A Performance Study of TCP on Ad hoc Networks

By
Umair Saeed Qureshi

A Project
Presented to Ryerson University
in partial fulfillment of the
requirements for the degree of

Masters of Engineering

In the Department of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2003

Author's Declaration

I hereby declare that I am the sole author of this research Paper. I authorize Ryerson University to lend this Research paper to other institutions and individuals for the purpose of scholarly research.

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

Borrow List

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign and give address and date.

Acknowledgements

I express my gratitude to my supervisor Prof. Dr. Jaseem-ud-Din and Prof. Dr. X.P.Zhang. The completion of this project would not have been possible without their valuable advice, continual guidance and technical expertise.

A Performance Study of TCP on Ad hoc Networks

Abstract

Ad-hoc networks, characterized by highly dynamic multi hop wireless connectivity, offer challenges related to unique issues of congestion, channel error, routing instability and network partitioning. Dealing with these issues requires precise detection of network states, which we accomplished by measuring appropriate metrics, such as packet out of order, inter-arrival delay differences, connection throughput, round trip time etc. We evaluated the performance of TCP under variety of network conditions running two important routing protocols namely Dynamic Source Routing (DSR) and Dynamic Sequential Distance Vector (DSDV) routing. These protocols belong to different class of routing protocols. DSR is an on-demand whereas DSDV is a link-state routing protocol. In this project, we carried out detailed simulations of a sizable ad-hoc network using NS2 to study the dynamics of the two routing protocols related to the performance of TCP by calculating the above metrics. We observed that congestion in ad-hoc network exhibits dynamic behavior and sometime it is not as bad as in case of fixed networks. For example we observed that node mobility introduces transience to congestion by dissipating congestion at bottleneck nodes. We observed in at least one scenario that node movement totally avoids congestion. We evaluated the performance under channel error conditions by measuring packets out of order and packet losses for both protocols. We also studied the routing characteristics of both protocols under identical mobility conditions. Finally, we evaluated the worst-case performance under extreme network condition by combining congestion, channel error and node mobility.

Contents:

List of Figures

Chapter 1

Chapter 2

Chapter3:

List of Tables:

Chapter 3

Chapter 1

INTRODUCTION

1.1 Project

TCP performance depends on routing protocol. Studies have indicated that it degrades significantly in mobile ad hoc network due to inherent problems of channel errors, frequent route changes and network partitions. In order to improve TCP throughput, these problems must be handled in a different manner from network congestion.

In this project, we investigated impact of routing protocols on TCP performance under these unique conditions of wireless environment. We selected two routing protocols, namely DSR and DSDV, for evaluating TCP performance. These protocols represent two different classes of Routing mechanisms of topology based algorithm. We simulated TCP session in mobile ad hoc network running these protocols under these network conditions. We used metrics which were defined in [2] to evaluate the TCP performance. Reference [2] used these metrics to detect network states by measuring at nodes, which is alternative to the network approach where measurements are performed within the network. By measuring these metrics we are able to derive the interesting results about the performance of transport protocol in an ad hoc network. The results may be useful for improving the performance of TCP layer.

The project is organized as follows: Chapter 1 provides an overview of TCP, its challenges in wireless domain and two approaches to encounter TCP issues in this

environment. Two Routing Protocols, DSR and DSDV are explained in Chapter 2. Network States and Metrics are discussed in chapter (1 or 2). Chapter 3 describes the simulation setup, discussion and results of simulations performed. Results are provided in Chapter 4.

1.2 Ad Hoc Networks

Wireless networking is an emerging technology that allows users to access information and services electronically, regardless of their geographic position. Wireless networks can be classified into two types: -

Infrastructure networks

Infrastructure network consists of a network with fixed and wired gateways. A mobile host communicates with the gateway called Base station via wireless medium. The wired Base station routes traffic originated from mobile and fixed nodes. The mobile unit can move geographically while it is communicating. When it goes out of range of one of base stations, it connects with another base station and starts communicating through the new base station. This is called handoff. In this approach the base stations are fixed.

Infrastructure less (Ad hoc) networks

In ad hoc networks, all nodes are mobile and can be connected dynamically in an arbitrary manner. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. Application of ad hoc networks are situations where a collection of users gather with their devices interconnected to perform a network with little or no prior plan. For example Ad hoc networks are very useful in emergency search-and-rescue

operations, meetings or conventions in which persons wish to quickly share information.

An ad hoc wireless network is self-organizing and adaptive. This means that a network can be formed on the fly without the need for system administration. The term "ad hoc" tends to imply "can take different forms" and "can be mobile, standalone, or networked." Ad hoc nodes or devices should be able to detect the presence of such other devices and to perform the necessary handshaking to allow communications and the sharing of information and services. Ad hoc wireless devices can have different computation, storage and communications capabilities. Ad hoc wireless communication can occur in several different forms.

1) The source and destination both are in ad hoc network. Both communicate with each other with out the support of base station. The source communicates with destination directly or through intermediate node each of which acts as a router and forward the traffic.
2) A Source node in ad hoc network can communicate with a destination connected to a wired network.
3) A Source node can communicate with a destination connected to another ad hoc network. The two ad hoc networks are interconnected through an infrastructure network. The infrastructure network (wired network) in this case routes the packets from the source to the destination.

1.3 Transmission Control Protocol

The Transmission Control Protocol (TCP) is one of the most popular and widely used end-to-end protocols for the Internet today. TCP provides reliable delivery of transport

level segments from a sender to receiver. It provides flow and congestion control functions. The main features of TCP are:

1) Acknowledgements of data packets from the receiver to the sender provide reliability in transmission.
2) Sequence number ensures in-sequence delivery of segments and identifies lost and corrupted packets.
3) Retransmission is used to resend lost or corrupted segments. Retransmission timer is needed to determine when to initiate a resend.

Each TCP segment has segment header, which contains the sender and receiver port numbers, sequence number, some control bits, options and payload. Application data are fragmented into segments and appended with a TCP segment header. At the TCP receiver, these segments are then reassembled back into messages.

1.3.1 TCP Flow Control

TCP provides reliable connection-oriented service. A virtual circuit connection must be established hop-by-hop from the source to the destination before data transmission begins. If acknowledgment of the previous data has been received successfully, then the source gradually increases data transmission. The TCP sliding window mechanism allows the sender to send multiple segments before waiting for an acknowledgment. The window size defines the number of packets that the sender can send before it receives acknowledgments back from the receiver. This window gradually opens wider when ACKs are successfully received. So by keeping track of which segments sent are ACKed and which are not, flow control is introduced since the sender cannot continue to send if the receiver has stopped responding with ACK. The window size can be made adaptive by varying it over time. If the receiver buffers are becoming

full, it sends a small window size advertisement to the sender. This results in the sender reducing its window size to avoid receiver buffer overflow.

1.3.2 TCP Congestion Control

TCP congestion control consists of three phases: a) slow start b) congestion avoidance c) fast retransmit/fast recovery. At starting a connection, or restarting after a packet loss, the congestion window size is set to one packet. The TCP sender gradually increases the congestion window (cwnd) size by one packet upon the receipt of an ACK, until the first sign of congestion is detected. Back off occurs and the window size is reduced to half the current window (down to a minimum of one segment); then slow start process begins gradually. The Slow Start mechanism starts when the SS threshold is introduced, which changes the increment gradient of segment transmission with respect to time. Each ACK received results in increasing the window by 1/cwnd-size. An additive increase (SS)/multiplicative decrease (back off) algorithm is used to avoid congestion in TCP.

1.3.3 TCP Challenges in Wireless Ad Hoc Environment

This fact creates a large percentage of internet traffic comprising of TCP. The TCP connection management is based on back off algorithm in order to overcome congestion and packet loss. TCP considers packet loss as signals for occurrence of congestion then the sender invokes congestion control mechanism. TCP assumes that nodes in the route are static and performs flow and congestion activities at the source and destination nodes. The protocol is designed to perform well over fixed /wired network. But it has been observed that on wireless, TCP suffers poor performance because of packet losses and corruption caused by wireless induced errors. TCP is unable to distinguish the presence of mobility and network congestion.

In ad hoc wireless networks, when a route is broken due to the mobility of nodes in the route, a route reconstruction or reconfiguration procedure is invoked. A delay is occurred during this time when the route is repaired. The TCP sender is unaware of this incident. Hence, it misunderstands the delay of ACK arrival, or the increase in RTT, as signs of network congestion. The source node begins to reduce its transmission window size and initiates slow start that significantly reduces throughput.

Similarly, TCP relies on the packet loss as an indication of network congestion and triggers efficient congestion control algorithms once congestion is detected. The purpose of this congestion control algorithm is to dynamically match the transmission rate of a connection to the currently available connection capacity. Additive Increase/Multiplicative Decrease (AIMD) is the congestion control algorithm use in the TCP. But the major drawback is that it reacts to all packet loss situations identically. AIMD assumes that packet losses are due to congestion and decreases the transmission rate. This algorithm is suitable for congestion-induced packet losses, but inappropriate behavior for non congestion-induced packet losses. Setting larger multiplicative decrease factor may cause larger decrease in window size that leads to inefficient utilization of network capacity. On the other hand, keeping multiplicative decrease factor smaller may lead to slower response to actual congestion and longer convergence time to the fair transmission rate. Hence, when nodes in the ad hoc network sense packet loss as a signal of congestion and mistakenly invoke congestion control algorithm (AIMD) it decrease resultant throughput.

1.3.4 Potential Problems with TCP in Ad Hoc Networks

If TCP is used without any modification in mobile ad hoc networks, a drastic drop in the throughput are experienced because of following possible reasons:

Effect of High BER

Bit error causes packets to get corrupted that result in lost TCP data segments or acknowledgments. When acknowledgments do not arrive at the TCP sender with in retransmit time out (RTO), the sender retransmits the segment, exponentially backs off its retransmit timer for the next retransmission, reduces its congestion control window threshold and closes its congestion window to one segment. Repeated errors will keep the congestion window of smaller size which decreases the throughput.

Effects of Route Recomputations

When an old route is no longer available, the network layer at the sender attempts to find new route to the destination. It is possible that a new route may take longer than RTO at the sender. This causes the TCP sender to time out, which retransmits a packet and invokes congestion control mechanism. Thus, when route is discovered, the throughput will continue to be small for some time as TCP invokes congestion control mechanism. If route computations are very frequent in network then TCP can not get chance to grow its window to its full size.

Effects of Network Partition

If source and destination are partitioned for several seconds then all the packets transmitted will be lost. So, if the partition occurs longer than RTO, the situation leads to consecutive retransmission of same segments to the receiver while the receiver is disconnected from the sender.

Effects of Multi path

Some protocols compute multiple paths between source and the destination in order to decreases frequency of route recomputation. This causes out of sequence packets arriving at the receiver. Another source out of order delivery is channel error. When error in the channel increases, the nodes tend to forward packets and acknowledgements through different routes. Data Packets reach their destinations out of sequence as these follow multiple paths. The effect of this is that the receiver generates duplicate acknowledgments which cause the sender to invoke congestion control.

1.3.5 Two Approaches for solution to Challenges

TCP invokes congestion control mechanism in case of packet loss as an indication of network congestion even if the loss is not due to congestion in the network. Since congestion is not the only reason of packet loss in wireless networks. Nodes connected to the network must be capable of differentiating different loss situations. Reaction of nodes to these situations should be different to their reaction to the congestion control; otherwise severe throughput degradation may occur otherwise severe throughput degradation may occur. An important issue is the capability of node to sense different network states that cause packet loss or expiry of retransmission time out. Appropriate filtering of network states enable nodes to take suitable action in response to a particular state. In order to differentiate network states, two approaches have been suggested.

1) Network Approach

2) End-to- End approach

Network Approach

In this approach, network implements a monitoring mechanism that generates a notification message when it detects an abnormal event so that TCP may react [11].

When mobility triggers network disconnection (called link failure event), the routing layer sends an Explicit Link Failure Notification (ELFN) to the TCP sender. On the other hand, an explicit loss notification is sent to the TCP sender when the router observes a wireless channel-induced packet loss. An obvious limitation of this approach is that these techniques need to be deployed at every node. Hence, this technique is difficult to adopt because of potential heterogeneity in network connectivity nodes.

End-to-End approach

This approach is used to identify the presence of various network states by developing suitable end-to-end measurements at the receiver end. These end-to-end measurements identify the presence of various network conditions that if left unchecked, will decrease the throughput. This approach detects congestion, disconnection, route changes and channel error. This approach is easy to implement and deploy and requires no network support and provides the flexibility for backward compatibility. Previous Research [2] has indicated that identifying the following network states which are necessary to improve TCP performance over ad hoc networks.

1.3.6 Network States

Congestion

When network congestion occurs, ad hoc transport should adopt the same congestion control mechanisms as Conventional TCP. Congestion is defined as packets loss due to buffer overflow at some nodes. [2]

Channel Error

Channel error may also cause packet loss at random. If random packet loss occurs, the sender should re-transmit the lost packets without invoking the congestion control mechanism that is without decreasing the transmission rate. [2]

Route Change

The delivery path between the two end hosts can change from time to time, with disconnections that are too short-term to result in TCP time out. Depending on the routing protocol, the receiver may experience a short burst of out-of order packet delivery or packet losses. It is recommended that the sender should estimate the bandwidth along the new route by setting its current sending window to the current slow start threshold and initiating the congestion avoidance phase [2].

Disconnection

This is the state when nodes cannot communicate as these are out of radio range of each other. This can also happen when any obstacle prevents communication between two nodes. If this delivery path is disconnected for the period of time greater than Retransmission Time Out (RTO), then TCP, sender backs off transmission exponentially. It is suggested that the sender should freeze the current state of congestion window and retransmission timeout and perform periodic probing until the connection is reestablished.

1.3.7 METRICS

The above network states can be determined by an end if it measures the following metrics, which are proposed in [2].

Inter-Packet Delay Difference :(IDD)

IDD is a measurement of delay difference between consecutive packets. It shows the congestion level along the forwarding delivery path. Upon each packet arrival, the receiver calculates the IDD value. Unlike the conventional inter-packet arrival delay (IAD), IDD is unaffected by random channel errors and packet sending behaviors. Figures 1.1, 1.2 and 1.3 are taken from paper [2].

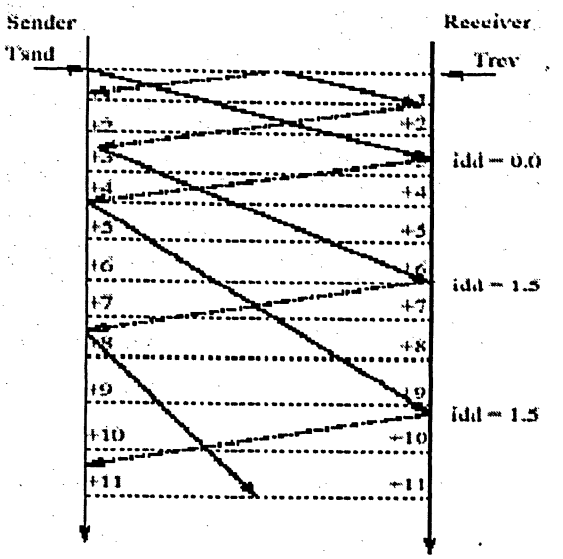

Figure 1.1: Congestion

Figure 1.1 shows the development of congestion and queue length is reflected by IDD values.

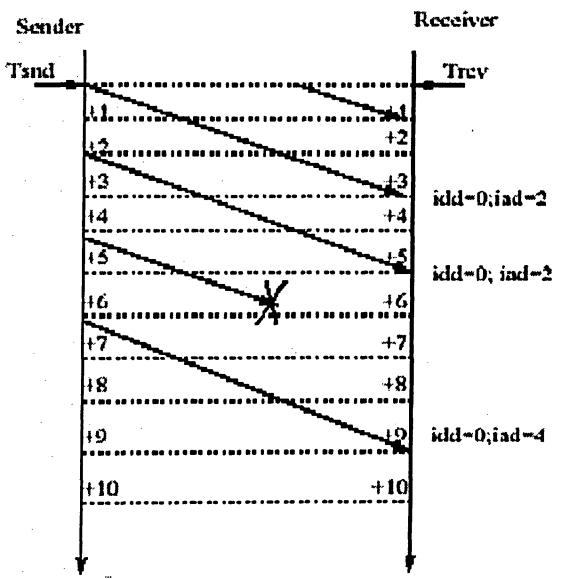

Figure 1.2: Channel Error

Figure 1.2 shows that IDD is not affected by random packet loss.

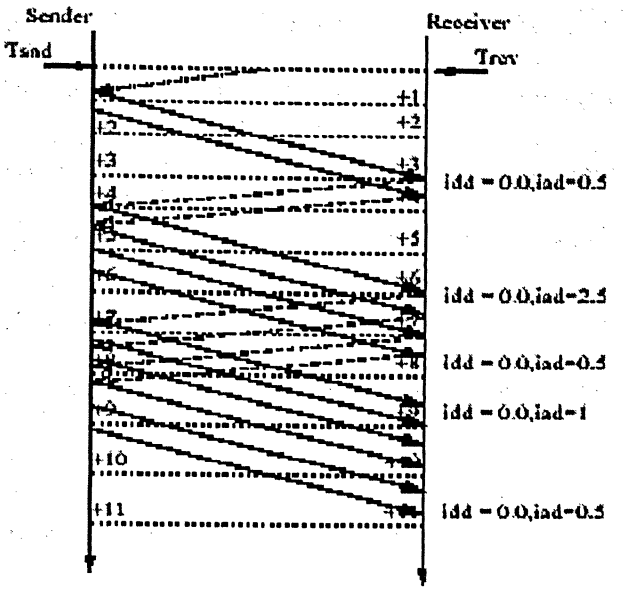

Figure 1.3: Packet Sending Behavior

Figure 3 shows that when there is no congestion, IDD remains unaffected by packet sending behavior. IDD can be calculated using the following formula:

$IDD = A_{i+1} - S_{i+1} - (A_{i+1} - S_{i+1})$

A= Packet Arrival time.

S = Packet Sending time.

Throughput

Throughput is measured as the number of packets received during time interval t. This metric is sensitive to Channel Errors, Disconnections and TCP source states. But it is not affected by out of order packets.

Throughput = No. of Packets received / Time Interval.

Packet Out-of-Order Ratio (POOR)

When a node moves from one point to another route transition may occur during which multiple delivery paths exist. Packets along the new route may reach destination earlier than those which are sent earlier along the old routes.

A packet is counted as being out of order if it arrives after a packet that was sent later by the same TCP sender. The receiver records the known maximum sending time for the entire received packet over the same the TCP connection, which is denoted by T. In order to measure POR the sender must time stamp every packet. When a packet is received that has time less than T, it is used to POR.

Packet out of Order Ratio = Number of out of order Packets / Time Interval.

Packet Loss Ratio (PLR)

This metric is used to measure intensity of channel error. It can be measured by keeping the number of missing packets during time interval T.

Packet Loss Ratio = Number of Lost Packet / Time Interval

Round Trip Time (RTT)

This metric calculates sum of time taken by data packet from real sender of packet to its destination and acknowledgement packet from destination to the sender.

Round Trip Time = $(T_{Data,Received} - T_{Data,Sent}) - (T_{Ack,Received} - T_{Ack,Received})$

$T_{Data,\ Received}$ = Time at which Data Packet is received.

$T_{Data,\ Sent}$ = Time at which Data Packet is Sent

$T_{Ack,\ Received}$ = Time at which Acknowledgment Packet is received.

$T_{Ack,\ Sent}$ = Time at which Acknowledgment Packet is Sent.

1.4 Related Work

With proliferation of mobile computing devices, the demand for continuous network connectivity regardless of physical location has spurred interest in the use of mobile ad hoc networks. This fact has initiated a lot of research in solving routing related issues in the area of mobile ad hoc wireless networks.

Paper [1] has compared performance of Dynamic Source Routing and Ad hoc on Demand Distance Vector protocols on the basis of three metrics which are: Packet Delivery Fraction, Average end-to-end delay of data delay of data packets, Normalized Mac Load and Normalized routing load. They have concluded that DSR outperforms AODV for smaller number of nodes and lower mobility incase of delay and throughput metrics.

An end-to-end approach is developed in paper [2] that relies on end-to-end measurements. Four metrics are suggested in order to detect different network states. A network event is signaled only if the relevant metric detect it. Standard TCP [2] uses end-to-end measurement of RTT and packet loss to detect congestion.

Paper [3] outlines problems with TCP in Ad hoc networks and suggests a thin layer between IP and TCP that ensures correct TCP behavior in order to maintain high throughput.

An Explicit link failure notification mechanism is suggested in paper [4] for each wireless node to inform TCP sender. This way the sender can distinguish link failure losses from congestion losses.

1.5 Summary

In this chapter, brief background of TCP performance degradation in mobile ad hoc network is discussed. We explained issues related to Transmission Control Protocol and network states of mobile ad hoc network environment. Two approaches which are suggested in research work in order to distinguish these network states are provided. Finally, we provided definitions of metrics used in our project.

Chapter: 2

Routing Protocols

The routing protocols designed for wired network cannot be used for mobile ad hoc networks because of mobility of nodes in the network. Numerous routing protocols have been developed for mobile ad hoc networks. These protocols must deal with nodes and links constraints, such as limited battery power, low bandwidth and error rates. These protocols are broadly categorized as: a) Table Driven protocol b) On-Demand Protocol. The characteristics of these routing protocols are quite distinct, which are described below.

a) On Demand Routing Protocols

These protocols take lazy approach to routing. In contrast to the table driven routing protocols where all up-to-date routes are maintained at every node, the routes are created when a source wants to send a packet to the destination. The routes remain valid till the destination is reachable or until the route is no longer needed.

b) Table Driven Routing Protocols

In table driven routing protocols, each node maintains one or more table containing routing information to every other node in the network. All nodes update these tables so as to maintain a consistent and up-to-date view of the network. When the network topology changes the node propagate update

messages through the network in order to maintain consistent routing information about the network. The protocols differ in this category differ in the method by which information about the topology changes is disseminated to all the nodes and the number of necessary routing-related tables.

2.1 Dynamic Source Routing (DSR)

The data packet forwarding technique by which source node determines sequence of nodes between itself and the destination node in a dynamically changing network topology is called "Dynamic Source Routing". By this technique, any node can compute a source route across multiple hops to any destination in a mobile ad hoc network. Each hop is identified by the address of the next node to which the packet is transmitted on its way to the destination. The sender maintains list of hops of this route in the packet's header. This protocol is designed keeping in view of requirements of multi-hop wireless ad-hoc networks. DSR determines and maintains all routing information which changes dynamically with joining or leaving of nodes and with alterations in transmission conditions.

2.1.1 Basic Operation

DSR is designed to ensure successful delivery of data packets in spite of node movement and other changes in network conditions. In order to transmit a data packet to another host, the sending node constructs a source route in the packets' header, giving the address of each node in the network through which the packet should be forwarded in order to reach the destination host. The sender transmits the packets over its wireless interface to the first hop to the next node identified in the source route. This node receives the packet and checks the source route in the packet's header. If this node is not the final destination

then it forwards the packet to the next hop over its wireless interface. Once the packet reaches destination, the packet is delivered to the network layer of the host.

2.1.2 Two Mechanisms

The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes in the ad hoc network. These are following:

a) Route Discovery Mechanism.

b) Route Maintenance.

Route Discovery Mechanism

Route Discovery allows any node in the ad hoc network to discover a route to any other host in the ad hoc network whether directly through transmission range or through intermediate network nodes. Each mobile node participating in the ad hoc network maintains a route cache in which it caches source routes that it has learnt. Before sending packet to the other node, the sending node checks its route cache for a source route to the destination node.

Route Discovery mechanism is initiated when the source node broadcasts a route request packets which may be received by those hosts that are within wireless transmission range of the sending node. The route request packet identifies the destination node as the target of the route discovery, for which the route is requested. If the route discovery is successful in finding route to the destination, the initiating host receives a route reply packet listing a sequence of network hops through which it may reach the target.

Route discovery mechanism works by flooding the network with route request (RREQ) packets [5]. Each ***route request packet*** contains address of the original Initiator of the request and address of the target of the request. Request-id is set by the initiator from a

locally-maintained sequence number to detect duplicate route requests received. This route request (RREQ) is received by those nodes which are in the transmission range of the initiator. Each Node in the network maintains a list of the initiator address mapped with request id that it has recently received on any route request. Following steps are performed by the node after receiving a route request:

1) If the address of host's own address does not match to the target address in the route record then host attaches its own address in the route record of the ***route request packet*** and re-broadcast the request, as shown in figure1.
2) If the target matches this host's own address, then the route record in the packet contains the route by which the request reached this host from the initiator of the route request. The copy of route is sent to the initiator in a ***route reply packet.***(refer figure 2.1)
3) In order to keep the mechanism ***loop free,*** Route Request Packet is discarded if the host address is already listed in the route record of the request or if senders' address and request id of this route request are already present in the hosts' list of recently received route request. It means that host has already received this request packet.

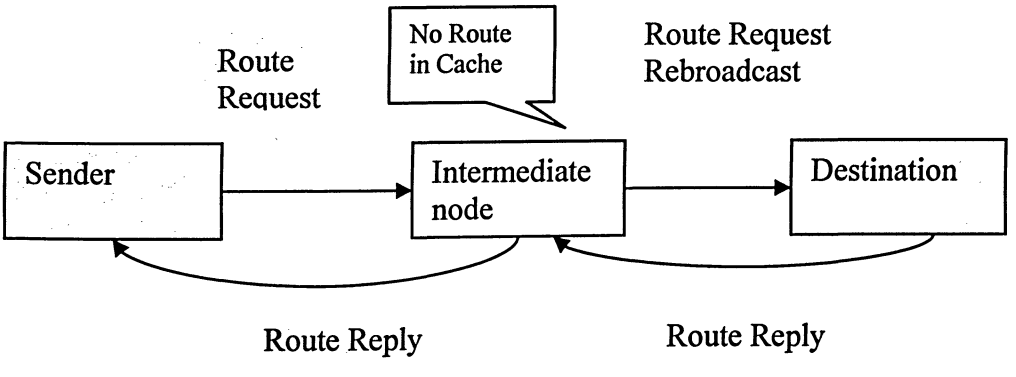

Figure 2.1: No Route in cache of Intermediate Node

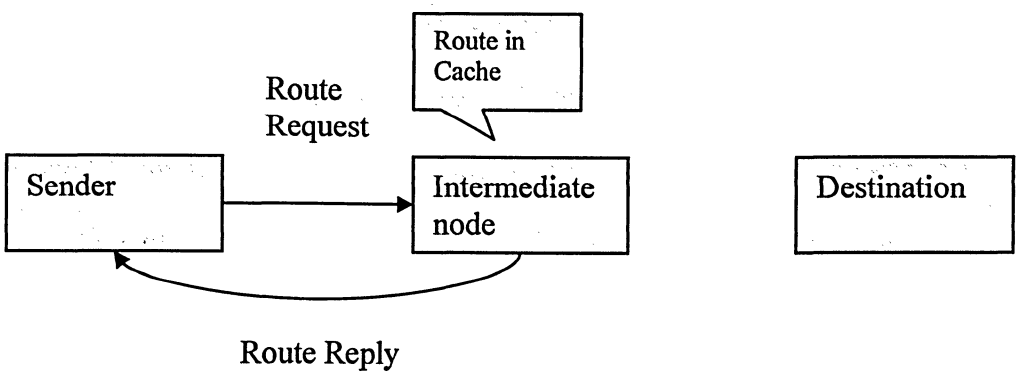

Figure 2.2: Route in cache of Intermediate Node

Hence, the route request packet broadcast by the sender can be received either by the destination node or an intermediate node. If it is received by an intermediate node then the node searches route to the destination in its cache. If it has route in its cache then it appends this route in route reply packet. If the route request packet is received by the destination node then the destination node sends route reply (RREP) packet to the sending node by reversing the route in the route record of route request packet the destination.

Route Cache

Nodes participating in the ad hoc network store routing information in their caches. These routes are learnt through route request, route reply and route error packets [5]. The collection of routes form tree of routes rooted at the hosts. A host can add entries to its route cache any time it learns a new route. Since wireless transmissions are inherently broadcast, a host can add to its route cache the routing information it gleans from overhearing from any data or route reply packet. Consider five nodes A, B, C, D and F, as shown in figure 2.3.

1) Mobile node A computes a route discovery for mobile host D and caches the route through B and C. Node A also learns the route to B and C.

2) When a host forwards a data packet as an intermediate hop on the route in that packet, the forwarding host is able to observe the entire route in the packet. When host B forwards packets from A to D, B can add the route information from that packet to its own route cache. If a host forwards a route reply packet, it can also add the route information from the route record being returned in that reply, to its own route cache.

3) If the host has a route cache entry for the target of the request, it may attach this cached route to the initiator without propagating the route request. If Mobile node F wants to send packet to D, it will initiate route discovery mechanism by broadcasting a route request packet. If this broadcast is received by A, A can return a route reply packet to F containing the complete route to D consisting of the sequence of hops A,B,C and D.

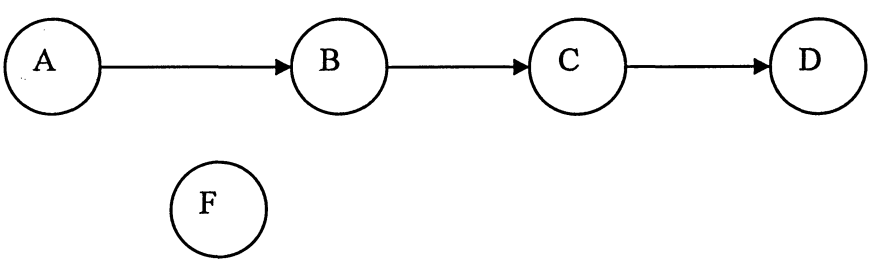

Figure 2.3: Route Discovery Mechanisms using Route cache

Route Maintenance

A host continuously monitors the correct operation of the source route while using this route. This monitoring of the correct operation of a route in use is called Route Maintenance. The route can no longer be used if any of the hosts along the route should fail or be powered off or if any node moves out of wireless transmission range of the next or previous hop along the route.

Since wireless networks are inherently less reliable than wired networks, route maintenance is performed frequently that requires reliable operation in dealing with packet losses. The two schemes End-to-End Acknowledgments and Hop-by-Hop Acknowledgments are proposed for making operations reliable. Hop-by-Hop Acknowledgements indicates particular hop in error in the route error packet but with End-to-End acknowledgments, the sender may assume that the last hop of the route to this destination is in error.

Route Error

If the data link level reports a transmission problem for which it cannot recover, this host sends route error packet to the original sender of packet. The Route Error packet contains:

- Addresses of the hosts at both ends of the hop in the error.
- Address of the host that detects the error.
- Address of the host to which it is attempting to transmit the packet on this hop.

When a host receives a route error packet, it removes the hop in error from its route cache as well as truncates all routes that contain this hop.

2.2 Destination-Sequenced Distance Vector Protocol (DSDV)

In DSDV protocol, each mobile node of an ad hoc network maintains a routing table that contains the following information about all available destinations, next hop to each destination and sequence number generated by the destination and metric of the path.

2.2.1 Basic Mechanism

Each node of the net work updates routing tables when new information is available that is significant to maintain the consistency of the routing table with the dynamically changing topology of the ad hoc network [6]. Nodes advertise routing information by

broadcasting a routing table update packets periodically or immediately when network topology changes. The update packets have metric of one for directly connected nodes. This shows that each receiving neighbor is one metric away from the node. After receiving the update packet, the neighbors update their routing table with incrementing the metric by one and broadcast the update packet to their corresponding neighbors. This process is repeated till all the nodes of the ad hoc network have received a copy of the update. A node receiving update packet keeps it for a while to wait for the arrival of the best route for each particular destination node before updating its routing table and retransmitting the update packets. The update route information contains metric, sequence number for each entry and addresses of the final destination and the next hop node.

Packet is forwarded if a node receives multiple update packets for the same destination during waiting period; the routes with more recent sequence number are preferred as the basis of packet forwarding. If the packets have the same sequence number then the route with least metric is selected. The advertisements of routes that are about to change may be delayed until the best routes have been computed. Delaying advertisements of unstable route can decrease e fluctuations in the routing table. This decrease in variation of routing table in turn reduces the number of rebroadcasts of routes entries with the same sequence number. Following are types of Update Packets:

a) **Full Dump**

Updates that carry complete routing table are called "Full Dump". Each node in an ad hoc network must periodically transmit its entire routing table to its neighbor most likely using multiple network protocol data units (NPDU). The full dump can also be transmitted in response to movement of mobile node.

b) Incremental Routing updates

Update packets that carry only the changed routing information between the full dumps is called "Incremental Routing Updates". The mobile nodes determine the significance of the routing information changes to be sent out with each incremental advertisement. For example, the change in route is given priority over the change in the sequence number while updating routing table.

2.2.2 Topology Changes

Links break when nodes move from one place to another or power of these nodes are shut down. The broken links with neighboring nodes are inferred by a host when no broadcasts have been received from the former neighbor. On detecting a broken link, the mobile host performs following operations [6]:

1) All the routes for which the next hop is reachable through the broken link are assigned infinity metric value.
2) Sequence number is updated.
3) The modified route is immediately disclosed by broadcasting an update packet.

When a link breaks, any mobile node other than destination node generates sequence number which is greater than the last sequence number. In order to report a change, a node generates an even sequence number for itself. The node generates an odd sequence number for reporting changes about its neighbor. This will help to avoid conflicting sequence numbers to be generated by nodes in response to network topology changes. The newly generated sequence number and metrics are put in an Update message and broadcast over the network.

The routes to a lost node will be reestablished when the lost node comes back to the network and broadcast its next update message with an equal or later sequence number

and finite metric. The update message will be disseminated through out the network to indicate that the broken link has come back into service. In any case, an entry with infinity metric is given priority over any entry with finite metric in the routing table.

2.3 Summary

We have compared the performance of two routing algorithms, on demand source routing and table driven routing in ad hoc networks. DSR uses source routing and route caches and does not depend on any periodic or time-based activities. DSR exploits caching and maintains multiple routes per destination. DSDV, on the other hand, uses routing tables to compute routes to other nodes in the network. These routing tables are maintained by periodic route updates. Routes are selected on the basis of latest sequence number or smallest metric. These criteria guarantee loop-free routes. Results obtained by using these two protocols in our simulations are discussed in Chapter 3.

Chapter 3

Simulation Results and Discussions

This chapter presents results and explanation of simulations executed while investigating the performance of topology based routing in Ad hoc networks. We attempted to evaluate reactive and proactive approaches. Thus, we selected DSR and DSDV protocols on the basis that DSDV belongs to Table Driven and DSR represents On Demand routing algorithm.

Following features and conditions are implemented for studying the behavior of these protocols:

1) Congestion:

   a) Full Path Intersection.

   b) Partial Path Intersection.

   c) No Path Intersection.

2) Channel Error

3) Routing

4) Composite Effect (Congestion + Channel Error + Mobility).

We evaluated performance of these protocols by computing Metrics defined in Chapter 1 (refer section 1.3.7) under above mentioned conditions.

3.1 Simulation Setup

In this project, NS-2 simulator with CMU wireless extension has been used. The area of simulation in which nodes can roam around freely is fixed as 800mX800m. Number of nodes is taken as 15. These nodes can move around randomly in this area. All these nodes run IEEE 802.11b. The physical layer has a data rate of 2 Mbps.

Nodes' mobility is an important parameter while evaluating ad hoc networks. Random way Mobility Model is selected as movement model for nodes. This model successfully captures all possible movements of nodes in all directions. It avoids geographic restrictions as nodes can move in all directions randomly at any instant of time. The movement pattern of one mobile node is independent of the other node. Similarly, the velocity of node does not depend on the velocity of the other node. This model provides different possible movement scenarios which matches closely to reality. It enables us to analyze ad hoc network behavior when nodes move around with random velocity towards their random destination as in realty the node movements are unpredictable. Hence, the analysis based on this model can be served as appropriate estimation of real time ad hoc network behavior.

In NS2, the effective communication range for each node is set to 250 meters. Since nodes can move randomly, we selected an area of 800mX800m. This area is sufficient to provide node isolations to simulate disconnections in different mobility scenarios. Following Table 3.1 summarizes all the common constant parameters of the simulations.

Table 3.1: Simulation Parameters

Parameters	Value
Terrain Size	800mX800m
Number of Node	15
MAC Protocol	802.11
Bit Rate	2 Mbps
Wireless Propagation Model	Free Space
Antenna Type	Omni directional
Mobility Model	Random Mobility Model
Speed	10 m/s

The speed of node is selected as 10 m/s which represent the speed of pedestrian. Antenna type is Omni directional as most of the mobile nodes. This scenario represents group of pedestrians moving in a square field e.g. rescue workers in an area hit by disaster. TCP is used with maximum window size of 8 packets. The packet size is 1460 bytes. TCP connection starts at 10^{th} second and ends at 150^{th} second between node 0 and node 7. Each simulation run lasts for 300 seconds.

Metrics

Following Metrics are calculated in these above conditions:

1) Number of TCP Packets received.
2) Inter Packet Delay Difference.
3) Number of Packets Dropped.
4) Number of Packets Out Of Order.
5) Round Trip Time.

Mobility Scenarios

In this project, different mobile scenarios are generated by using Random Mobility Model. We generated 38 different scenarios out of which we picked five interesting scenarios for analysis.

Congestion Scenario

In order to analyze effect of congestion, we presented no congestion and congestion conditions. In congestion conditions case, we used Constant Bit Rate traffic to simulate congestion in the TCP path. We arbitrarily selected node 3 and node 12 as source and destination of CBR traffic.

CBR traffic has following parameters:

Inter Packet spacing = 0.0025 second

Packet Size = 1460 bytes

Bit Rate = 1460X (1/0.0025) X8 = 4.672 Mbps.

High volume of CBR traffic caused congestion by creating queues at buffers of intermediate nodes. TCP packets were being routed through these nodes which were also forwarding CBR packets. Thus, TCP traffic experienced congestion at these nodes. We scheduled CBR connection during interval 60^{th} second to 80^{th} second of simulation time.

We took following two intervals during TCP connection period and calculated mean and instantaneous values of Inter Packet Delay Difference (IDD) during both of these intervals:

1) Before Congestion Interval. = 40 second to 60 second.
2) During Congestion Interval. = 60 second to 80 second.

Channel Error Scenario

In order to simulate the effects of error prone nature of wireless medium, we used the error model provided with ns-2 to simulate channel error.

3.2 Congestion

Simulations for recording TCP throughputs are performed using DSR and DSDV under the same mobility scenario as generated by Random Way Mobility Model. In order to study effects of congestion on the TCP throughputs, we simulated Congestion and No Congestion Conditions. Congestion is created by CBR traffic introduced during time interval between 60^{th} and 80^{th} second. TCP throughputs, recorded using DSR as routing protocols for five Mobile scenarios, are shown in figure 3.1. We showed effects of congestion on TCP throughputs in case of DSR protocol in figure 3.2.

DSR

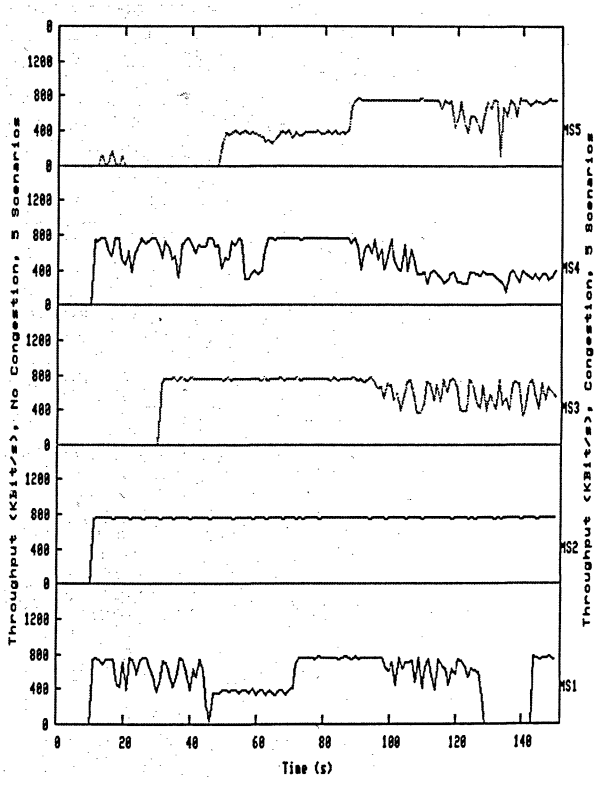

Figure.3.1: TCP Throughput for DSR

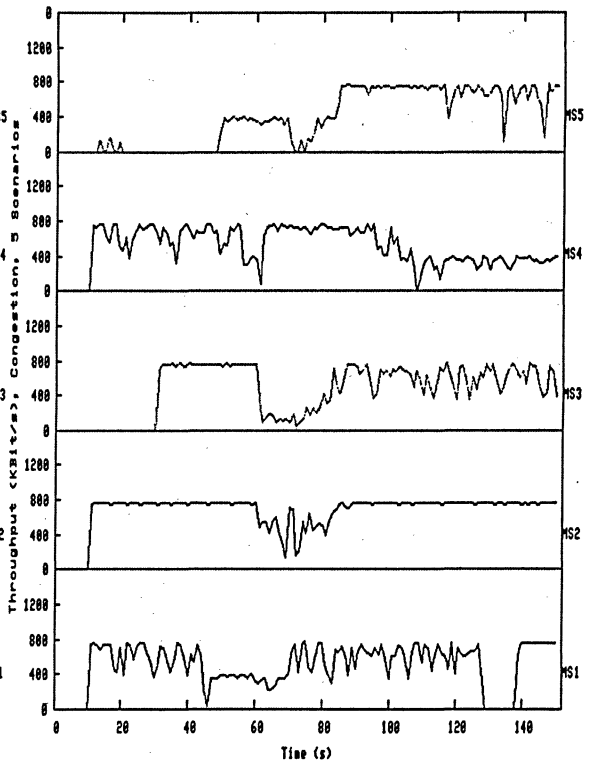

Figure.3.2: TCP Throughput for DSR With Congestion

DSDV

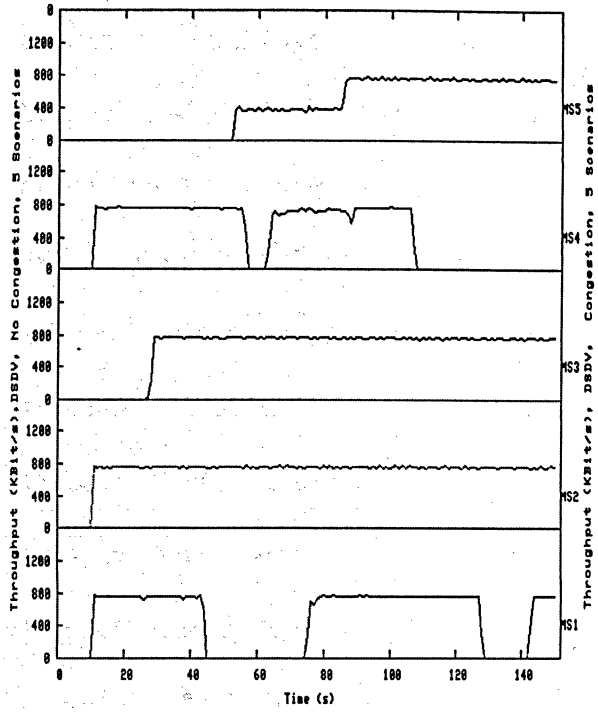

Figure 3.3: TCP Throughput using DSDV

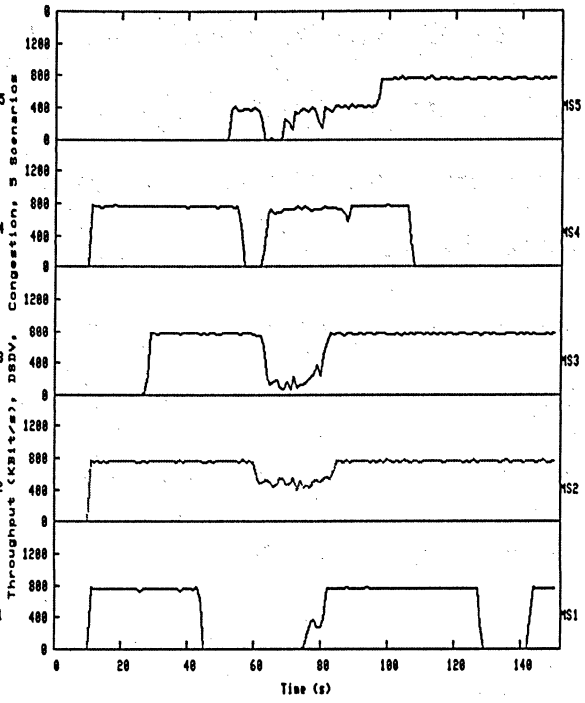

Figure 3.4: TCP Throughput using DSDV With Congestion

Similarly, we repeated these set of experiments with DSDV routing protocol and recorded TCP throughput with and without effects of congestion in figure 3.3 and figure 3.4. The number of each mobile scenario is tagged on the right side of each graph. The observations made in these scenarios are explained in the following sections. Three following phenomenon are observed in MS3, MS5 and MS4 respectively while simulating TCP throughputs for mobile ad hoc networks:

1) Full Path Intersection.
2) Partial Path Intersection.
3) No Path Intersection.

3.2.1 Full Path Intersection

When TCP and CBR traffics both flow along the same path i.e. share same nodes, then TCP throughput decreases drastically during this period of sharing the route. CBR traffic

created bottleneck during 60th second to 80th second of simulation time. In order to represent a scenario in which both transmission Control Protocol traffic and Constant Bit Rate traffic share the same path, Mobile Scenario 3 is selected. When both of these traffics share same intermediate node for whole duration of either of traffics connection life time, then throughputs are affected to their maximum. The phenomenon in which both traffics flow through the same intermediate node during the whole connection time of either of traffics can be termed as Full Path Intersection. We recorded TCP throughputs for DSR and DSDV protocols under congestion and no congestion conditions. These conditions are discussed in following sub sections.

DSR: No Congestion

We calculate TCP throughput when there is no CBR traffic which can create congestion. TCP communicating nodes are having route stability from 10th second to 90th second of simulation time as shown in figure 3.5. During this period, packets are being forwarded over the stable routes. After 90th second, the routes of the packets are changing continuously which is represented by high fluctuations in TCP throughput. Inter Packet Delay Difference which is metric designed for congestion, remains approximately the same for no congestion and congestion intervals. These IDD values are graphically represented for no congestion and congestion cases in figure 3.6 and figure 3.7 respectively. IDD values are approximately the same because no traffic such as CBR is flowing in the network which can occupy buffers and develops queues at intermediate nodes. The TCP packet is forwarded without waiting in queues at these nodes.

TCP Packets received = 6711 packets

Mean IDD value (40th second - 60th second) No Congestion Interval = 0.0089 s.

Mean IDD value (60th second - 80th second) Congestion Interval = 0.0090 s.

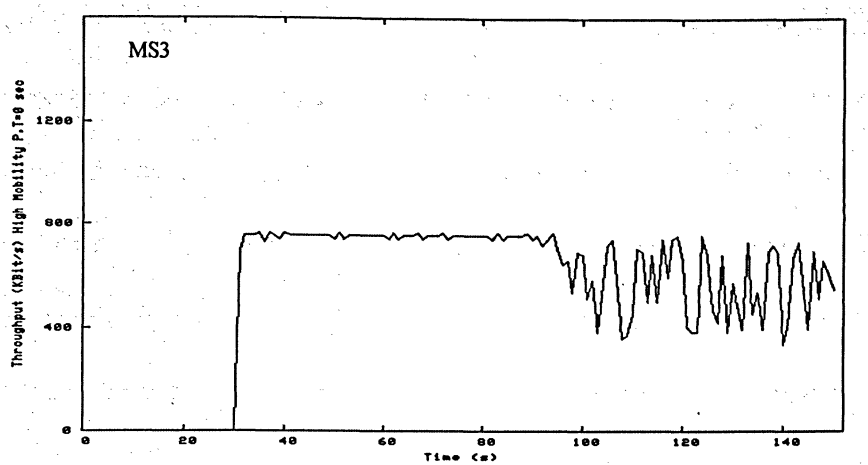

Figure 3.5: TCP Throughput for DSR without Congestion

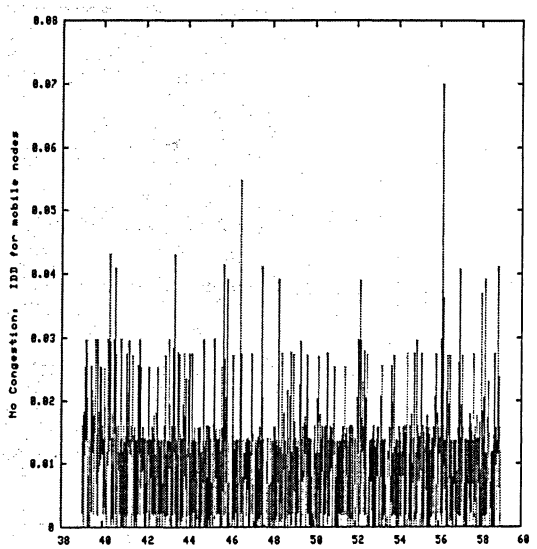

Figure 3.6: No Congestion Interval

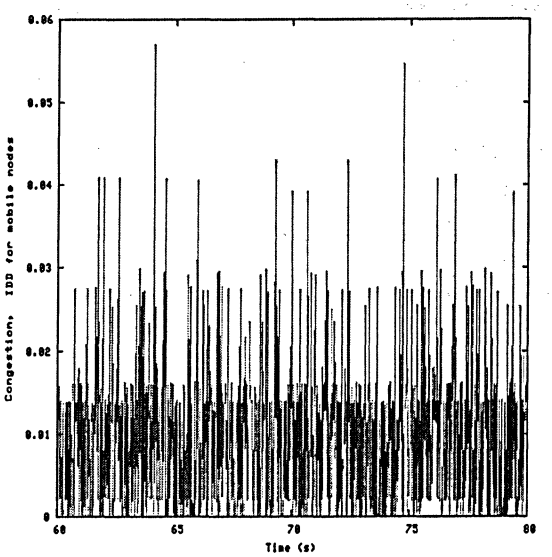

Figure.3.7: Congestion Interval

DSR: Congestion

CBR traffic is introduced between 60th seconds to 80th seconds. In the previous section of no congestion, we have noticed that this is the period when TCP throughput remains stabilized. But as soon as CBR traffic is introduced into it, TCP throughput suffers degradation as shown in figure 3.8. This CBR traffic created queues at those nodes where TCP packets are forwarded towards their destination. This phenomenon continues until

80^{th} second when CBR traffic stops flowing after this instant route become unstable. This causes throughput to fluctuate. During congestion period, IDD values change drastically as shown in figure 3.10. IDD graph, (refer figure 3.9) during no congestion period is same as IDD metric graph of no congestion interval of previous section (refer 3.2.1.1) as shown in figure 3.6. Mean IDD values from 40^{th} second to 60^{th} second validates the fact that TCP connection is not experiencing any congestion during this interval. But mean value of IDD for congestion period (from 60^{th} second to 80^{th} second) increases to 10 times of the IDD values for no congestion period (from 30^{th} second to 60^{th} second). This increase in mean value of IDD shows the presence of deep congestion during 60^{th} second to 80^{th} second. Two consecutive packets take more time to reach the same destination because of the blockade created on their path. IDD values have large high values (spikes). These values can be observed during interval 68^{th} seconds to 74^{th} second. Mean IDD value is higher for congestion interval than that of no congestion interval.

TCP Packets received = 5679 packets

Mean IDD value (40^{th} second to 60^{th} second) No Congestion Period = 0.0090 s.

Mean IDD value (60^{th} second to 80^{th} second) Congestion Period = 0.081 s.

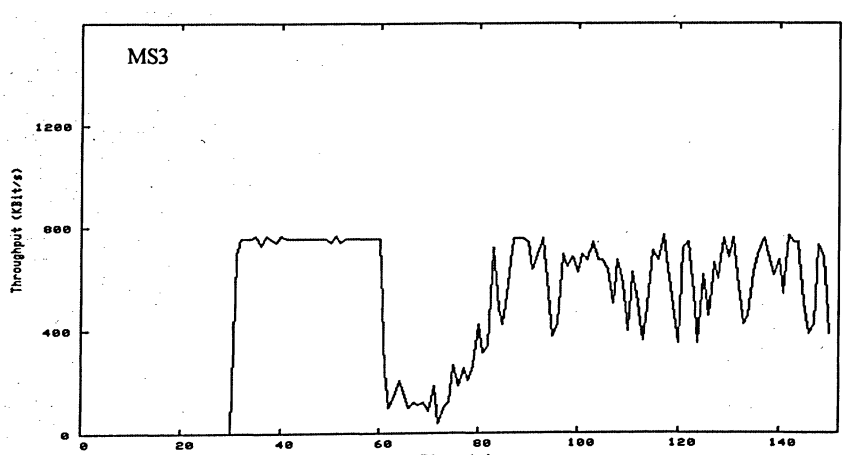

Figure 3.8: TCP Throughput for DSR with Congestion

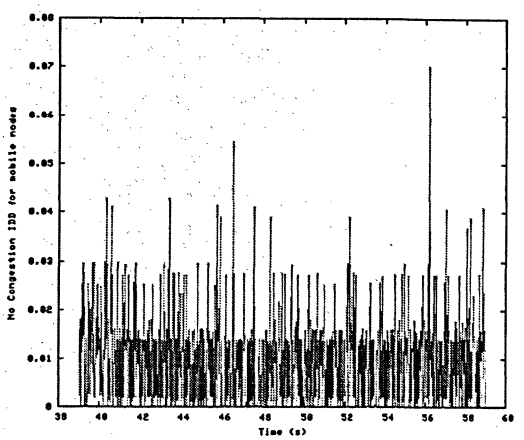

Figure 3.9: No Congestion Interval

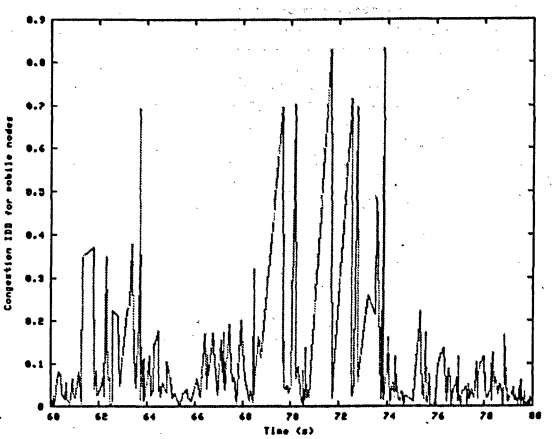

Figure 3.10: Congestion Interval

DSDV: No Congestion

When DSDV is used as the routing protocol, TCP remains stabilized during 25th second to 150th seconds as shown in figure 3.11. Since same mobility model is used for computing TCP throughput, the sender took approximately 15 second to build routing table by starting sending packets at 25th second of simulation whereas TCP started at 10th second of simulation.

TCP Packets received = 7671 packets

Mean IDD value (40th second to 60th second) No Congestion Interval = 0.00976 s.

Mean IDD value (60th second to 80th second) Congestion Interval = 0.00971s.

Figures 3.12 and 3.13 show that Mean values of IDD remain the same for both No Congestion and Congestion period. This indicates that there is no congestion in the network.

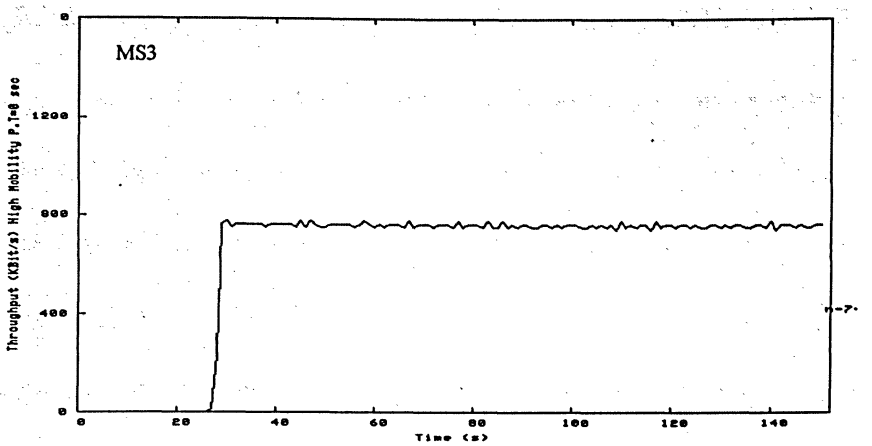

Figure 3.11: TCP Throughput for DSDV

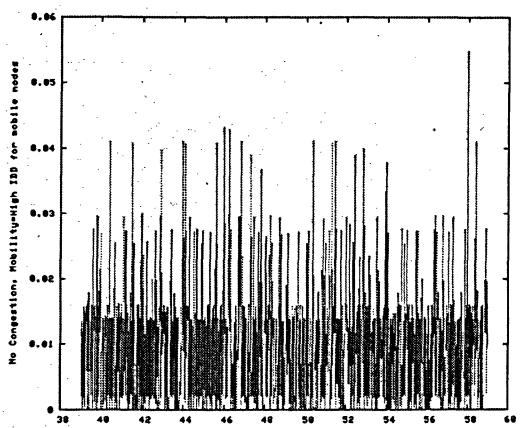

Figure 3.12: No Congestion Interval

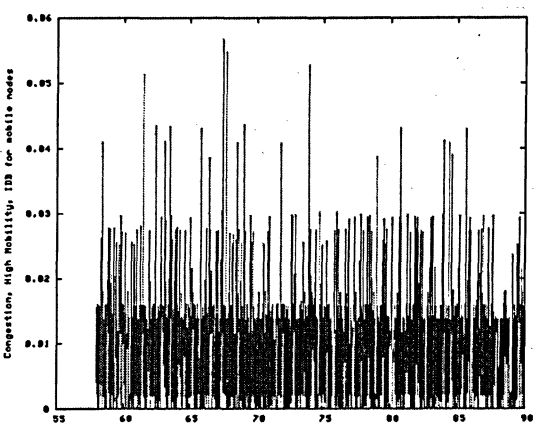

Figure 3.13: Congestion Interval

DSDV: Congestion

We observed that under no congestion conditions (refer previous section) TCP throughput was constant (refer figure 3.11) during the whole connection period. We, now, introduce CBR traffic between 60th second and 80th second of simulation time, TCP throughput decrease during this period as shown in figure 16. TCP and CBR traffics are sharing the same path. Mean IDD value for congestion period (60th second to 80th second). This happens because both grow ten fold from its value in no congestion period (38th second- 58th second). IDD values are high during congestion period that is why the scale along y-axis has been changed.

TCP packets are being forwarded by those intermediate nodes which are also forwarding CBR packets. CBR traffic is high in volume which developed long queues at these nodes. TCP packet and their respective acknowledgement are delayed at these queues. IDD values for congestion period are shown graphically in figure 3.16.

TCP Packets received = 6788 packets

Mean IDD value (40^{th} second to 60^{th} second) No Congestion Interval = 0.00976 s.

Mean IDD value (60^{th} second to 80^{th} second) Congestion Interval = 0.043 s.

Full path intersection of TCP and CBR traffic when DSDV is used as routing protocol decreases TCP throughput by 11.5% of no congestion scenario.

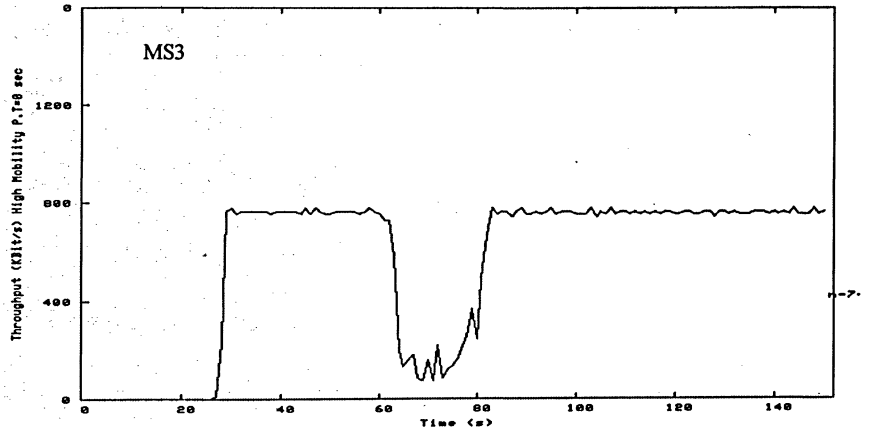

Figure 3.14: TCP Throughput for DSDV with Congestion

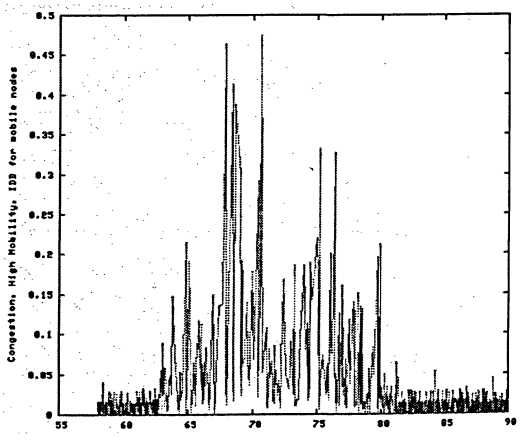

Figure 3.15: No Congestion Interval

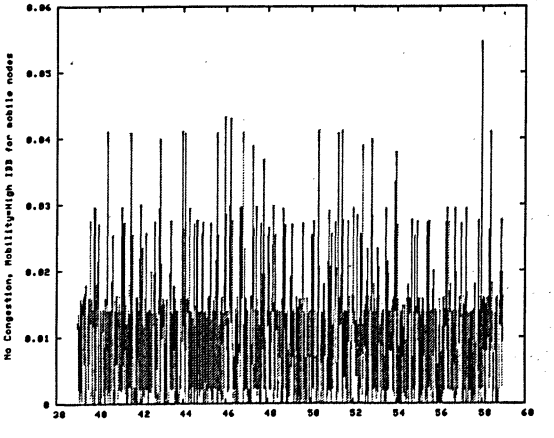

Figure 3.16: Congestion Interval

3.2.2 Partial Path Intersections

The phenomenon when TCP and CBR traffic share the same path for some time during congested period is termed as Partial Path Intersection. TCP sender sends the data packets to the receiver through a path which is shared by CBR packets. Node mobility in the network changes the status of congested node. In this case, CBR and TCP traffic were originally sharing the congested node. The path of either one or both changes due to the movement of some nodes along the path including the congested node causing the congested node no longer lying on the intersection of the two paths. This results in releasing congestion at the previously congested node. This shows that the traffic congestion in an ad hoc network may dissipate because of node mobility.

DSR: No Congestion

We observe that TCP connection starts at 10^{th} second after the simulation begins. At this instant, the sender invokes route discovery mechanism that computes route for TCP data packets to their destination. But because of high mobility, there is TCP transmission for very short period. At approximately 47^{th} second, the TCP sender computes route and starts transmitting data packets. The route change occurs again at 78^{th} second when throughput achieves its peak. The path remains stable for the next 40^{th} seconds indicating DSR gets a route through those nodes which respond route request packets from their caches. The topology does not change until about 120^{th} second. After that the increasing mobility of nodes creates fluctuations in TCP throughput. These fluctuations are mainly caused by the delay in route computation as neighboring node take time to reply the route request packets initiated by the TCP sender. Inter Packet Delay Differences are calculated for two Intervals as shown in the figure 3.18 and figure 3.19. Mean IDD values for both of these intervals are following:

TCP Packets received = 4823 Packets

Mean IDD value (50^{th} second-70^{th} second) No Congestion interval = 0.028 seconds

Mean IDD value (70^{th} second -80^{th} second) Congestion Interval = 0.0279 seconds

Both of these mean values are approximately same. These values indicate that TCP traffic is not experiencing any congestion on its route to its destination as CBR traffic has not been introduced.

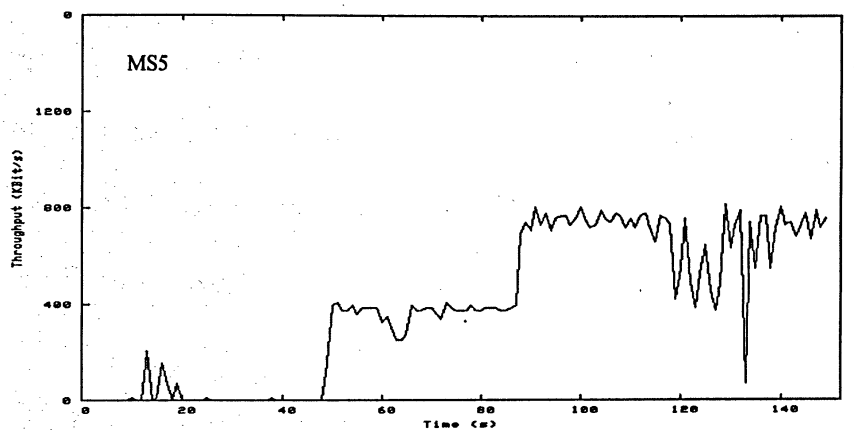

Figure 3.17: TCP Throughput for DSR

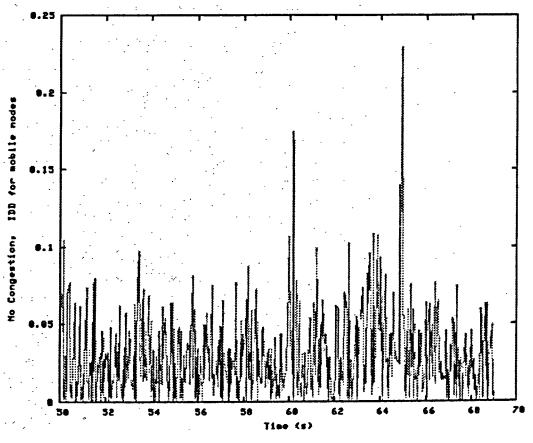

Figure 3.18: No Congestion Interval

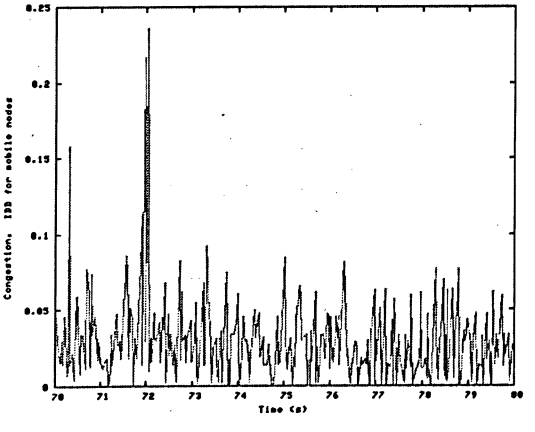

Figure 3.19: Congestion Interval

DSR: Congestion

We observe that TCP traffic started at 47th second of simulation time as shown in figure 3.20. CBR traffic flows between node 3 and 12 for 60th -80th second interval. The sender computes TCP traffic through that node which is not used by CBR until 72nd second. IDD value increases at this instant as shown in figure 3.22. This shows that the sender has computed a route that share same node with the CBR traffic. As soon as both traffics start sharing the same node, the TCP throughput decreases. This phenomenon lasts until node 3 stops transmitting CBR traffic. Mean values of IDD for this interval is same as that of previous one. This shows that TCP is not experiencing any congestion during this interval (50th second-70second). IDD values are graphically represented in figure 3.22 for Congestion case (interval 70 second to 80 second). Increases in IDD values are noticeable in the figure as these become 3.3 when there is maximum congestion.

TCP Packets received = 4765 Packets

Mean IDD value (50th second-70th second) No Congestion Interval = 0.028 seconds

Mean IDD value (70th second -80th second) Congestion Interval = 0.1771 seconds

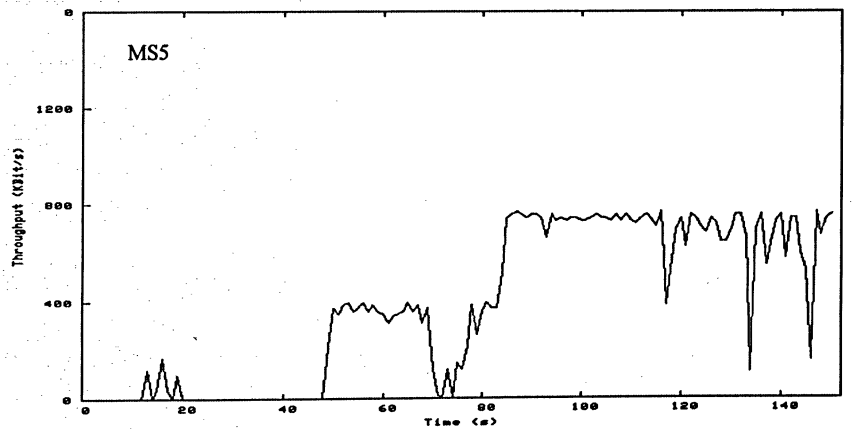

Figure 3.20: TCP Throughput for DSR With Congestion.

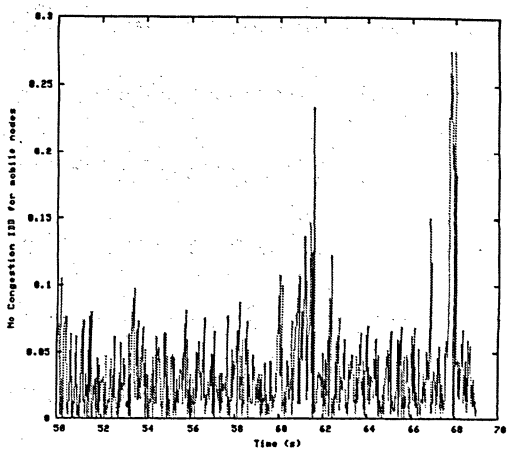

Figure 3.21 .No Congestion Interval

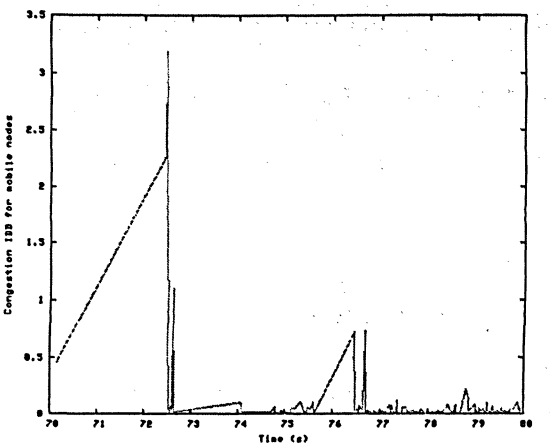

Figure 3.22: Congestion Interval.

Mean IDD value for this period of time is 0.11717 which is 4.2 times of mean IDD value of no congestion case. TCP Packet Received decreases by 1.2 % whereas for Full Path Intersection TCP Packet received decreases by 15.37 % (refer Table 3.2).

DSDV: No Congestion

The sender takes 46 seconds to develop routing tables for the network topology because it Starts transmitting TCP packets at 56^{th} second of simulation time as shown in figure 3.23. At 82nd second, the update packets from the neighboring nodes enable the TCP sender to compute a shorter route through which it attains its peak throughput. DSDV path gives the TCP sender stable throughput till the end of connection. Throughput fluctuations have been observed in case of DSR during interval 120^{th} second to 150^{th} second. The node mobility causes DSR to compute route every time it needs to send data thus route discovery mechanism creates these fluctuation. But in case of DSDV, the nodes promptly report route changes by incremental packets. Hence routing information is incorporated into routing table much faster that makes the TCP throughput more stable. We change time interval for which IDD values are to be recorded. Since TCP starts communicating at 53^{rd} second of simulation, no congestion interval has been change to 50^{th} second. In order to observe the effect of CBR, congestion period (60^{th} second to

80th second) remains unchanged. Figures 3.24 and 3.25 provide graphical representation of IDD values for both of these periods selected. Following are mean values recorded for IDD during congestion and no congestion intervals.

TCP Packets received = 5113 Packets

Mean IDD value (53rd second - 60th second) No Congestion Interval = 0.02 s

Mean IDD value (60th second - 80th second) Congestion Interval = 0.023 s

It has been observed that Mean IDD values are approximately same. TCP traffic is not experiencing any congestion during its connection time as CBR traffic is not flowing.

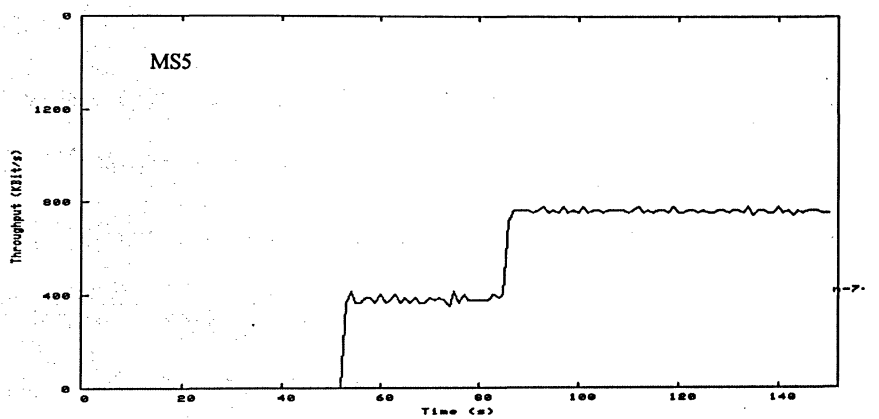

Figure 3.23: TCP Throughput for DSDV

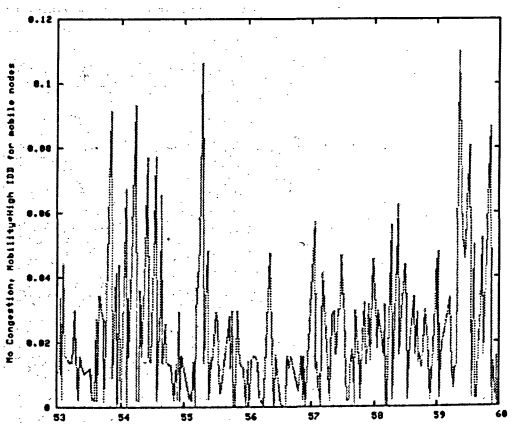

Figure 3 24: Congestion Interval

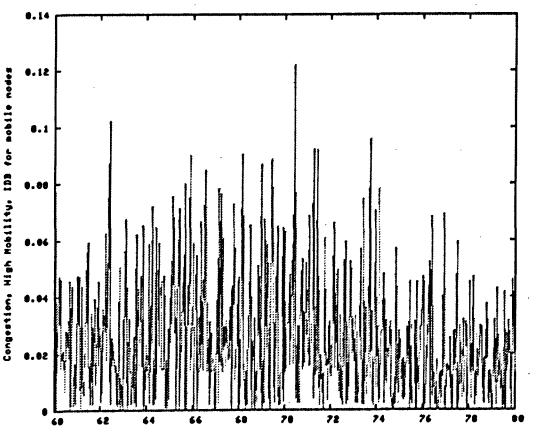

Figure 3.25: Congestion Interval.

DSDV: Congestion

CBR traffic is introduced in the ad hoc network when the node 3 starts sending constant

Bit rate packets to the node 12 between intervals 60th second to 80th second. TCP traffic is affected by the CBR traffic from interval 62nd second to 69th second as shown in figure 3.26. IDD values become 1.2 to 1.4 showing deep congestion at those nodes which are both forwarding TCP and CBR traffic. But the node mobility causes the TCP traffic to get separated from the CBR traffic. Routing information reported by incremental and full dump packets in response to network topological changes enable the TCP sender to compute a route through those nodes which are not being used by CBR traffic. That is why the TCP throughput regains its initial rate as node mobility causes congestion to dissipate.

IDD values are calculated for two intervals of time as mentioned in (No Congestion DSDV case). These values are represented graphically in figures 3.27 and 3.28. IDD values jump 1.2 to 1.4 showing deep congestion (refer figure 3.28) at those nodes which are forwarding TCP and CBR traffic.

TCP Packets received = 4846 Packets

Mean IDD value (53rd second - 60th second) No Congestion Interval = 0.02 second

Mean IDD value (60th second - 80th second) Congestion Interval = 0.036 second

Mean IDD value from interval (53rd second-60th second) is 0.0199 which is same value observed in previous no congestion case (refer previous section). Since there is no CBR traffic during this interval that is why mean IDD value is the same as that of previous one. CBR traffic started flowing during interval 60th second to 80th second which created congestion for TCP throughput. Mean IDD value recorded (0.036 s) was higher than that of no congestion case.

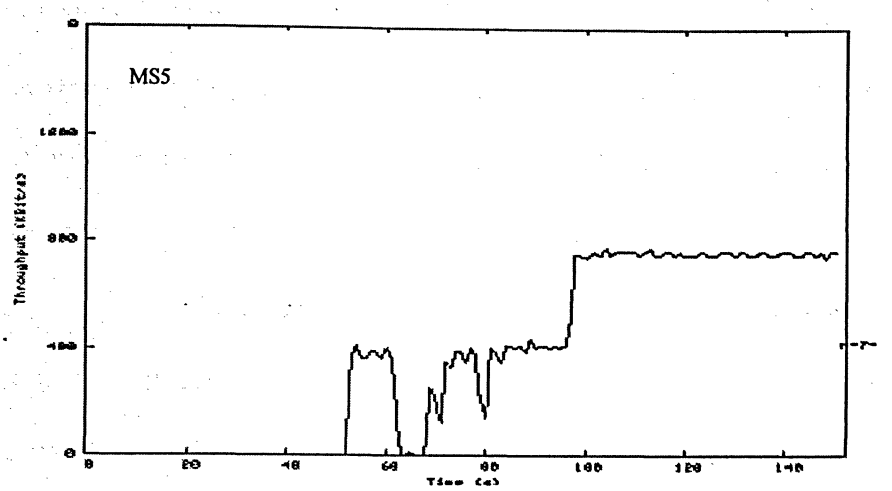

Figure 3.26: TCP Throughput for DSDV with Congestion.

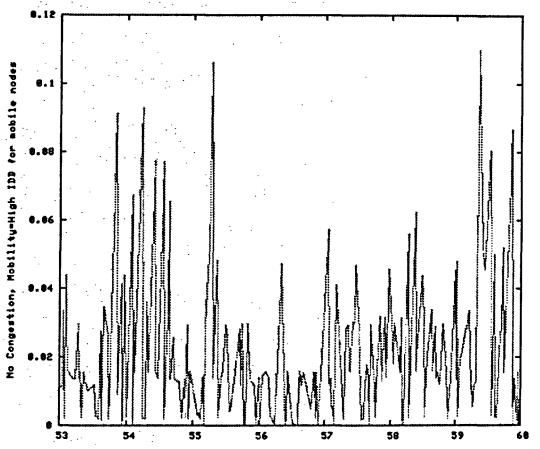

Figure 3.27:No Congestion Interval

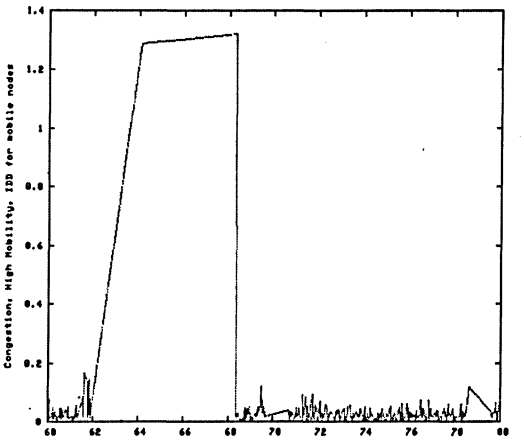

Figure 3.28: Congestion Intervals

3.2.3 No Path Intersection

In this scenario (refer MS4), TCP traffic was not being affected by CBR traffic which was introduced into the ad hoc network between interval 60th second to 80 second. Both these traffics were not sharing the same node that is why CBR traffic could not create congestion at nodes. The TCP throughput is not being affected by CBR traffic. This scenario shows that in ad hoc network, node mobility may avoid congestion at all even in the presence of highly congesting traffic.

DSR: No Congestion

In this scenario, TCP throughput fluctuations shown in figure: 3.29 during interval 10th seconds to 60th second show route instability during this period. We know that incase of DSR, TCP sender computes new route each time it needs to send packet. Node mobility is changing network topologies at a rate faster than the rate with which routing mechanism of DSR computes route for destination of TCP packet. TCP throughput remains stabilized during interval 60th second to 90th second. After this interval, the node mobility causes the TCP through to change. Mean IDD values remain the same for both no congestion and congestion period.

TCP Throughput = 6399 packets

Mean of IDD value (40th second - 60th second) No Congestion Period = 0.0186 s.

Mean of IDD values (60th second - 80th second) Congestion Period = 0.0185 s.

Mean IDD values remain the same for both no congestion and congestion period though their instantaneous values are different for congestion and no congestion intervals shown in figures 3.30 and 3.31. There is no congesting traffic which can cause bottle neck nodes for TCP traffic.

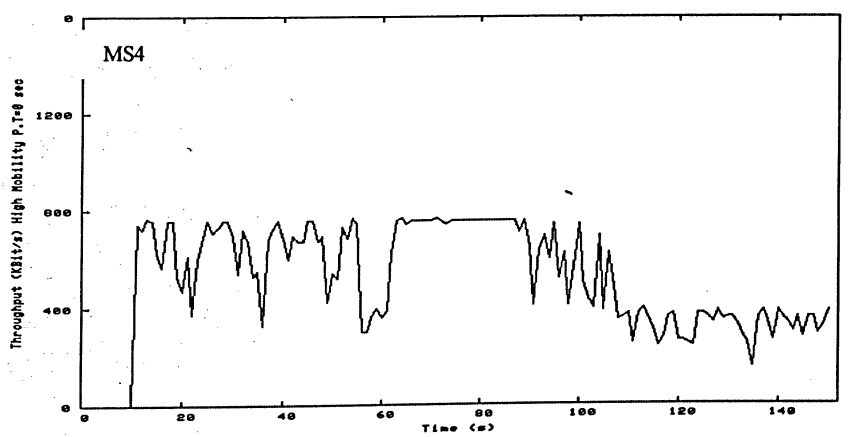

Figure 3.29: TCP Throughput for DSR

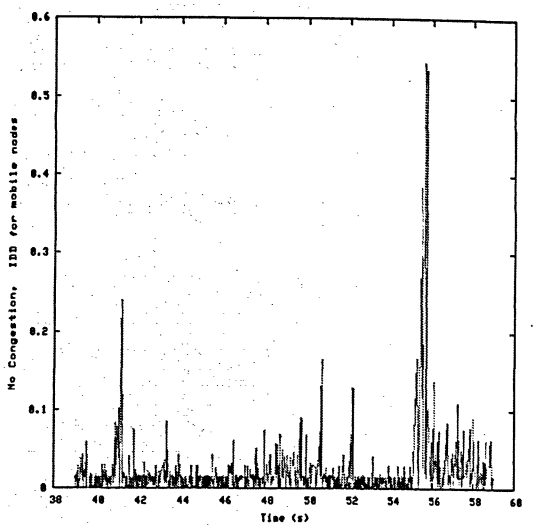

Figure 3.30: No Congestion Interval

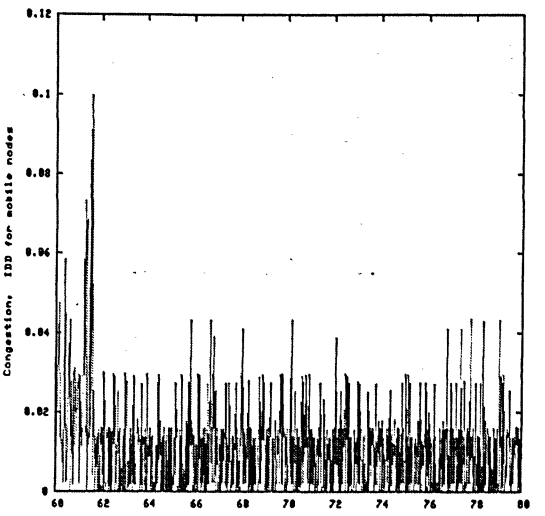

Figure 3.31: Congestion Interval

DSR: Congestion

Congestion is introduced between 60th second and 80th second by starting CBR connection between two nodes. But the effect of this connection is not visible on the TCP throughput graph as shown in figure 3.32. This shows that TCP communicating nodes are using those routes and intermediate nodes which are not being used by CBR traffic that is why there is no difference among graphical observations made during Congestion and previous No Congestion conditions. IDD values remain the same for both congestion and no congestion interval even though CBR traffic is flowing between node 3 and node 12. Throughput remains the same for both congestion and no congestion cases.

TCP Throughput = 6399 packets

Mean of IDD value (40th second - 60th second) No Congestion Interval = 0.0186 s.

Mean of IDD value (60th second - 80th second) Congestion Interval = 0.0185 s.

Number of TCP packets received and IDD values for Congestion case are same as those of No Congestion case (refer previous section). This shows that TCP traffic remains unaffected even after the initiation of congestion creating CBR heavy traffic.

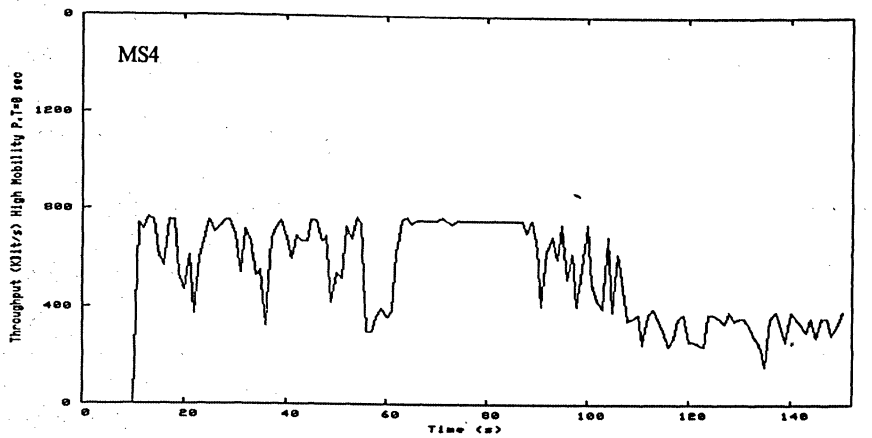

Figure 3.32: TCP Throughput for DSR

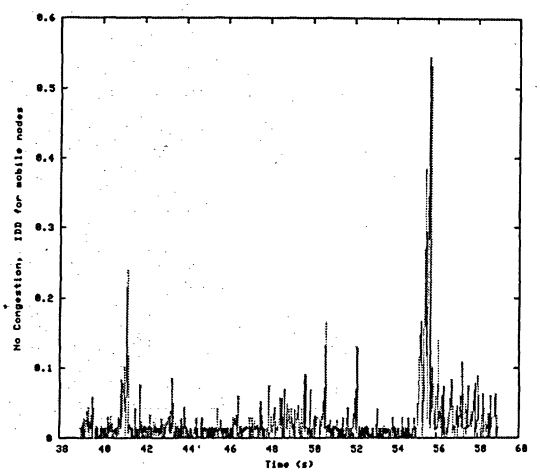

Figure 3.33: No Congestion Interval

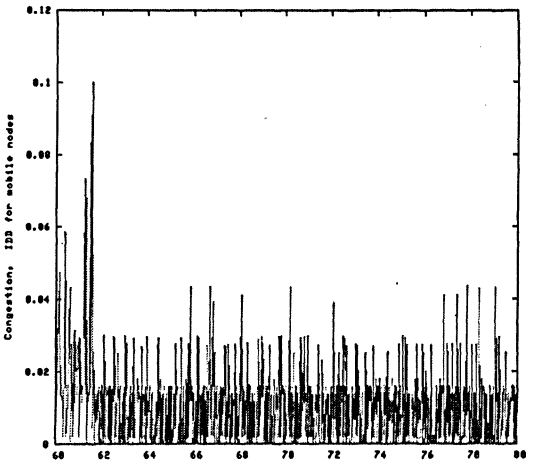

Figure 3.34: Congestion Interval

DSDV: No Congestion

The node 0 sets up TCP connection with node 7 during time interval 10th second to 58th second as shown in figure 3.35. During this interval, the TCP Through remains stabilized. There is no TCP transmission between 58th second and 62nd second. This interval can be network partitioning but since DSR has throughput during this period therefore the zero throughput is because of DSDV routing outage. But after 62nd second, the sender immediately regains its throughput. The throughput decreases slightly 88th second. IDD values for no congestion and congestion period shown in figure 3.36 and figure 3.37.

Both graphs have similar patterns for these mentioned periods. This shows that there is no congestion during interval. IDD values for no congestion and congestion period are shown in figures 36th and 37th respectively. Both graphs have similar pattern for these mentioned periods. Mean IDD values are also the same for both of these periods. This shows that there is no congestion during interval.

TCP Throughput = 5489 packets

Mean of IDD value (40th second - 60th second) No Congestion Interval = 0.0103 s.

Mean of IDD value (60th second - 80th second) Congestion Interval = 0.0109 s.

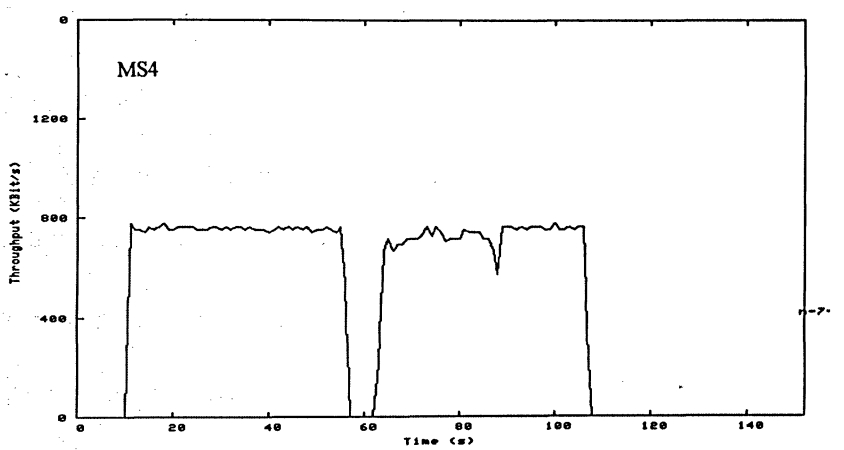

Figure 3.35: TCP Throughput for DSDV

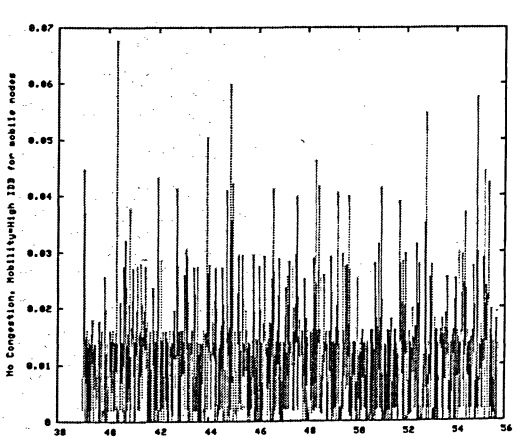

Figure 3.36: No Congestion Interval

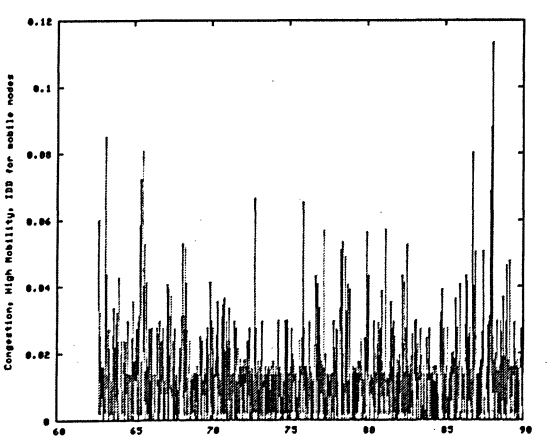

Figure 3.37: Congestion Interval

DSDV: Congestion

In this case, the TCP throughput remains unaffected even after the introduction of CBR traffic flowing in the ad hoc environment (refer figure: 3.38). The node 3 starts sending CBR connection with node 12 between time interval 60 second and 80 second interval. But TCP throughput remains unaffected by the CBR traffic as shown in figure 3.38. It has been observed in previous congestion cases that TCP throughput suffered degradation whenever CBR traffic is introduced into network, the TCP through put decreases during this interval. This congestion is shown by high IDD value. But in this scenario, TCP sender routes packets which are not being used by CBR traffic.

TCP Throughput = 5489 packets

Mean of IDD value (40^{th} second - 60^{th} second) No Congestion Interval = 0.0103 s.

Mean of IDD value (60^{th} second - 80^{th} second) Congestion Period = 0.0109 s.

Throughput and mean IDD values of both intervals remain the same for congestion and No congestion scenarios. IDD values are graphically represented in figures 3.39 and 3.40. It means that the routing protocol has computed the route for TCP traffic flow which is not affected by CBR traffic. Hence, node mobility in an ad hoc Network can prevent congestion.

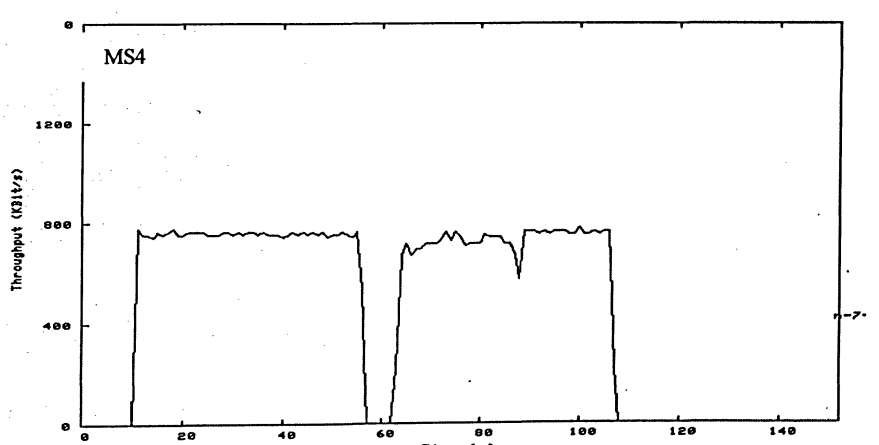

Figure 3.38: TCP Throughput for DSDV with Congestion

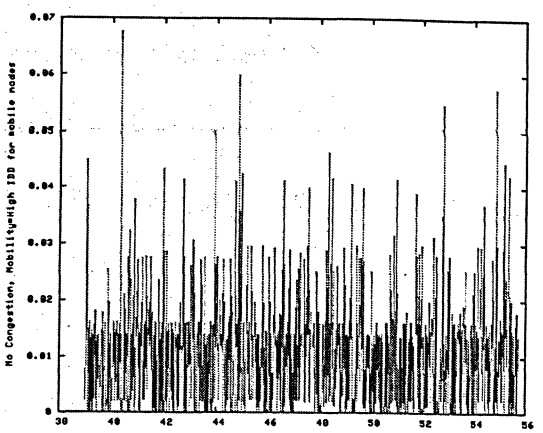

Figure 3.39: No Congestion Interval

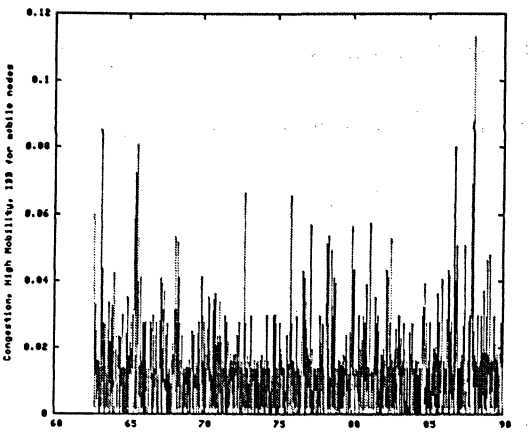

Figure 3.40: Congestion Interval

3.2.4 Discussion of Results: Congestion

TCP packets received and mean IDD value for congestion and no congestion conditions are tabulated in Table 3.2. Full Path Intersection of TCP and CBR traffic decreases TCP packet received by 15.37% of no congestion scenario when DSR is used as routing protocol as shown in Table 3.2. But incase of DSDV, the decrease in TCP packets received is (11.51%) less than TCP packets received for DSR. Mean IDD value for congestion interval is 9 folds of that of no congestion period for DSR whereas mean IDD value for DSDV is 5 times of no congestion interval. This shows that congestion is worse for DSR than DSDV.

In Table 3.4, we provide TCP Packets received and mean of IDD values during congestion and no congestion interval for Partial Path Intersection. TCP Throughput in case of DSR decreases by 1.2% whereas for DSDV the decrease is 11.5%. These decrements in throughput are less than those of Full Path Intersection. We notice that the degree of congestion indicated by mean IDD values is less for Partial Path Intersection than those of Full Path Intersection.

In case of No Path Intersection, it is evident from values recorded in Table 3.4 that TCP throughput and mean IDD values remain constant before and after introduction of CBR traffic in the ad hoc network. CBR traffic does not create congestion points on the TCP flow. Both traffic flows are following separated paths.

Table 3.2: Full Path Intersection of TCP and CBR flows for DSR and DSDV

Mobile Scenario MS3	DSR			DSDV		
	No Congestion Condition	Congestion Condition	Remarks	No Congestion Condition	Congestion Condition	Remarks
Packets Received	6711 Packets	5679 Packets	15.37% Decrease	7671 Packets	6788 Packets	11.51% Decrease
Mean IDD value (No Congestion Interval)	0.0089 second.	0.009 second.	Constant	0.009 second.	0.009 second.	Constant
Mean IDD value (Congestion Interval)	0.009 second.	0.0813 second	9 times of No congestion interval	0.009 second.	0.043 second.	4.7 times of No Congestion interval

Table 3.3: Partial Path Intersection of TCP and CBR flows for DSR and DSDV

Mobile Scenario (MS5)	DSR			DSDV		
	No Congestion Condition	Congestion Condition	Remarks	No Congestion Condition	Congestion Condition	Remarks
Packets Received	4823 Packets	4765 Packets	1.2%	5113 Packets	4846 Packets	Constant
Mean IDD value (No Congestion Interval)	0.028 second	0.029 second	Constant	0.028 second.	0.029 second	Constant
Mean IDD value (Congestion Interval)	0.029 second	0.1171 second	4.2 times of No Congestion interval	0.0279 second	0.036 second	1.2 times of No congestion Interval

Table 3.4: No Path Intersection of TCP and CBR flows for DSR and DSDV

Mobile Scenario (MS4)	DSR			DSDV		
	No Congestion Condition	Congestion Condition	Remarks	No Congestion Condition	Congestion Condition	Remarks
Packets Received	6399 Packets	6399 Packets	Constant	5489 Packets	5489 Packets	Constant
Mean IDD value (No Congestion Interval)	0.0186 second.	0.0186 second	Constant	0.0103 second.	0.0103 second.	Constant
Mean IDD value (Congestion Interval)	0.0186 second	0.0186 second	Constant	0.0109 second.	0.0109 second.	Constant

3.3 Channel Error

In order to investigate the effect of channel error on TCP performance in mobile ad hoc environment, we incorporated error model into our simulations. This error model, provided in ns2, simulates link-level error or loss by either marking the packet's error flag or dumping the packet. The packets are dropped randomly. Random variable generating errors is uniformly distributed from 0 to 1. We selected packet error rate as 3 percent. We record TCP throughput by taking DSR and DSDV as routing protocols and using the same five scenarios generated by Random Way Mobility Model as in case of Congestion. These simulations executed are termed as "No Channel Error Conditions". We incorporated error model into our simulations and recorded TCP throughput using the same parameters. These conditions are termed as "Channel Error Conditions".

TCP throughputs for five mobile scenarios using DSR as routing protocol are recorded. These conditions are termed as "No channel error conditions" are shown in figure 3.41. We present TCP throughput using same routing protocol but under the influence of "channel error" in figure 3.42.

TCP throughputs using DSDV as routing protocol under "No Channel Error" and "Channel Error" conditions respectively are exhibited in figures 3.43 and 3.44.Out of these five scenarios, scenario MS3 is considered as the best example for investigating the effects of channel/medium errors. Metrics packets out of order and packet losses are calculated for this scenario (MS3) for both protocols.

DSR

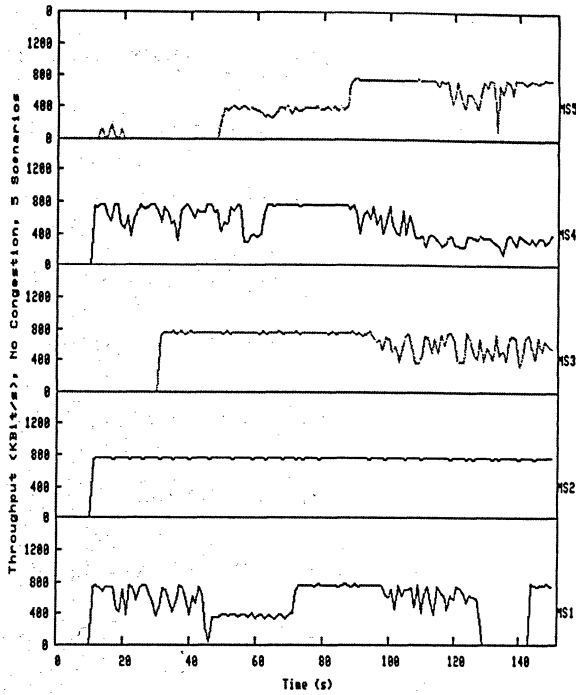

Figure 3.41: TCP Throughput with No Channel Error.

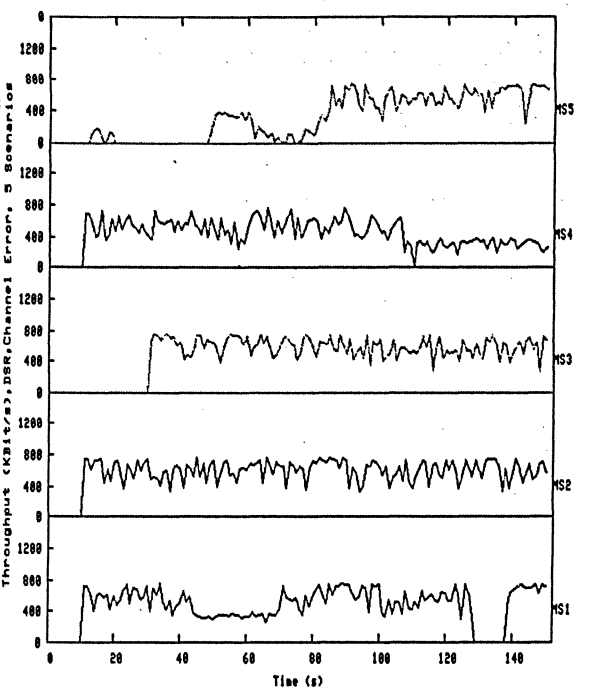

Figure 3.42: TCP Throughput with Channel Error.

DSDV

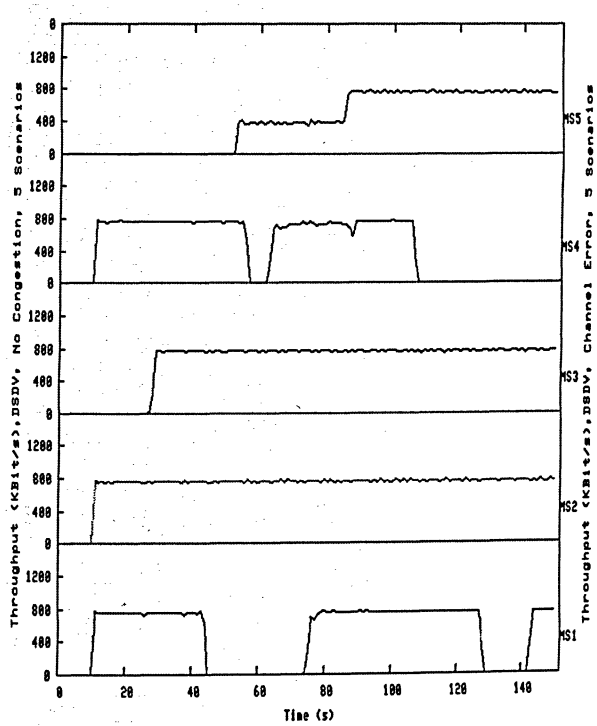

Figure 3.43: TCP Throughput with No Channel Error

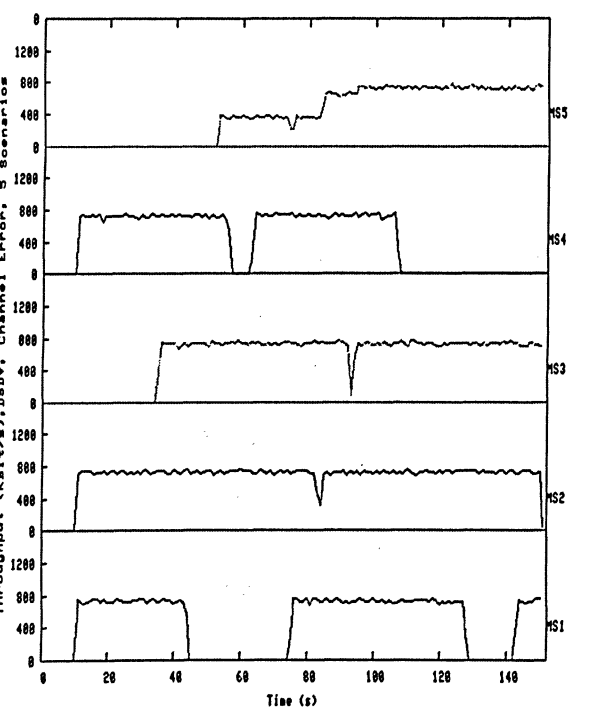

Figure 3.44: TCP Throughput with Channel Error.

3.3.1 DSR: No Channel Error

In this scenario (refer MS3), the nodes routing TCP packets are having route stability from 10^{th} second to 90^{th} second of simulation time as shown in figure 3.45. During this time, packets are forwarded over the stable route. After 90^{th} second, node mobility is reflected by frequent changes in the TCP throughput. DSR computes routes by invoking route discovery mechanism which causes data packets to travel through multiple routes before reaching their destination. Packets traveling through these routes reach destination out of sequence. Figure 3.47 shows packets out of order after 97^{th} second of simulation time. This is the instant after which TCP throughput experiences instability. The routes of the packets are changing continuously which is represented by Packets out of Order graph. These out of order packets reach node 7 are shown during period between 97^{th} second and 150^{th} second in figure 47.

TCP Packets delivered at Receiver = 6711 packets.

Packets Dropped = 1 packets.

Packets out of order = 51 packets.

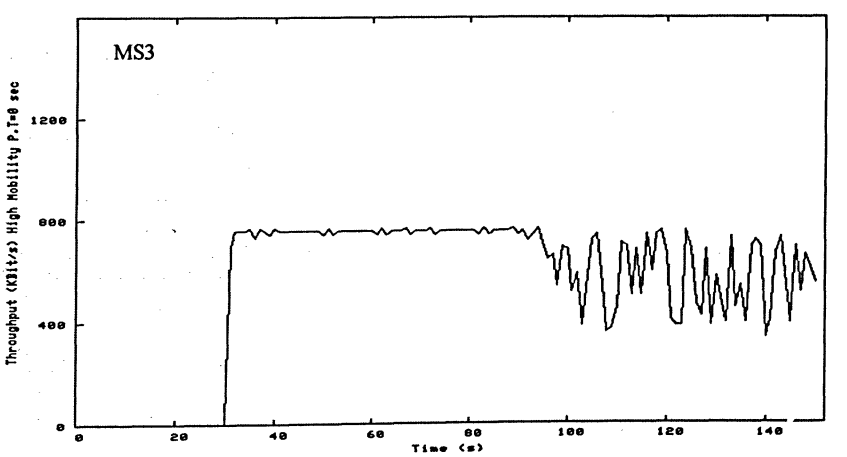

Figure 3.45: TCP Throughput

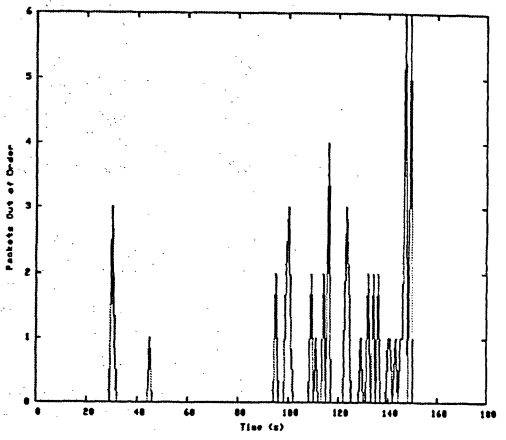

Figure 3.47: Packets Out of Order

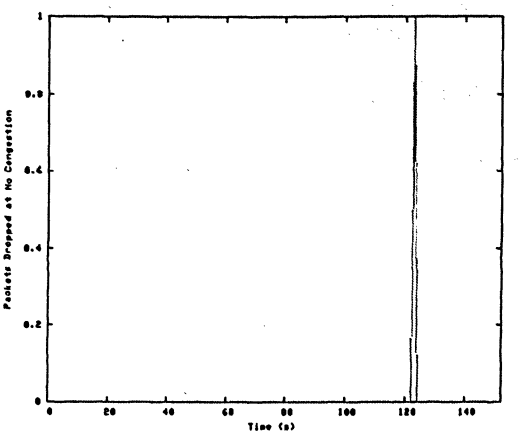

Figure 3.46: Packets Dropped

3.3.2 DSR: Channel Error

Channel error model is incorporated in the simulations. As soon as the TCP connection is set up between TCP sender and receiver, channel error model starts introducing packet error. The 802.11 MAC use acknowledgements in order to provide early detection and retransmission of corrupted packets. In this case, route maintenance can be easily provided, since at each hop, the host transmitting the packet can determine if that hop is still working. If the data link level reports a transmission problem for which it cannot recover, this host sends route error packet to the original sender of the packets encountering the error. When a route error packet is received, the hop in error is removed from the host's route cache. All routes which contain this hop must be truncated. Every time the TCP sender invokes route discovery mechanism the TCP throughput fluctuates. In the absence of channel error (refer previous case), packets out of order are not observed in figure 3.50 during interval 42nd to 95th second of simulation time. But in the presence of channel error, refer figure 49 these packets reaching destination out of sequence are observed during interval 42nd second to 95th second. This is because of this the routes stored in the caches of the neighboring nodes become stale as neighboring nodes do not get fresh routes as a result of channel errors as mentioned earlier. Every

time TCP sender computes route to the destination through different set of nodes. Data packets travel through different routes to reach the destination. Hence, the channel Error creates packets out of sequence. The number of out of sequence packet is enhanced when network topology changes because of node movement as this increase during interval 97 second to 150th second. Channel Error both contribute in incrementing out of sequence packets. Errors in the medium cause the packets on the fly to their destination to be dropped. Packets dropped increases by

TCP Packets received = 5912 Packets

Packets Dropped = 14 packets.

Packets out of order = 199 packets.

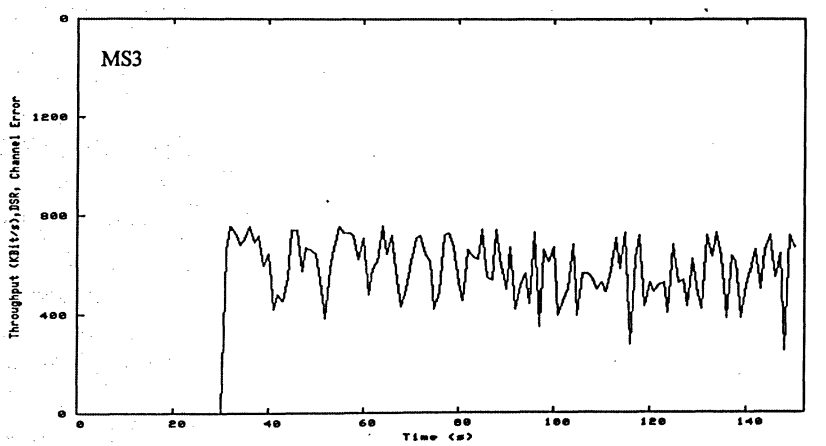

Figure 3.48: TCP Throughput

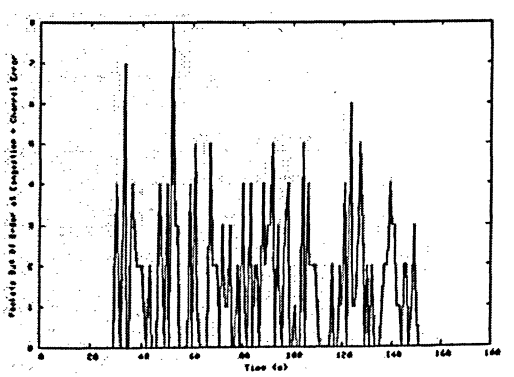

Figure 3.50: Packets Out of Order

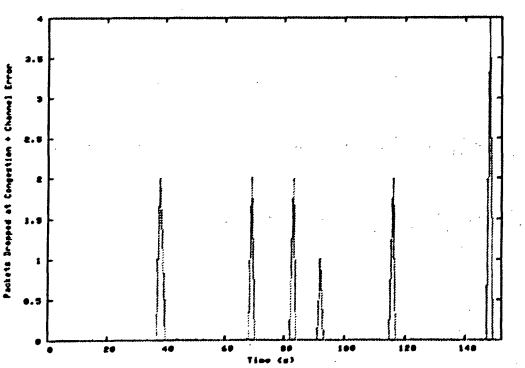

Figure 3.49: Packets Dropped

3.3.3 DSDV: No Channel Error

The TCP throughput is stabilized when DSDV is used as routing protocol as shown in figure 3.51. During interval between 30^{th} seconds and 95^{th} second of simulation time, there is no node mobility. We know that DSDV uses incremental packets and full dump for updates. Routes are reported by these updating packets. Packets are routed through the same routes as these updating packets report route without change of metric. Hence, old routes do not stale when there is no node movement that is why packets keep on following same paths. After 95 seconds, the network undergoes topological changes because of node mobility. The elements in the routing table of each mobile node change dynamically to keep consistency with dynamically changing topology of an ad hoc network. To reach this consistency, the routing information advertisement must be frequent or quick enough to ensure that each mobile node can always locate all the other mobile nodes in the dynamic ad hoc network. These changes in the network topology are reported by incremental packets. Routing table contains these updated routes with corresponding metric and sequence number. Upon the updated routing information, each node has to relay data packet to other nodes upon request in the dynamically created ad hoc network [6]. Packets out of order are created because of availability of multiple paths from source to the destination. These packets follow different paths because each time that route is selected by the packet which has newer sequence number or better metric than the previous route. Packets reaching node 7 (destination/receiver node) out of sequence are shown in figure 3.53. These packets are observed during node mobility period i.e. 95^{th} second to 150^{th} second of simulation time.

TCP packet delivered at Receiver = 7671 Packets

Packets Dropped = 31 packets

Packets out of order = 33 packets

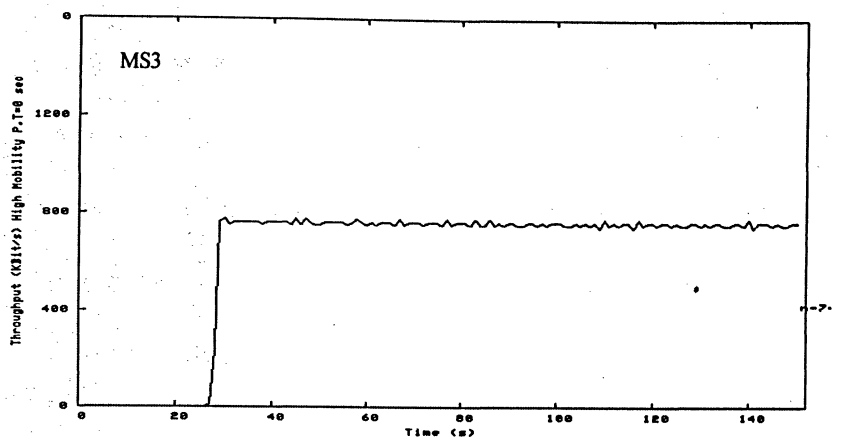

Figure 3.51: TCP Throughput

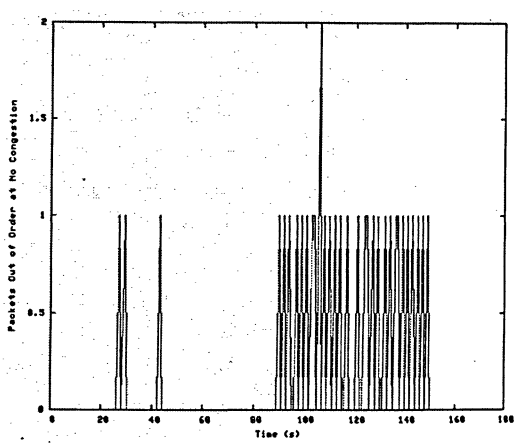

Figure 3.53: Packets Out of Order

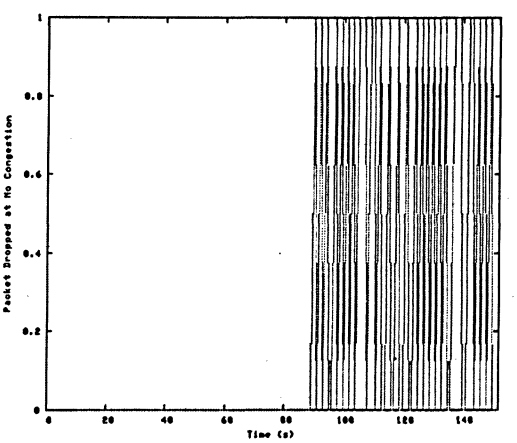

Figure 3.52: Packets Dropped

3.3.4 DSDV: Channel Error

When channel error is introduced, TCP throughput undergoes slight variations. Packets out of sequence are observed in figure 3.56 right from the start of TCP connection. The number of these packets increases after 97th second of simulation time as node mobility also contributes packets out of order. In DSDV, we know that [7]

1) Routes with older sequence are discarded and routes with new sequence numbers are preferred.

2) A route with a sequence number equal to that of an existing route is preferred if it has better metric and the existing route is discarded or stored as less preferable.

Metric of routes are changed by the channel error. Routes with better metric and newer sequence number are computed in order to route packets. Hence, channel error changes metric of routes. Change in metric makes TCP sender to select different routes for packets to forward to the next hop. Packets traversing different routes cause packets to reach destination out of sequence. So, these out of order packets are observed in figure 3.56. After 95th second of simulation time, node mobility further increases the packets out of order as shown in figure 3.56. Channel error causes packets to be dropped as shown in figure 3.55.

TCP packet delivered at Receiver = 6872 Packets

Packets Dropped = 118 Packets

Packets out of order = 112 Packets

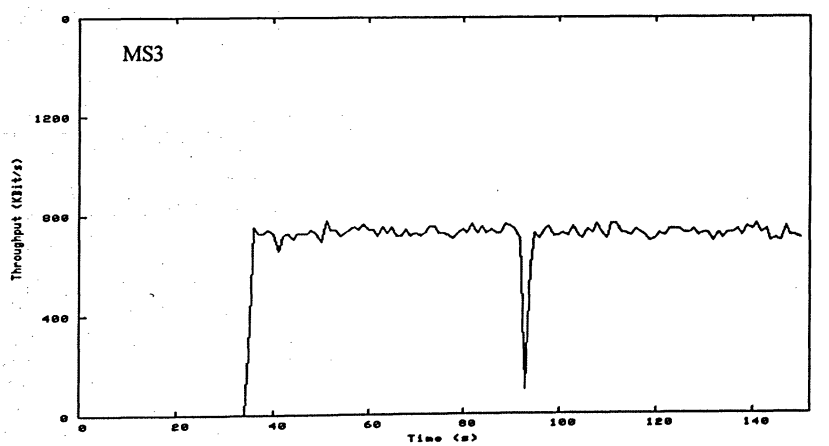

Figure 3.54: TCP Throughput

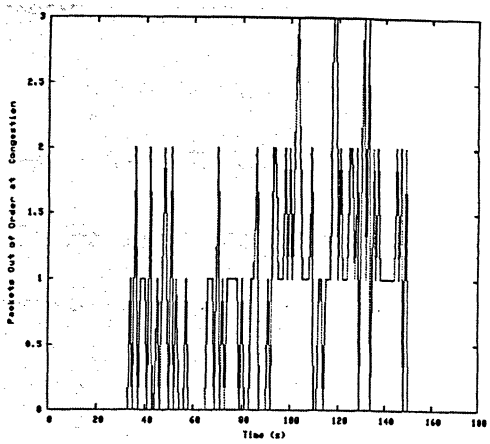

Figure 3.56: Packets Out of Order

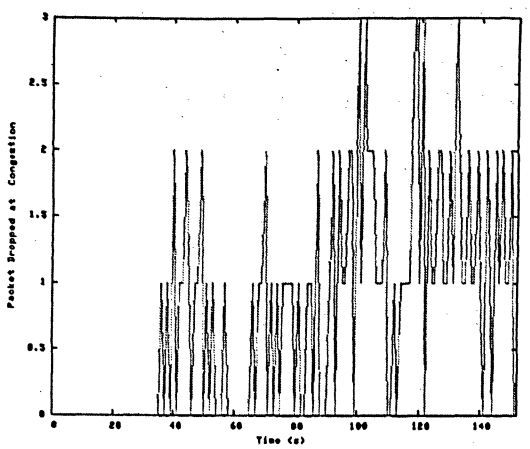

Figure 3.55: Packets Dropped

3.3.5 Discussion of Results: Channel Error

We observe that when we use DSR as routing protocol, TCP throughput decreases by 11.9% in the presence of channel error and packet out of order ratio becomes 4 folds and packets drop increase by 12 times of those at no channel error condition as shown in Table 3.6. For DSDV, channel error decreases TCP packets received by 10.41% and increases packet losses by 3.8 times and packets out of order by 3.4 times.

It can be noticed that packet losses for DSR are less than packets dropped incase of DSDV under the influence of Channel Error Model. But packets out of order ratio is higher for DSR than that of DSDV. This because of the fact that whenever packet is to be sent by a node, it first consults its cache, if it cannot find route in cache then it broadcasts route request packet by invoking Route Discovery Mechanism. The route reply packets refresh routes in the cache of nodes under normal conditions. But channel errors corrupt these route reply packets. Routes in the caches of nodes are not refreshed. So, nodes cannot use a stale route that is why whenever the node needs to send the data packet, it has to invoke Route Discovery mechanism. Hence, a node computes different routes for every packet. This causes high packet out of order ratio for DSR. But in case of

DSDV, we observe that route is computed once for forwarding of data packets on the basis of least metric or latest sequence number [6]. That is why we get less packet out of order ratio for DSDV than that in case of DSR.

Comparison of packet loss ratios shows that DSR happens to be more reliable than DSDV. In DSR, node computes route for forwarding of every data packet whereas in case of DSDV routing tables are refreshed by routing advertisements called increment packets. The channel errors corrupt these advertisement packets. The routing table entries are not properly refreshed by these advertisements. This situation leads to increase packet losses.

Table 3.5: TCP Packet Received, POOR and PLR for DSR and DSDV

Mobile Scenario (MS3)	DSR			DSDV		
	No Channel Error	**Channel Error**	**Remarks**	**No Channel Error**	**Channel Error**	**Remarks**
TCP Packets Received	6711 Packets	5912 Packets	11.9% Decrease	7671 Packets	6872 Packets	10.41% Decrease
Packet Out of Order Ratio (POOR)	51 Packets 0.76%	199 Packets 3.3%	3.9 times Increase	31 Packets 0.40%	118 Packets 1.7%	3.4 times increase
Packet Loss Ratio (PLR)	1 Packet 0.015%	12 Packets 0.2%	12 times Increase.	33 Packets 0.43%	112 Packets 1.6%	3.8 times increase

3.4 Routing

In order to understand routing in case of DSR and DSDV, 38 simulation runs are executed. Out of these, one mobile scenario (MS1) is selected for studying both of these protocols by using Packets out of Order (POO) and Round Trip Time (RTT) as metrics. We have shown graphical representation of TCP throughput and RTT values of these five scenarios in figure 3.57 and figure 3.58 respectively.

DSR

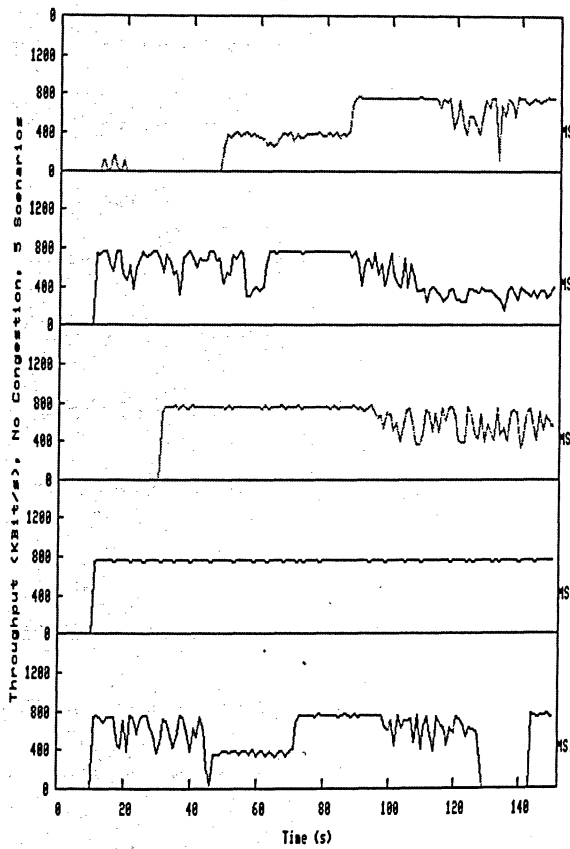

Figure 3.57: TCP Throughput of Five Scenarios Using DSR

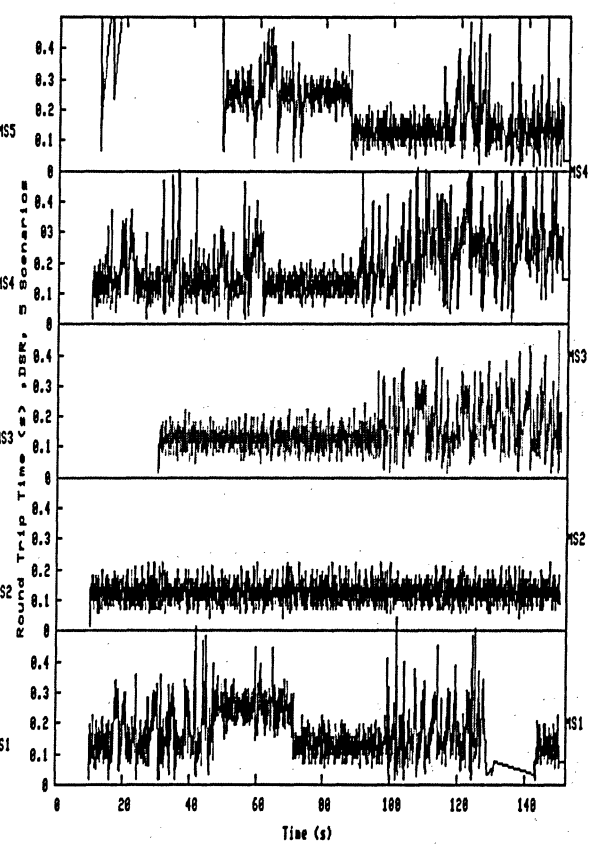

Figure 3.58: RTT of Five Scenarios

3.4.1 DSR: Routing

In mobile scenario (MS1), TCP connection is set up at 10^{th} second of simulation time as shown in figure 3.61. The TCP throughput remains constant for interval 10-18 second of

simulation time. We observe fluctuations in throughput during period 18 – 45 second. Following can be the reasons for fluctuations 1) Packet Drops due to link level Errors. 2) Path changes.

We calculated packet drops taking place during the TCP connection period shown in figure 3.64. We do not observe any packet drop during this interval. The first probable reason can be eliminated. We know that in the presence of high mobility, link failures can happen very frequently. Link failures trigger new route discoveries. This can cause frequency of route discoveries directly proportional to link failures. In order to avoid massive route discoveries, routes learnt through route request or route reply packets are stored in caches. The route discovery is delayed in DSR until all cached routes fail. But with high mobility, the chances of cache routes being stale are quite high. This results in initiating a route discovery. In response to this, a large number of replies with MAC overhead are received. Hence cache staleness and high MAC overhead together result in degradation. High node mobility causes this TCP variation observed in time region 18-45 second as shown in figure 3.61.

Since node mobility causes staleness of route which initiates route discoveries frequently resulting in receiving of large number of route reply packets. Each node learns multiple routes to forward packets. That is why packets are propagated through different paths under high mobility conditions. These packets reach their destination out of sequence. We observe packets out of order during time interval 18^{th} second 45^{th} second shown in figure 3.63. Packets out of order observed in this time region are because of high node mobility.

RTT variations during this time region are observed in figure 3.62. This is because of the fact that path length through which data packets and their respective acknowledgments

propagate is changing. High node mobility triggers route discovery frequently which causes each node to learn multiple routes for forwarding packets to their destination. Therefore, nodes forward these packets through different routes. Different propagation time of each packet along each path results in RTT variations. The TCP throughput decreases during time interval 50^{th} seconds to 70^{th} seconds shown in figure: 3.61. During this interval, we parsed the mobility file (scen-15-test) generated by Random Mobility Model. We concluded that during this interval nodes mobility. Node 9 was moved as shown below:

$ns_ at 61.164485692206 "$node_(9) setdest 274.488808096776 288.704936031864 5.148047437145"

Paper [1] by Charles E. Perkins describes that Nodes usually get clustered with low mobility, an artifact of random mobility model. This leads to network congestion in the certain regions. The possibility of link failure is low with low mobility (refer paper). But Congestion in turn causes link layer feedback to report link failures even when the nodes are relatively static and physical link exists between them. DSR does not invoke Route Discovery mechanism when a spurious link failure is reported. DSR caches are nearly up to date in low mobility. DSR takes advantage from caching considerably by salvaging at intermediate nodes and using alternate routes at the sources. The TCP source using DSR degrades its output because of this congestion.

This congestion has been validated by IDD metrics. We calculated IDD values for two equal intervals: 1) interval 50^{th} to 70^{th} seconds 2) Interval 70^{th} to 90^{th} second as shown in figures 3.59 and 3.60 respectively.

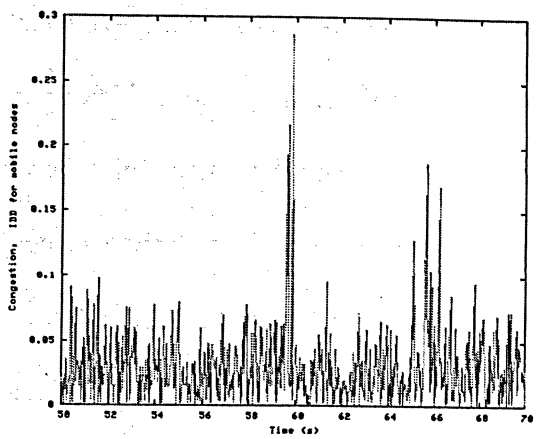

Figure 3.59: IDD values (50-70 second)

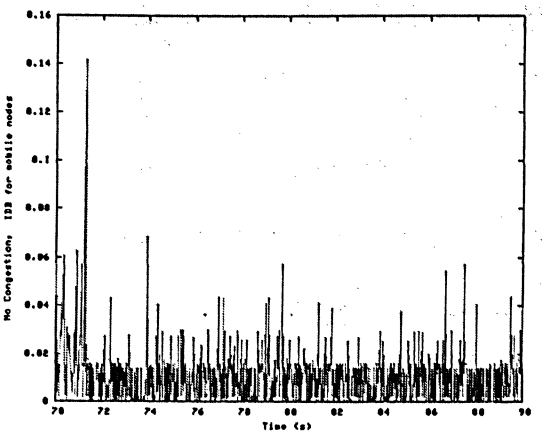

Figure 3.60: IDD values (70-90 second)

Mean IDD value for (50 to 70 seconds) interval = 0.027 second.

Mean IDD value for (70 to 90 seconds) interval = 0.01 second

We can observe that IDD value for the interval 50th second to 70th second is higher than that of interval 70th to 90th second. It is evident that TCP throughput is degraded during interval 50th second to 70th second because number of hops between source and destination has increased but it is not because of path length (number of hops). We observed that RTT values also increased but it is not because of path length. We know that congestion can increase RTT values as packets have to wait in queues caused by the congestion. RTT and IDD metrics indicate that TCP traffic experiences congestion during this interval because of which the throughput decreases during this interval.

During interval 70th second to 98th second, TCP throughput achieves its peak and remains stabilized during this interval. We have noticed through our mobility files that nodes have mobility during this time period. Nodes 11, 14 and 1 move with 5.35, 5.0 and 7.12 m/s. But this mobility does not cause staleness of routes. That is why paths are computed from route cache. RTT values show minimum variations during this interval as shown in figure 3.62. We do not observe packets out of order during this interval (refer figure 3.63). This shows that routes are not changing frequently which could alter the sequence of packets

reaching destination. This stabilized path also gives congestion free TCP throughput as indicated by mean IDD value given as 0.01 second.

During interval, TCP throughput fluctuates during interval 98th second to 127th second. Random mobility file scen-15-test, we noticed that nodes 2, 3, 13, and 12 are moving with 7.2, 9.37, 9.83 and 7.18 m/s. Because of this average speed of 8.5 m/s, we can say that the high node mobility causes TCP throughput to fluctuate as shown in figure 3.61. RTT values also show variation in response to change in path length shown in figure 3.62. IDD mean value is observed to be 0.013 second. This shows that this period can be termed as "no congestion" and RTT values do not increase because of congestion but these are increasing because of increase in number of hops between source and destination. High node mobility during this interval has caused packets out of order as shown in figure 3.63. At 127th second, the TCP connection source and destination get disconnected as shown in figure 3.61.As there is not TCP transmission during this interval. The analysis of the mobility file, scen-15-test, shows this command at this instant which caused the node to go into partition.

$ns_ at 127.912338756752 "$god_ set-dist 0 6 16777215"

Node 0 (source node) is forwarding its packets through node 6 whose path length has become 16777215 which is infinity. This shows that network is partitioned during interval 127th second to 144th second. After 144th second, there is TCP transmission till the end of connection.

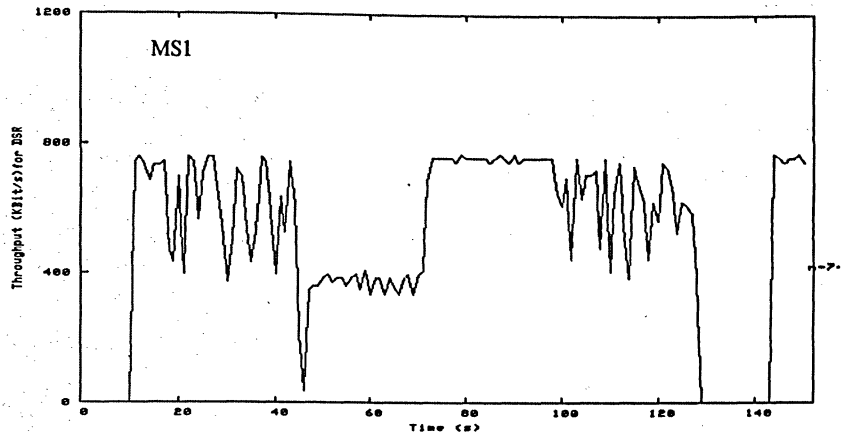

Figure 3.61: TCP Throughput using DSR

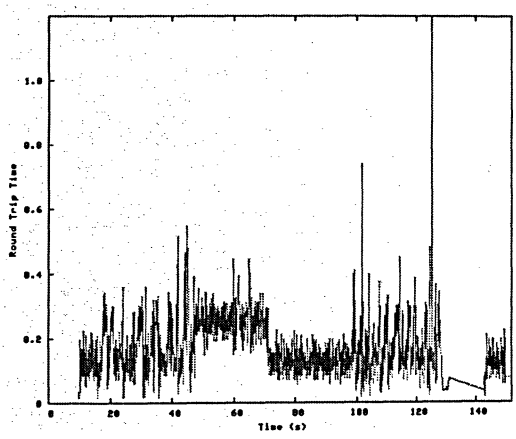

Figure 3.62: RTT for DSR

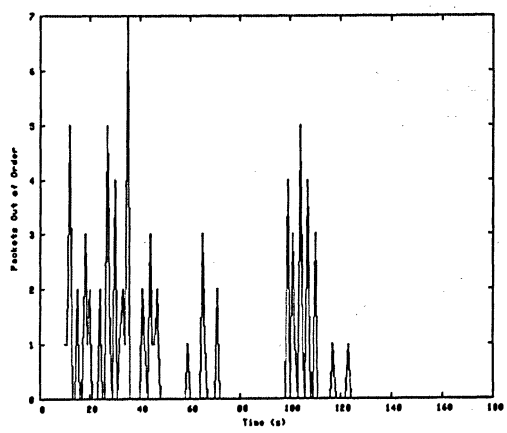

Figure 3.63: Packet Out of Order

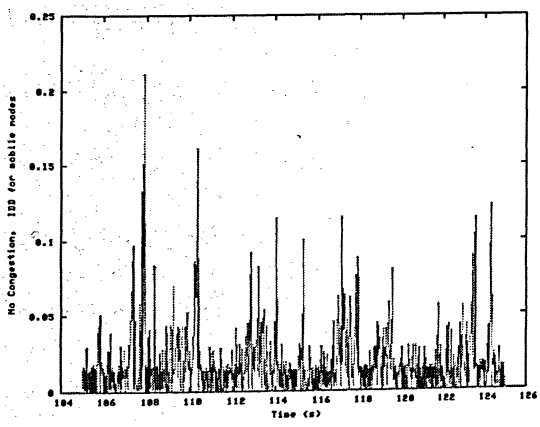

Figure 3.65: IDD values (104 – 127 second)

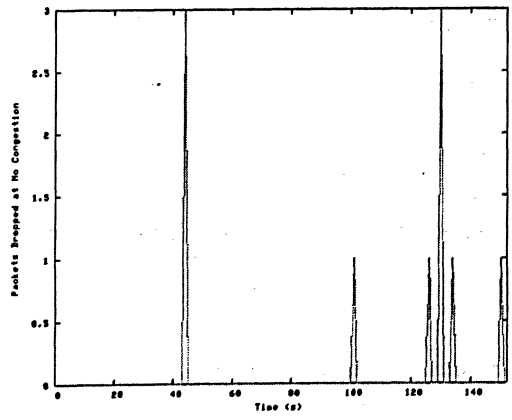

Figure 3.64: Packets Drop

DSDV

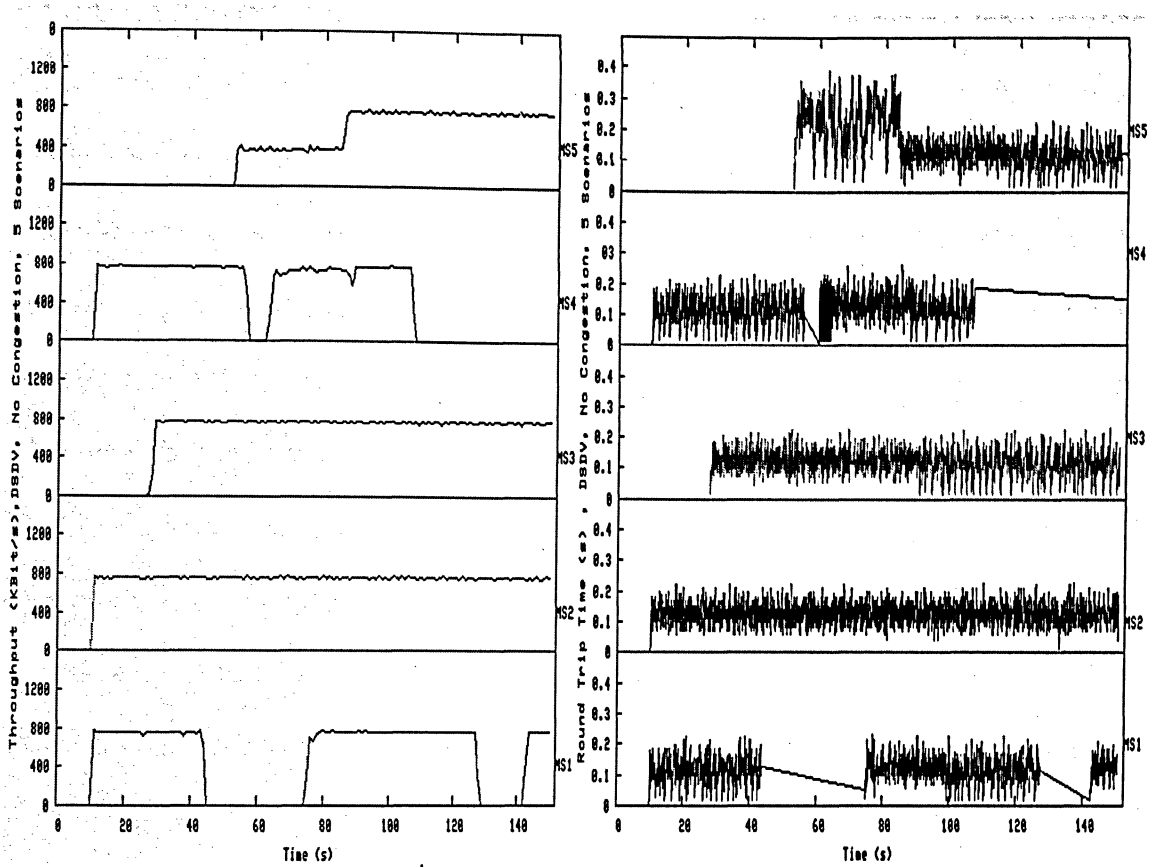

Figure 3.66: TCP Throughput of Five Scenarios using DSDV

Figure 3.67: RTT of Five Scenarios

3.4.2 DSDV: Routing

For mobile scenario (MS1), DSDV is used as routing protocol. TCP throughput remains stabilized during intervals when source is successful in establishing connection with its destination. TCP connection between source and destination is set up at 10^{th} second of simulation time. TCP throughput remains constant during interval (10^{th} second to 45^{th} second) as shown in figure 3.68.

We used the same mobility model for analyzing routing of DSDV as in case of DSR. In case of DSR (previous case), it has been observed that there is no node mobility during 10^{th} second to 18^{th} second. The TCP throughput remains stabilized during this interval.

But topological changes are inferred from mobility file during interval 18 second to 45 second. TCP source, using DSDV as routing protocol, keeps the routing table updated during interval when nodes are moving. These changes are reported by routing update packets called incremental packets. These packets carry routing information changed since the last full dump. Incremental packets update packets are transmitted between the full dumps for partial changes of routing table such as receiving new sequence number and significant route changes.

We know that node mobility starts after 18^{th} second. This fact is shown by small dip in TCP throughput (refer figure 3.68). RTT values fluctuate after this instant. Spikes of these values are touching 0.2 second. RTT fluctuations show that TCP data and acknowledgements for different transmissions between source and destination are taking place through different routes. Since incremental packets report topological changes in the routing tables of the nodes in the network carrying different sequence number and metric, therefore routes with newer sequence number or better metric are selected each time by the nodes for forwarding of these data packets towards their destination. The source node determines the significance of routing information sent out each incremental packet. The TCP through remains stabilized during this period of path variation that is why packets out of order are observed during interval 18 second to 45 second of simulation time.

Routing Outage

During time interval (between 45 second. and 72 second) of simulation; there is no TCP throughput incase of DSDV shown in figure 3.68. But incase of DSR, TCP throughput is observed during this period as shown in figure 3.61. This routing outage is explained as follows: We are using the same mobility file for executing DSDV simulations as used

incase of DSR simulations. There is low mobility during this interval. We have set pause time equal to zero in our simulations but random mobility model can make nodes to move with less speed and nodes can be static too. These observations can be made from mobility file used in our simulation. We inferred that this interval can be categorized as low mobility period. As mentioned in the previous section (refer paper 1), nodes get clustered when there is low mobility i.e. when the nodes are moving with less speed or at rest. This leads to network congestion in certain regions of network. Congestion in turn causes link layer to report link failure even when the nodes are relatively static and the physical link exists between nodes. Such spurious link failure can cause unnecessary invalidation of route entries in the routing table of source nodes. This makes TCP source to go into disconnection even though the nodes are in the communication range of one another.

The presence congestion has been confirmed by IDD metric incase of DSR (previous case) during interval (18^{th} second to 45^{th} second). Mean value of IDD during this time interval is higher than that of interval (72^{nd} second to 127^{th} second). Hence congestion can lead TCP throughput to be discontinued when nodes are moving with low speed. There are no RTT values during this interval as shown in figure 3.69. Since there is no TCP transmission that is why no packets out of order are observed during this period as shown in figure 3.70.

After 80^{th} second of simulation time, we observe in the mobility file that the nodes become mobile as the average speed of nodes increases. This node movement in network significantly changes the network topology which in turn dissipates the congestion in the intermediate nodes. The source node starts getting incremental packets about the path changes. These topological changes make the source node to achieve its maximum

throughput. The TCP throughput is stabilized till the 127th second of simulation time. The This TCP throughput disconnection is caused because of following mobility file command:

$ns_ at 127.912338756752 "$god_ set-dist 0 6 16777215"

The source node cannot forward its TCP packets through node 6 because of 16777215 numbers of hops. This number represents infinity metric.

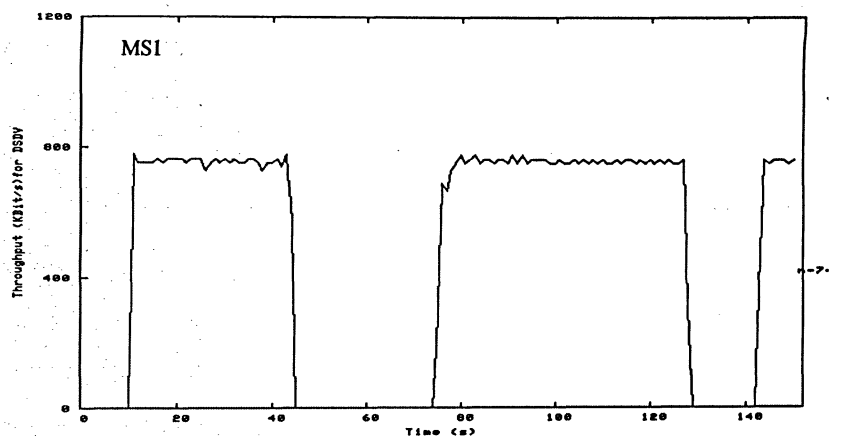

Figure 3.68: TCP Throughput using DSR

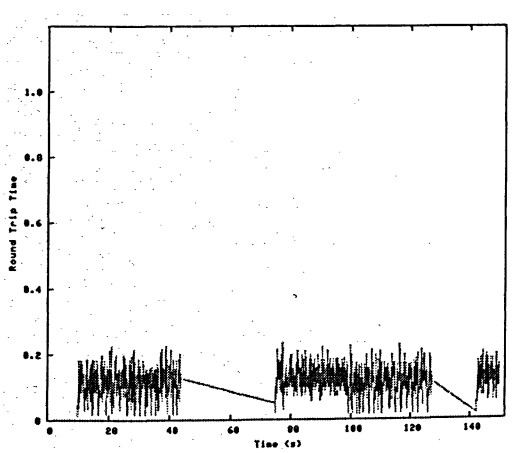

Figure 3.69: RTT for DSDV

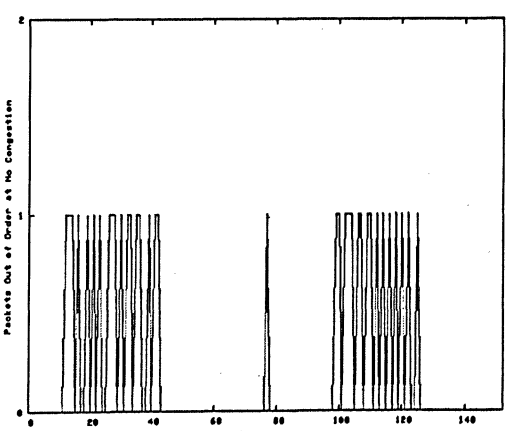

Figure 3.70: Packet Out of Order

3.4.3 Discussion of Results: Routing

DSR and DSDV are compared under same conditions of node mobility. (Refer Table 3.8). It has been observed that DSDV has 6% throughput less than that of DSR. TCP performance is better when DSR is used as routing protocol. Packet Out of Order Ratio for DSR is higher than that of DSDV. This phenomenon is observed because of the fact that routes available in the cache of node get stale because of node mobility. In order to forward data packets, node initiates route discovery mechanism for every data packet. This causes node to compute different route every time. Data packets are forwarded along different routes which cause to increase packets out of sequence. In DSDV, route is selected on the basis of least metric or newest sequence number from routing tables. These route entries are refreshed by routing advertisements. So, data packets follow the same route till the broad casting of route advertisements of new topology. Moreover, delaying of route advertisement for unstable routes also decrease packets out of order in case of DSDV [6].

The packet loss ratio for DSR is less than that of DSDV. This is because of reliable data delivery mechanism for DSR. Although changing topology of network because of node mobility cause routes in the cache to get stale, the node computes route for every data packet by route discovery mechanism. Hence, DSR has less packet loss ratio and higher throughput than those of DSDV.

Table 3.6: Comparison of Routing for DSDV and DSR

Mobile Scenario (MS1)	DSR	DSDV	COMPRISON
Packets received	6248 Packets	5870 Packets	6% less than DSR
Packet Out of Order Ratio (POOR)	78 (1.2%)	35 (0.5%)	2.3 times for DSR
Packet Loss Ratio (PLR)	16 (0.2%)	51 (0.8%)	3.2 Times more for DSDV.

3.5 Composite Effect of Mobility, Congestion and Channel Error

3.5.1 DSR

In order to study combined effect of congestion and channel error, mobile scenario (MS3) is selected. Congestion is created during interval 60^{th} second and 80^{th} second of simulation time. Error Model is incorporated in the simulator. TCP throughput is shown in figure 3.71. The sender starts communicating with its receiver at approximately 30 second of simulation time. We have observed that in case of no channel error and no congestion case, TCP throughput remains constant during period between 30^{th} second and 60^{th} second as shown in the figure 3.5. But when channel error model is incorporated, TCP throughput undergoes fluctuations. Channel error generates route error by using hop-by-hop acknowledgment mechanism. (Refer Chapter 2)This mechanism provides early detection of corrupted packets. These route error messages remove routes from the caches of neighboring nodes. These nodes compute different routes whenever these nodes receive route requests from the TCP source. Since we know that route discovery mechanism is initiated on demand to transmit data packet. Hence, every data packet is transmitted through different routes as the previous routes can no longer be used as these routes are removed from the caches of nodes by route error messages. Packets forwarded along different paths reach their destination out of sequence. Packets out of order, corresponding to TCP fluctuations, are more during interval between 30^{th} second and 60^{th} second than those out of order packets during the rest of TCP connection period as shown in figure 3.71. Since every data packet and corresponding acknowledgment traverse through route different from the previous packet and acknowledgement, this causes RTT

values to undergo variations corresponding to TCP fluctuations as shown in the figure 3.72.

During period 60th second to 80th second when CBR packet uses that node as their route which are already being used as TCP packets. It becomes full path intersection of both traffics. Mean value of IDD during this period is 4.8 times of non congested period. This metric indicates deep congestion during this interval. Since throughput has been decreased that is why less number of packets out of order is being observed during this interval in figure 3.71.

In figure 3.72, RTT values show a large increment up to the value of 2.7 second corresponding to deep congestion period. It shows that sum of time required by data packet to travel from source to destination and acknowledgements to travel from destination to source during congestion period is more than that of during no congestion period. Congestion makes data and acknowledgement packets to wait for long time in the queues of the intermediate nodes to be forwarded to their destination nodes. This causes to increase RTT of each transmission. Three packets are dropped when congestion dissipates approximately at 80th second of simulation time.

TCP Throughput decreases at 95 second in response to packet loss as shown in figure 65. TCP throughput keeps on fluctuating between intervals 90th second to 150th simulation time because of presence of channel error which creates packets out of order till the termination of TCP connection.

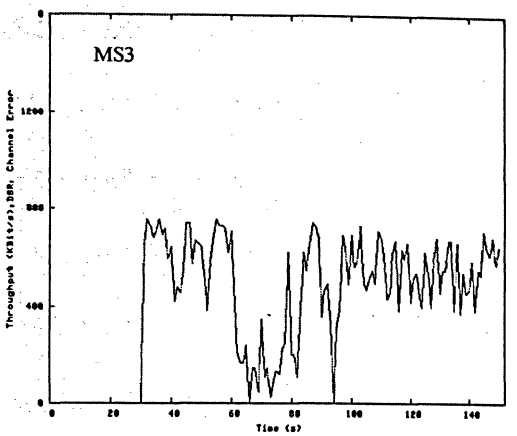

Figure 3.71: TCP Throughput using DSR

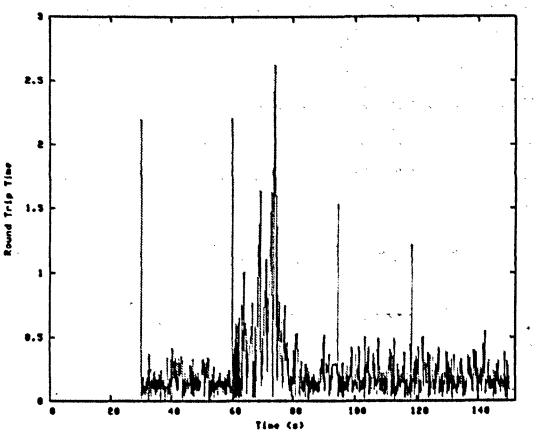

Figure 3.72: Round Trip Time

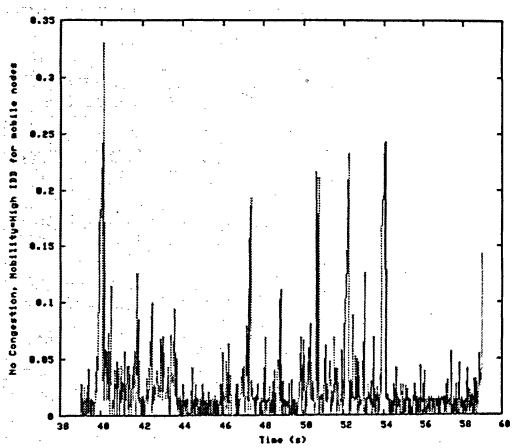

Figure 3.73: IDD for No Congestion. Interval

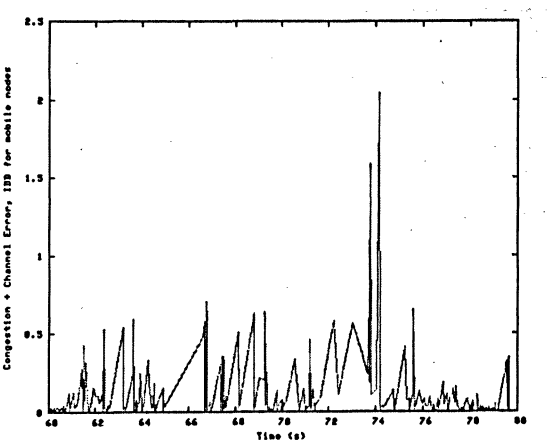

Figure 3.74: IDD for Congestion Interval

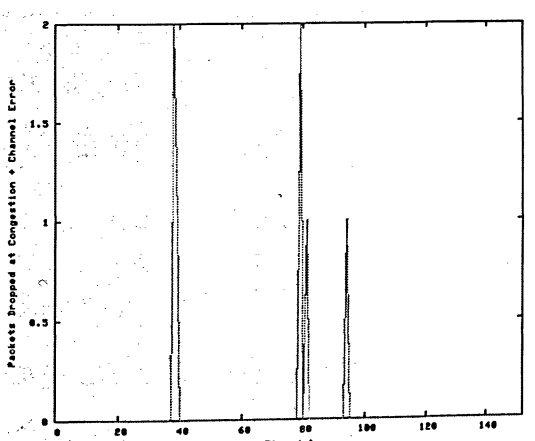

Figure 3.75: Packets Dropped

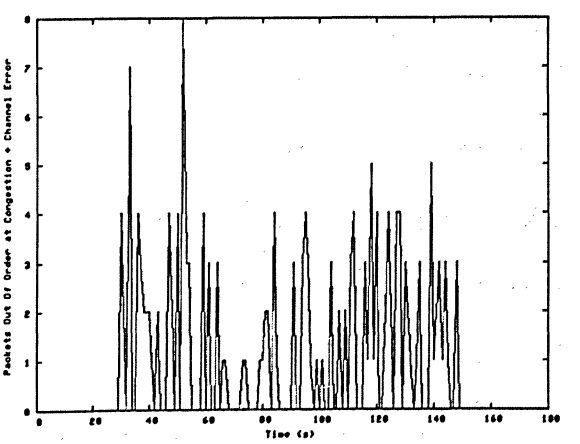

Figure 3.76: Packets Out of Order.

3.5.2 DSDV

We used Mobile Scenario (MS3) generated by random way mobility model to analyze effects of channel error and congestion on the mobile nodes using DSDV as routing protocols. TCP throughput is constant in case of no congestion and channel error. (Refer DSDV: No congestion case). When we introduced congestion by CBR traffic and channel error model to simulate random errors of wire less unpredictable medium, TCP throughput experiences variations as shown in figure 3.77 during period between 34^{th} second and 60^{th} second of simulation time. Packets out of order observed during this interval are because of channel error. Incremental packet report routes with better metric and newer sequence number. TCP sender forwards packets by choosing different routes depending upon metric of route and sequence number. Packets out of order in the absence of channel error are shown in figure 3.53 (refer section 3.3.3, DSDV with No Channel error Case). Packets are dropped during period between 35^{th} second and 60^{th} second of simulation time. Inter packet delay of this period shown in figure 3.79 is same as that of section 3.3.3 when there is no channel error (No Congestion case) showing that IDD values are not affected by packet drops.

During interval, CBR source, node 3 starts transmitting packets to the node 12 (destination). These packets are forwarded through those nodes which are also forwarding TCP packets. Both of these traffics build up queues at the buffers of these intermediate nodes. This phenomenon causes in TCP throughput degradation during interval 60^{th} second and 80^{th} second. TCP throughput undergoes variation after 60^{th} seconds of simulation period. RTT values show increase with time as congestion of data packets keeps on building at buffers of nodes as shown in figure 3.78. Data packet and acknowledgement packet take more time to traverse routes because with the passage of

time, congestion will keep on building as more and more packets will be coming from source and will be queued up at these bottle neck nodes. RTT value becomes 1.1 second at the 63 second of simulation time where as TCP throughput is lowest at this instant shown in figure 3.77.

IDD values are recorded for this interval in the graphical form shown in the figure 3.79 and 3.80. The mean IDD values are given as below:

Mean IDD value for (40 second - 60 second) No Congestion Interval = 0.0113 s

Mean IDD value for (60 second - 80 second) Congestion Interval = 0.055 s

Mean IDD value during congestion period is approximately 5 times the mean IDD value for No Congestion interval. Graphical representation of IDD values for congestion period shows that IDD values increase exponentially with time (refer figure 3.80) as congestion starts building up at the intermediate nodes. CBR source stops sending packets after 80^{th} second of simulation time. IDD values decreases as congestion dissipates. TCP throughput increases to its maximum value as at 80^{th} seconds shown in figure 3.77. Variations are shown on the TCP throughput. These fluctuations are caused because of channel error.

In mobile scenario, we have observed that there is node mobility during interval 93 second till the end of TCP connection. Variations in TCP throughput during this interval (93rd second to 150^{th} second) in case of DSDV are not as high as in case of DSR (refer figure 3.77). Update packets of DSDV report these node movements in the routing tables of nodes. Routes are computed with newer sequence number and metric. Packets are forwarded by these nodes on the routes with better metric and sequence number. Node mobility causes consecutive packets to travel through different routes. This phenomenon is responsible for RTT values to undergo variations during period of node mobility as

shown in figure 3.78. Packets reaching node 12 (destination) following different routes will be out of sequence as shown in figure 3.81. Number of Packet drops are during interval (93 second and 150 second) is more than that occurring during congestion interval (60 second to 80 second) as shown in figure 3.82. This shows that combination of node mobility and channel error causes more packets to drop than packet dropped occurred because of combination of congestion and channel error. Congestion decreases TCP throughput therefore fewer packets are dropped during congestion interval.

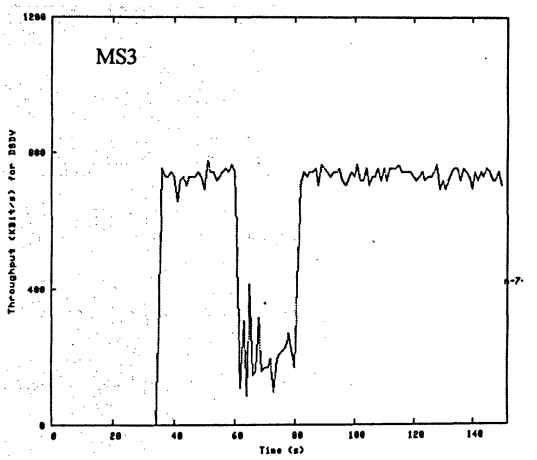

Figure 3.77: TCP Throughput using DSDV

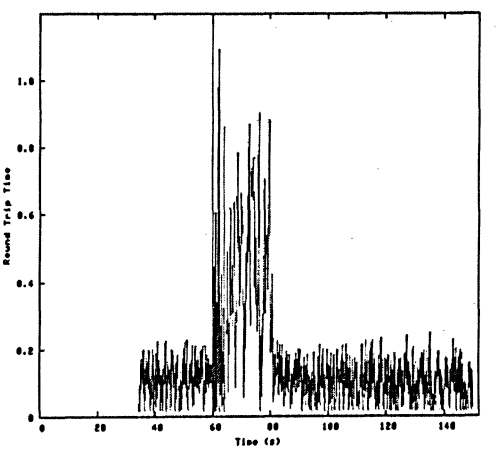

Figure 3.78: Round Trip Time.

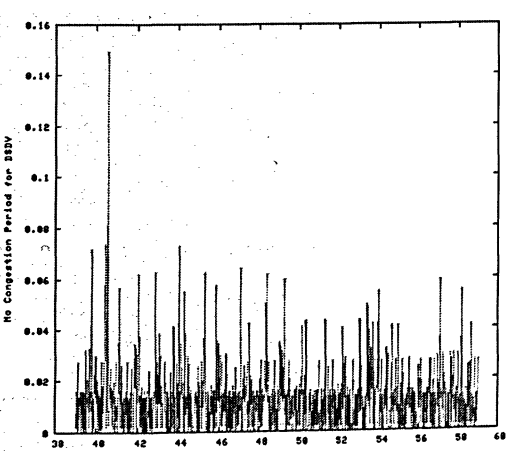

Figure 3.79: IDD for No Congestion Interval.

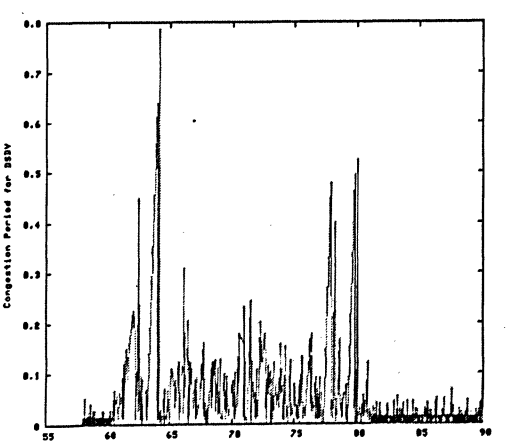

Figure 3.80: IDD for Congestion Interval

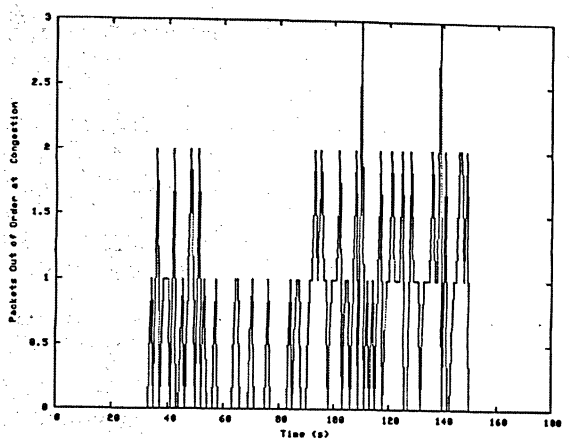

Figure 3.81: Packets Out of Order.

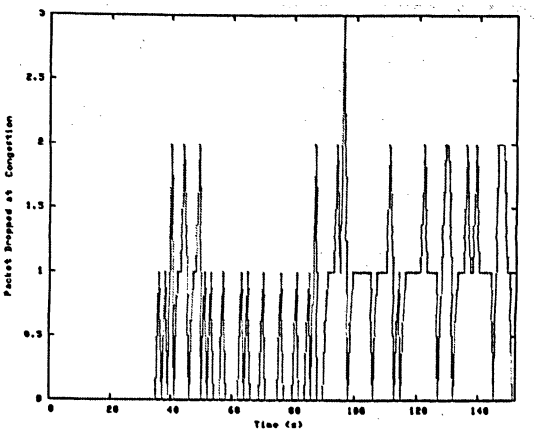

Figure 3.82: Packets Dropped

3.5.3 Discussion of Results: Composite Effect

We have observed that combined effect of mobility, channel error and congestion decreases TCP Packets received, increases Packet Out of Order Ratio and Packet Loss Ratio. Table 3.7 shows that TCP packets received for DSR are more than those for DSD where as less packets are dropped for DSR than incase of DSDV. Packets reaching destination out of sequence are more for DSR than those of DSDV.

Table 3.7: Comparison of DSDV and DSR under composite effect.

Mobile Scenario (MS3)	DSR	DSDV	Remarks
TCP Packets Received	5046 Packets	4718 Packets	6.5% Less than DSR
Packet Out of Order Ratio (POOR)	167 (3.3%)	86 (1.8%)	2 Times more for DSR
Packet Loss Ratio (PLR)	12 (0.23%)	92 (1.9%)	13 Times more for DSDV
IDD (60 – 80 second)	0.097 second	0.055 second	1.8 Times more IDD values for DSR

3.6 Summary

We selected 5 runs out of 38 simulations executed. We observed the effects of network states on TCP throughput by running DSR and DSDV routing protocols. We have noticed that congestion exhibits dynamic behavior because of node mobility. In case of both of routing protocols, the number of TCP packets received is much higher incase of full path intersection than the number of packets received when there is partial path intersection. Similarly, mean IDD value for partial path intersection is less than mean IDD value for full path intersection.

We have observed that channel errors not only affect TCP throughput adversely but also increase packets out of order ratio and packet loss ratio for both protocols. We compared TCP performance of both of these routing protocols and found out that DSR performed better than DSDV in terms of TCP throughput and packet loss ratio. But channel error conditions cause more packets out of sequence in case of DSR than those of DSDV.

Chapter 4

Conclusions and Future work

In this project, we simulated TCP session in mobile ad hoc network running different class of routing protocols under a variety of network conditions. We used metrics which are defined in [2] to evaluate the TCP performance. Reference [2] used these metrics to detect network states by measuring at nodes, which is alternative to the network approach where measurements are performed within the network. Each network state, namely congestion, channel error and route change, is filtered out if relevant metric signals out. By measuring these metrics we are able to derive the interesting results about the performance of transport protocol in an ad hoc network. The results may be useful for improving the performance of TCP layer.

4.1 Congestion

It is known about fixed networks that congestion occurs when data traffic build up queues at intermediate nodes when they are at the cross roads of the traffic path. Congestion changes only when traffic load varies. But in our simulations we observed that in addition to data traffic load, node mobility plays a significant role in dissipation and development of congestion at the bottleneck intermediate nodes. We chose three scenarios to study the phenomenon of congestion in mobile ad hoc networks.

We simulated congestion conditions by injecting CBR traffic flow in the network at a data rate much higher than that of TCP and link capacity. In scenario of interest the CBR

traffic causes congestion at intermediate nodes. The congestion decreases TCP throughput. Mobile Scenario (MS5) shows interesting feature of node mobility when TCP and CBR traffic path intersect at some intermediate nodes for some time and the congestion dissipates when the node is no longer at the intersection of the two paths. This could happen due to the movement of the intermediate node. Queues of data packets are formed at the intermediate node, but as the nodes start moving out of the path of TCP traffic including the congested node the TCP throughput improves. TCP source computed new route through those nodes that are not forwarding CBR traffic. Hence after some time, the paths of each traffic become separated. This topological change dissipated congestion at buffers of nodes previously used as routes. Some of these nodes cease to route traffic after some time even during our designated congestion period.

In mobile scenario (MS4), node mobility totally avoided congestion potentially created by CBR. The movement pattern of intermediate nodes forwarding TCP packets was such that these were not used by CBR traffic as its route during the life time of CBR connection (Congestion interval , refer Chapter 3). Hence, buffers of intermediate nodes were not filled up by CBR packets which prevented these nodes from experiencing congestion. That is why we do not observe degradation in TCP throughput during congestion interval. In case of fixed network, congestion depends on the queue length of node and volume of data traffic which it is forwarding. It has been observed that congestion in the ad hoc network can be altered by node mobility. Congestion can be created or dissipated by node movement.

4.2 Channel Error

In order to simulate the effects of error prone nature of wireless medium, we used the error model provided with ns-2 to simulate channel error. We introduced error in all five scenarios, and chose MS3 to study effects of channel error.

It has been observed that TCP throughput when the source was using DSR as routing protocol in the presence of channel error decreases by 11.9 % from the throughput without the channel error. The channel error caused Packet drops to increase by 14 times the packet drops in the absence of error. Similarly, channel error also caused Packets out of sequence to increase by 4 times of its value in the absence of channel error.

Similar observations can be made in case of DSDV when TCP source used this protocol as routing protocol. TCP throughput in the presence of channel error decreased by 10.45% of throughput recorded in the absence of channel error. Packet out of order and Packet drops increased 3.4 and 3.8 time of their values without channel error, respectively.

It is evident from above results that channel error causes packet loss, which eventually result in the drop in TCP throughput. In DSR, channel error corrupts data, route reply and route request packets. Intermediate nodes may not contain routes in their caches. On initiation of route discovery mechanism, sender may compute different routes to forward packets to the destination, which causes data packets to travel through different routes. These packets reach destination out of sequence. In case of DSDV, channel error bring about changes in topology due to link failure , which requires routing table updates nodes by incremental and full dump packets. This results in packets traversing through different routes as indicated by increase in packets out of order under channel error.

Asymmetry in routes can cause variation in RTT. The Round Trip time is defined as the sum of time taken by packet to traverse through a route from source to destination and the time taken by its acknowledgement to travel from destination to source. Incase of route asymmetry caused by channel error the two times are different for each packet showing high variation in RTT. It has been observed that the effect of channel error on packet out of order for DSR and DSDV are approximately same but number of packet drops incase of DSR is more than that of DSDV.

4.3 Routing

In order to understand routing behavior of both of these protocols, mobile scenario (MS5) is selected. Two end-to-end metrics, round trip time (RTT) and number of packet out of order are used to analyze both protocols.

It has been observed that during interval when there is few node movements, packets are forwarded through stable routes, number of packets out of sequence and Round Trip Time remain approximately the same during this interval. But period during which topology of network changes due to node mobility, number of packets out of order becomes more than the number of packets out of sequence during time period when there is less node movements. Similarly, RTT values recorded during node movement period exhibits high fluctuations. Hence, node movement causes unstable routes which make packets to reach their destination out of sequence and this phenomenon results in high RTT variations.

4.4 Concluding Remarks

We have studied the throughput of TCP under different network conditions and routing protocols. Congestion exhibits more dynamic behavior due to node mobility. Congestion

duration varies depending upon the path intersection whereas congestion level varies due to session intersections. Channel error causes packet losses and packets out of order because of routing changes. Asymmetry routes result in RTT variations. Combined effects of channel error, congestion and mobility result in increasing congestion level, packets out of order ratio and packet loss ratio.

4.5 Future Work

In the future, we intend to investigate congestion level changes due to node mobility. Design of new congestion control algorithm for TCP that accounts for network states leading to packet losses should be undertaken.

Bibliography

[1] Charles E. Perkins and Elizabeth M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," IEEE Personal Communications, February 2001, pages 16-28.

[2] Zhenghua Fu and Benjamin Greenstein, "Design and Implementation of a TCP-Friendly Transport Protocol for Ad Hoc Wireless Networks," IEEE International Conference on Network Protocols (ICNP'02), 2002, pages 216-228, Paris, France.

[3] J. Liu and S. Singh, " ATCP: TCP for Mobile Ad Hoc Networks," In IEEE J-SAC, vol. 19, no. 7, pp.1300-1315, July 2001.

[4] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP performance over wireless networks," In Proc. MOBICOM, pages 2-11, Berkeley, CA, 1995.

[5] Johnson, D. B. and Maltz, D. A.; "Dynamic Source Routing in Ad hoc wireless networks," Mobile Computing, edited by Tomas Imielinski and Hank Korth, Kluwer Academic Publishers, ISBN: 0792396979, 1999, Chapter 5, pages 153-181.

[6] Charles E.Perkins and Pravin Bhagwat, " Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Proceeding of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, London, UK, pages 234-244.

[7] Zhenghua Fu, Xiaoqiao Meng and Songwu Lu, "How Bad TCP can perform in Mobile Ad-Hoc Networks," Proceedings of the Seventh International Symposium

on Computers and Communication, IEEE Computer Society, July 2002, pages166-176.

[8] Joseph P.Macker and M.Scott Corson, "Mobile Ad Hoc Networking and the IETF, " Mobile Computing and Communications Review , Volume 3, Number2.

[9] Hari Balakrishnan, "How Network Asymmetry affects TCP," IEEE Communications Magazine, April 2001, pages 66-69.

[10] Vaidyanathan & Raghupathy Sivakumar, "A Microscopic analysis of TCP performance over wireless Ad-hoc Networks," Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 270 – 271.

[11] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," Mobicom'99, pages 76- 87.

[12] Kevin Fall and Kannan Varadhan, "The ns Manual", 2003. [http://www.isi.edu/nsnam/ns/doc]

[13] Leon-Gracia and Widjaja, "Communication Networks". [The McGraw-Hill Companies, Inc.,2000]

[14] C.-K. Toh, "Adhoc Networks". [Prentice Hall, 2001]

[15] H.M. Dietel, "Perl , How to Program" Edition II, 2002. [The McGraw-Hill Companies, Inc.,2002]

[16] Judith Samson, "Teach Yourself Red hat Linux". [Sams Publishing,2001]

[17] Marc Gries, " Tutorial for Network Simulator". [www.isi.edu/nsnam/ns/tutorial]