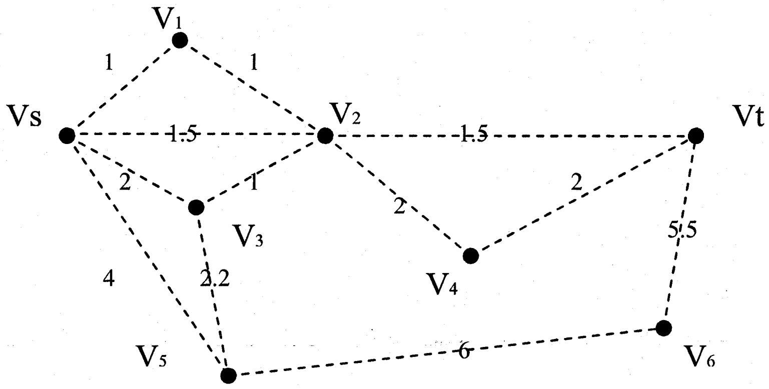


Figure 15: Network topology of application-aware routing adapted from [25].

The route discovery is performed in the following steps:

Step 1: Define a set I= $\{R_1, R_2, ...R_k\}$ including all the routes with the shortest delays satisfying the delay requirements.

Step 2: Select the subset $A \subset I$ whose elements satisfy the bandwidth requirement. If set A is null, then go to Step 4.

Step 3: From set *A*, select the route *R* with the minimum variance of the transmission delays during a predefined period.

Step 4: Select the route *R* with the maximum allocated available bandwidth. If there is sufficient available bandwidth for a multimedia application, the most robust QoS route is selected using this scheme. If there are no routes that meet the bandwidth requirement, the route with the highest available channel capacity, which satisfies the delay constraint, is selected.

Figure 15 shows a MANET including eight mobile nodes. The dashed line between two nodes represents the wireless connection. The number tag of each dashed line denotes the estimated transmission delay in the unit of 10 ms. We could compute four routes that satisfy the delay requirement from *Vs* to *Vt*: 1, *Vs->V2->Vt*, 2, *Vs->V1->V2->Vt*, 3, *Vs->V2->V4->Vt* and 4, *Vs->V3->V2->Vt*. From step 2, we could eliminate those routes that don't satisfy the bandwidth requirement. We assume that route 1 and 2 can satisfy the bandwidth requirement. Then from step 3, we could choose the route that has the minimum delay variance as the QoS route. If none of the routes satisfy the bandwidth requirement, the route with the maximum available bandwidth will be selected.

A major advantage of this routing protocol is that no extra overhead is incurred for QoS routing, since the existing transport layer packets are used for QoS metric estimation. Additionally, both delay and throughput constraints may be considered. However, the use of RTP is assumed, and therefore the range of application scenarios for this protocol is obviously limited.

3.5 Comparison of QoS Routing Protocols

There are different ways to classify the QoS routing protocols in MANETs. Some classify the protocols by the network topology (flat, hierarchical, hybrid). Some classify the protocols by different approaches to solve the QoS issues (ticket-based probing, predictive, more node state information). Some classify the protocols by route discovery approach (proactive, reactive, hybrid). Other typical classifications include by the interaction with MAC layer (independent or dependent), and also by the QoS requirements (delay, bandwidth, security, energy). In this report, the classification of QoS routing protocols is based on the approaches to QoS routing in MANETs. The following table lists the representative QoS routing mechanisms discussed in this report. It includes the QoS metrics, the node information, the requirement from MAC layer and other assumptions to make the protocols feasible.

Routing Algorithms	QoS Metrics	Architecture & Reactive	Network/Node Information	MAC Layer	Other Assumptions
Ticket based QoS Routing (TBR) [18]	Bandwidth or delay	Flat / Proactive	Available channel capacity; delay estimates; global state information	Resource reservation	DSDV routing
Bandwidth guaranteed Routing (CCBR) [19]	Bandwidth	Flat / Proactive	Time slot schedule Neighbor nodes status	CDMA over TDMA; resource reservation	DSDV routing, call admission control
On demand QoS routing [20]	Bandwidth	Flat / Reactive	Node states Neighbor nodes status	CDMA over TDMA; resource reservation	AODV routing
On-demand Delay-constrained Routing Protocol (ODRP) [21]	Bounded delay	Flat / Reactive	Distance vector consisting of |V|-1 entries (identifier of V, shortest path, next hop)	Resource reservation	AODV routing but proactive state dissemination
Predictive Location-based QoS Routing [24]	Bounded delay	Flat/ Reactive	Node relative positions and velocities	None	Relative location awareness; relative speed awareness; source-routing
CEDAR (Core Extraction Distributed Ad-hoc Routing) [25]	Bandwidth	Hierarchical/ Partially	Link residual capacity	Link residual capacity estimation	RTS/CTS is cached for the purpose of core broadcasting
Application-aware QoS Routing [26]	Bounded delay and bandwidth	Flat/ Reactive	RTCP information	None	RTP is needed

Table 6: Comparison of QoS routing protocols.

4 A Simulation Study on the Performance of SWAN Model

SWAN is a stateless network model which uses distributed control algorithms Additive Increase Multiplicative Decrease (AIMD) to deliver service differentiation in mobile wireless ad hoc networks in a simple, scalable and robust manner. The SWAN model includes a number of mechanisms used to support rate regulation of best effort traffic. A classifier and a shaper operate between the IP and MAC layers. The classifier is capable of differentiating real-time (RT) and best effort (BE) packets, forcing the shaper to process BE packets but not RT packets. The shaper represents a simple leaky bucket traffic shaper. The goal of the shaper is to delay BE packets in conformance with the rate calculated by the rate controller. Each source node uses admission control for RT sessions. When a new RT session is admitted, the packets associated with the admitted flow are marked as RT traffic. The classifier looks at the marking and, if the packet is marked as RT, the packet will bypass the shaper mechanism, remaining unregulated.

4.1 Introduction

In this report, the study will be based on the simulations completed with NS-2 [28] which requires NS-2 version 2.26. The simulation environment and tools (cygwin, setdest, cbrgen, tcl, GAWK, GNUPLOT) will be introduced first. Then QoS metrics which will be used to measure the performance of the SWAN model in different scenarios will be presented. Next, an extensive simulation will be executed using different parameters of the network, the model itself and different traffic patterns (TCP, UDP, different number of connections/streams). In this simulation, CBR traffic will be modeled with different packet length and interval to simulate the voice and video flows and TCP traffic will be modeled as best effort traffic. First, we will vary the different number of traffic flows and connections in the network to measure the impact of the traffic pattern on QoS performance of the model. Then the parameters of SWAN model itself (increment rate and decrement rate) will be varied to expose the characteristics of the model. At last, the QoS metrics will be compared with different node mobility (speed and pause time). Results and analysis will be presented after each simulation.

4.2 Simulation Environment

4.2.1 Overview

The SWAN simulator environment requires the NS-2 simulator version 2.26. The NS-2 simulator runs under UNIX (e.g., Linux, FreeBSD, SunOS, Solaris) and Window environments. In this simulation, the NS-2 is installed in Windows environments using Cygwin. Cygwin provides a Linux-like environment under Windows.

The simulation consists of generating the following input files to NS-2: a scenario file that describes the movement pattern of the nodes and a communication file that describes the traffic pattern in the network. These files can be generated by the built-in tools in NS-2: cbrgen and setdest which is located at /ns-src/indep-utils/cmu-scen-gen directory. These files are then used for the simulation and as a result of it, a trace file is generated as output. Prior to the simulation, the parameters that are going to be traced during the simulation are selected. The trace file can then be scanned and analyzed for the various parameters that one wants to measure. This can be used as data for analysis with GAWK and plots with GNUPLOT. The trace file can also be used to visualize the simulation with NSNAM.

4.2.2 Tools

NS-2

NS-2 is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS-2 has gained popularity among participants of the research community, mainly because of its simplicity. It allows simulation scripts to be easily written in a script-like programming language, OTcl. More complex functionality relies on C++ code that either comes with NS-2 or is supplied by the user. This flexibility makes it easy to enhance the simulation environment as needed, although most common parts are already built-in, such as wired nodes, mobile nodes, links, queues, agents (protocols) and applications (i.e. ftp). Most network components can be configured in detail, and models for traffic pattern and errors can be applied to a simulation in order to increase its reality. Simulations in NS-2 can be logged to trace files, which include detailed information about received and transmitted packets and allow for post-run processing with some analysis tools. Figure 16 outlines some of the basic components of NS-2 - in particular those that are important for the implementation of the SWAN framework described later in this chapter. The NS-2 distribution version discussed within this report is ns-allinone-2.26.

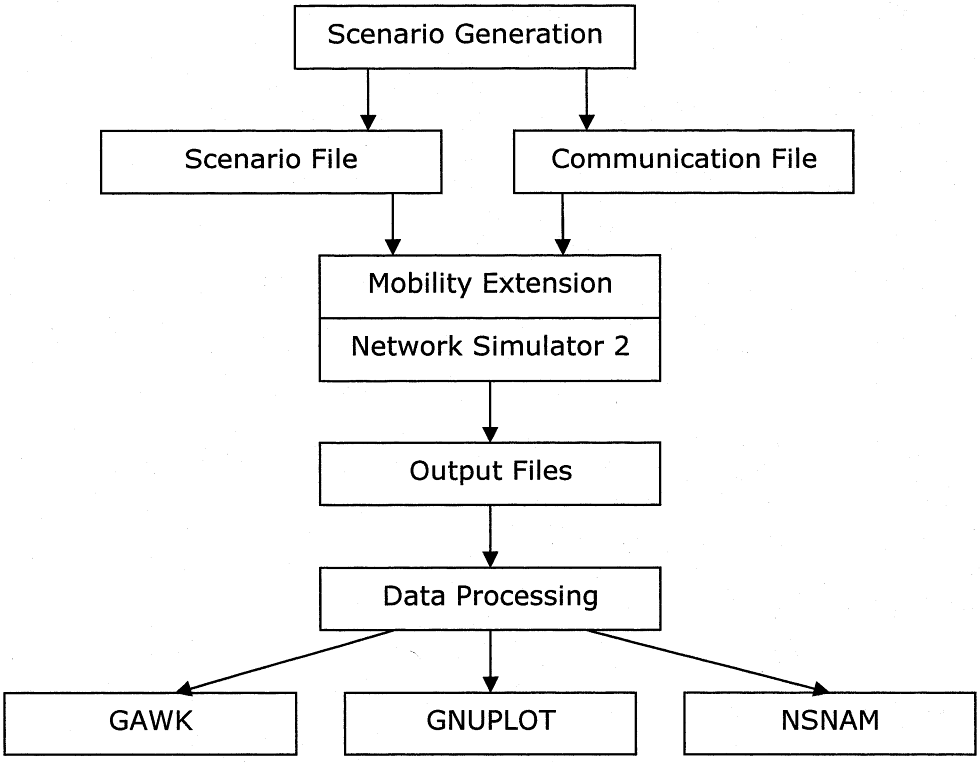


Figure 16: Simulation process.

GAWK

AWK is a computer program that is designed to process text-based data. GAWK is AWK developed by GNU (GNU is a recursive acronym for GNU's Not UNIX.). Using AWK, a command file and an input file should be given. A command file can be a file or a command line input. The command file would tell AWK how to deal with the input file.

It is composed of patterns and actions. For an input file, every line of the input file will be examined to judge whether this line matches the pattern. If this is the case, this line will be processed by the corresponding action. During the processing of the input file, GAWK will first separate the input file into pieces of records. The record separator is "\n" by default. That is the reason AWK normally parses the file line by line. Each record is composed of several fields; different fields are separated by white space by default. In the command file $1 represents the first field of a record. In actions, GAWK uses printf to print out the processing result. BEGIN and END are two special patterns in GAWK. Their corresponding actions are executed only at the beginning and the ending of the execution of a command file.

GNUPLOT

GNUPLOT is a portable command-line driven interactive data and function plotting utility for UNIX, IBM OS/2, MS Windows, DOS, Macintosh, VMS and many other platforms. It was originally intended as to allow scientists and students to visualize mathematical functions and data. GNUPLOT supports many types of plots in either 2D and 3D. It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text. It also supports various specialized plot types. GNUPLOT supports many different types of output: interactive screen terminals (with mouse and hotkey functionality), direct output to pen plotters or modern printers (including postscript and many color devices), and output to many types of file (eps, fig, jpeg, LaTeX, metafont, pbm, pdf, png, postscript, svg ...). GNUPLOT is easily extensible to include new devices. In this simulation, GNUPLOT will be used to plot the QoS performance statistics of different routing protocols with different parameters.

Mobile Node Movement: The generator for creating node movement files are found under ~ns/indep-utils/cmu-scen-gen/setdest/ directory. Setdest tool is used to generate the positions of nodes and their moving speed and moving directions. The tool use a random waypoint model. Run setdest with arguments in the following way:

```
./setdest -n <num_of_nodes> -p <pausetime> -s <maxspeed> -t <simtime>
-x <maxx> -y <maxy> > <outdir>/<scenario-file>
```

Generating Traffic Pattern Files: The traffic generator is located under ~ns/indep-utils/cmu-scen-gen/ and are called cbrgen.tcl and tcpgen.tcl. They may be used for generating constant bit rate (CBR) and TCP connections respectively.

To create CBR connecions, run: ./ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]

To create TCP connections, run ns tcpgen.tcl [-nn nodes] [-seed seed]

4.2.3 Measurements

A. End-to-end delay

Average End-to-end delay = total delay of received pkts/ total # of received pkts

The average end-to-end delay calculates the delay of the packet which is successfully transmitted from the source to the destination. This end-to-end delay includes all possible delays caused by buffering during route discovery latency, queuing in the interface queue, retransmission delays at the MAC, propagation and transfer times. It is the duration of the time a packet travels from the application layer of the source to the destination. End-to-end delay is one of the most important metrics in analyzing the performance in QoS aware routing protocols. The average end-to-end delay is averaged out of all the end to end delay of successfully transmitted packets.

In this simulation, the end-to-end delay will be calculated separately for video, voice and TCP traffic to evaluate the performance of SWAN model and analyze how it controls the best-effort traffic to give the real-time traffic high priority.

B. Network Throughput

Throughput = bits per second delivered to destination

It is defined as the total number of data delivered to destination divided by the simulation time. The throughput for both real-time and non-real-time traffic will be considered independently. In the trace file, the logic

```
if (($1 == "r") && ($7 == "cbr" ) && ( $4=="AGT" )
```

*throughput = throughput + packet_size ($6) *8/ measurement interval*

will be used to indicate whether a CBR packet has been delivered to the destinaction and if yes, the packet will be counted as throughput. $7 is the different traffic type in which

CBR stands for real-time traffic and TCP stands for best-effort traffic. In this way, we can calculate the real-time traffic and non-real-time traffic separately.

C. Goodput

goodput = (# of pkts received by the receiver in sequence) × *packet_size/ measurement interval*

In this simulation, the packet size is fixed, for voice traffic, the size is 80 bytes and for video traffic, the size is 521 bytes. The term "goodput" narrows the definition of throughput to be the empirical amount of data actually usable by applications over time. Goodput is typically some fraction of the total throughput. Dropped frames, retransmission of data, and the addition of protocol headers in the IP stack are all elements that reduce the usable throughput over a network link. Bit error is especially common over wireless links and typically results in a loss of TCP throughput. In this simulation, we are interested in both throughput and goodput.

D. Control packet overhead ratio

Control packet overhead ratio = (# of routing pkts/ # of pkts sent).

Control packet overhead ratio will be calculated as the ratio between the total number of routing packets and the total number of packets sent. The overhead packets in the routing layer include packets both for route discovery and route maintenance e.g. Hello messages, RREQs, RREPs and RERRs.

4.2.4 Simulation Parameters

Network simulator NS-2 (v 2.26) is used to run the experiments due to its extensive support for MANETs and ability to support QoS SWAN module. In this simulation, 50 nodes are considered, distributed over a 1500 x 300 m area and moving randomly. Routing is handled by the AODV with cache on and DSDV routing protocols which are mature routing solution in MANETs. The channel bandwidth is 11 Mbps and the traffic sources are chosen to be CBR with background traffic of up to 32 low priority TCP flows. Each simulation run lasts for 120 seconds in order to allow the network to experience some congestion. The offered load could be varied by changing the CBR packet size, the number of CBR flows or the CBR packet rate. High priority flows are set to 200kbps and 32kbps. Several pairs of source and destinations of real-time and non-real-time traffic flows are manually selected to guarantee the fairness of the simulation. Figure 17 is a screen shot of the simulated MANET with 50 nodes moving randomly in a selected area.

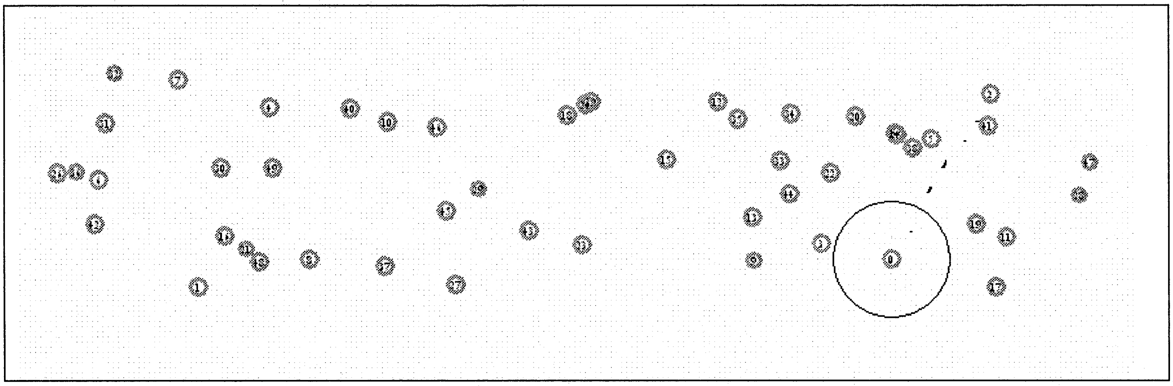


Figure 17: A screen shot of the simulated network.

The simulation parameters used in NS-2 V.2.26 during the network simulation are configured as follows. The channel type is a wireless channel and the radio propagation model is two-ray ground. The two-ray ground model reflects both the direct path and a ground reflection path. MAC layer based on CSMA/CA (as in IEEE 802.11) is used with RTS/CTS mechanism. The data rate at physical layer is 11Mbps. Queue type is "drop tail" and the maximum queue length is 50 packets. The mobile nodes are using the energy model. The energy model in a node has an initial value which is the level of energy the node has at the beginning of the simulation. In this simulation, the initial energy level is set as 100 Joules (1 watt-hour = 3600 J). The rxPower is set to 0.3W (the energy usage for every packet the node receives) and txPower is set to 0.6W (the energy usage for every packet the node transmits). We could calculate that a node can sustain 167s if it transmits packets constantly. When the energy level at a node goes down to zero, no more packets can be transmitted or received by the node. SWAN model parameters are listed below. In this simulation, QoS measurements will be compared with SWAN mechanism ON and OFF to get the performance of the model.

SWAN model parameters:

```
set opt(swan_rc)        "ON"          ;# rate controller ON/OFF
set opt(swan_ac)      "ON"          ;# admission controller ON/OFF
set opt(dir)          "result/test" ;# result directory
set opt(band)           "100kb"      ;# initial rate
set opt(ssthresh)     "1Mb"         ;# slow start threshold
set opt(segment)       "50kb"       ;# increment segment
set opt(mdrate)       "50"         ;# decrement rate
set opt(gap)          "1.2"        ;# gap control
set opt(minband)       "100kb"       ;# minimum rate
set opt(acrate)       "2000kb"      ;# admission control rate
set opt(thrate)       "4000kb"      ;# threshold rate
```

Several pairs of source and destinations of high priority (RT) and low priority (TCP) flows are manually selected. The simulation analysis is based on the average results over different runs. The selected source-destination pair of four high priority video traffic flows are (9, 11), (6, 45) (42, 37) and (20, 23) and voice traffic flows are (28,25), (36,32), (22,41) and (7,14). These pairs are manually selected to create multi-hop paths across the network. The hop counts for both real-time traffic and non-real-time traffic are 2-3. The simulation parameters and the traffic parameter of high priority flows and low priority flows are summarized in the following tables respectively.

Simulation Parameter	**Value**
Number of nodes	50
Simulation area	1500m×300m
Maximum node speed	10 m/s
Simulation time	120s
Routing protocol	AODV, DSDV
MAC protocol	IEEE 802.11
Node initial energy	100 Joules
RxPower / TxPower	0.3W / 0.6W
MAC bandwidth	11 Mbps

Traffic	Value
Offered load	4 video flows, 4 voice flows with 4, 8, 12, 16, 20, 24, 28, 32 low priority TCP flows respectively
Start time for high priority flows	5 seconds after simulation
Packet length and packet interval	Voice: 80 bytes, 0.02s (80×8/0.02=32kbps) Video: 512 bytes, 0.02s (512×8/0.02=200kbps)
Start time for low priority flows	2 seconds
Rate of low priority packet	128kbps

4.2.5 Rate and Admission Control

1. Rate control of BE traffic

Each node in the mobile ad hoc network independently regulates best effort traffic using AIMD rate control algorithm based on feedback from MAC. This feedback measure, used by the rate controller, represents the packet delay measured by the MAC layer. These packet delays, based on a certain threshold, trigger the operation of AIMD. It aims to maximize the transmission rates of best effort traffic under the constraint of packet delay, while providing sufficient bandwidth for real time traffic.

2. Source-based admission control of RT traffic

Admission control applies radio resource monitoring and uses packet delay as feedback result to decide if a flow can be admitted into the network. Signaling, through the use of probe request and probe response packets, is used to determine if the flow can be supported by the network. Because nodes are unaware of mobility and flow re-routing, resource conflicts can arise and persist. False admission is a result of multiple source nodes simultaneously initiating admission control at the same instance and sharing common paths and nodes between source-destination pairs.

4.3 Simulation Results and Analysis

In this section, simulation results are presented. The following QoS metrics are analyzed with SWAN mechanism ON and OFF: average delay, throughput, control packets overhead ratio and goodput. The results are also compared by varying the SWAN parameters to analyze the impact of these parameters on QoS metrics.

4.3.1 Delay Analysis

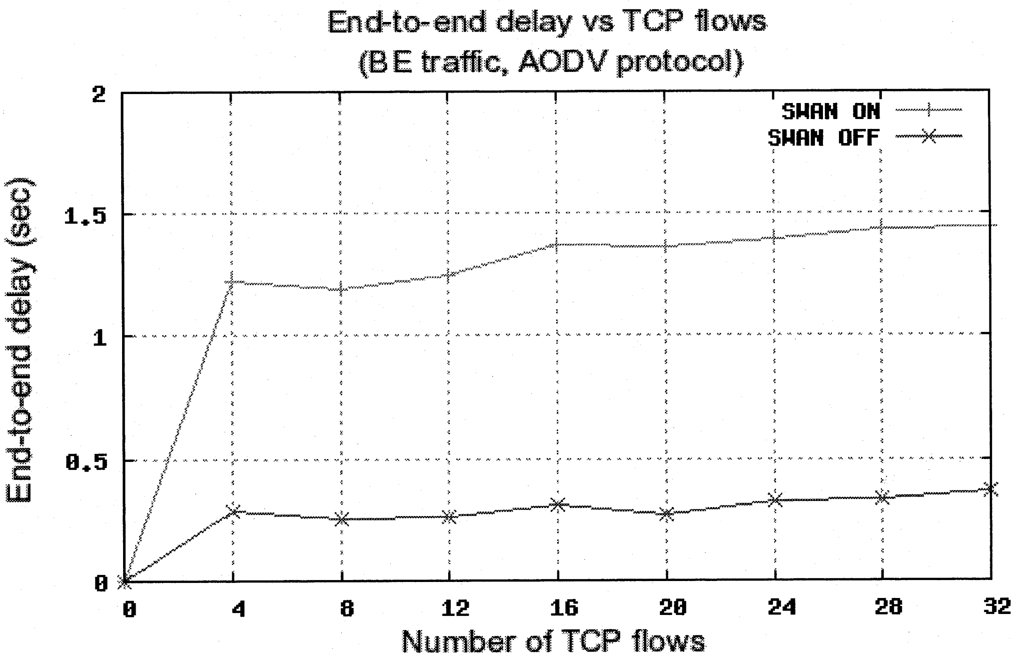


Figure 18: End-to-end delay of best effort traffic vs TCP flows.

TCP Flows	4	8	12	16	20	24	28	32
SWAN ON	1.226620516	1.190979691	1.251883262	1.373105473	1.359692216	1.396165923	1.432991674	1.444313273
SWAN OFF	0.284363797	0.253227183	0.258509987	0.308114018	0.271490079	0.323472652	0.337894147	0.370259218

Figure 18 shows the impact of SWAN model on the delay of BE traffic with growing number of TCP flows from 4 to 32. In this simulation, AODV is used as the routing protocol. The average end-to-end delay of BE traffic with SWAN ON and OFF is compared. We observe that the average delay of BE traffic with SWAN mechanism OFF is much lower compared to the revised system with SWAN mechanism ON. For example, when the number of TCP flows in the network is 24, the delay for the BE traffic in the original system is 0.323s while in the revised system is 1.396s, which is almost 1s longer than in the original system. Although the network load which is varied by the number of background TCP flows, doesn't have much impact on the average delay for BE traffic, we notice that the average delay in the revised system with SWAN ON is always about 300% longer than in the original system. That means the BE traffic is rated controlled by the SWAN mechanism, and when the real-time application detects the RT packets delay become excessive, the rate controller will regulate the BE traffic transmission to give RT packets higher priority to be transmitted. This is in conformance with the SWAN mechanism in which BE traffic is controlled by AIMD algorithm.

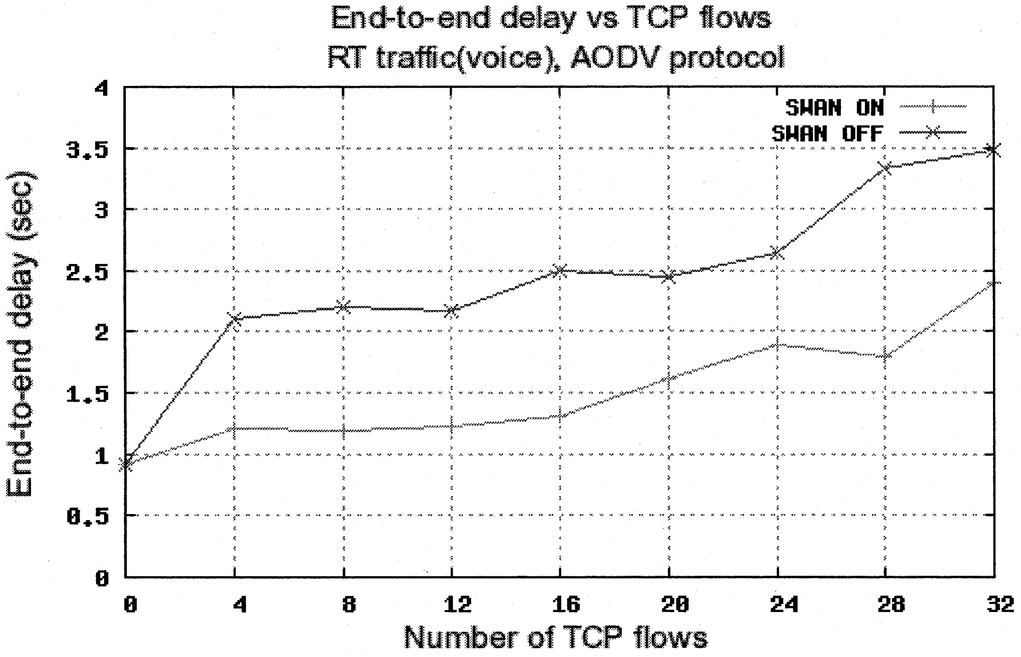


Figure 19: End-to-end delay of voice traffic vs. TCP flows.

TCP Flows	0	4	8	12	16	20	24	28	32
SWAN ON	0.906191622	1.21396202	1.190979691	1.226066226	1.312407749	1.614246943	1.885875603	1.788158584	2.401538898
SWAN OFF	0.906191622	2.104054494	2.210839433	2.169897368	2.495478558	2.454066991	2.652250453	3.338613307	3.473108531

In contrast, the average delay of the RT traffic (voice, video) in the revised system performs much better than the original system as in Figure 19 and 20 depict. It is observed that the average delay in the revised system grows smoothly as the network load grows by varying the number of background TCP flows We notice that in the original system without any regulations about BE traffic, the RT traffic experiences much longer delay compared to the revised system. For example, in Figure 19, when the TCP flows in the network is 24, the average delay for the voice traffic in the original system is 2.65s and in the revised system is only 1.88s. The average delay is 29% shorter in the revised system than in the original system. The numbers in the simulation show that the average delay for the RT traffic is about 30% lower with SWAN ON, but it is at the cost of the delay of the BE traffic which is increased by about 300%. The results presented in this section imply that the revised system can support RT traffic with consistently low delay by controlling the rate of BE traffic for a single shared media channel.

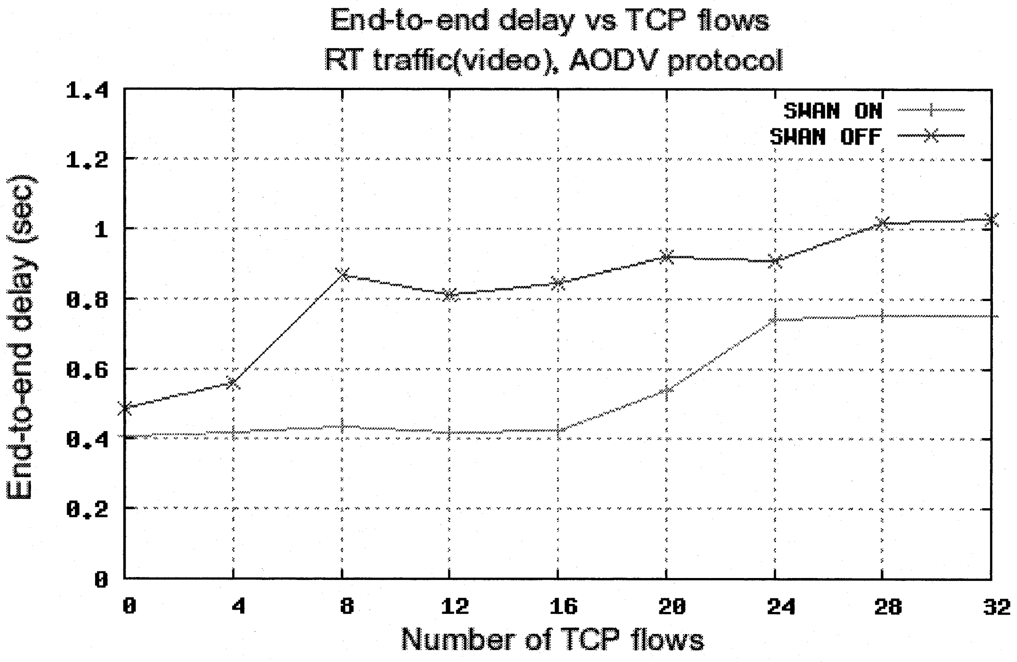


Figure 20: End-to-End delay of video traffic vs. TCP flows.

TCP flows	0	4	8	12	16	20	24	28	32
SWAN ON	0.4080 73281	0.4160 19852	0.4326 86383	0.4181 49459	0.4244 06083	0.5368 98688	0.7413 97221	0.7565 12669	0.7518 74464
SWAN OFF	0.4880 55816	0.5580 0214	0.8681 45782	0.8136 38103	0.8459 21846	0.9208 39013	0.9074 74962	1.0170 15241	1.0306 45405

Figure 20 is similar to Figure 19, but it shows the average delay of the video traffic with SWAN ON/OFF. The average delay for the video traffic is also about 20%

lower with the SWAN mechanism ON when the number of TCP flows equals to 24. It again demonstrates that using SWAN model can achieve better QoS performance for real-time applications.

In the next simulation, we use a different MANET protocol (DSDV) for the network to test the compatibility of the SWAN model working with other MANET protocols. The traffic pattern remains the same, 4 voice and 4 video traffic flows as high priority RT traffic and a different number of TCP traffic flows (4-32) as background BE traffic to simulate the load of the network. Figure 21 and 22 show that SWAN model can also work with DSDV protocol.

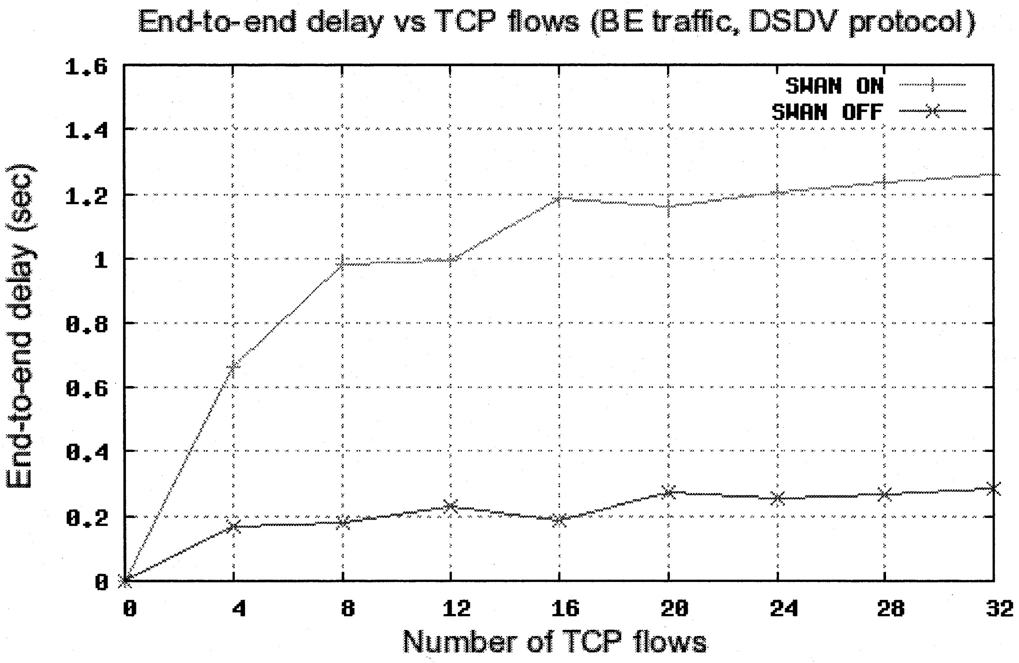


Figure 21: End-to-end delay of TCP traffic with DSDV protocol.

TCP flows	0	4	8	12	16	20	24	28	32
SWAN ON	0	0.663216698	0.977569322	0.977569322	1.182183781	1.15662215	1.200298406	1.235045556	1.260217215
SWAN OFF	0	0.167388908	0.177383563	0.228500555	0.186045738	0.271956266	0.25395226	0.264748594	0.282862335

Figure 21 shows the impact of the SWAN mechanism on the average end-to-end delay in the networks using DSDV protocol. It shows that the BE traffic is sacrificed to guarantee a more stable and lower delay for the RT traffic, as using AODV protocol. In the simulation, such as for 24 TCP traffic flows, the average delay in the original system is only 0.254 seconds for BE traffic while with the SWAN model to control the BE traffic rate, the average delay reached about 1.200 seconds which is 372% higher than in the original system. We also observe that when the network load is lighter such as the number of TCP flows is less than 16, the average delay for BE traffic grows linearly with the network load in SWAN network, but when the traffic density is higher, the delay for BE traffic grows very smoothly which demonstrates a little bit different behavior compared to AODV protocol in which the network load doesn't have much impact on the average delay for BE traffic. That is because AODV protocol finds routes by on-demand request while DSDV protocol stores the route table on each node.

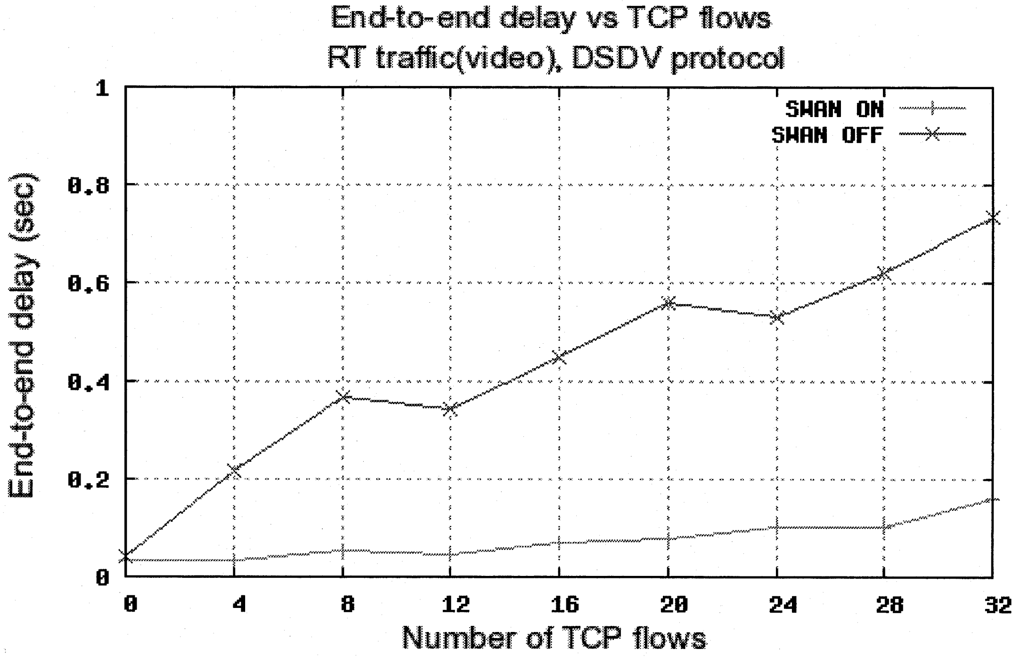


Figure 22: End-to-end delay of real-time traffic with DSDV protocol.

TCP flows	0	4	8	12	16	20	24	28	32
SWAN ON	0.033281726	0.030824889	0.052352591	0.045927517	0.068444092	0.078713361	0.102008933	0.101800708	0.159826709
SWAN OFF	0.038965119	0.214460311	0.366813021	0.340867299	0.449535079	0.560731384	0.531662224	0.621871142	0.735199784

Figure 22 shows that the RT traffic (video) delay is controlled effectively by the SWAN mechanism using DSDV protocol. We observe that the average delay of the RT traffic shows a significant difference between the original and revised systems. In the original system, the delay of RT traffic grows from 0.21 to 0.74 second as the number of TCP flows increases from 4 to 32 flows. In contrast, the average delay of RT traffic in the revised system always remains around 0.05s to 0.10s and grows smoothly when the background BE traffic grows. In addition, the average delay of the RT traffic remains consistently and stably low and about 75%-86% lower than in the original system. It again demonstrates that the RT traffic delay is effectively controlled by AIMD algorithm in the SWAN network.

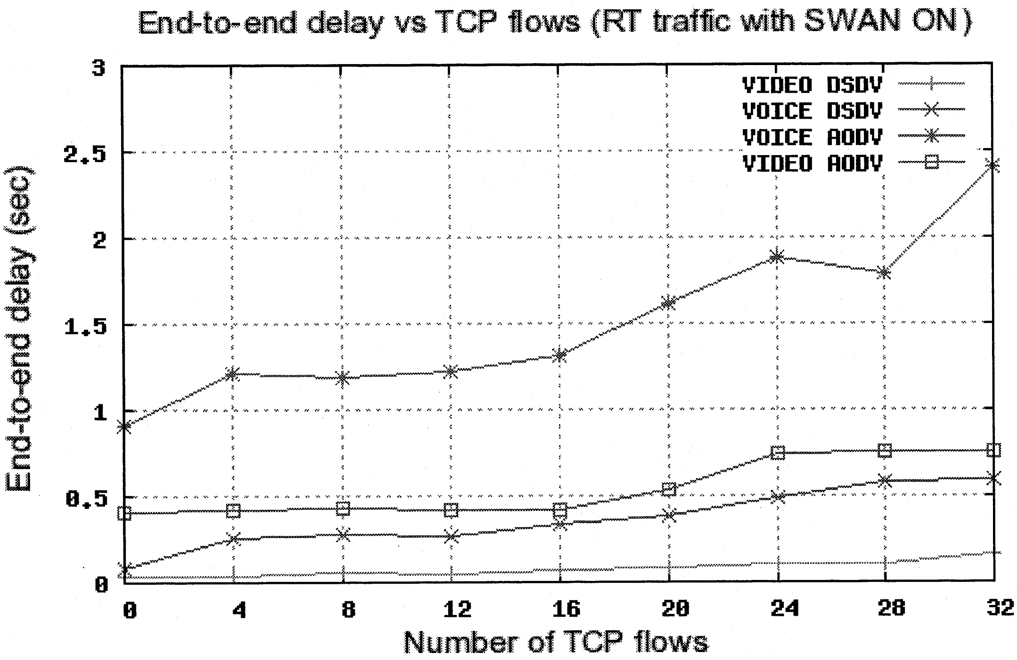


Figure 23: The comparison of DSDV and AODV protocol in end-to-end delay.

In Figure 23, we compared the average end-to-end delay of RT traffic (both voice and video) using AODV and DSDV protocol with SWAN ON. It shows that DSDV outperforms AODV protocol with SWAN mechanism in terms of controlling the RT traffic delay. The two protocols with SWAN ON both have the capability to control the BE traffic to guarantee a better QoS for the real-time applications. For example, in high traffic density environment such as when the number of TCP flows is 24, the delay of video traffic is 0.102s using DSDV protocol while it is 0.741s in the network using AODV protocol. The reason is that in AODV, the routing is discovered on demand and more control packets are needed, thus it takes more time to establish a route in AODV. We also observe that even in the SWAN OFF system, DSDV outperforms AODV protocol in terms of traffic delay, but that does not mean DSDV is better than AODV in terms of delivering the RT traffic. If the number of nodes grows largely, the number of routes will increase rapidly. This effect together with the mobility of nodes may increase the control packets overhead due to the maintenance of the routes, consuming the scarce bandwidth in the MANETs and therefore reducing the throughput of such pro-active protocols as DSDV and hence decrease its performance.

4.3.2 Throughput Analysis

In the next simulation, we compare the throughput of the network with SWAN model ON/OFF using DSDV protocol since DSDV protocol is relatively stable in route selection. The simulation network and traffic pattern remains the same with 50 nodes moving randomly in a 1500m×300m area. The throughput is calculated separately for BE traffic and RT traffic.

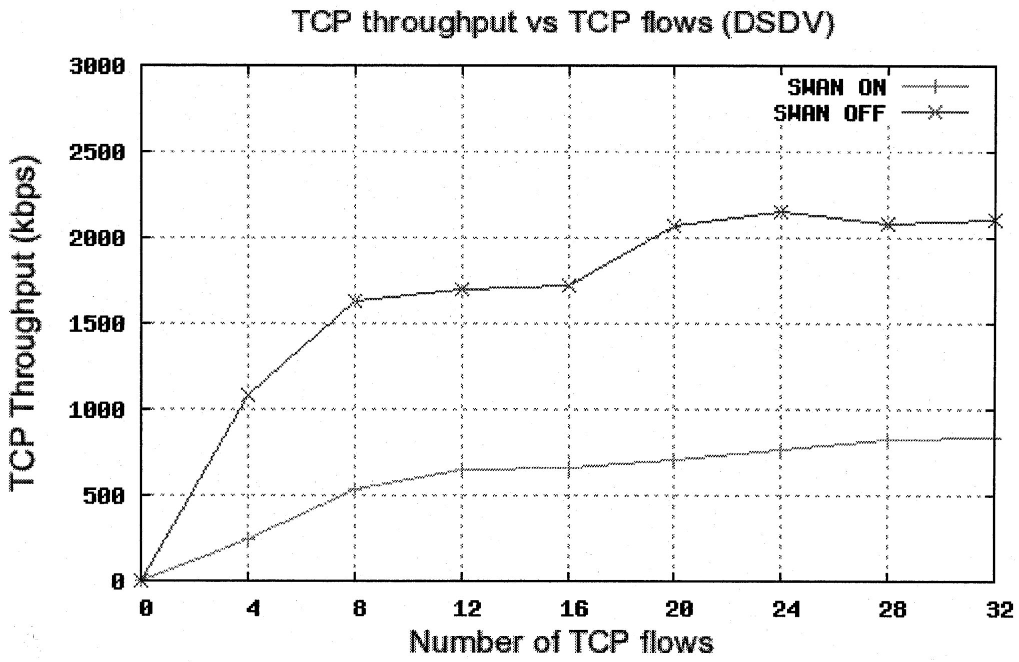


Figure 24: BE traffic throughput vs. TCP flows using DSDV protocol.

TCP flows	0	4	8	12	16	20	24	28	32
SWAN ON	0	240.9590	533.0488	652.6036	667.282	709.4049	767.9821	826.4909	838.3873
SWAN OFF	0	1085.9353	1623.3317	1702.3213	1715.9339	2066.2413	2147.13	2086.1027	2102.7695

Figure 24 depicts the BE traffic throughput vs. the number of TCP flows with SWAN ON and OFF using DSDV protocol. It shows that the BE traffic throughput is impacted by BE traffic rate control mechanism in SWAN model. For example, when the number of TCP flows equals to 24, the average BE traffic throughput with SWAN ON is about 768 kbps, while in the original system, it could reach 2147 kbps which is 180% higher. By adopting the SWAN control mechanisms, we observe that the reduction in the average delay of the RT traffic is at a cost of the loss of the BE traffic throughput. AIMD Rate control is designed to restrict BE traffic yielding the necessary bandwidth required to support RT traffic. Rate control also allows the BE traffic to efficiently utilize the bandwidth that is not utilized by the RT traffic at any moment. Due to these reasons, the network with SWAN mechanism will get lower throughput than the original network.

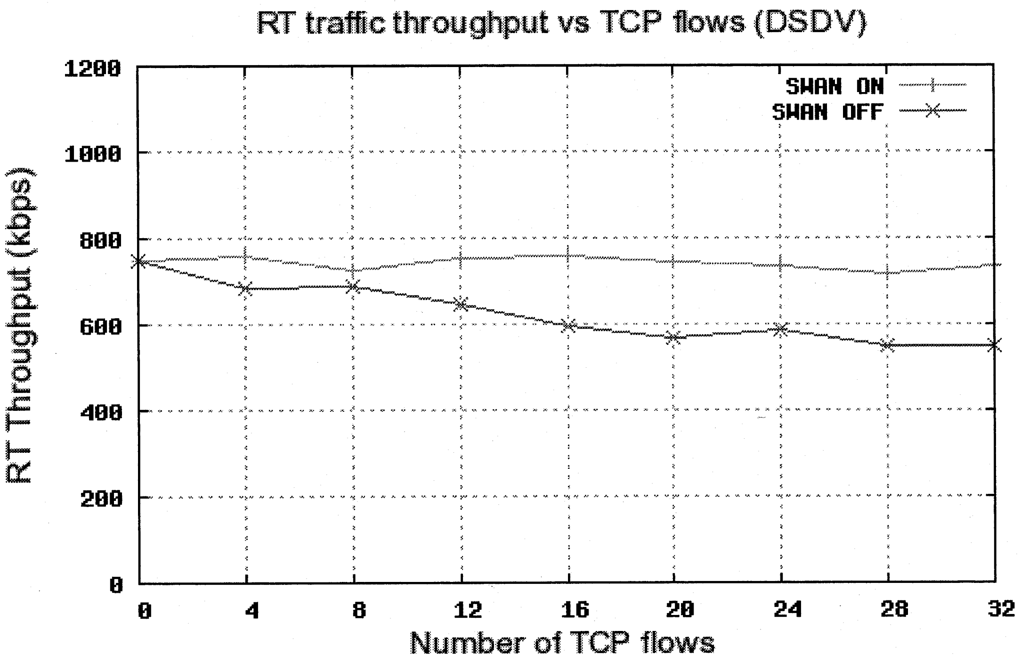


Figure 25: RT traffic throughput vs. TCP flows using DSDV protocol.

TCP flows	0	4	8	12	16	20	24	28	32
SWAN ON	749.9512	758.8528	726.91587	751.29667	758.40773	744.14387	736.84813	716.7376	735.79227
SWAN OFF	747.8472	684.40453	686.50107	644.7464	597.04213	569.6912	585.2752	550.99173	549.57867

Figure 25 shows the impact of the SWAN mechanism on the RT traffic throughput using DSDV protocol. It shows that the RT traffic throughput with SWAN

ON is higher than the original system. When the number of TCP flows is 24, the throughput of the original system is 585 kbps while in the revised system, it is 737 kpbs which is 26% higher. The increase of the RT traffic throughput is at the cost of the BE traffic throughput. This is a promising result as the SWAN model controls the BE traffic to guarantee the bandwidth for delivering the RT traffic. The entire network throughput is decreased mainly due to the drop of the BE traffic throughput.

4.3.3 Control Packet Overhead Ratio

Reactive protocols such as AODV find routes when needed by a source. They usually rely on flooding when no topology information is available. Proactive protocols such as DSDV proactively discover the topology with every node emitting regular "hello" packets and an optimized mechanism is used to broadcast local topology information. These two approaches have different characteristics with regard to control traffic overhead. Reactive protocols generate overhead only when a new route is needed, while proactive protocols continuously generate control traffic. Link failure, mainly due to mobility, will produce additional overhead with both approaches. In a reactive protocol, routes either have to be repaired or rediscovered. In a proactive protocol, the broadcasted topology in the network has to be updated to reflect the change.

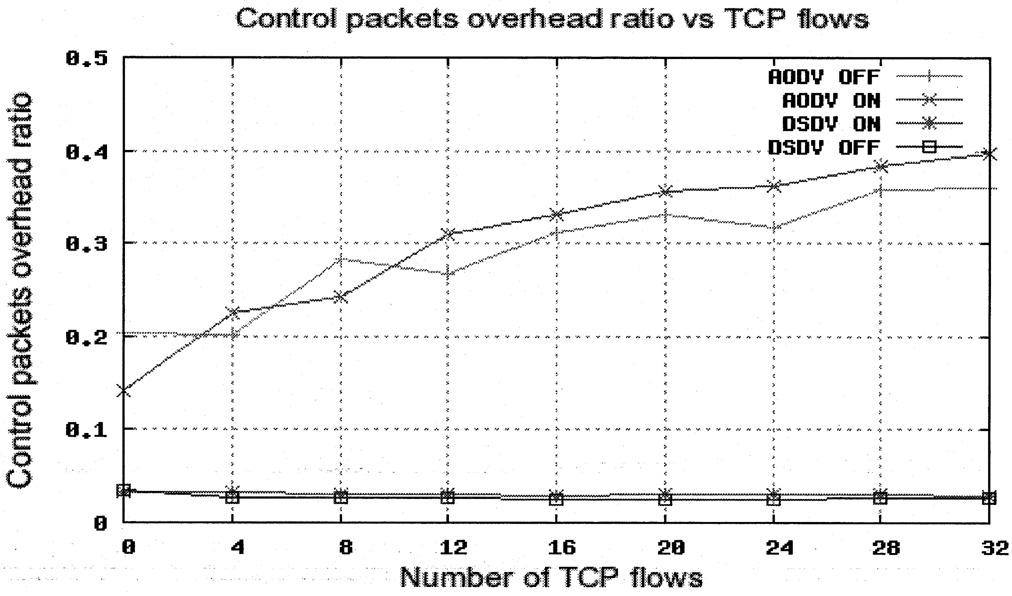


Figure 26: Normalized control packets overhead as a function of TCP flows.

TCP flows	0	4	8	12	16	20	24	28	32
AODV SWAN ON	0.1424111	0.2245909	0.242167	0.3095908	0.3320372	0.3567525	0.3632485	0.3827916	0.397564
AODV SWAN OFF	0.2025857	0.2012113	0.2829045	0.2668881	0.2910099	0.3310615	0.3182851	0.3581822	0.3598396
DSDV SWAN ON	0.0324712	0.0327264	0.0301549	0.0309648	0.0298171	0.0301304	0.0310299	0.0302328	0.0287862
DSDV SWAN OFF	0.034485	0.0278684	0.0270958	0.0267848	0.0258447	0.025995	0.0258758	0.0278752	0.0263999

Figure 26 shows the normalized control packets overhead with SWAN mechanism ON and OFF using AODV and DSDV protocol. The normalized control packet overhead is calculated as the ratio between the total number of routing packets and the total number of packets sent. The network topology and traffic pattern remain the same as the previous simulation. From Figure 26, we observe that DSDV has constant control packets overhead while AODV has much higher control packets overhead and as the number of background TCP flows grows, the control packets overhead grows accordingly. For example, when the number of TCP flows is 24, for DSDV protocol, the SWAN ON has 0.031 control packets overhead ratio and SWAN OFF is 0.026. And it remains constant as the traffic density grows, as always around 10% increase when the SWAN mechanism is ON for DSDV protocol. For AODV protocol, the control packets overhead ratio grows linearly, 0.363 for SWAN ON and 0.318 for SWAN OFF. SWAN incurs about 14% increase. In general, SWAN incurs about 10% increase of control packets overhead for both pro-active (DSDV) and reactive (AODV) protocols in this simulation.

SWAN mechanism incurs higher overhead because of its source admission control and rate control mechanism, which result in less data packets sent across the network. For AODV protocol, as the data sources increase, more routes need to be discovered and hence, more routing packets are needed, that explains the linearly increased control packets overhead for AODV protocol. Whereas in DSDV, the number of routing packets sent is almost constant since it generates the routing packets periodically to maintain the routing information no matter whether the sources are sending data packets or not. But DSDV protocol is not scalable. In this simulation, the network has only 50 nodes which is regarded as a small size network. For large network topology, AODV protocol will outperform DSDV.

4.3.4 SWAN Model Parameters

SWAN uses AIMD rate control algorithm to control transmission rate of BE traffic. The algorithm works like this: every T seconds, each mobile device increases its transmission rate gradually (additive increase with increment rate of ***c*** Kbps) until the packet delays become excessive. The rate controller detects excessive delays when one or more packets have greater delays than the threshold delay ***d*** sec. As soon as the rate controller detects excessive delays, it backs off the rate (multiplicative decrease by ***r%***). It could simply be understood as this:

```
If (n>0) /* one or more packets have delays
    $S = S * (1-r/100)$ /* multiplicative decreased by r%
Else
    $S = S + C$ /*additive increased by C Kbps
```

TCP flows are rate controlled with parameter ***c*** and parameter ***r***, while voice and video flows are not rate controlled once admitted through the source-based admission control process.

<u>Increment Rate</u>

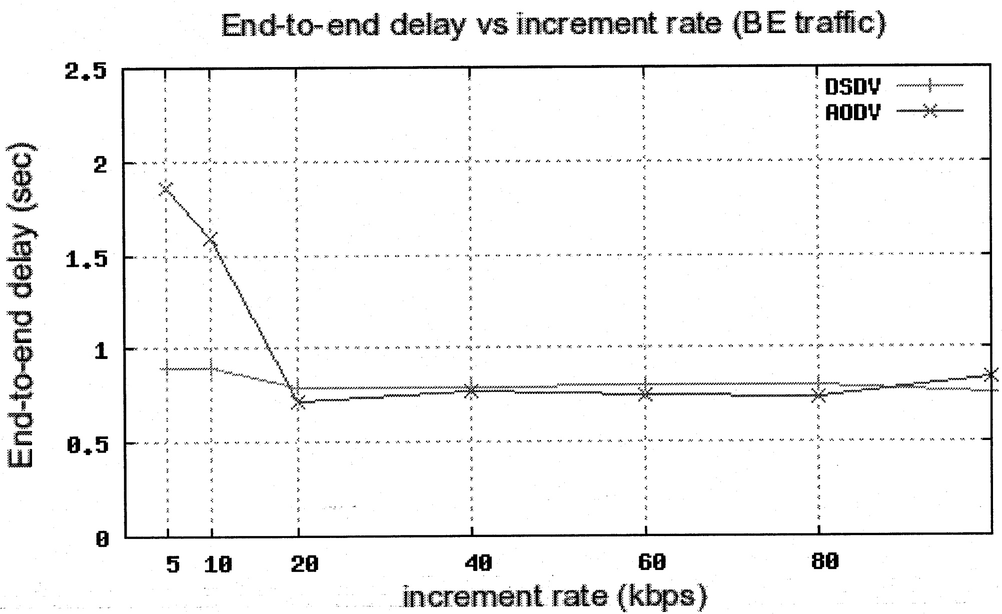


Figure 27: The impact of increment rate on end-to-end delay for BE traffic.

Increment rate	5	10	20	40	60	80	100
AODV	1.85433	1.58670	0.71681	0.76894	0.75010	0.74220	0.84078
DSDV	0.89761	0.89949	0.79561	0.78691	0.80310	0.79904	0.76018

Figure 27 shows the impact of increment rate *c* on the average end-to-end delay of BE traffic. The simulation is done in this environment: there are 32 BE traffic flows and 4 video flows as RT traffic. Each node is moving at 10m/s in the network. We observe that when the increment rate is more than 20kbps, it does not have much impact on the average end-to-end delay for both AODV and DSDV protocols. But when the increment rate is less than 20kbps such as 5kbps, the delay for AODV protocol is 1.85s which is almost 160% higher than the delay at the other increment rates. For DSDV protocol, the delay at 5kbps is 0.89s which is 11% higher. The increment rate is for regulating the BE traffic. When there is enough available bandwidth for BE traffic, the BE traffic transmission rate will be increased by the increment rate. It is a promising result since when the increment rate is small, the BE traffic is limited not to be increased accordingly and hence incurs higher delay. When the increment rate is chosen properly, the delay for BE traffic is controlled properly as well.

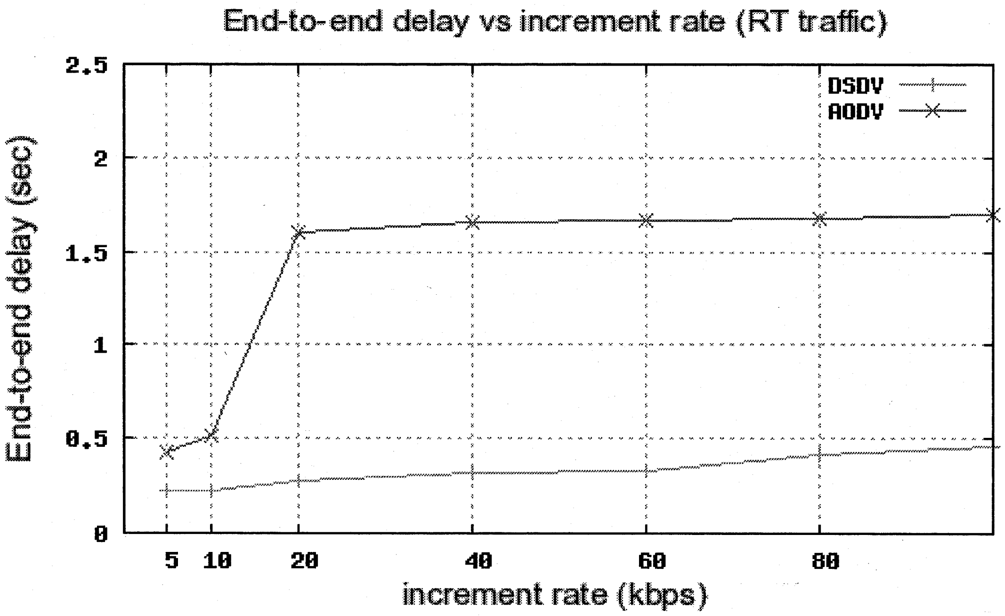


Figure 28: The impact of increment rate on end-to-end delay for RT traffic.

Increment rate	5	10	20	40	60	80	100
AODV	0.43064	0.51019	1.59810	1.65264	1.66688	1.67606	1.69383
DSDV	0.22615	0.22702	0.27520	0.32183	0.32848	0.41825	0.45627

Figure 28 shows the impact of increment rate ***c*** on the end-to-end delay of RT traffic. Still, we observe that when the increment rate is above 20kbps, the delay for RT traffic remains almost constant (around1.6s). But when the increment rate is less than 20kbps, which means the BE traffic is only increased at 20kbps every ***T*** seconds when there is not much delay in the network, the RT traffic delay becomes smaller (0.43s at 5kbps, and 0.51s at 10kbps). The small delay for the RT traffic is at the cost of the delay of the BE traffic. Therefore, we need to choose carefully the increment rate to keep the balance between the delay of RT traffic and delay of the BE traffic.

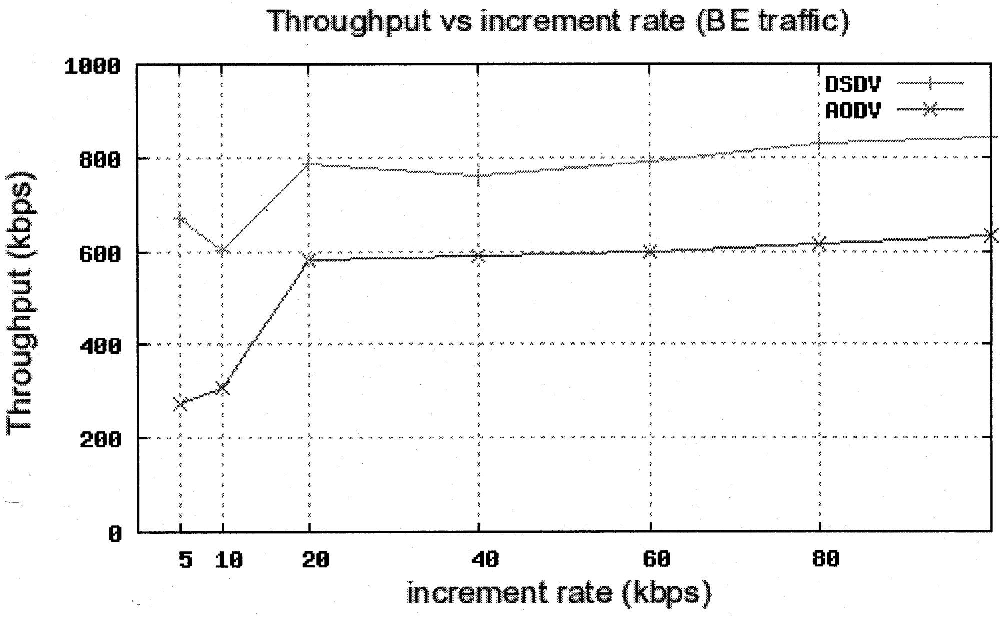


Figure 29: The impact of increment rate on throughout for BE traffic.

Increment Rate	5	10	20	40	60	80	100
AODV	272.6976	307.5546	579.2854	590.6646	597.7837	613.8426	631.5034
DSDV	670.7453	603.4931	784.6109	759.2870	792.2870	830.5971	840.9344

Figure 29 shows the impact of increment rate on the throughput of the BE traffic. It shows that the total throughput of BE traffic is noticeably decreased when a small value of parameter *c* – increment rate is chosen. Particularly for AODV protocol, the throughput of TCP flows at small increment rate such as 5kbps is only 273kbps, which is only 46% of the throughput at increment rate 40kbps. So again, we need to choose carefully the increment rate for the SWAN model to balance the throughput of the BE traffic and RT traffic.

Decrement Rate

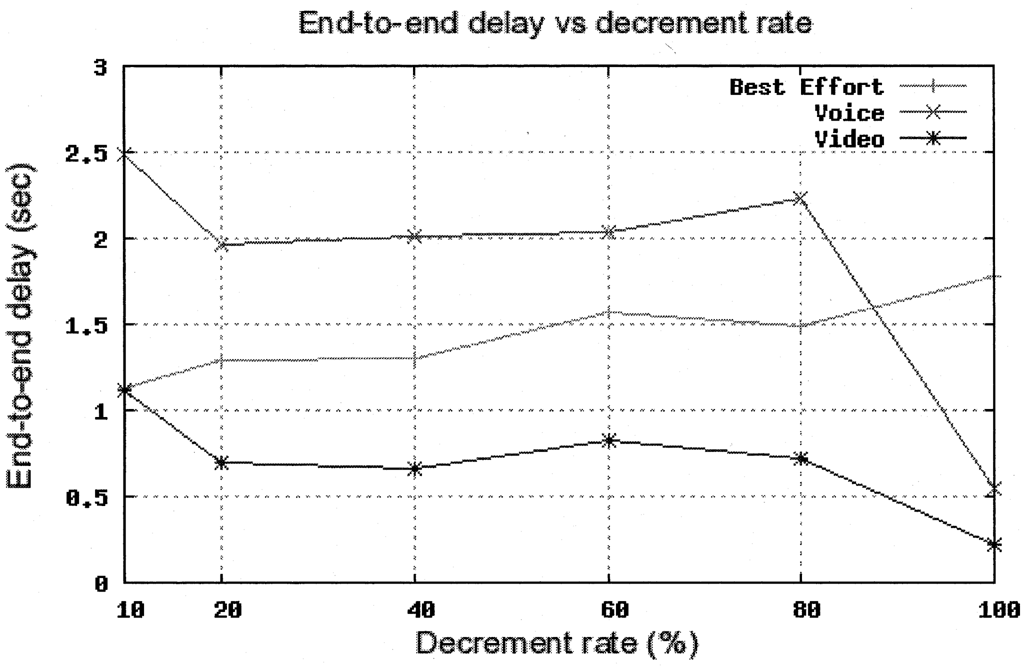


Figure 30: The impact of decrement rate on the end-to-end delay.

Decrement rate	10	20	40	60	80	100
Best effort	1.1301825	1.2957750	1.3061663	1.5677888	1.4848169	1.7808164
Voice	2.4885650	1.9648064	2.0153547	2.0329153	2.2292350	0.5503388
Video	1.1108730	0.6934064	0.6583377	0.8239492	0.7152962	0.2179786

Figure 30 depicts the impact of decrement rate on the end-to-end delay of BE traffic and RT traffic. The x-axis in this figure represents the value for parameter ***r*** (decrement rate, %). The simulation is taken with AODV protocol and 32 background TCP flows. From this simulation, we can see that when small decrement rate (10%) is chosen, the end-to-end delay of BE traffic is also small but the RT traffic delay is negatively impacted by the small decrement rate since it doesn't control much of the BE traffic to yield the bandwidth for RT traffic. As the decrement rate goes up from 80% to 100%, the delay of BE traffic is, on the contrary, affected by being increased 15% (from 1.48s to 1.78s). But for RT traffic such as voice traffic, its delay is as small as 0.55s while the average delay for the decrement rate (40%-80%) is around 2.0s. So choosing a small decrement rate will give the network a better throughput and better performance for BE traffic and choosing a large decrement rate will give a better performance for RT applications. The combination of increment rate and decrement rate should be chosen carefully to satisfy the requirements for different applications.

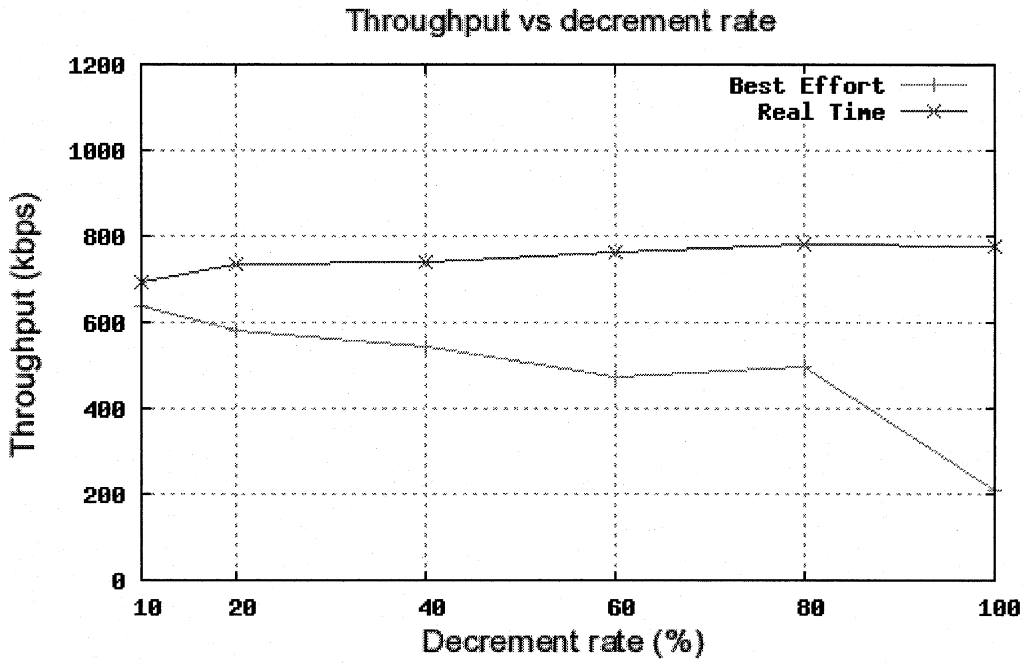


Figure 31: The impact of decrement rate on throughput.

Throughput	10	20	40	60	80	100
Best effort	637.75013	581.996	543.13067	472.63707	499.99013	207.76693
RT traffic	691.47573	736.13253	741.0896	762.80827	782.32373	777.29787

Figure 31 depicts the impact of decrement rate on the networks throughput. When a small decrement rate is chosen such as 20%, the BE traffic throughput is 581 kbps, but as the decrement rate increases, the BE traffic throughput is controlled (by the decrement rate *r%*)and dropped. When the decrement rate hits 100%, the BE traffic throughput is declined as low as 207 kbps, which is only 36% of the BE traffic throughput at 20% decrement rate and 38% at decrement rate 40%. For RT traffic, it steadily increases as the decrement rate increases. Therefore, although choosing large decrement rate can improve the QoS performance for real-time applicatoins (both for delay and throughput), it is at the cost of the throughput and delay of the BE traffic.

Admission Control Rate

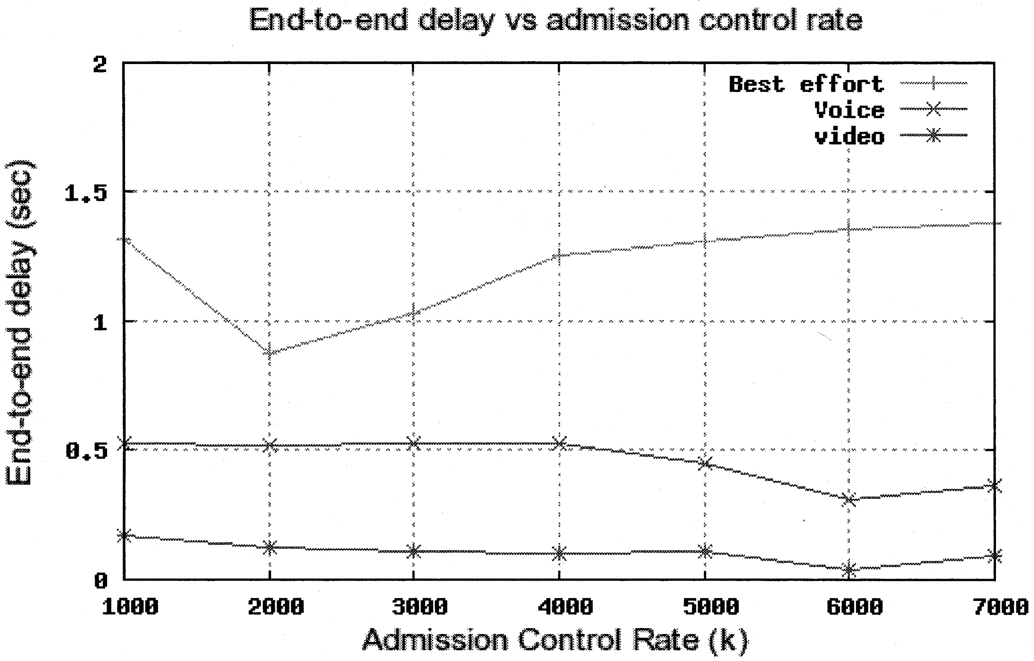


Figure 32: The impact of admission control rate on the average end-to-end delay.

Admission Control Rate	1000k	2000k	3000k	4000k	5000k	6000k	7000k
Best effort	1.319809	0.874886	1.031097	1.256037	1.308109	1.358908	1.383715
Voice	0.530385	0.52317	0.523861	0.525274	0.447506	0.306585	0.364878
Video	0.170458	0.124212	0.107528	0.096965	0.104722	0.041611	0.090705

In this simulation, source admission control mechanism for RT traffic is analyzed. DSDV is chosen as the network protocol and there are 32 TCP flows and 4 video, 4 voice RT traffic flows in this network. The admission control rate is varied from 1000k to 7000k for RT traffic. Another parameter - threshold rate is also varied accordingly in this simulation. The bandwidth availability for BE traffic is the bandwidth difference between the threshold rate and the current rate of the RT traffic. To guarantee the transmission of

BE traffic, we always keep the difference between the threshold rate and admission control rate (for RT traffic) to 2000k. The simulation results show that when small admission control rate -1000k and threshold rate -3000k are chosen, the delay for both the RT traffic and BE traffic is higher than when the other control rate is chosen. This is because the admission control rate and threshold rate are chosen so conservatively that the RT traffic will take all the available bandwidth up to the admission control rate and only leaves the rest for BE traffic which makes the BE traffic starved. But when the admission control rate is increased to a reasonable amount, the RT traffic may not take all the bandwidth available up to the admission control rate and hence the rest bandwidth can be used by BE traffic. And we also observe that when the admission control rate is increased which means that more RT traffic can be allowed to be transmitted in the network, the delay for RT traffic decreases accordingly and BE traffic increases slightly when large number of RT traffic flows is allowed. But due to the dynamics of the mobile network channel, when the admission control rate is large, for example, when the admission control rate is 7000k and threshold rate is 9000k, the delay for both RT traffic and BE traffic is increased. That is because although there is enough bandwidth available for RT traffic and hence it is allowed to be transmitted by the source admission controller, due to the network dynamics, the transmission has to be dropped when there is a bottleneck node or the network can not guarantee the bandwidth anymore due to the dynamics. Hence choosing large control admission rate doesn't necessarily mean more RT traffic can be granted, it also needs to be taken into consideration such variables as network dynamics, node mobility, etc.

4.3.5 Node Mobility

The nodes in a MANET may move completely independently and randomly as far as the communications protocols are concerned. This means that topology information has a limited lifetime and must be updated frequently to allow data packets to be routed to their destinations. Again, this invalidates any hard packet delivery ratio or link stability guarantees. Furthermore, QoS state which is link or node position dependent must be updated with a frequency that increases with node mobility. An important general assumption must also be stated here: for any routing protocol to be able to function properly, the rate of topology change must not be greater than the rate of state information propagation. Otherwise, the routing information will always be stale and routing will be inefficient or could even fail completely.

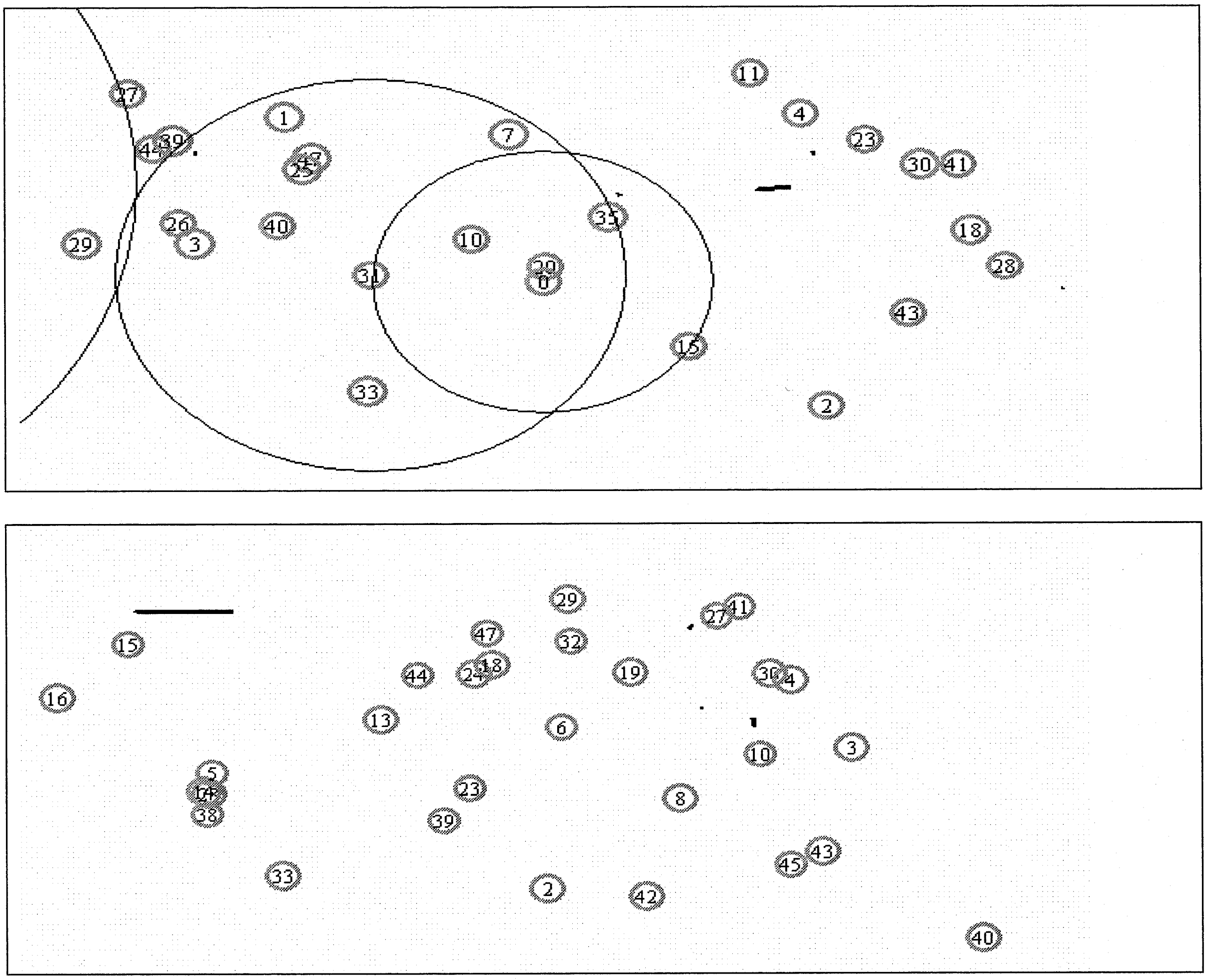


Figure 33: Illustration of node mobility in NS-2.

In this section, we consider the impact of node mobility on the performance of SWAN model. DSDV is used for routing protocol in the simulated network. The network topology and traffic pattern remain the same as the previous simulation. Figure 33 is a screen shot that depicts the moving nodes in a way-point moving model. In this simulation, the node mobility will be simulated by varying the nodes moving speed and pause time. The initial position of the 50 nodes is the same for both simulations.

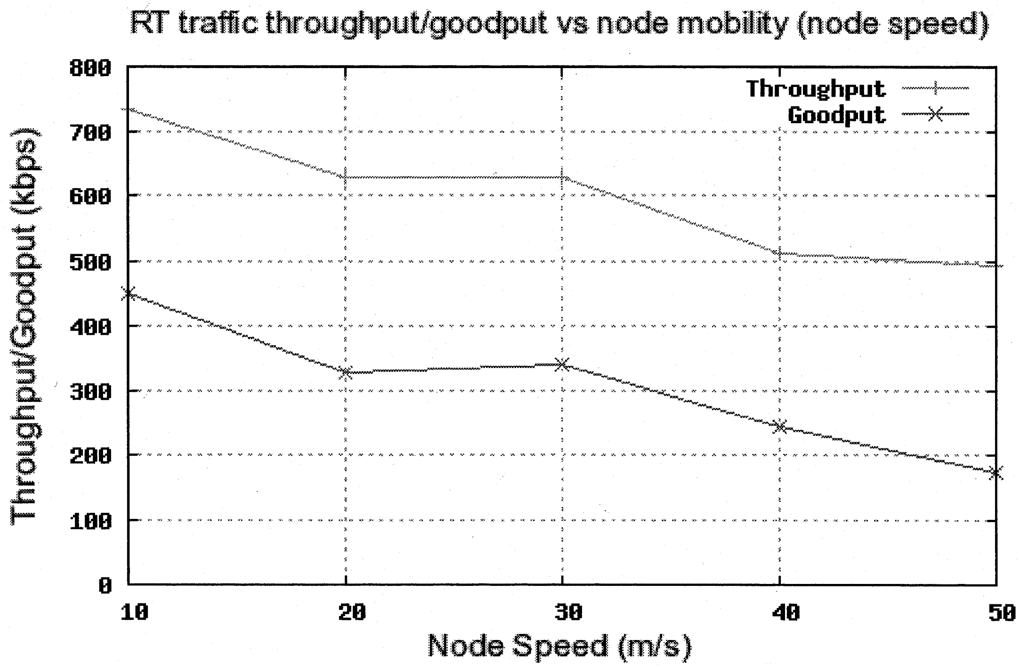


Figure 34: Goodput of the real-time traffic as a function of node speed.

Node speed	10	20	30	40	50
Throughput	735.9216	629.8826	629.0445	510.8445	492.0162
Goodput	451.0698	328.5290	340.7946	246.4373	174.288

Figure 34 shows the impact of node moving speed on the throughput and goodput of the RT traffic. Each mobile node selects a random destination and moves with a random speed from 10 m/s up to a maximum speed of 50m/s when the destination is reached. As shown in Figure 34, both the average goodput and throughput for RT traffic in the system with SWAN mechanism are decreased as node mobility increases. When the node speed increases from 10m/s to 50m/s, the goodput of the real-time traffic drops almost 61% (from 451 kbps to 174 kbps) and 33% for throughput (from 735 kbps to 492 kbps for). Hence we conclude that as the node speed increases, the goodput of the RT traffic drops due to the network dynamics which means the real-time applications' quality will suffer from the node mobility.

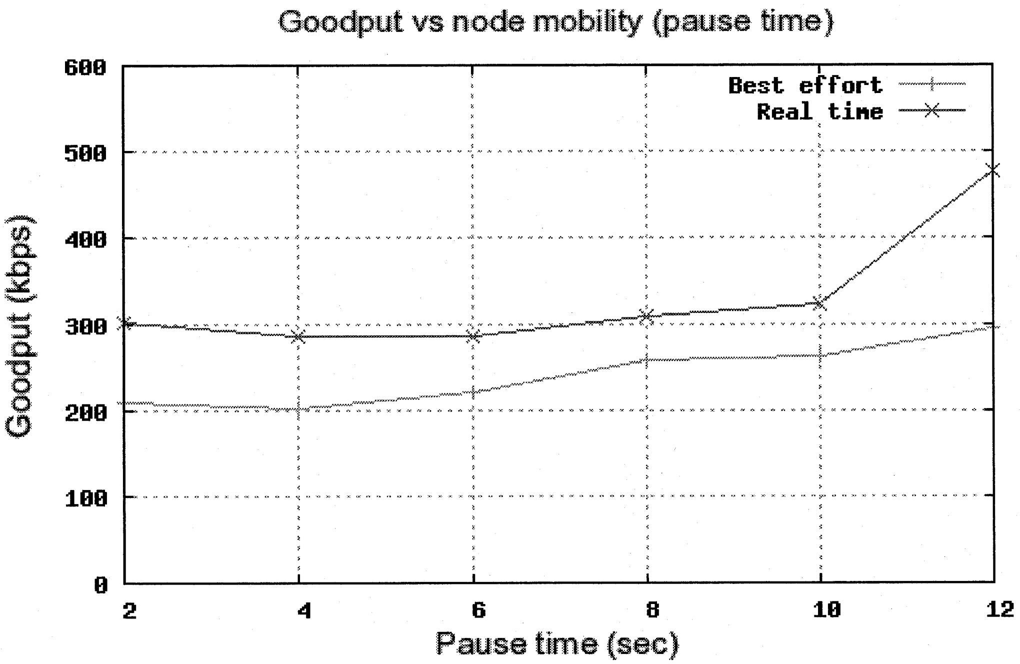


Figure 35: Goodput vs node mobility (pause time).

PAUSE TIME	2	4	6	8	10	12
Best Effort	208.6570	203.1274	220.3989	258.7648	263.0997	295.936
Real Time	302.9514	286.7904	286.1216	308.8426	322.5610	475.8677

Figure 35 shows the impact of node mobility (pause time) on the goodput for both RT traffic and BE traffic. We also observe that when the pause time decreases which means the node mobility increases, the goodput of real-time traffic drops accordingly. For example, when the pause time equals to 12 second, the goodput for RT traffic is 475 kbps, and it drops 36% to 302 kbps when the pause time is only 2 second. For BE traffic, the goodput also suffers as the network mobility increases. But for specific applications, the goodput for real-time traffic is more important than goodput for BE traffic since if the packets arrive at the destination out of order for RT applications, they may not even be useful and hence have to be dropped. The impact of mobility on the goodput is due to route discovery latency and congestion along the new route.

4.4 Conclusions

From the simulation study, we confirm that the SWAN mechanism improves the performance of the ad hoc network in terms of end-to-end delay when there is large traffic in the network (that is when the number of background traffic flows is high). In addition, the SWAN model shows compatibility of working with different routing protocols such as AODV and DSDV and the reliability in a dynamic network environment with different node mobility with varying moving speeds and pause time. The decrease of the end-to-end delay of real-time traffic is at the cost of the delay of the best-effort traffic and the sacrifice of the throughput of the network. By adopting a proper control mechanism, we observe that up to 20%-30% reduction for the average delay of the real-time traffic (voice and video), but it is at a cost of the BE traffic delay and throughput. But considering the rate control mechanism and source admission mechanism of the SWAN model, it provides reasonable and promising results. The reason is that, in real time transmissions, only the packets that arrive at the destination in time are useful, and the packets that arrive late are useless for the application; hence, the SWAN model guarantees the delivery of the real-time traffic. With the SWAN mechanism, source admission control insists on trying to find a route which has enough data rate to send the traffic to ensure that the packets arrive at the destination on time. In addition, on-going TCP traffic is decreased when the promised data rate cannot be provided by the routes any more. These strategies make more packets to be dropped during the simulation and hence reduce the throughput and goodput of the network. Without these strategies, the real time traffic will keep on sending even when the data rate of the route cannot satisfy the request. As a result, those packets sent during this period are more likely to subject to more delay and might be useless when they arrive to the destination. To sum up, it is reasonable to drop packets at the source when the QoS can not be guaranteed in the system. It also helps to decrease the traffic in the network. The good performance with the SWAN mechanism is achieved at the expense of the throughput and delay of the best-effort traffic.

5 Conclusions

5.1 Summary

In this project, we reviewed the current research on QoS support in MANETs. Although all of the research focus on different problems, they are highly related to each other and have to deal with some common difficulties, which include mobility, limited bandwidth and power consumption, and broadcast characteristic of radio transmission.

In Chapter 2, several solutions proposed for QoS provisioning in MANETs were discussed. First the issues and challenges involved in providing QoS in MANETs were identified. Then the existing QoS approaches were examined. Finally, some of the important QoS frameworks for MANETs were described.

In Chapter 3, a thorough overview of QoS routing metrics and design considerations were provided. Then many of the major contributions to the QoS routing solutions are reviewed. The protocols were selected in such a way as to highlight many different approaches to QoS routing in MANETs. The operation, strengths and drawbacks of these protocols in order to enunciate the variety of approaches proposed and to expose the trends in designers' thinking were summarized in this chapter.

And last, our simulations revealed that the QoS model - SWAN is suitable for the guarantee of QoS in MANETs when the traffic of the network is higher. It could help to ensure the end-to-end delay of the transmission as well as constrain the useless transmissions in the network. Simulation, analysis, and results from an experimental wireless show that real-time applications experience low and stable delays under various multi-hop, traffic and mobility conditions.

5.2 Future Work

Recently it has become evident that a traditional layering network approach (separating routing, scheduling, rate and power control) is not efficient for QoS routings in MANETs. The main building blocks of a wireless network design are routing, rate control, medium access (scheduling) and power control. These building blocks are divided in layers. Typically, routing is considered in a network layer and medium access in a MAC layer, whereas power control and rate control are sometimes considered in a PHY and sometimes in a MAC layer. Nowadays, the cross-layer design approach is the most relevant concept in mobile ad hoc networks which is adopted to solve several the open issues for QoS. It aims to overcome ad hoc networks performance problems by allowing protocols belonging to different layers to cooperate and share network status information while still maintaining separated layers. In particular, the mechanisms on how to access the radio channel are extremely important to guarantee QoS and improve application performance.

Cross-layer design remains a growing research area. Information from other layers is shared between the layers to facilitate optimal configuration. In general, the protocol stack is treated as a single mathematical construct that must be jointly optimized due to complex interdependencies. Most deployed techniques are singularly-adaptive and do not typically address the complete problem space of 1) adaptation in support of application QoS requirements, 2) adaptation in reaction to harsh, time-varying channel conditions, and 3) adaptation of technology to support seamless interoperability in a heterogeneous network. This is the problem space facing the design of a cross layer approach. Unfortunately, the concept of a comprehensive cross-layer approach still largely exists only within the network research domain, and is considered to only be viable in the long-term given sufficient research and development efforts.

Most of the protocols proposed only provide QoS in terms of specific metrics, such as bandwidth, delay, or reliability. However, it may be necessary to develop mechanisms to support QoS in terms of multiple metrics. For instance, when searching for QoS paths that have the required bandwidth, it is desirable to find reliable paths. Given the faulty nature of MANETs, constructing a QoS route that meets the bandwidth requirements while also meeting certain reliability requirements would result in better performance. Another example, for some video applications, it not only requires small delay but also enough bandwidth to transmit the images and videos.

Other approaches to satisfy the end-to-end QoS requirements include adaptive waveform design (power, modulation, coding, interleaving) in order to attempt to maintain consistent link performance across a range of channel conditions and adaptive

MAC techniques that monitor traffic conditions and modify MAC parameters to increase throughput, adaptive network layer techniques, and adaptive host-based transport and application layer techniques.

In this simulation, only small size network which contains 50 nodes is taken into consideration. Future work of this simulation may include the QoS performance of SWAN model in medium to large size networks. And also other traffic patterns with different real-time application requirements can be tested in SWAN network. Other simulations can be done in SWAN network with different routing protocols and network dynamics such as varying network topology, different nodes moving pattern, etc.

Appendix:

I: Trace file formats

Trace file is one of the text based results that the user gets from a simulation. It records the actions and relevant information of every discrete event in the simulation. There are a variety of forms for trace files. Simulations using different simulation networks or using different routing protocols could get trace files having different trace file formats. In the trace file, actions of different layers in the network can be traced. It includes agent trace, router trace, MAC trace and movement trace. All of these traced events can be written to a file in a predefined format. When the user simulates large events, the trace file can be very large. It will not only require time to generate the trace file during simulation but also need space to store it. As a result, user should always choose part of the choices to trace.

An example of one record in the wireless trace file is listed as follows:

r 5.000000000 _3_ RTR --- 2 cbr 512 [0 0 0 0] ------- [3:1 4:0 32 0] [0] 0 0

The first field can be r, s, f and D for received, send, forward and drop. The second field gives occurring times for the event. The third field is the node number. The fourth field is the trace name that can be AGT, RTR, MAC, and IFQ. AGT, RTR and MAC represent transport, routing and MAC layer separately. IFQ indicate events related to the interface priority queue. The number after the dashes is a globally unique sequence number of a packet. The letters after the number give the traffic type. Traffic types can be CBR (Constant Bit Rate), TCP (Transport Control Protocol) and ACK. The number right after the packet type is the packet size in bytes. The following two square brackets separated by the dashes are MAC and routing layer information such as source and destination addresses. With the information recorded in each event, performance metrics such as end-to-end delay, throughput, packet loss can be calculated with the help of some additional programs, e.g. Gawk, Perl, GNUPLOT and Tracegraph.

References:

[1]. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," IETF RFC 2501, 1999.

[2]. S. Chakrabarti and A. Mishra, "QoS Issues in Ad Hoc Wireless Networks," Communications Magazine, IEEE, Volume 39, Issue 2, Feb. 2001, Page(s):142 – 148.

[3]. P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," Multi Topic Conference, IEEE INMIC 2001, Dec. 2001, Page(s):62 – 68.

[4]. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," ACM press, Volume 24, Issue 4, Oct. 1994, Page(s). 234 – 244.

[5]. Z. J. Haas and M. R. Pearlman, "ZRP: a hybrid framework for routing in Ad Hoc networks," Ad Hoc Networking, 2001, Page(s): 221 – 253.

[6]. R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture- an Overview," IETF RFC1663, Jun. 1994.

[7]. L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A New Resource ReSerVation Protocol," Volume 7, Issue 5, Sep. 1993, Page(s):8 – 18.

[8]. S. Blake, "An Architecture for Differentiated Services," IETF RFC2475, Dec. 1998.

[9]. X. Hannan, W.K.G. Seah, A. Lo and K.C. Chua, "A Flexible Quality of Service Model for Mobile Ad Hoc Networks," Vehicular Technology Conference Proceedings, IEEE 51st, Volume 1, 15-18 May 2000, Page(s):445 – 449.

[10]. S-B. Lee and A.T. Campbell, "INSIGNIA: In-band Signaling Support for QoS in Mobile Ad Hoc Networks," Proceedings of 5th Intl. Workshop on Mobile Multimedia Communication MoMuc 98, Berlin, Oct. 12-14 1998.

[11]. A. Gahng-Seop, T. Campbell, V. Andras and S. Li-Hsiang, "Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)," Mobile Computing, IEEE Transactions, Volume 1, Issue 3, July-Sept. 2002, Page(s):192 – 207.

[12]. V.S.Y. To, B. Bensaou, and S.M.K. Chau, "Quality of Service Framework in MANETs Using Differentiated Services," Vehicular Technology Conference, IEEE 58th, Volume 5, Issue 6-9, Oct. 2003, Page(s): 3463 - 3467.

[13]. H. Chia-Hao, K. Yu-Liang; W. E.H.-K, C. Gen-Huey, "QoS Routing in Mobile Ad

Hoc Networks Based on the Enhanced Distributed Coordination Function," Vehicular Technology Conference, VTC2004-Fall, IEEE 60th, Volume: 4, Sep. 2004, Page(s): 2663- 2667.

[14]. Y. Seung, N. Prasad and K. Robin, "Security-Aware Ad hoc Routing for Wireless Networks," ACM Symp, on Mobile Ad Hoc Networking and Computing, 2001.

[15]. C.R. Lin, "QoS Routing in Ad Hoc Wireless Networks," Local Computer Networks, 1998, Proceedings, 23rd Annual Conference, 11-14 Oct. 1998, Page(s):31 – 40.

[16]. S. Chakrabarti and A. Mishra, "Quality of Service Challenges for Wireless Mobile Ad Hoc Networks," Wiley J. Wireless Commun. and Mobile Comput., Volume 4, Mar. 2004, Page(s): 129-153.

[17]. Z. Chenxi, M.S. Corson, "QoS Routing for Mobile Ad Hoc Networks," INFOCOM 2002, 21st Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings. IEEE, Volume: 2, Page(s): 958- 967.

[18]. R. Renesse, P. Khengar, V. Friderikos and H. Aghvami, "QoS Conflict Resolution in Ad Hoc Networks," Communications, IEEE International Conference, Volume 8, Jun. 2006, Page(s):3826 – 3831.

[19]. T. B. Reddy, I. Karthigeyan, B. Manoj, and C. S. R. Murthy, "Quality of Service Provisioning in Ad Hoc Wireless Networks: a Survey of Issues and Solutions," Ad Hoc Networks, Volume 4, Issue 1, Jan. 2006, Page(s): 83-124.

[20]. S. Chen and K. Nahrstedt, "Distributed Quality-of-service Routing In Ad Hoc Networks", IEEE JSAC, Volume 17, Issue 8, Aug. 1999, Page(s) 1488–1505.

[21]. C.R. Lin and J. Liu, "QoS Routing in Ad Hoc Wireless Networks," IEEE Journal on Selected Areas in Communications, Volume 17, Issue 8, Aug. 1999, Page(s): 1426–1438.

[22]. C.R. Lin, "On-demand QoS Routing in Multihop Mobile Networks," in Proceedings of IEEE INFOCOM 2001, Volume 3, Apr. 2001, Page(s): 1735–1744.

[23]. L. Wen-Hwa, W. Shu-Ling and S. Jiang-Ping, "A Multi-path QoS Routing Protocol in a Wireless Mobile Ad Hoc Network," Lecture Notes In Computer Science, Proceedings of the First International Conference on Networking-Part 2, Volume 2094, Page(s): 158 – 167.

[24]. B. Zhang and H. T. Mouftah, "QoS Routing for Wireless Ad Hoc Networks: Problems, Algorithms and Protocols," Communications Magazine, IEEE, Volume 43, Issue 10, Oct. 2005, Page(s):110 - 117

[25]. S. H. Shah and K. Nahrstedt, "Predictive Location-based QoS Routing in Mobile Ad Hoc Networks," Volume 2, 28 April-2 May 2002, Page(s):1022 – 1027.

[26]. P. Sinha, R. Sivakumar and V. Bharghavan, "CEDAR: A Core-extraction Distributed Ad Hoc Routing Algorithm," IEEE JSAC, Volume 17, Issue 8, Aug. 1999, Page(s): 1454–1465.

[27]. W. Min and K. Geng-Sheng, "An application-aware QoS Routing Scheme with Improved Stability for Multimedia Applications in Mobile Ad Hoc Networks," Vehicular Technology Conference, IEEE 62nd, Volume 3, Issue 25-28, Sep. 2005 Page(s): 1901 – 1905.

[28]. Network Simulator - NS – 2, Available at http://www.isi.edu/nsnam/ns/.

BL-11-58