PONGALIUR M.S. 2005

K. M. PONGALUR M.S. 2005


K. M. PONGALIUR
M.S.
2005

This is to certify that the
thesis entitled

ADAPTIVE SCTP FOR IMPROVED PERFORMANCE IN
MOBILE AD-HOC NETWORKS

presented by

Kanthakumar Mylsamy Pongaliur

has been accepted towards fulfillment
of the requirements for the

M.S. degree in Computer Science and Engineering

Lionel D. Ni

Major Professor's Signature

8-09-05

Date

ADAPTIVE SCTP FOR IMPROVED PERFORMANCE IN MOBILE AD-HOC NETWORKS

By

Kanthakumar Mylsamy Pongaliur

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science

2005

ABSTRACT

ADAPTIVE SCTP FOR IMPROVED PERFORMANCE IN
MOBILE AD-HOC NETWORKS

By

Kanthakumar Mylsamy Pongaliur

This thesis studies the comparative performances of the transport protocols using the different routing strategies, to identify the bottleneck parameters in SCTP to be optimized for improved performance. The state of the path is modeled using transport layer parameters. To achieve this, an adaptive algorithm is designed, which observes the changes in the path condition. The preliminary evaluation of these protocols through analysis and simulations using ns-2 showed that the average delay for SCTP is high compared to TCP. The efficiency of packet delivery is excellent with regards to the parameters- packet delivery ratio, routing overhead and goodput. Once the path condition is determined to be degrading, we identify a backup path to be switched to. This switching of the primary path due to path quality degradation has resulted in the reduction of average delay. Consequently there is increased bandwidth utilization resulting in a significant increase in the goodput. The kinds of applications that benefit from increased bandwidth utilization are bandwidth-constrained multimedia transport over MANET. Another class of application benefiting from reduction in average delay is real-time applications.

I dedicate this thesis to my parents...

ACKNOWLEDGEMENTS

This thesis dissertation would not have completed, if not for the guidance and perseverance of Dr. Herman Hughes, who not only encouraged me but also challenged me through out my academic program and was never satisfied with the second best.

I would like to thank Dr. Li Xiao for the critical assessment of the work throughout.

I would also thank Dr. Charles Owen for accepting to be on the committee.

Last but not the least I would like to thank my wife for providing me with enduring support during my good as well as difficult times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abbreviations

AODV	Ad-Hoc On Demand Distance Vector
DSDV	Destination Sequenced Distance vector
DSR	Dynamic Source Routing
DUP-TSN	Number of duplicate chunks received
ECN	Explicit Congestion Notification
GAP-ACK	Number of Gap Ack blocks in SACK
MANET	Mobile Ad-Hoc Network
Ns2	Network Simulator 2
RTO	Retransmission Time Out
RTT	Round Trip Time
SACK	Selective Acknowledgement
SCTP	Stream Control Transmission Protocol
SSN	Stream Sequence Number
TCB	Transmission Control Block
TCP	Transmission Control Protocol
TSN	Transmission Sequence Number
UDP	User Datagram Protocol

Chapter 1: Introduction

Wireless networks have been in the forefront of research and development for the past decade. They can be broadly classified into infrastructured networks and infrastructure-less networks. In infrastructured wireless networks, there is a basic need for infrastructure to be present (example- base stations or access points) for the mere existence of the network. The infrastructure-less networks do not place any such demand and mobile ad-hoc networks (MANET) fall in this category. MANETs have been the focus of many research applications ranging from military operations to rescue missions. The fundamental issues involved in mobile ad-hoc networks are that the wireless link interfaces have unique routing interface characteristics and the node topologies within a wireless routing region may experience increased dynamics, due to motion [1]. As a result, the physical layout of mobile ad-hoc networks changes continuously. Hence, the routing algorithm forms a critical part of the ad-hoc network. There are various routing protocols like Dynamic Source Routing (DSR) [2], Destination Sequenced Distance vector (DSDV) [3] and Ad hoc On Demand Distance Vector (AODV) [4] routing. The common features of these protocols are that they are lightweight and provide loop-free operations and responsive routing information. As mentioned earlier, ad-hoc networks are prone to link failures due to the movement of nodes. One of the basic problems with the existing transport layer protocol is its inability to distinguish between the link-failures due to the movement of the mobile nodes and congestion in the network. Consequently, the throughput degrades when the nodes move [5]. Several studies [5][6][7][16] have focused on studying TCP and UDP performance in MANET and, as a result, proposed some modifications to these existing protocols. These studies have been largely done on how the MANET performance is affected by the different routing protocols, while assuming the de facto transport layer protocol; namely, transmission control protocol. Since TCP was primarily designed for wired networks, it suffers performance degradation in the mobile ad hoc scenario. Another protocol, which is a new entrant into the family of transport protocols, is the stream control transmission protocol (SCTP). Its performance in ad-hoc networks has not been studied as exhaustively as TCP, but results from the investigations [8][13][14][15] have been promising. In this thesis we focus on comparing transport protocol performance, while running routing strategies like AODV and DSDV over MANET.

The selection of the routing protocols is made such that AODV is a reactive protocol, while DSDV, a proactive protocol. From our preliminary study we identify the sphere of SCTP, which can be improved for enhanced performance in MANET. We identified the delay in the packet delivery to be high in the case of SCTP, compared to TCP. We had comparable performance in case of data packet delivery percentage. The routing overhead for SCTP is comparatively lower than that of TCP. As a result, we have higher goodput in the case of SCTP. To reduce the delay of packet delivery, we do a switch of activity to a hot-standby path (called co-primary). It has been observed that, by switching we can reduce the average delay by 25% resulting in an increased goodput of 20%. Further, in this chapter we will discuss the various routing and transport protocols used while, doing a detailed summary of SCTP in chapter 2.

1.1 Routing and Transport Protocols

1.1.1 Routing Protocols

In this section we briefly discuss the various routing protocols under consideration for our work.

1.1.1.1 Destination Sequenced Distance Vector (DSDV):

DSDV [3] is a table driven proactive protocol. In this protocol, each node maintains the distance and the next hop information to each destination node. Also maintained is a sequence number for each route table entry, originated by the destination node. The routing information is transmitted by broadcast and the updates are transmitted periodically or immediately when a significant topology change is available. On receipt of new routing information, the route with a more recent sequence number is used or, if it has the same sequence number, then the route with a better metric is chosen. Each node periodically broadcasts its routing information. A broken link can be identified if we don't receive a broadcast from the node for a certain period of time. On such identification, all nodes reached through that link are assigned an infinite metric value. It gives good performance when the node mobility is low.

1.1.1.2 Ad-Hoc On Demand Distance Vector routing (AODV):

AODV [4] is a hybrid of DSR and DSDV, borrowing the salient features of both. It provides a loop free route even while repairing the routes. When a route is to be discovered, AODV uses a discovery mechanism similar to DSR, but instead of source routing, AODV relies on dynamically establishing route table entries at intermediate nodes. To maintain the routes it uses a concept similar to DSDV, but each node maintains a monotonically increasing sequence number counter, which is used to supersede stale cached routes. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out. It is eventually deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate a route discovery.

1.1.2 Transport Protocols

1.1.2.1 User Datagram Protocol (UDP):

UDP [9] provides just the bare minimum functionalities that are to be provided by the transport layer. There is no connection setup feature like handshaking and hence it is a connectionless protocol. Another characteristic is the lack of a congestion control mechanism. It gives good performance for real-time communication, but does not guarantee QoS.

1.1.2.2 Transmission Control Protocol (TCP):

TCP [10] is the de-facto protocol for wired networks and uses a 3-way handshake to set up a connection. TCP is connection oriented, full duplex, and end-to-end communication protocol. Once a connection is established, the two application processes can send data to each other. It supports a congestion control mechanism, error control, guaranteed QoS and delivery of the data packet unlike UDP. Another important feature is that TCP is a byte streaming protocol.

1.1.2.3 Stream Control Transmission Protocol (SCTP):

SCTP [11] is a general-purpose transport protocol used for the transport of telecommunication signaling messages over an IP based network. The primary purpose of SCTP is to provide a reliable end-to-end message transportation service over IP-based networks. It performs this service within the context of an association between two SCTP endpoints. SCTP is connection-oriented in nature, but the SCTP association is a broader concept than the TCP connection. SCTP provides the means for each SCTP endpoint to provide the other endpoint (during association startup) with a list of transport addresses (i.e., multiple IP addresses in combination with an SCTP port) through which that endpoint can be reached and from which it will originate SCTP packets. The association spans transfers over all of the possible source/destination combinations, which may be generated from each endpoint's lists. Thus, the two new capabilities that are designed into SCTP are the support for multi-homed hosts and the support for multiple streams in a single SCTP association. The SCTP data transportation service is message-oriented as compared to the byte oriented data transfer of TCP. The multi-homing feature allows multiple source and/or destination addresses in one SCTP connection ("association" is used in SCTP terminology). When one interface/address fails, the traffic can be automatically transferred to another interface without interrupting the ongoing association.

The thesis is organized as follows. We have a detailed description of SCTP in chapter 2. This is followed by the problem statement and preliminary investigation in chapter 3. Related work also forms a part of chapter 3. Chapter 4 details the adaptive algorithm. Chapter 5 has the results followed by conclusion in chapter 6. Then we have the appendices containing the graphs for the preliminary investigation and the references follow it.

Chapter 2: Stream Control Transmission Protocol

In this chapter[1] we discuss the core concepts of SCTP from association establishment through transmission of data to association termination. SCTP was developed keeping in mind the drawbacks of TCP, but still it has been designed for wired networks. Hence it too has certain drawbacks, which we will be demonstrating through simulations. We then propose solutions to overcome these drawbacks.

2.1 SCTP Overview

SCTP is a transport layer protocol that uses a four-way handshake process to set up an association with another node. Another feature of SCTP is its ability to support multiple IP addresses and this gives rise to the possibility of load-sharing, seamless mobility support in MANET. Since the nodes can support multiple IP addresses, there could be multiple sessions with an association, which can live simultaneously. In the current SCTP, at a given time the two nodes will be communicating over just one pair of addresses. This connection is called the primary

---

[1] Parts of this chapter are adapted from rfc2960.

connection. To check the connection status among the other addresses currently not in use, we have a feature called heartbeat in SCTP. The PING utility in UNIX can be an analogy to the heartbeat. Unlike TCP there can be no half open connections in SCTP.

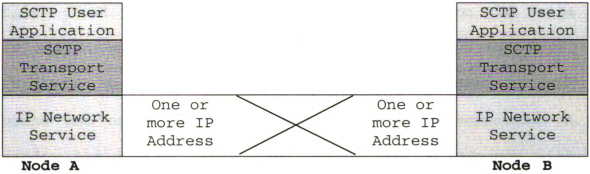

Figure 2.1: SCTP in protocol stack

Figure 2.1 depicts the overall position of the SCTP in the SCTP-IP stack and also gives a pictorial understanding of how more than one IP addresses per node can be associated in the formation of a connection (association in SCTP).

2.2 SCTP Association Setup

The association setup process in SCTP is a 4-way handshake process. This is mainly done to prevent the SYN attack that is possible in TCP. The security from SYN attacks is achieved by using a cookie mechanism in the establishment phase. Let us consider two nodes (node A and node B), where node A (from now on referenced just as A) wants to establish an association with node B (called B). A sends an

INIT chunk (message) to B, with its initiate tag set to a particular value. This tag parameter is used to identify the association. B responds with an INIT-ACK chunk with the initiate tag parameter of A copied into the verification tag field of INIT-ACK chunk. Since the INIT chunk is the first chunk in the association, the verification tag value is set to zero. Another important function of the INIT, INIT-ACK chunk exchange is to negotiate the number of out-bound and number of in-bound streams to be supported. A send's its request using INIT chunk, and B responds with INIT-ACK chunk containing the number of in-bound and out-bound streams acceptable. The smaller of the two sets of request is accepted. The INIT and INIT-ACK chunks are used for the exchange of the IP addresses at which each of the nodes can be reached. The INIT-ACK chunk includes a state cookie. The state cookie is used to prevent the SYN-attack.

The third stage in the association establishment is the sending of COOKIE ECHO at the receipt of INIT-ACK by A. Node A copies the state cookie from the INIT-ACK as it is, into the COOKIE ECHO chunk and sends the COOKIE ECHO to B. The entire operation is shown in figure 2.2

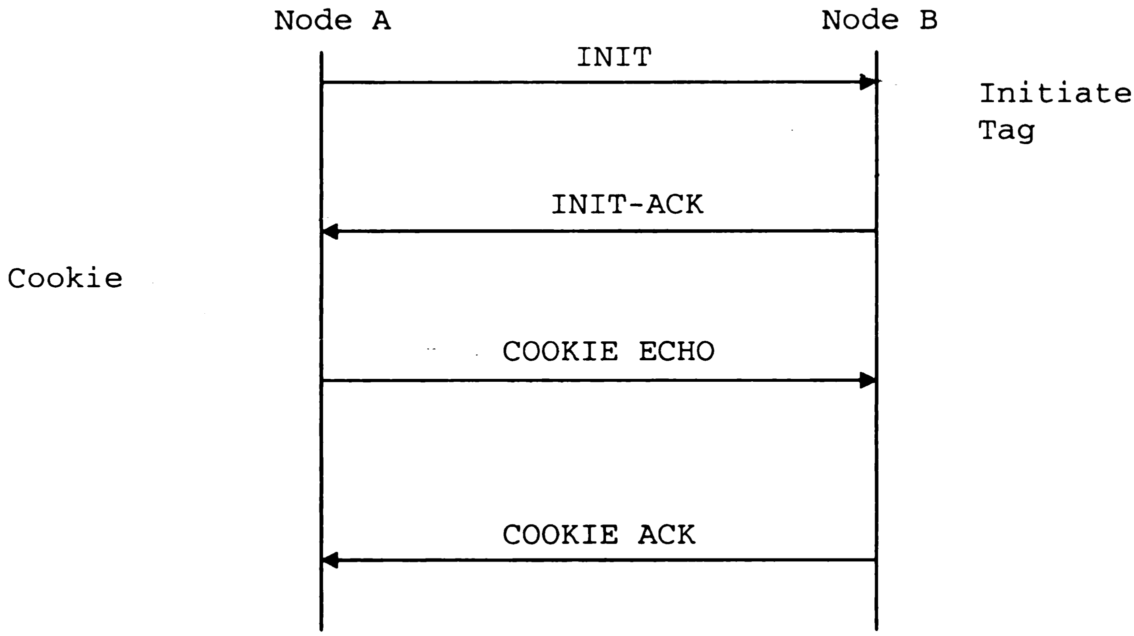

Figure 2.2: Establishment Phase of SCTP

On the receipt of the COOKIE ECHO, B will create the TCB, and send back the COOKIE ACK chunk. Another important feature of the establishment phase is that in the last two steps, data can be exchanged, and hence the wait period before the actual transmit of data is one RTT, rather than two RTT's.

2.3 Data Transfer

Once the association is established, the address that was used to set up the association is marked the primary address (primary connection) and the transfer of data takes place on this connection. The transfer of data is facilitated by two kinds of chunks, namely the data chunks and SACK chunks. As the name specifies the data chunks are used to transfer data while SACK is used to acknowledge the data received. Another characteristic of SCTP is its use of two types of sequence numbers, namely TSN (transmission sequence number) and SSN (stream sequence number). Data chunks are identified by the TSN, which is unique for the entire association. The SSN is used within a stream for sequenced delivery. The stream identifier identifies each individual stream within the association.

Example: [11]

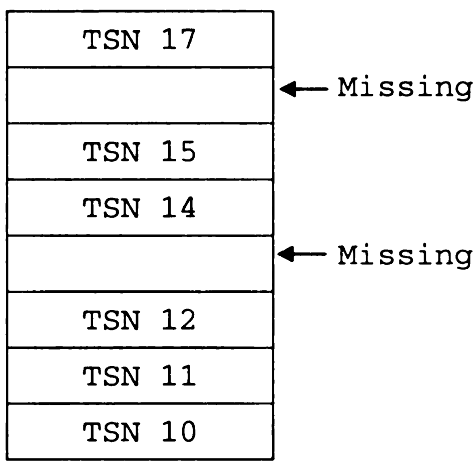

Figure 2.3: Data chunks received, before a SACK

SACK is a cumulative acknowledgement scheme, which acknowledges the receipt of all the data chunks up to the last received TSN. SACK contains the Gap Ack blocks (representing the chunks received after a gap in TSN) and the duplicate TSN blocks.

Suppose that the chunks represented in figure 2.3 are the data chunks received at the time of sending the SACK chunk. Chunks 10, 11, 12, 14, 15, 17 are received while 13, 16 are missing. The representation of the SACK chunk will be as shown in figure 2.4.

Cumulative TSN ACK = 12	
Num of Block = 2	Num of Dup = 0
Block #1 strt = 2	Block #1 end = 3
Block #2 strt = 5	Block #2 end = 5

Figure 2.4: Partial SACK chunk showing GAP-ACK blocks

The duplicate TSN (dup TSN) are the chunks, which were received in duplicate due to retransmission.

The computation and management of RTO in SCTP follows closely the way TCP manages its retransmission timer. To compute the current RTO, an endpoint maintains two state variables per destination transport address: SRTT (smoothed round-trip time) and RTTVAR (round-trip time variation). If the computed RTO is less than RTO.Min seconds then it is rounded up to RTO.Min seconds. The reason for this rule is because, RTO's that do not have a high minimum value are susceptible to unnecessary timeouts.

2.4 Heartbeat Mechanism

Heartbeat is the mechanism by which the nodes keep track of the status of the path between addresses that are not in use. The address that is used for the data transfer is called the primary address (connection), the rest are backup address pairs (connection). The backup connection is used if the primary address fails. To use the backup connection, one should know the state of the path as it is rarely used. This is where the path heartbeat mechanism comes in handy. The heartbeat mechanism is used to probe the destination transport address defined in the current association. The sender node sends the HEARTBEAT chunk and the receiver replies with a HEARBEAT ACK. A HEARTBEAT ACK is always sent to the source IP address of the IP datagram containing the HEARTBEAT chunk to which the destination node is responding.

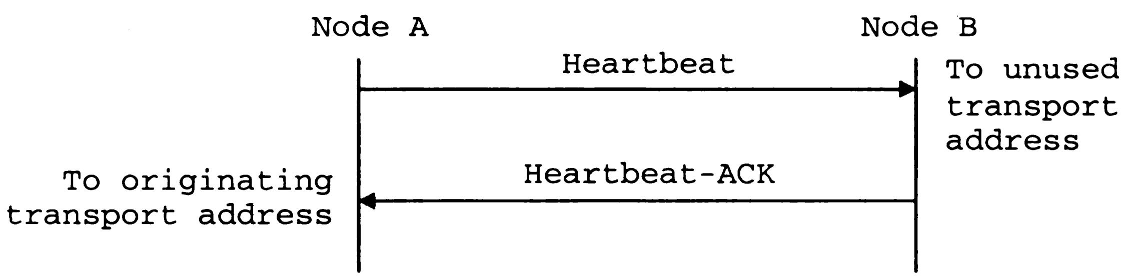

Figure 2.5: Heartbeat mechanism in SCTP

The heartbeat chunk is largely undefined, hence it can be used for different purposes like gathering statistics, etc.

2.5 Association Termination

SCTP supports two types of association termination, the graceful termination called SHUTDOWN and the abrupt termination called ABORT. The ABORT is called under different types of error circumstances. The abort chunk may contain the cause parameters to inform the receiver the reason for abort. In case of abort, the node puts the verification tag in the outbound abort packet. It deletes its TCB and goes into closed state. If the abort packet is lost, then it takes a long time for the other peer to realize that the association has been terminated. Data chunks may not be bundled with abort, while it may contain control chunks.

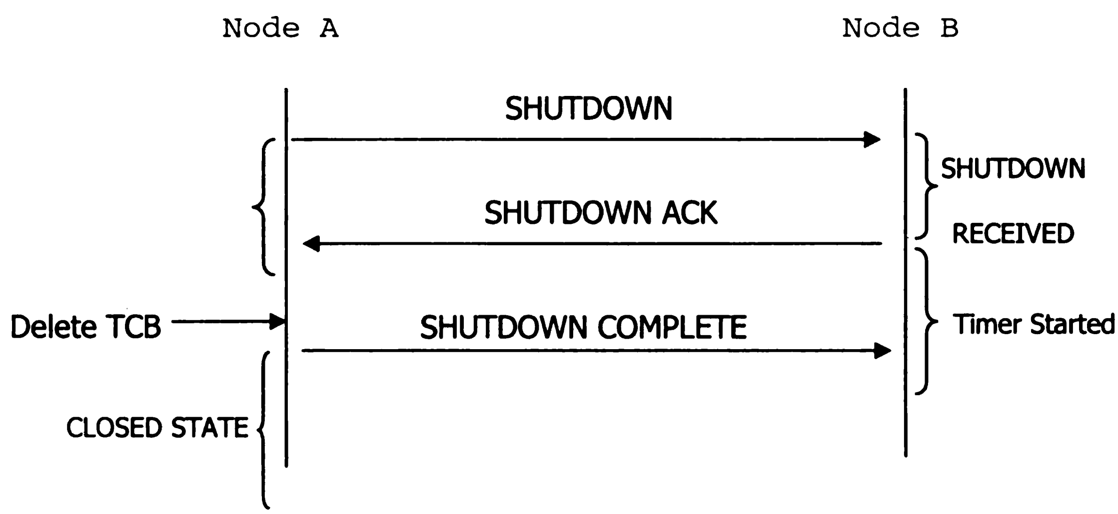

Figure 2.6: Graceful termination of association

Graceful shutdown is depicted in figure 2.6. A sends a SHUTDOWN chunk and waits for a SHUTDOWN ACK. On receiving the SHUTDOWN ACK, it deletes the TCB, and sends a SHUTDOWN complete and goes into the closed state. On receipt of SHUTDOWN COMPLETE by the B, it also goes into the closed state. Suppose the SHUTDOWN COMPLETE gets lost, then B sends the SHUTDOWN ACK a few times before it marks A unreachable.

2.6 SCTP differences from TCP congestion control

GAP-ACK Blocks in the SCTP SACK carry the same semantic meaning as the TCP SACK. TCP considers the information carried in the SACK as advisory information only. Similarly SCTP considers the information carried in the GAP-ACK Blocks in the SACK chunk as advisory. In SCTP, any DATA chunk that has been acknowledged by SACK, including DATA that arrived at the receiving end out of order, are not considered fully delivered until the Cumulative TSN Ack Point passes the TSN of the DATA chunk (i.e., the DATA chunk has been acknowledged by the Cumulative TSN Ack field in the SACK). Consequently, the value of cwnd (congestion window) controls the amount of outstanding data, rather than (as in the case of non-SACK TCP) the upper bound between the highest acknowledged sequence number and the latest DATA chunk that can be sent within the congestion window. SCTP SACK leads to different implementations of fast-retransmit and fast-recovery than non-SACK TCP.

The biggest difference between SCTP and TCP, however, is multi-homing. SCTP is designed to establish robust communication associations between two endpoints, each of which may be reachable by more than one transport addresses. Potentially different addresses may lead to different data paths between the two endpoints, thus ideally one may need a separate set of congestion control parameters for each of the paths. TCP guarantees in-sequence delivery of data to its upper-layer protocol within a single TCP session. This means that when TCP notices a gap in the received sequence number, it waits until the gap is filled before delivering the data that was received with sequence numbers higher than that of the missing data. On the other hand, SCTP can deliver data to its upper-layer protocol even if there is a gap in TSN, provided the Stream Sequence Numbers are in sequence for a particular stream (i.e., the missing DATA chunks are for a different stream) or if unordered delivery is indicated.

Chapter 3: Problem Statement and Preliminary Investigation

3.1 Problem statement

The performance of SCTP is at par with TCP except for the delay it incurs in delivering the packet [14]. This is verified in the preliminary simulations. The delay in the delivery of packets is almost double of that in TCP. The reduction in delay of packet delivery will result in better utilization of the network's bandwidth. The bandwidth utilization is directly proportional to goodput. Hence, an increase in bandwidth utilization will result in increase of goodput. We also want to model the network path condition based on the transport layer statistics. This will enable us to take decisions at a higher level in the protocol stack. In this thesis, we are going to look at modeling the network path based on transport layer parameters and reduce the average delay. The motivational applications are the family of applications which use bandwidth constrained MANETS to transport large quantities of data such as multimedia data. We can also extend the advantages to cater to real-time applications. Hence by reducing average delay in packet delivery, we are able to cater to real-time applications and by increasing the bandwidth utilization, we should be able to cater to bandwidth constrained multimedia applications.

3.2 Related Work

Though not much work is done to adapt SCTP for wireless/ mobile networks, a lot of research has gone into adapting TCP for similar networks. Since SCTP borrows the congestion control mechanism directly from TCP, most of the optimizations for TCP congestion control can be applied to SCTP as well. Further on, we discuss some of the related work that has been done to improve the performance of SCTP in wireless/mobile environment.

In [8], A. Argyriou et al. do a performance evaluation between SCTP and TCP over DSR and AODV. The authors conclude the overall performance of SCTP is far better than TCP. In the paper they specify some modifications to SCTP, but these modifications are source routing (DSR) protocol specific. They modify DSR to give disjoint routes to SCTP, so that when SCTP intends to switch paths, it has the disjoint path. This is advantageous because SCTP maintains congestion control parameters separately for each individual path. This strategy will not work with the table driven routing strategies. The table-driven routing strategy may not know the underlying path. They also conclude that the transport layer allows faster path selection, in case a number of paths exist, leading to improved overall throughput.

In [13] G. ye et al. use the existing explicit congestion notification (ECN) as an indicator of congestion loss. It assumes that, if the loss is not accompanied by an ECN, the loss is caused by the wireless link dynamics (e.g. could be corruption of packet). It does not deal with the loss of the ECN messages, and this could lead to the sender believing that the packet loss is not due to congestion and will keep sending data packets at the current high rate. This can lead to drastic deterioration of the network. The authors don't take any advantage of the alternate paths available in SCTP.

DS-SCTP [14] represents a SCTP handover scheme, which selects the path, based on the moving average of end-to-end delay. The scheme considers multi-media traffic over wireless media but in a static environment. The authors assume the presence of a better path and handover the transmission to the second path; basically start using the alternate path. They don't suggest any method or strategy to take this decision. But they do corroborate our results that the delay in SCTP is one of the factors limiting the goodput.

In Collaborative-SCTP [15], the authors propose a cross-layer optimization of SCTP which is adaptive to variable bit errors of wireless channel.

3.3 Performance Metrics

The performance metrics used for the comparison of the protocols are the average routing overhead, packet delivery ratio (expressed as percentage), average delay (expressed in seconds) and goodput.

We define the average routing overhead as the ratio of the total number of routing bytes sent to the total number of data bytes sent, averaged over all the sources.

$$\boxed{\text{Routing Overhead} = \frac{\text{Routing Bytes Sent}}{\text{Data Bytes Sent}}}$$

The ratio of the total number of data packets received to the total number of data packets sent is termed as the useful packet delivery ratio.

$$\boxed{\text{Packet Delivery Ratio} = \frac{\text{Data Packets Received}}{\text{Data Packets Sent}}}$$

The average delay is measured as the difference between the time when the packet was received and the time when the agent sent the packet and averaged over all the packets.

$$\text{Average Delay} = \frac{\sum_{1}^{n}(\text{Receiving time} - \text{Sending time})}{n}$$

where n is the total number of useful packets received.

Goodput is measured as the total amount of data bytes delivered to the upper layer protocol (excluding the retransmissions) per unit time.

$$\text{Goodput} = \frac{\text{Total Data Delivered}}{\text{Connection Time}}$$

3.4 Preliminary Investigations

3.4.1 Simulation Setup

We present a preliminary investigation, which strengthens our stand on high average delay in SCTP. In this investigation, we performed simulation runs in ns2 along with the wireless extensions provided by the MONARCH project at the Carnegie Mellon University, to study the performance of three transport layer protocols. ns-2 is developed by the Lawrence Berkley National Laboratory (LNBL) [12].

At the physical layer, we used a radio propagation (Two-ray Ground Reflection) model, with omni-directional antennas and a shared media interface. The IEEE 802.11 medium access protocol was used as the link layer protocol. The random waypoint model was used to generate the movement of the nodes. 50 mobile nodes move in a flat rectangular topology of size 1500m x 1500m, with each node having a transmission range of 250m. The bandwidth of each link was 2 Mbps and the total simulation time was 200 seconds. The number of connections was restricted to a maximum of 25 over the 200 seconds simulation period.

The mobility of a node is characterized by two parameters, the mean speed and the pause time. At the start of the simulation, each node waits for a pause time. It then starts moving towards a randomly selected destination with the mean speed. Once the selected destination is reached, the node remains stationary for a pause time before moving again towards a new destination. This process is repeated until the end of simulation is reached. The simulations were carried out for mean speeds of 1m/s and 25m/s and pause times of 0sec, 50sec, 100sec and 200sec. This resulted in eight different sets of scenarios and 10 simulations were carried out for each scenario. The results obtained were averaged over the 10 simulation runs. A pause time of 200sec, which is equal to the total simulation time, implies that there is no movement of nodes. On the other hand, when the pause time is 0 seconds, the nodes are continuously moving. Thus, the varying factors in the preliminary simulation studies are the transport protocol, the routing protocol, the mean speed and the pause time.

3.4.2 Discussion of preliminary results

The figures referred to, in this analysis are produced in the appendix-B at the end of the report. We see the average delay associated with the packets of TCP to be around 0.4 seconds, while that of SCTP and UDP to be around 0.8 seconds. This is depicted in figures B1.1, B1.2, B1.3 and B1.4 in appendix B. Another observation is that with an increase in the mean speed the average delay for UDP on DSDV is considerably reduced to be around 0.4 seconds. This is due to the fact that the useful packet delivery of UDP reduces drastically with the increase in the mean speed and here we are calculating the average delay on only those packets that were delivered successfully. The average delay is seen to be consistently the same irrespective of the speed of the node movements. The variation in the overall packet delivery ratio is shown in figures B2.1, B2.2, B2.3 and B2.4 in appendix B. It is clear from the figures that the packet delivery ratio of TCP and SCTP (in high 90's) is far better than that of UDP (hovers between 35 and 50). One of the primary factors responsible for such high packet delivery percentage of TCP and SCTP is the connection oriented-ness of two the protocols. A connection is established between the two end-nodes before the packet is sent, whereas in UDP, the packet is transmitted and it is hoped that the destination receives it. In figure B2.4, we see the packet delivery percentage of UDP increases monotonically with the increase in the pause times. In these figures, we observe the high data delivery percentage of SCTP, which is just a couple of percentages lower than that of TCP. Even though SCTP delivers a slightly smaller percentage of its packets, it is able to deliver higher numbers of them, which is shown by the figures B4.1, B4.2, B4.3 and B4.4, i.e. the overall goodput of SCTP is higher than TCP. Figures B3.1, B3.2, B3.3 and B3.4, depict the routing overhead involved in sending the data bytes. SCTP outperforms TCP with regards to the overhead involved. This is because SCTP was designed as a lightweight protocol for PSTN signaling. On an average the amount of routing bytes required by TCP is 5 times the amount of routing bytes required by SCTP to deliver 1 byte of data. The amount of routing information required for UDP is less than TCP, but the number of packets delivered by UDP is far less compared to TCP.

From the above discussion it is seen that, the performance of SCTP, is at par with TCP except for the delay it incurs in delivery of packets, which is almost double of that of

TCP. We are going to look at methods to reduce the average delay, which is expected to increase the goodput of the network, i.e. increase the bandwidth utilization.

Chapter 4: Adaptive SCTP

The acknowledgement mechanism in SCTP is different from that of TCP and this could be one of the many reasons for delayed packet delivery. In SCTP, the acknowledgement mechanism is of a cumulative type and implying that we don't reply to every packet. By the time the cumulative acknowledgement is sent, the network dynamics may have changed, especially in a mobile network. Another reason for the delay in packet delivery is the retransmission time out (RTO) parameter, for which we don't have a satisfactory solution. The RTO mechanism is borrowed from TCP and currently these two mechanisms are not using the multihoming feature of SCTP to their advantage.

4.1. Statistical inference

We do a statistical analysis on the relation between the two parameters; namely, the GAP-ACK parameter and DUP-TSN. The correlation between the two parameters (GAP-ACK parameter and DUP-TSN) is seen to be +0.61. The correlation between the delay in the delivery of packets and the values of DUP-TSN is calculated (refer to table 4.1). Similarly, we calculated the correlation between DUP-TSN and the average delay. The results are presented in table 4.2.

		Mean Speed(m/sec) 1	25
Pause Time (Sec)	0	0.71	0.69
	50	0.79	0.73
	100	0.53	0.74
	200	0.76	0.74

Table 4.1: Correlation between # DUP-TSN and Delay

		Mean Speed(m/sec) 1	25
Pause Time (Sec)	0	0.53	0.57
	50	0.71	0.66
	100	0.58	0.65
	200	0.72	0.68

Table 4.2: Correlation between GAP-ACK and Delay

From tables 4.1 and 4.2, it can be seen that delay has a strong correlation with GAP-ACK as well as DUP-TSN. The observed delay values seem to be more strongly correlated with DUP-TSN compared to GAP-ACK. The corresponding graphs for the above tables are given in appendix A.

4.2. Modeling the state of connection using SACK parameters

The structure of SACK chunk has two parameters as discussed in section 4.1 (refer to figure 4.1). The GAP-ACK block specifies the chunks in the stream that have not yet reached the destination. The study of this parameter over the period of the connection, gives a very good approximation of the variation in path condition. Another parameter that is included in the SACK chunk is the DUP-TSN. DUP-TSN is the number of duplicate chunks received due to retransmission.

Type = 3	Chunk Flags	Chunk Length
Cumulative TSN Ack		
Advertised Receiver window Credit (a_rwnd)		
# of Gap Ack Blocks = N		Number of Duplicate TSN = X
Gap Ack Block #1 start		Gap Ack Block #1 end
....		
Gap Ack Block #N start		Gap Ack Block #N end
Duplicate TSN 1		
....		
Duplicate TSN X		

Figure 4.1: SACK Chunk

We study the path condition by analyzing the variation in these parameters over a period of time.

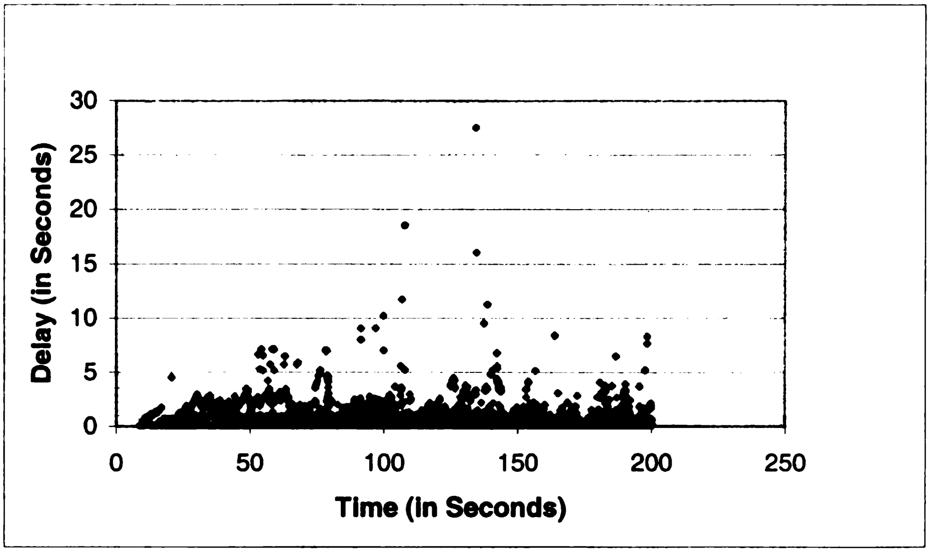

Figure 4.2a: Distribution of packets delay. AODV, Pause time=50Sec, Speed =1m/sec

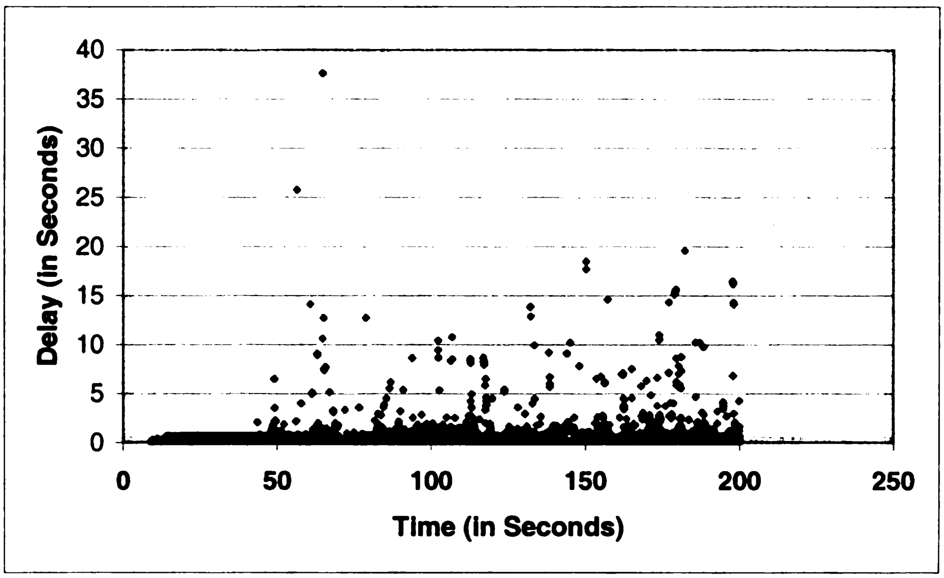

Figure 4.2b: Distribution of packets delay. AODV, Pause time=50sec, Speed =25m/sec

Figures 4.2a, and 4.2b represent the distribution of packet delays for AODV with pause time equal to 50 seconds and two different mean speeds (1m/sec and 25m/sec). It is evident from the figures that there is an increase in the number of delayed packets just after 54 seconds, as indicated on the x-axis. We averaged the delay in packet delivery by 0.5-second durations between 55 second and 60 second mark. The figures 4.3a and 4.3b represent the same. The average delay for this period is 0.9 seconds for mean speed 1m/sec and 0.6 seconds for mean speed 25m/sec. This variation can be the result of multiple reasons. It could be that, most of the packets never reached the destination, when mean speed was 25m/sec.

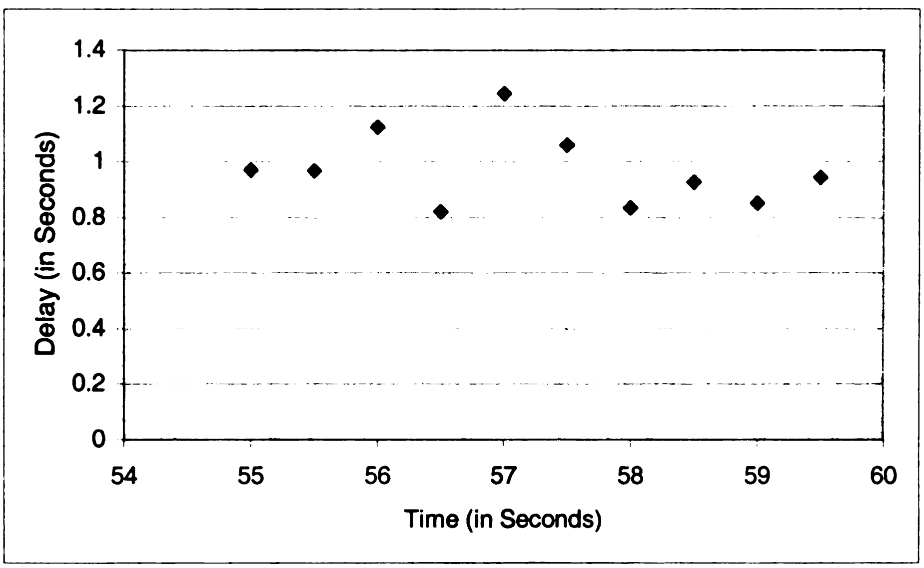

Figure 4.3a: AODV, averaged delay of packets. Speed 1m/sec, pause time 50 Sec.

Secondly, the faster mobile nodes reached their destination soon. Once they reached the destination, they were stationary for the next 50 seconds; hence they were able to receive more packets quicker compared to the other case.

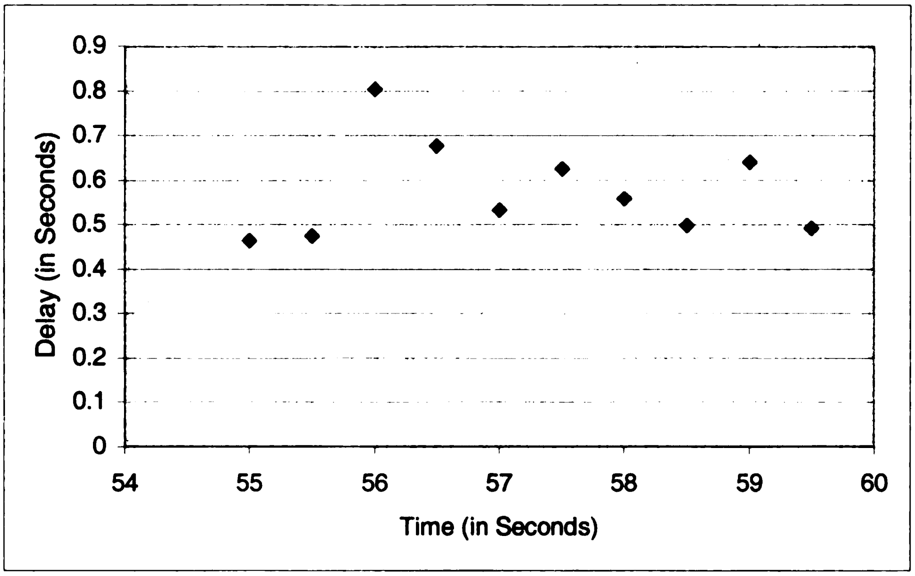

Figure 4.3b: AODV, Averaged Delay of packets. Speed 25m/sec, pause time 50 Sec.

Though we have depicted the graphs for pause time 50 seconds, we study the behavior for two pause times, namely, 50 and 100 seconds. With pause time 50 seconds, the nodes have just started moving when the time crossed 50 seconds and for pause time 100 seconds they are still stationary in our study interval. The mean speeds are 1m/sec and 25 m/sec. Figure 4.4 represents a graph with a positive slope[2] when the mean speed is 25m/sec. It indicates an increase in the number of GAP-ACK blocks. This implies that either the

[2] Slope refers to the corresponding regression line.

packets are getting lost in the network or they are getting delayed in the network. Similarly, when the mean speed of the node is slow (i.e. 1m.sec), the number of GAP-ACK blocks increases more rapidly. This is attributed to the fact that the nodes are still moving and have not reached the destination location yet, due to their slower speed. We get similar graphs with speeds 1m/sec and 25 m/sec with pause time of 100 seconds (study interval is 55 seconds to 60 seconds).

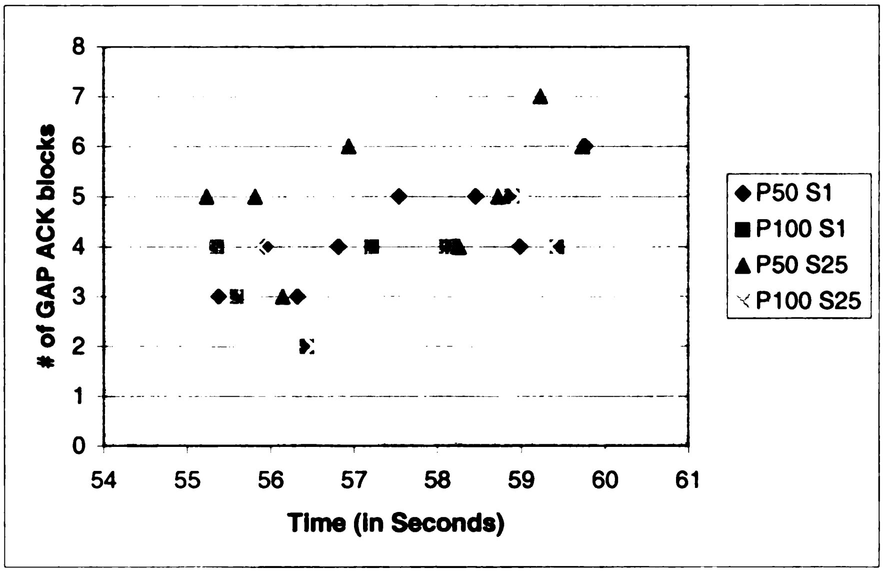

Figure 4.4 GAP-ACK distributions

To determine if the increase in GAP-ACK blocks is due to packet loss or packet delay we study the number of DUP-TSN in the SACK chunk. The interval of study is over the association between node 7 and node 23 for the time duration of 55 - 60 sec. Figure 4.5 represents the number of DUP-TSN in the SACK chunk. We see that the DUP TSN's are varying without a consistent slope, representing the variation in the network condition. The fall in number of DUP packets while the number of GAP-ACK blocks increases during that period of time, represents the loss of the original packets in the network. The region in the graph where the number of DUP chunks increase is because of delayed out of order delivery of the original packet. For pause time 100, the speed doesn't matter for our study. The graph of DUP-TSN for the case with speed 1m/sec is comparable with 25m/sec.

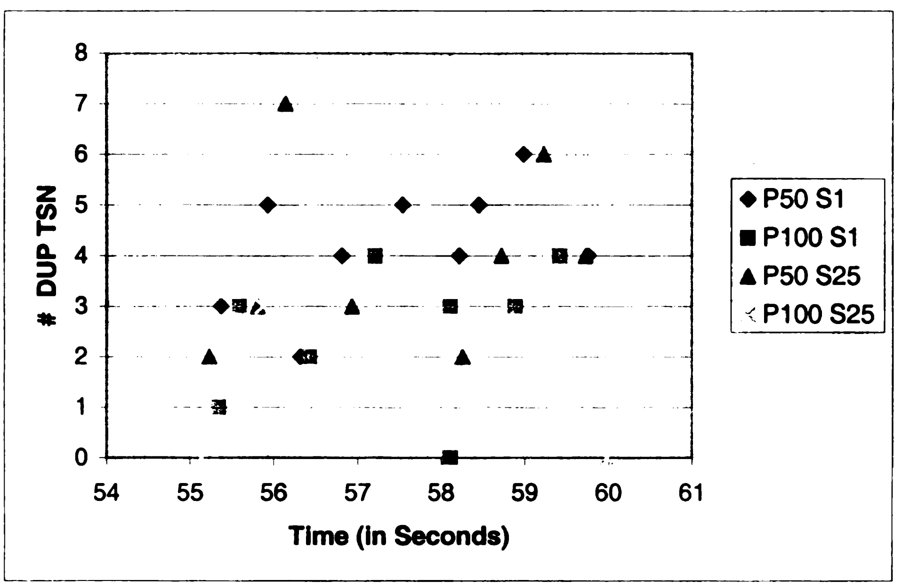

Figure 4.5 DUP-TSN distributions

This led us to the conclusion, that the variation in the DUP-TSN and the GAP-ACK parameters of the SACK chunk can give an appropriate status of the path condition.

4.3 Adaptive Algorithm

In SCTP we have multiple connections available within the association. But we use only one connection at a time. The addresses associated with this connection are called the primary address. This is called the primary connection in the discussion. We maintain four parameters, namely Network_Quality, O_Num_Gap, O_Num_Dup and SACK_COUNT. The O_Num_Gap Parameter keeps track of the number of GAP-ACK blocks in the last SACK chunk received, O_Num_Dup tracks the DUP-TSN in the previous SACK chunk and the SACK_COUNT is a variable which when crosses the threshold value of 3, indicates degradation in link condition. Consequently, we shift from the primary connection to the co-primary connection. The condition in which the SACK_COUNT reaches the value of 3 follows an algorithm depicted in figure 4.6. The algorithm is executed at the sender node on the receipt of a SACK chunk. The Num_Gap and Num_Dup represent the values from the currently received SACK Chunk.

```
If ((O_Num_Dup <= Num_Dup)
{
     if (O_Num_Gap <= Num_Gap)
     {
          if (Network_Quality == true)
               {
                    SACK_COUNT = SACK_COUNT + 1;
               }
          else { Network_Quality = true; }
     }
     else
     {
          if (Network_Quality == true)
          {
               SACK_COUNT = SACK_COUNT + 1;
               Network_Quality = false;
          }
         else { Network_Quality = true; }
     }
else
{
     if (O_Num_Gap <= Num_Gap)
     {
          if (Network_Quality == true)
          {
               SACK_COUNT = SACK_COUNT + 1;
               Network_Quality = false;
          }
          else { Network_Quality = true; }
     }
     else { SACK_COUNT = 0; Network_Quality = false}
}

If (SACK_COUNT == 3)
{
     Shift to Co-Primary
}
else
{
     O_Num_Dup = Num_Dup;
     O_Num_Gap = Num_Gap;
}
```

Figure 4.6: Adaptive Algorithm

We shift to a backup connection, when SACK_Count reaches the value of 3. The logic behind the algorithm is that if, for three consecutive cases, the GAP-ACK block and the DUP-TSN's are increasing, it shows that we are having congestion in the network and should use one of the backup connections. The value of threshold (3) is decided on the basis of results from simulation runs with varying threshold values. With the threshold set to 2, we noticed that we were shifting too soon; this occurred very frequently. Setting the threshold value to 4 caused in the primary path to fail. Having decided that we need to shift to the backup path, there is another algorithm, which gives us the co-primary path (also called hot-standby path). This decision is taken based on the following parameters, the roundtrip time (RTT), and the last hop information. If it were a source routing protocol like DSR is being used, we can use a disjoint path as the backup path [8]. If it is not a source routing protocol, we take into consideration the last hop node.

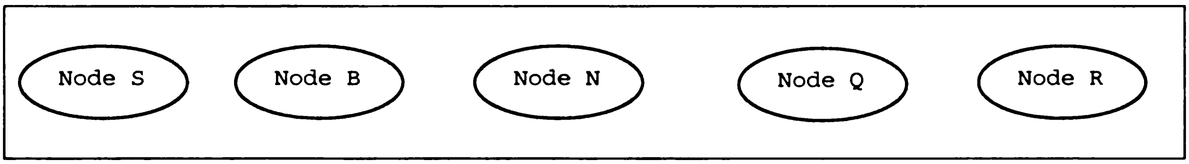

Figure 4.7: Last hop node depiction:

First we consider the RTT by selecting the path with the lowest roundtrip time and mark it t_a. We also consider all those paths, which have roundtrip time less than or equal to $(1.1)*t_a$. Among these paths we select the path with the lowest roundtrip time and have a disjoint last hop node with the primary path. The last hop node is node Q in figure 4.7, where node S is the sender and node R is the receiver. This is because, in a high percentage of times, it is the node movement of the receiver, which is responsible for the packets getting dropped. The paths in an association that share the last hop have some common intermediate nodes too. It is found that, most of the paths are not longer than 4-5 hops. Consequently, the consideration of the last hop information plays a crucial role in the selection of the co-primary path.

Once the co-primary path is selected as the primary path, we start a timer called the path_shift timer. This timer expires in time 1.1*RTT where RTT is the roundtrip time of the primary path. This setting of the initial value is strict in the sense that, it is set to a fixed value slightly above RTT. We require this because, if the path does not give us improvement with regards to the delay in packet delivery, then there is no use in changing the path. If we receive the acknowledgement before the expiration of the timer, we keep this co-primary path as the primary path, else we change back to the previous primary path.

Chapter 5: Results and Analysis

The simulation setup is as described in chapter 3. We use ns2 as the simulator. At the physical layer, we use a radio propagation (Two-ray Ground Reflection) model, with omni-directional antennas and a shared media interface. The IEEE 802.11 medium access protocol is used as the link layer protocol. The random waypoint model is used to generate the movement of the nodes. In our simulations, 50 mobile nodes move in a flat rectangular topology of size 1500m x 1500m, with each node having a transmission range of 250m. The bandwidth of each link was 2 Mbps and the total simulation time was 200 seconds.

The number of connections was restricted to a maximum of 25 over the 200 seconds simulation period. This restriction is introduced to have at least some connections which were long enough to be affected by the network dynamics. Without such a restriction, we would have had a large number of small connections, which may not have given us the true performance picture.

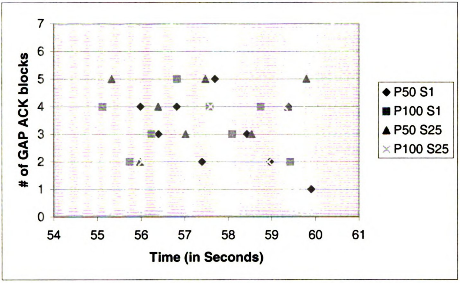

Figure 5.1: GAP-ACK distributions

We counted the number of GAP-ACK blocks and the DUP-TSN in the SACK Chunk, which is as shown in figure 5.1. It can be observed that the number of GAP-ACK blocks does not increase as much when compared to the plain vanilla SCTP. This could imply that either the packets are delivered in sequence or there is loss of packets, hence, no DUP-TSN. This will be clear after the analysis of the average delay of packets and the goodput measurement. The distribution of DUP-TSN in the SACK chunks is denoted in figure 5.2. Here again in accordance with the results depicted in figure 5.1, we see that the DUP-TSNs in the SACK chunks is consistent and not increasing as in plain vanilla SCTP.

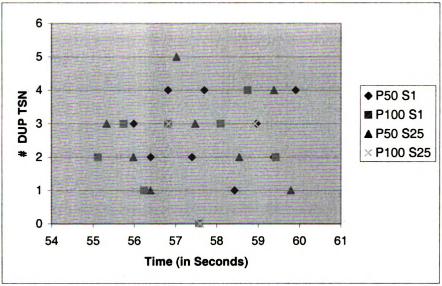

Figure 5.2: DUP-TSN distributions

In plain vanilla SCTP, we were using a connection, which was getting congested as the lifetime of the association increased. At the same time, we had better connections within the association. These better connections could have given a better path, but we never explored for them till our primary path failed completely. Another reason for the improvement in the performance is the retransmission time out parameter. In the plain vanilla case, if the sender doesn't get the acknowledgement within the RTO, the chunk gets retransmitted and the RTO is increased. This in turn reduces the bandwidth utilization resulting in reduced goodput.

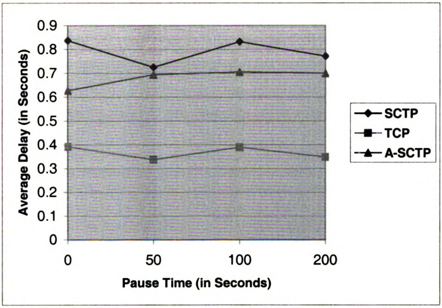

Figure 5.3a: Average Delay AODV, Mean Speed 1m/s

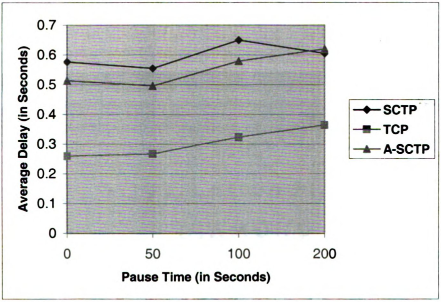

Figure 5.3b: Average Delay AODV, Mean Speed 25m/s

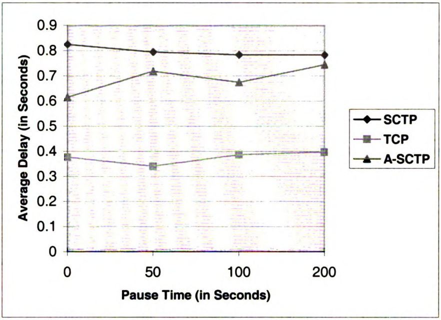

Figure 5.3c: Average Delay DSDV, Mean Speed 1m/s

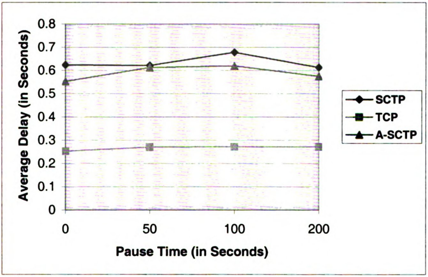

Figure 5.3d: Average Delay DSDV, Mean Speed 25 m/s

From the figures 5.3(a, b, c, d), it is seen that we gain an advantage of 25% reduction in average delay in packet delivery. It is also observed that when the pause time is 200sec, i.e. when there is no mobility of nodes, we don't gain much advantage. This is because, in case of no mobility, there is not as much packet loss due to link deterioration (i.e. one of the intermediate nodes move out of the path). So our algorithm, which uses the transport layer parameters like duplicate packet delivery and the gaps in packet delivery, is not very efficient in such cases. So for a stationary wireless network, this solution may not fare as well as it does for MANET. The reduction in average delay results in the increase in bandwidth utilization. Increase in bandwidth utilization reflects in the increase in the goodput. On an average we get a 20% increase in the goodput.

The other two parameters; namely, the percentage packet delivery ratio and routing overhead are not analyzed in this section for the following reasons. If we are providing a better and more stable path to the packet, the packet delivery ratio is bound to increase. This is also depicted from the goodput graphs in figure 5.4(a, b, c, d).

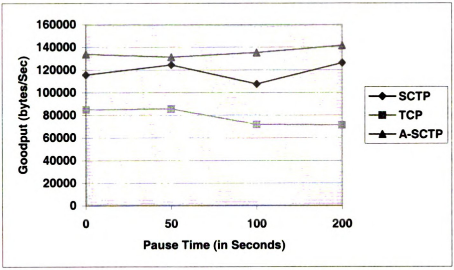

Figure 5.4a: Goodput AODV, Mean Speed 1m/s

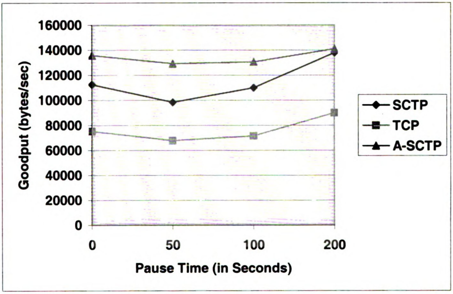

Figure 5.4b: Goodput AODV, Mean Speed 25m/s

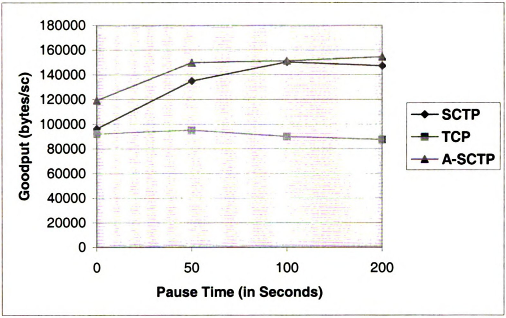

Figure 5.4c: Goodput DSDV, Mean Speed 1m/s

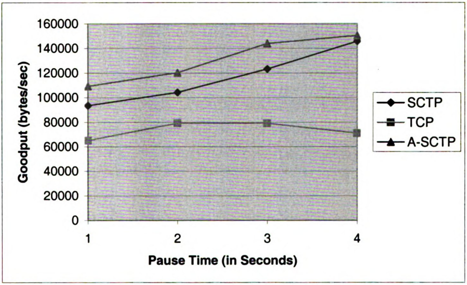

Figure 5.4d: Goodput DSDV, Mean Speed 25m/s

The routing overhead is not analyzed for A-SCTP. This is because we do not introduce any extra overhead in routing.

We use the existing parameters at a node to gauge the condition of the network and take the necessary steps at the node. We are not introducing any new packet into the network to gather statistics and hence not increasing the routing overhead. The only overhead that we are introducing at the node is the additional number of CPU cycles for executing our algorithm. Our algorithm has a constant time complexity (i.e. O(constant k)). A study on sensor networks has revealed that transmitting one bit of information takes as much energy as 800 CPU cycles. Hence the overhead introduced by our algorithm is easily offset by the amount of energy saved by not requiring retransmission of a packet.

Chapter 6: Conclusion

The goals of the thesis were to compare the performance of SCTP in MANET and identify the bottleneck parameters to be optimized to increase goodput. Using simulations we discovered that the average delay was high for SCTP compared to TCP. By reducing the average delay, the bandwidth utilization can be improved. It was seen that the SACK parameters (namely GAP-ACK and DUP-TSN) are strongly correlated to the delay in packet delivery. Consequently, we model the path condition using the SACK parameters. Based on an adaptive algorithm, when the path condition was seen to deteriorate, we use the multi-homing feature of SCTP to switch to another available path. This way we do not wait for the link to deteriorate completely (break down) before we switch. This has resulted in the reduction of packet losses and the overall delay in the packet delivery. The consequence of this can be seen in increased bandwidth utilization, leading to increased goodput. Up to now, researchers have modeled the state of the network using the routing layer or the link layer parameters. We have been successful in modeling the path condition using the transport layer parameters and taking decisions at a higher layer. Another prominent feature is that we do not increase any routing overhead, which most of the other approaches will require in order to gather information.

In the future we wish to consider feedback from the link layer queues, this can be used to model the network. The queue length of the link layer can give a better estimate of the network conditions. Using this information the transport layer can decide to switch to a different path. One of the drawbacks of doing this is that it will lead to cross layer design and the dependence of the transport layer decisions on the link layer.

There is not much performance gain when the nodes are stationary. In the future we would want to model this algorithm to cater to networks, which are not affected by the network dynamics caused by node mobility.

APPENDICES

Appendix A

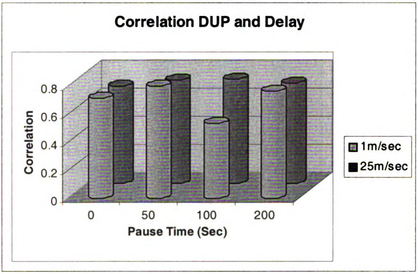

Figure A1: Correlation between DUP-TSN and Delay

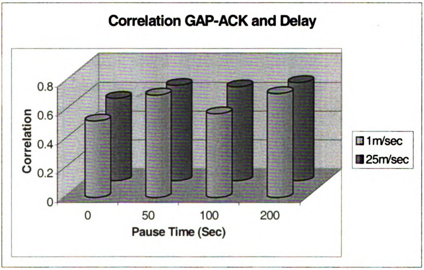

Figure A2: Correlating between # GAP-ACK and Delay

Appendix B

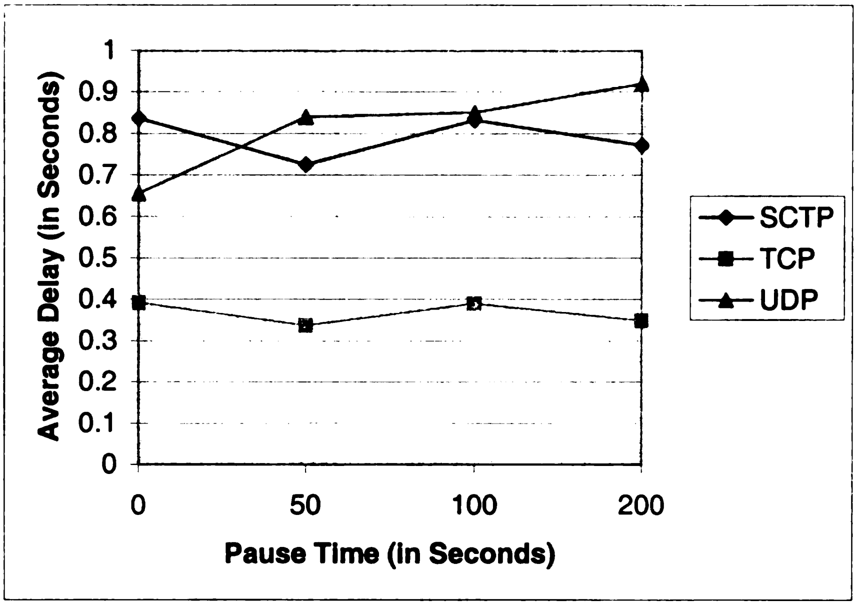

Figure B1.1: Average Delay. AODV, Mean Speed = 1m/s

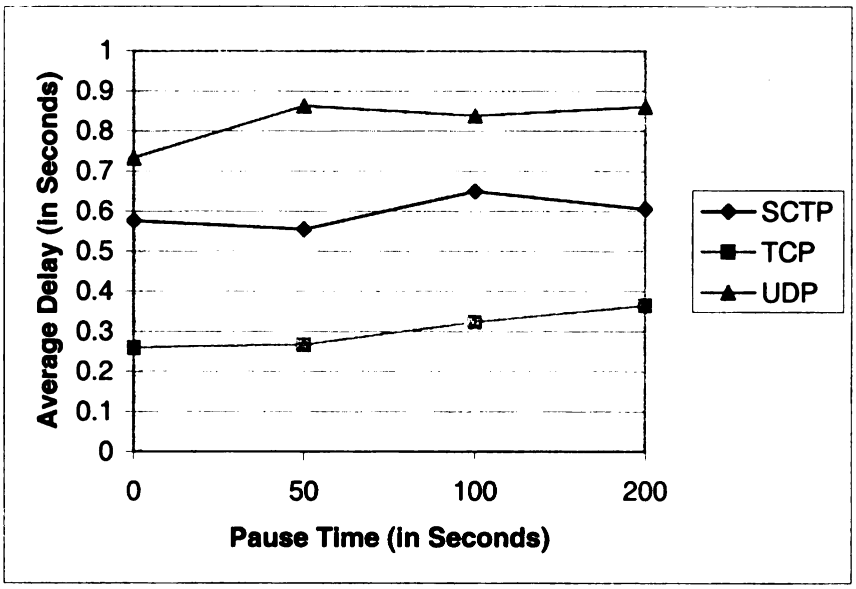

Figure B1.2: Average Delay. AODV, Mean Speed = 25m/s

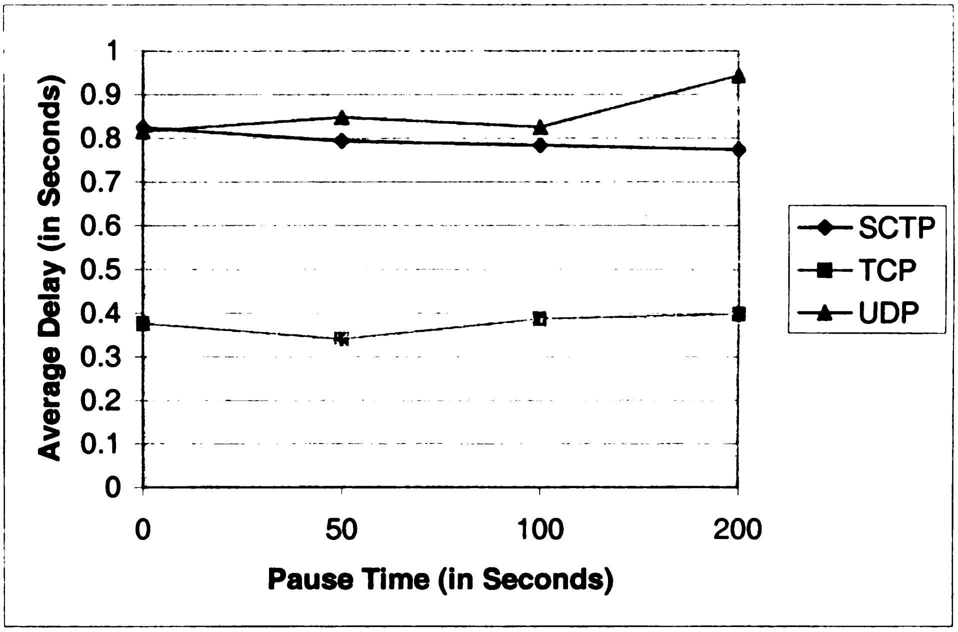

Figure B1.3: Average Delay. DSDV, Mean Speed = 1m/s

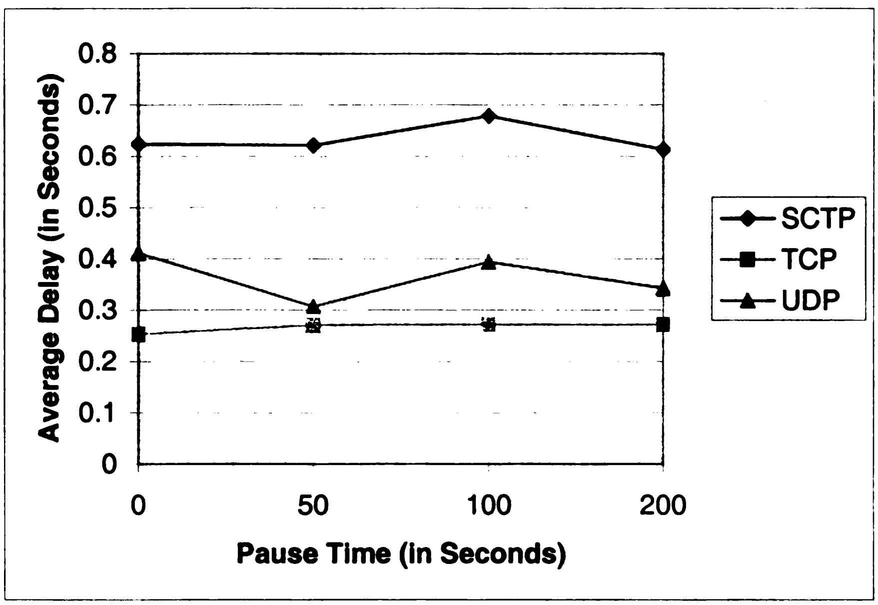

Figure B1.4: Average Delay. DSDV, Mean Speed = 25m/s

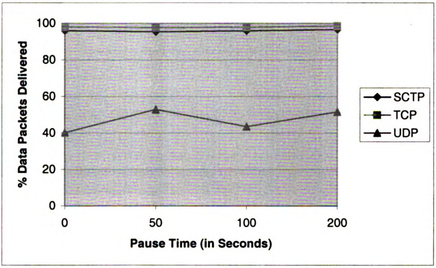

Figure B2.1: % Delivery. AODV, Mean Speed = 1m/s

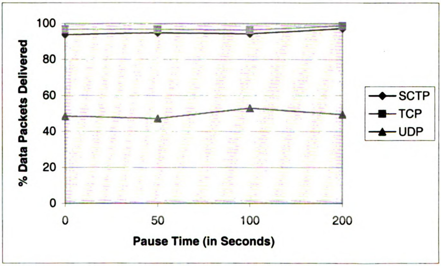

Figure B2.2: % Delivery. AODV, Mean Speed= 25m/s

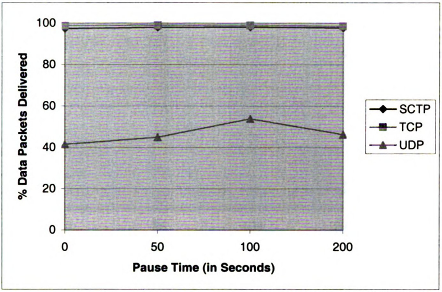

Figure B2.3: % Delivery. DSDV, Mean Speed = 1m/s

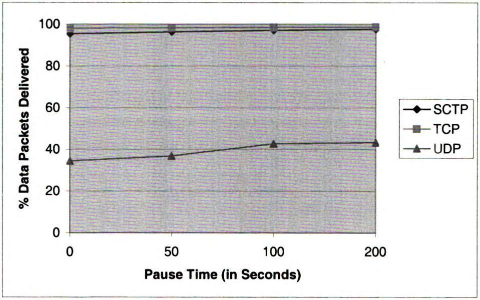

Figure B2.4: % Delivery. DSDV, Mean Speed = 25m/s

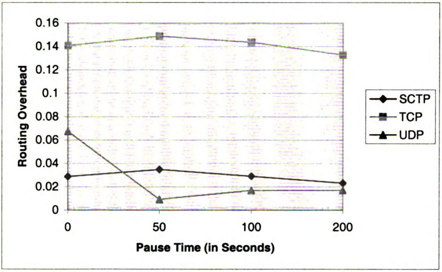

Figure B3.1: Routing Overhead AODV, Mean Speed=1m/s

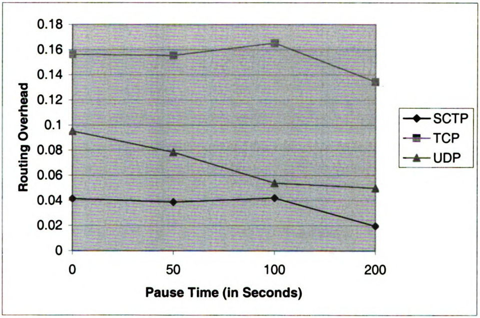

Figure B3.2: Routing Overhead AODV, Mean Speed=25m/s

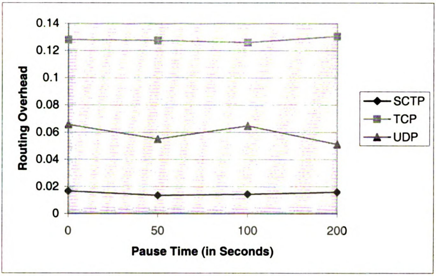

Figure B3.3: Routing Overhead DSDV, Mean Speed=1m/s

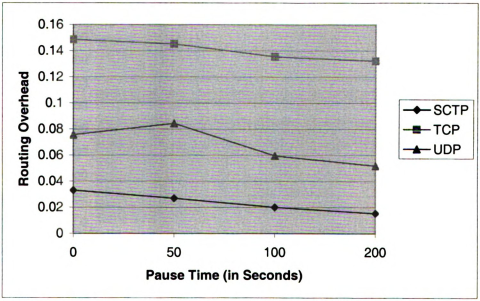

Figure B3.4: Routing Overhead DSDV, Mean Speed=25m/s

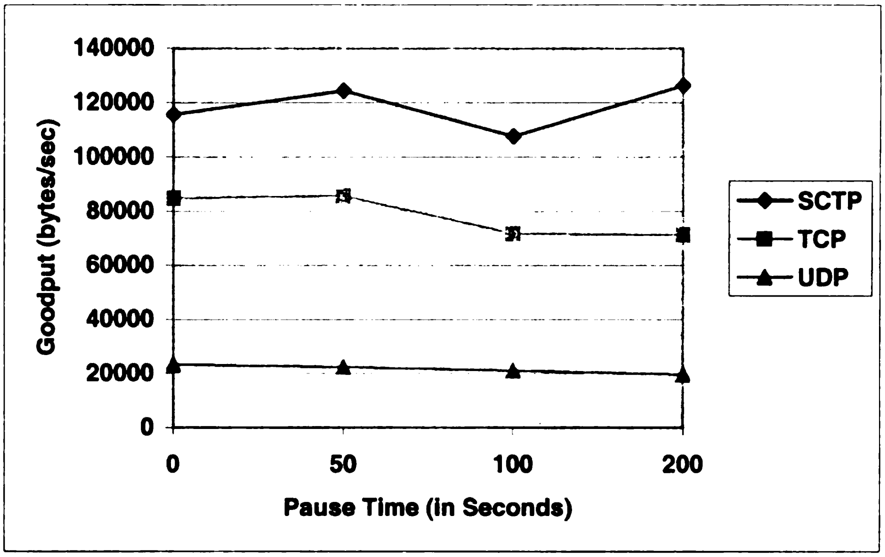

Figure B4.1: Goodput AODV, Mean speed 1m/sec

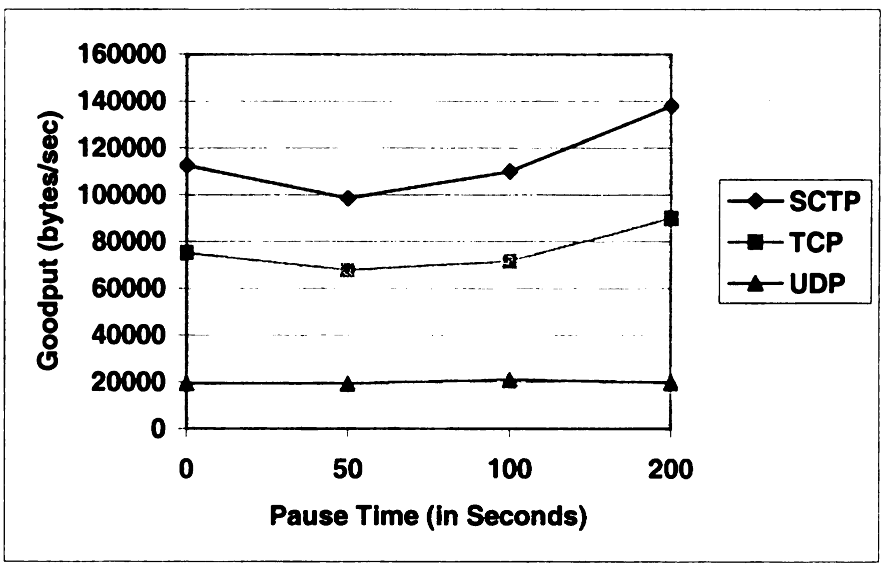

Figure B4.2: Goodput AODV, Mean speed 25m/sec

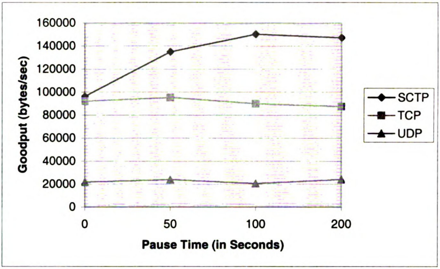

Figure B4.3: Goodput DSDV, Mean speed 1m/sec

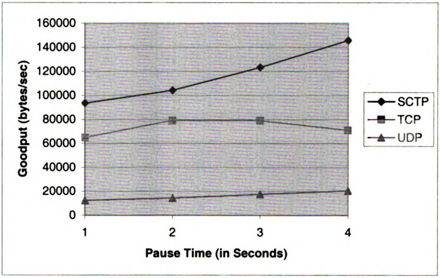

Figure B4.4: Goodput DSDV, Mean Speed 25m/sec

BIBLIOGRAPHY

[1] http://rtg.ietf.org/wg/manet/charter

[2] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinsinki and H. Korth, Editors. Mobile Computing. Kluwer Academic Publishers, 1996.

[3] C. E. Perkins and P.Bhagwat. Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proc. ACM SIGCOMM 94, London, pp. 234-244, 1994

[4] C. E. Perkins and E.M. Royer. Ad-hoc On-Demand Distance Vector Routing. In proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and A pplications, New Orleans, LA, February 1999, pp. 90-100.

[5] Gavin Holland and Nitin Vaidya, Analysis of TCP performance over Mobile Ad-hoc Networks. In proceedings of ACM/IEEE MOBICOM '99, pp. 219-230, August 1999.

[6] Ashish Ahuja, Sulabh Agarwal, Jatinder P. Singh and Rajeev Shorey, Performance of TCP over Different Routing Protocols in Mobile Ad-hoc Networks. In proceedings of IEEE Vehicular Technology Conference (VTC 2000), May 2000.

[7] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad-hoc Network Routing Protocols. In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98), pp. 85-97.

[8] Antonios Argyriou and Vijay Madisetti. Performance Evaluation and Optimization of SCTP in Wireless Ad-Hoc Networks. In proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03)

[9] User Datagram Protocol http://www.faqs.org/rfcs/rfc793.html

[10] Transmission Control Protocol
http://www.faqs.org/rfcs/rfc793.html

[11] R. Stewart et al. Stream Control Transmission Protocol
http://www.faqs.org/rfcs/rfc2960.html

[12] The Network Simulator ns2 http://www.isi.edu/nsnam/ns/

[13] G. Ye, T. Saadawi, and M. Lee, "Improving Stream Control Transmission Protocol Performance over Lossy Links", IEEE Journal on Selected Areas in Communications, Volume 22, Issue 4, pp. 727-736, May 2004.

[14] James Noonan, Andrew Kelly, Philip Perry, Seán Murphy, John Murphy, "Simulations Of Multimedia Traffic Over SCTP Modified For Delay-Centric Handover", In proceedings of World Wireless Congress '04.

[15] Shaliu,Shoubao Yang, Weifeng Sun, "Collaborative STCP: A Collaborative Approach to Improve the Performance of SCTP over Wired-cum-Wireless Networks"

[16] Perkins, Dmitri D. and Hughes, Herman D., "A Survey on Quality of Service Support in Wireless Ad Hoc Network", Journal of Wireless Communications and Mobile Computing (WCMC); Special Issue on Mobile Ad Hoc Networking, Trends and Applications, 2(5) (2002) pp. 503-513.