Implementing and Testing of Anchor Point Node (APN) Hybrid Network Architecture for Remote Areas with Simulation of Transmission Control Protocol (TCP)

1Abdul Razaque 2Khaled Elleithy 3Qassim Bani Hani
1arazaque@bridgeport.edu 2elleithy@bridgeport.edu 3qassim_banihani@yahoo.com
Computer Science and Engineering Department
University of Bridgeport, USA

Abstract
Transmission Control Protocol (TCP) is one of the popular transport-level protocols for distributed applications like FTP, http, telnet, email applications due to its rich set of desirable features, including a reliable, ordered, duplex byte stream, flow control, and congestion control. The performance of TCP is affected due to various factors including the congestion window, maximum packet size; retry limit, recovery mechanism, backup mechanism and mobility. The mobility is the major factor to affect the performance of transport protocols in mobile wireless networks. Quantification of losses with varying amount of mobility in different variants of TCP is one of the performance parameter need to be considered while designing the networks. Unfortunately we could not find such information in the published literature. To quantify the effect of mobility for different variants of TCP, I have designed and implemented APN hybrid network, consisting of three segments wired, wireless and MANET particularly for remote areas. Our simulated testing of APN hybrid network is around the performance of TCP variants by using Random Waypoint Mobility model with varying mobile speeds, which shows the speed of walking person to vehicle in mountainous and deserted areas in order to analyze throughput, RTT fairness, In-order delivery of data, effect of mobility on Goodput, End-to-End delay and control overhead from mobility point of view.

General Terms
Theory, Design Architecture, Experimentation, Performance.

Key words: Mobility, Random Way Point Mobility Model, MANET, TCP, Wireless LAN, Anchor Point Node, DSR Routing Protocol.

1. Introduction and Background Study

Deployment of wireless networks in last several years has motivated many researchers to study and make effort for improving the performance of TCP which was originally designed for fixed wired networks. This work confirms that TCP in its present structure is not an optimal transport service for mobile and wireless networks. Some variants of TCP have been proposed and implemented in order to optimize the performance in wireless and mobile networks. We will evaluate some of these variants, and we will demonstrate what they can increase the performance on wireless networks. This study will be based on implemented APN hybrid network particularly MANET. The MANET can be set up in any remote areas without infrastructure support. The nodes, which are part of MANET and they require data and information from database but database is available in the wired network, therefore MANET can be integrated with wired network to obtain the required data and information. Some applications run over the database and these applications are supported by Transmission Control protocols.

This contribution aims to exhibit the flaws for TCP compared to Hybrid network especially MANET. Exhibition will be done with ns2. This simulator will provide the outcomes for different protocols’ throughput, effect of mobility on Goodput, overhead etc…that we will analyze. TCP uses some congestion control parameters, which include congestion window, recovery mechanism, retry limit, maximum packet size and back up mechanism for IEEE 802.11 retransmission (Fernando Tapia, 2004), (A.O. Oluwatope et al., 2006). To minimize the congestion problems, as different TCP Variants have been introduced & simulated on various schemes in order to identify the performance for each TCP Variants and analyzing which variant has considerable performance due to mobility. To
The slow start threshold (ssthresh) decides whether cwnd may move to slow start or congestion avoidance phase, as it depends on available network capacity, if cwnd moves again to slow start phase that the value of cwnd in the slow start phase, MUST be less than or equal to 2*SMSS bytes and MUST NOT be more than 2 segments” (J. Postel., RFC 793, 1981) & (RFC 2581, Allman et al., 1999). If it is decided to move cwnd into congestion avoidance phase then next algorithm of congestion avoidance starts to function. (J. Postel., RFC 793, 1981) & (RFC 2581, Allman et al., 1999). This is explained in next section.
(Fernando Tapia et al., May 2004), which is explained in the next section.

1.1.3 Fast Retransmit & Fast Recovery Mechanisms

When packets start to receive an out of order, then that duplicate acknowledgment is generated and sender has to wait for three Dupack or four Acks with similar sequence number. Dupack is considered as sign for loss of packets. The sender retransmits the segment without waiting for timeout. Therefore, ssthresh = max (Flight Size / 2, 2*SMSS) & cwnd = ssthresh+3MSS (RFC 2581, Allman et al., 1999).

If Ack is received, then cwnd is increased with1 MSS. To recover the lost segments; the fast recovery algorithm start to function. IMSS is recovered in congestion window against each RTT, this process continues until all lost segments are recovered. The rest of paper is arranged as: First, the focus of our contribution is to introduce the behaviour of TCP Congestion Control Variants. second, we study related work with respect to other proposed techniques. Third, APN Hybrid Network Architecture and Essential Components. Fourth, Connectivity Process of APN Architecture. Fifth, Description of APN Hybrid network Scenario. Sixth, Overview of Random Waypoint Mobility Model. Seventh, Simulation Setup. Eighth, Simulation Result. Ninth, Discussion of Result. Finally, we give conclusion and future work, which identify better TCP Variant from mobility point of view in hybrid network and future directions.

![Figure 1: State Transition for TCP Variants (Allman et al., RFC 2581, 1999).](image)

1.2 TCP Congestion Control Variants

We have studied some TCP congestion control variants with respect to characteristics and summarized the overview of variants from mobility point aspect.

1.2.1 TCP Tahoe

The mechanism of TCP Tahoe consists on three transmission phases: slow start, congestion avoidance, and fast retransmit ((Laxmi Subedi et al., 2003). Fast Recovery phase does not allow the cwnd to increase exponentially. Instead, it concerns the much slower Congestion Avoidance mechanism. TCP Tahoe does not handle the multiple losses of segments in network, there is very slight difference between 1 and more segments drop in the TCP Tahoe (Jacobson.V 1988).When small numbers of duplicate acknowledgments are received, sender concludes that segments have been lost and start to retransmit the segments without waiting for timer to expire, which leads to higher channel utilization (Kevin Fall and Sally Floyd 1998).

1.2.2 TCP Reno

TCP Reno modifies the “Fast Retransmit” mechanism by incorporating the Fast Recovery algorithm. In TCP Reno, the window size changes at regular intervals in a typical Situation. It continues to be increased until segment losses occur. The window size of TCP Reno depends on two phases: slow start phase and congestion avoidance phase. When TCP receives acknowledged packet at the sender side within time t+tA sec, the current window size is adjusted as cwnd (t+tA) Here “t” is time to reach the packet at receiving side and
“tA” is the time to receive an acknowledgement at sender side (W. Richard Stevens et al., 1994).

When packet loss occurs and sender receives the number of duplicate ACKs then, it enters into Fast Recovery and retransmits the loss packet. When Fast Recovery algorithm is triggered, the sender’s cwnd is set to half of the window. When TCP Reno retransmits the loss packet it waits half a window to be acknowledged. Subsequently it sends the new packet against every received duplicate ACK. When sender receives the acknowledgment for newly sent packet, it exits Fast Recovery. On exit of Fast Recovery the TCP Reno sender shrinks its usable window to cwnd. The fast recovery algorithm of TCP Reno improves the performance of TCP in case of a Single segment losses occur within a window of data. However, performance of TCP Reno experiences in case of multiple packet losses occur within a window of data (Laxmi Subedi et al., 2003) & (S. Floyd and K. Fall July 1996).

1.2.3 TCP New-Reno

TCP New Reno provides slight modification of TCP Reno. It improves the fast recovery algorithm of TCP Reno. It has capability to identify multiple segments of losses in window. It does not exit the fast recovery algorithm until all lost segments are acknowledged (H. Lee et al., 2001). The Reno TCP has a flaw of forcing unnecessary RTO when there are multiple packets losses occur in a single cwnd. TCP Reno exits the Fast Recovery upon receiving the ACK whereas The New-Reno TCP does not exit the Fast Recovery upon reception of partial ACK, instead it consider it as an indication, that the packet immediately subsequent acknowledged packet has been lost. TCP New-Reno recovers from multiple packet losses in a single cwnd without waiting for RTO. It retransmits each lost packet one by one per each RTT. A TCP New Reno set congestion window size to slow start threshold and persists the congestion avoidance phase and retransmits the next segment when it receives a partial acknowledgment (S. Floyd and T. Henderson April 1999).

1.2.4 TCP SACK

The TCP SACK is extension of TCP Reno. TCP SACK algorithm allows a receiver to acknowledge out of order Segments selectively rather than cumulatively because mechanism of TCP SACK is based on selective acknowledgement, which makes it more robust to packet losses. The Sack blocks presented in SACK TCP are used to notify the sender about non-contiguous data segments, which have been received and queued at receiver. The SACK TCP has introduced a new variable “PIPE” that makes the Fast Recovery period more efficient. The variable PIPE is used to keep the record of estimated outstanding packets in transmission path. The sender sends the data when cwnd > PIPE. The value of PIPE is incremented when sender sends any packet (either retransmitting lost segment or transmitting a new segment) and decrement the value on each acknowledged segment. TCP SACK does not exit the Fast Recovery process on receiving partial ACK. It exits Fast Recovery when all of the outstanding data segments are acknowledged (M. Omueti and Lj. Trajkovic July 2007).

1.2.5 TCP Vegas

TCP Vegas is interoperable variant, which augment the throughput performance; reduce packet loss, while it does not affect the fairness. The edge of Vegas is to calculate the available bandwidth in network. TCP Reno calculates the available bandwidth by increasing the number of packets, until the network is congested and start to loss segments. This process is objectionable, as it can make TCP overshoot and generate buffer overflows in routers and gateways in wired network. It also leads to MAC layer conflict in a wireless Multihop network. TCP Vegas firstly congests the network then adjusts it. TCP Vegas determines the bandwidth on the basis of difference (diff) between expected and actual throughput, to avoid packet loss (Brakmo et al., 1999). Difference in the flow rates can easily be interpreted into the difference between the window size and number of acknowledged segments during the Round Trip Time (RTT) by using the following statement.

\[
\text{Difference} = (\text{Expected throughput} - \text{Actual throughput}) \times \text{Base RTT} \quad \text{(I)}
\]

Here “Base RTT” shows the minimum round trip time. TCP Vegas algorithm functions as follows:

The expected rate can be obtained as \( \text{Expected} = \frac{\text{cwnd}}{\text{Base RTT}} \quad \text{(II)} \)

Here, cwnd is current window size and Base RTT is minimum Round Trip Time (RTT).

The Actual Round Trip Time can be calculated as \( \text{Actual} = \frac{\text{cwnd}}{\text{RTT}} \quad \text{(III)} \)

The basis of difference the source updates window size as follows:

\[
\text{cwnd} = \begin{cases} 
\text{cwnd} + 1 & \text{if diff} < \alpha \\
\text{cwnd} - 1 & \text{if diff} < \beta \\
\text{cwnd Otherwise} & \text{Otherwise}
\end{cases} \quad \text{(IV)}
\]
TCP Vegas attempts to identify and use extra bandwidth when it becomes available without congesting the network (Jeonghoon Mo et al., 13 July 1998).

1.2.6 TCP Westwood

TCP Westwood is a sender side modification of the TCP Reno and enhances the performance in wired and wireless segments of networks. It gives more significant improvement in wireless networks with lossy links (Saverio Mascolo et al., 2001). TCP Westwood estimates bandwidth, which can be employed by the congestion Control algorithm performed at the sender side of TCP connection. The slow start and congestion avoidance phases are unaffected in congestion window that is why they increase linearly and exponentially. When TCP Westwood gets number of n duplicate Acknowledgment (DupAck) and timer is expired, it functions like this.

\[
\text{If (numbers of DupAck are received)}
\]
\[
\text{Ssthresh} = \frac{\text{BWE} \times \text{min RTT}}{\text{Seg_size}}
\]
\[
\text{If (cwin > ssthresh)}
\]
\[
\text{cwin} = \text{ssthresh};
\]
\[
\text{endif}
\]
\[
\text{endif}
\]
Here seg_size is the length of TCP segment in bits (Claudio Casetti et al., 2002).

TCP Westwood sets the value of ‘RTTmin’ as the smallest round-trip time (RTT), which is estimated by TCP by using its own RTT estimation mechanism. It retransmits the lost segments after receiving three identical DupAck. TCP Westwood uses Additive Increase and Additive decrease mechanism. This variant has mobility support with some extend but initially takes more time for bandwidth estimation that is why it transmits lesser packets than TCP Vegas (Casetti.C et al., 2001). The purpose of this study is to confine the slight interaction of the various TCP Variants and gives better starting point for designers to design new transport protocols that should be more robust to the different packet delivery patterns during the mobility and provide better services than existing transport protocols from mobility point of view.

1.3 Congestion Management algorithms for TCP Variants

Substantial research has been done to investigate the outcomes of various algorithms used to improve and maintain congestion window in different network scenarios. The performance of Hybrid networks is unpredictable and, therefore, no algorithm achieves better results in all cases. Many algorithms do not execute well in deployed networks because constraint and network limitations affect the performance. These algorithms can be analyzed in simulated environments. However, some algorithms enhance network performance and aim to attain higher utilization of end-to-end resources. TCP New Reno is a modified version of TCP Reno, which handles multiple packet losses occur within a window of data. SACK performs better in case of multiple losses of segments. The case of Hybrid networks, where disconnections may occur due to mobility, has not been considered (S. Floyd and K. Fall 1996). TCP Reno with SACK gets minimum bandwidth utilization in case of congested links and obtains lower Goodput than TCP Reno and TCP New Reno (R. Paul and Lj. Trajkovic Aug., 2006).

TCP Vegas with other TCP Variants gets lower share of bandwidth utilization, therefore attain lower Goodput than all of TCP Variants. TCP Westwood shares better bandwidth utilization with TCP Tahoe, Reno, New Reno, SACK and Vegas. Hence, In case of noncongested network TCP SACK, Tahoe, New Reno and Reno are considered as comparable performer (H. Lee, S. Lee, and Y. Choi Feb.,2001). TCP New Reno outperforms TCP Reno and TCP Reno with SACK when no losses of segments happen during the slow-start phase. Study of various TCP algorithms over wireless networks with interrelated packet losses indicate that TCP New Reno often performs worse than TCP Tahoe because of the ineffective fast recovery algorithm (F. Anjum and L. Tassiulas June 2003). In summary, reported studies indicate variable performance of TCP algorithms.

These differences may be due to a variety of hypothesis, network parameters, network topologies and mobility. The reason of designing these algorithms is to improve and maintain the congestion window in order to enhance the utilization of network resources in case of congestion. The performance of west wood and Vegas in Hybrid networks has not fully been analyzed in detail from mobility point of view. We compare the performance of these TCP Variants how they manage the congestion and produce the outcomes in
mobility.

1.4 The performance-affecting factors for TCP

In addition to the conventional problems of wireless networking, TCP suffers due to timeouts and duplicate acknowledgments. TCP does not differentiate losses due to congestion or a link error, and in resulting the performance of TCP is affected. There are different factors, which are affecting the performance of TCP Variants. We present analysis of some factors, which cause degradation in the performance of TCP in wireless networks particularly Hybrid network.

1.4.1 Round Trip Time (RTT)

RTT is set against congestion window to compute the Retransmission timeout (RTO) for each packet. The Variation of RTT reduces the performance of TCP. The RTT has much greater inconsistency in wireless networks than the wired networks. The wireless links suffers due to limited power, weak signals and mobility, these factors lead to variation of RTT, and performance of TCP Variants is affected (Vassilis Tsaoussidis et al., 2001).

1.4.2 Packet Error Rate

The packet losses indicate the congestion. The ratio of packet error rates in wireless links is greater than wired links. In Hybrid network, occurrence of Packet reordering is related with handoff (mobility). Whenever mobile node hands over from one base station to other, the some transmitted packets sent by mobile node, are still received at the previous station, then routed to new base station and some part of transmitted packets is delivered at new base station. The traveling of packets through different paths may take different times to reach at the destination (node) that causes of packet error rate or reordering.

The same situation also happens in Hybrid network because it may be the integration of wired and wireless segments of network. TCP congestion control mechanism cannot handle the handoff (mobility) issue; resulting segments are mainly dropped and throughput is highly affected. This assumption leads to study the performance of TCP Variants in APN Hybrid network from mobility point of view.

1.4.3 Bandwidth Utilization

TCP tries to find out existing bandwidth in the slow start phase by doubling its congestion window against each RTT. TCP uses existing bandwidth by increasing the cwnd by 1MSS against each RTT. TCP identifies how much highest amount of data should be sent against each RTT. There is different signal to Noise Ratio (SNR) available on wireless links, which reduces the bandwidth of channel. TCP does not also sense the fluctuation of SNR, and continuously sends the pace of the packet at the current rate, which causes the losses of packets. If ratio of SNR also falls in Hybrid network due to mobility, the performance of TCP suffers a lot that causes for degrading the performance. Due to different affecting factors, our target is to identify the performance of each congestion control mechanisms in presence of random waypoint mobility model with Hybrid network.

We conclude that TCP performs well in wired networks where Packet losses occur mainly due to congestion. The rapid growing need of wireless networks specifies that wireless networks will play more important role in future networks but TCP degrades the performance in Wireless networks due to channel contention, signal fading, mobility, and limited energy and power (KA-Cheong Leung et al., 2006). The congestion mechanism of TCP has no capability to differentiate losses due to congestion and other losses (Antonio Annese et al., 2004). TCP experiences reduced performance because of non-congestive losses, which include burst loss, random loss and packet reordering, leading high decline in throughput. Mobility causes Link failures in wireless networks.

Introduction of wireless technologies has increased the use of computing and communication services in our everyday business and personal lives. At the same time, inconsistent bandwidth, handover, re-association, latencies and limited power constraints, or lack of infrastructure create the problems for applications. Understanding the opportunities and challenges, the researchers of wireless network have contributed through major improvement in emerging technologies, leading toward the visualization of consistent mobile communication. Most of legacy applications use TCP for transport of data between sender and receiver which perform well in the wired networks, but suffers from performance degradation due to link-layer packet losses and mobility.

2. Related Work

This section reviews and evaluates different proposal for TCP Variants configured for Hybrid network. The focus of review is to find the benchmark for evaluation of TCP performance parameters include throughput, Round Trip Time (RTT), End-to-End delay, an effect of mobility at Goodput, In-order delivery of data, Control overhead and Broken links. This section provides the background information,
and set the context in which paper is written. We will evolve sound understanding of TCP Variants in Hybrid networks. The focal point of study is to implement the designed Architecture of APN Hybrid network and testing the performance of TCP Variants from mobility point of view over.

Log Westwood + TCP (Maroje Delibasi et al., November, 2006) have compared the performance of Log Westwood + TCP with New Jersey, New Reno and Westwood+ and analyzed some performance metrics of TCP Variants through simulation using Single-flow and multiple flows scenario; Friendliness scenario, Wireless scenario and satellite scenario. Authors have proposed Log Westwood + TCP and claimed that log Westwood+ TCP can perform better than Westwood+. The maximum window size ‘$W_{\text{max}}$’ and Varying parameter ‘$\alpha$’ are introduced for improving the performance of log Westwood+. As ‘$W_{\text{max}}$’ is congestion window value at which last packet loss was identified and also represents overall pipe size estimate. Function proposed in Log Westwood+ TCP works during congestion avoidance phase whereas slow start phase is not changed. If ACK is received then congestion window is updated according to given formula.

$$W \leftarrow W + \max \left( \frac{W_{\text{max}} - W}{\alpha W} \cdot \frac{1}{W} \right)$$

Authors introduced the concept of new ‘$W_{\text{max}}$’ and ‘$\alpha$’ parameters for improving the performance of log Westwood+ and simulation results show marginal improvements. Though paper explains the reasons for performance improvements, it lacks sound reasoning and causes of performance degrading.

comparison of TCP variants across a hybrid Wireless (A.O. Oluwatope et al., 2006) presented the simulation of TCP SACK, TCP New Reno and Westwood on the basis of realistic scenario between two Nigerian Universities. Authors have obtained throughputs for simulated TCP Variants, which were analyzed at MATLAB 6.5 and introduced two existing possible solutions for improving the performance of TCP in wireless lossy networks.

The first approach conceals the con-congestion losses from TCP sender, which needs no modification to existing sender implementations. The most of losses seen by TCP are due to congestion. The second approach is to make aware the sender from existence of wireless hops and understand it that some packet losses are not due to congestion. Authors have also classified many schemes into three basic, which is based on link-layer proposals, end-to-end proposals and split-connection proposals. Authors have simulated the scenario and obtained the results at 5% packet error rate.

In this scenario, TCP flows were in slow start phase for first 10 seconds. TCP Westwood has obtained better throughput performance, which is 0.45 Mbps and reason of producing the better throughput performance of Westwood is bandwidth estimation mechanism whereas TCP Reno and TCP SACK have gained 0.35 Mbps and 0.38 Mbps respectively.

Gateway Adaptive Pacing for TCP (Sherif M et al., October 2007) have proposed rate-based packet scheduling scheme within TCP congestion window in wireless environment for improving the fairness and Goodput performance for TCP connections across Multihop IEEE 802.11. Some transport layer Functionalities have been incorporated to the IP layer at the Internet gateway for obtaining transparent decoupling. An efficient congestion control mechanism for TCP over Hybrid wireless networks gives adoptive pacing scheme for wired-to-wireless at Internet gateway.

TCP-PCP over Wired/Wireless Hybrid networks (Jin Ye et al., August 2008) proposed the concept of congestion probability (CP), which depends on Explicit Congestion Notification (ECN). Authors have claimed that TCP-PCP can improve the performance of TCP more than Westwood and Jersey. ECN signal points out the forthcoming congestion rather than Congestion, which is already happening. Since network dynamically changes. In order to distinguish serial ECN feedbacks, Authors have introduced new concept of CP (Congestion Probability).This scheme has limitations because it can function only for particular scenarios. Long-lived FTP application runs over single TCP connection to deliver data from source to destination. It is not discussed if more than one FTP flows are used then proposed concept of Congestion Probability (CP) whether produces better performance or not.

### 2.1. Analysis of Existing Work

In this contribution, we have evaluated 8 Variants of TCP which are TCP-PCP, Westwood TCP, New Jersey TCP, SACK TCP, Westwood TCP +, Log Westwood TCP +, New Reno TCP and New Reno AP proposed to improve the performance parameters in Hybrid network environment. On the basis of some performance metrics, we have made a benchmark of existing work and compared
the work with our Analysis of TCP Variants in Hybrid network given in Table No. 1.

Table 1: Analysis of Existing work in Hybrid network with respect to our Analysis.

<table>
<thead>
<tr>
<th>Existing work</th>
<th>Throughput/Goodput</th>
<th>RTT</th>
<th>In-order delivery of data</th>
<th>Effect of Mobility</th>
<th>End-to-End delay</th>
<th>Control overhead</th>
<th>Broken links</th>
<th>Mobility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maroje Delibasi et al</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>A.O. Oluwatope et al</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sherif M et al</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Only Discussed</td>
</tr>
<tr>
<td>Jin Ye et al</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>APN Hybrid Network</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2.2. Conclusion of Existing Work

We conclude the existing work found in literature survey on basis of Hybrid network. Maroje Delibasi et al work has proposed scheme for improving the performance of Log Westwood + TCP and produced simulation based study of Log Westwood + TCP, New Jersey, New Reno and Westwood+ on the basis of different static scenarios in Hybrid network. A.O. Oluwatope et al have used realistic packet error rate static scenario between two Nigerian universities. TCP Westwood, TCP SACK and TCP New Reno have been simulated and analyzed their performance with MATLAB 6.5.

Sherif. M et al have proposed rate-based packet scheduling scheme within TCP congestion window in wireless environment for improving the fairness and Goodput performance for TCP connections across Multithop IEEE 802.11. TCP-GAP and New Reno have been simulated on Chain topology, two parallel-chain topology, Cross Topology and Random Topology. Jin Ye et al have proposed a novel TCP-PCP, which is based on the prediction of Congestion Probability. It can increase TCP throughput in Hybrid networks because the sender monitors the network state by using serial ECN response other than single ECN response. Finally Authors have evaluated the performance of TCP-PCP, Westwood TCP and Jersey TCP Variants on topology-1 and topology 2. Impact of mobility is not thoroughly studied for example a fast moving mobile nodes can quickly raise handovers or packet loss ratio may be high due to link losses or reconfiguration of routing may not complete in acceptable time causes higher drop rate.

The goal of literature survey is to identify the performance of each TCP Variants in Hybrid network. In this connection, we have focused the performance of some TCP Variants from mobility point of view and used Random Waypoint Mobility model. The speed and number of mobile node (mobility ratio) increases and found an impact of mobility on the performance of simulated TCP Variants in
hybrid network.

3. APN Hybrid Network Architecture and Essential Components

With emergence of mobile network, require well-organized communication architecture to combine wireless and wired components. Generally the design of central protocols of the wired network does not report for wireless architecture. End-to-end services of wired network are obtained, which is based on the functionality of TCP. Few studies as (P.M. Ruiz et al., October 2005) have been conducted while connecting the MANET to the Internet (wired network). Previous studies have focused particularly on UDP. UDP does not provide guaranteed communication because it does not need to re-establish new connection either addresses are changed or lost the packets.

UDP is not affected due to change of address in a MANET as much as TCP does. TCP uses a connection oriented services where packets may flow from a sender to a receiver. Each flow is identified by different metrics, which include IP-address and change of the IP-address cause a drop of connection. Congestion control is another issue that is not adapted to MANET environment and resulting TCP sometimes does not use entire available network resources. Hence, this study will focus on TCP performance in hybrid network. MANET does not use subnets as wired IP network can use. MANET depends on flat topology and keeps unique address within MANET. So, routing protocols must handle new addressing structure. The node, which makes coordination between MANET and wired network is said to be Anchor Point Node (APN). APN possesses the functionality of translating address structures, routing protocols and physical interfaces between two networks given in figure 2.

![Flow of data from wired node to MANET node in APN hybrid network](image)

APN plays role of proxy and accomplish address translation. It also replies to all route requests, which it knows, pertain to Internet address. Due to flat addressing structure, MANET nodes in MANET cannot differentiate between wired and MANET nodes. The APN must maintain an entry in the routing table for each of the diverse Connections. Due to hidden forces, the mobile node sends a route request every time and forwards the packets to new wired node. This causes flood of route request in MANET and routing tables will be increased (Alba Batlle Linares et al., 06 June 2008).

This common solution, although hypothetically is reasonable but causes inefficiency with mobility. Even if an additional APN is available fewer hops away from the mobile node, packets are always routed through the same APN. New path may not be best path and having low bandwidth. In a more practical scenario, the node changes the APN either it losses the link or improve the
performance. The nearest APN are chosen to minimize the MANET traffic and mobile nodes are allowed to change the APN, resulting in breaking of connections. Wired node only identifies the IP address of the gateway, which plays a role as proxy for mobile node. Mobile IP is used to continue the TCP connection. If we use Mobile IP that each node in MANET preserves the same IP address of the (APN) to which it is attached. If Mobile IP is not used, it creates another problem, what should be an IP address of mobile node in wired segments. In this connection lot of research has been done on MANET connectivity, although some solutions are already available in literature (C. Jelger et al., April 2004) & (R. Wakikawa et al. Mar 2006) but still needs to improve given in (P.M. Ruiz et al., October 2005).

3.1. Addressing
The scheme of assigning IP address is unique but assigned IP addresses to mobile nodes and it does not mean to be interconnected with wired infrastructure directly. As IPv6 has some extensive capabilities, e.g. extended address space and address auto configuration over IPv4, which some schemes use it more easily to avoid the variations between flat structures and hierarchical (P.M. Ruiz et al., October 2005).

3.2. Flat and Hierarchical Addressing
The major problem is hierarchical addressing in wired IP network and nodes belong to subnets are connected with (APN). IP address not only plays a role as unique identifier but also manages the resources for the nodes, which are part of network hierarchy. Therefore each node routes the segments through APN. Nodes, which possess their IP addresses, take the decision to route the packets to APN. Routes and links are changed frequently in MANET due to high mobility and increasing the number of nodes. Flat address topology is used, if IP address is only employed as unique identifier in MANET. Common address field of an IPv4 header is given in figure 3.

Figure 3. Showing Address field of IPv4 Header.

3.3. Address Auto configuration
Address auto configuration is the process by which a node configures its IP address. DHCP server performs the functionality to assign IP-addresses in conventional wired IPv4. DHCP server has low overhead in fixed network because congestion will not create problem (Alba Batlle Linares et al., 06 June 2008). There is different situation in MANET because network partitions and merging occur frequently whenever nodes move. It is not simple process to obtain DHCP server due to flat addressing and mobility.

3.4. Routing Protocols in MANET
Different routing protocols are used in MANET due to mobility and flat addressing structure. Some routing protocols are obtained from wired protocols and some are evolved on need and fitness basis (S. Pal Chaudhuri et al., 2000). MANET protocols follow some particular rules and having adaptive nature due to scalable and dynamic topology. The main challenge, which routing protocols face is to inform about the current topology. Routing protocols are classified into three categories, which are reactive, Proactive and hybrid. The classification of routing protocols is made on this base how they find suitable route to reach existing destination, as some MANET routing protocols are given in figure 4.
3.5. DSR routing protocol

DSR is simple and an efficient on-demand reactive routing protocol, which is designed for multiple wireless-ad-hoc networks based on source routing. DSR performs route discovery, route maintenance and guarantees loop free routing. Each node maintains route cache. Routing overhead of DSR can automatically be scaled (Xin Yu et al., December 20, 2004). Whenever mobile node sends the packets, it checks its own route cache. If route is found in cache to reach for destination node then all the packets are destined. In case route is not found then it forwards route request message to other node with destination address. DSR does not depend on other fundamental protocols in the network. It has no periodic routing advertisement, neighbor detection and link status sensing (David. B et al., 2001).

If we compare DSR with other routing protocols as DSDV and OLSR, which are table driven proactive routing protocol. In DSDV, each node maintains a routing table for next hop up to destination node and broadcast updates periodically. In case of OLSR, it selects neighbor node as multi point relay (MPR) and forwards the traffic (Philippe Jacquet et al., 18 November 1998). AODV is upgrading of DSDV and it is considered as table driven protocol (Xin Yu et al., December 20, 2004). The basic purpose of DSR is to determine short-cut routing paths if possible (C. Venkatesh et al., 2005). We use TCP Variants with DSR routing protocol in hybrid network to analyze the performance for each TCP variants.

4. Connectivity Process of APN Architecture

The section presents outline of initial connection setup and handoff process for Manet Mobile Node (MMN). Figure 5 shows timing diagram and describe the signals involved in it. Initial connection setup and MMN hand off process can be defined in the following steps. Initially the nodes, which are the part of Manet, intending to communicate with corresponding node (CN). They should establish initial connection setup and send the message through Current MANET Anchor Point Node (CMAPN) “Request for connection setup with CN”.

a. When CMAPN obtains the Request for connection setup from MMN and forwards the message “coordination request for connection setup” to Infrastructure Based Anchor Point Node (IBAPN). In response CMAPN also sends back message “Reply for connection setup with CN” to (MMN). When MMN obtains the message from CMAPN then it will be waiting till initial connection is established.

b. IBAPN forwards the message “forwarding coordination request for connection setup” to respective HA/FA within wired area. HA/FA informs the IBAPN with message “Accept coordination request” to CMAPN.

c. HA/FA forwards message with “forwarding initial connection setup” to (CN) and sends back response to IBAPN “Accept forwarding coordination request”. When CN receives the message then inform the HA/FA “Accept initial connection setup” with (MMN). With establishment of initial connection setup between CN and MMN then data exchange process is started.

d. When MMN changes the location and moves to other MANET then it sends the request for handoff to new MANET Anchor point node (NMAPN) with message “request for joining”.

Figure 4. Overview of some MANET routing protocols
e. NMAPN sends the message “location change forwarding message” (LCFM) to IBAPN for informing the handoff process and similar message is forwarded to HA/FA and finally to CN for location update.

f. NMAPN forwards the LCFM to IBAPN and also “update” to CMAPN. In response, CMAPN sends acknowledgement (ACK) to NMAPN for location update.

g. When CN gets the message LCFM then it sets the connection again with MMN and message is forwarded with “new connection setup in change of location”.

h. With the establishment of new connection, the data exchange process is started.

![Figure 5: Initial connection setup and Hand off process.](image)

**Algorithm 1 (connection connectivity & data exchange)**

Input: MMNID: ID of Manet Mobile Node, Data ID: ID of the requested Data, |DHCP|: Assign temporary IP address to MMN |dtr|:data transmission rate, Free Channel: the free channel, Output: Issue a message to MMNID, (IBAPN,MAPN ): the IDs of the Infrastructure Anchor Point Node, Manet Anchor point Node

1: The MAPN finds the values of the following variables: v: Data v with ID = DataID, Last Regular Channel: channel ID of the last regular multicast of ‘s’ Last Regular Time: the final time of connectivity ’s’,W(s): the window size for ’s’ [s]: the transmission duration of ’s’

2: Set IBAPN = NULL, MAPN = NULL, and Workload=0

3: if none of the existing regular multicast currently serving ’s’ then

4: IBAPN = FreeChannel

5: MAPN = NULL

6: Jump to step 15

7: end if

8: skew = currentTime – LastRegularTime

9: if drr < |dtr| then

10: tempWL = dtr
11: else
12: tempWL = |s| − min{|dtr|, |s| − skew}
13: end if
14: if skew > W(s) then
15: IBAN = FreeChannel
16: MAPN = NULL
17: else
18: IBAN = LastRegularChannel
19: MAPN = FreeChannel
20: Duration of the data transmission
Workload = tempWL
21: end if
22: Issue a data message containing (IBAPN, MAPN) to MMN

5. Architecture Design of APN Hybrid network

Random Waypoint mobility model is used in Hybrid network, which combines the features of wired network with wireless and MANET in order to make reasonable communication. The nodes, which make the possible communication between different segments of network, are called APN. An APN can play a role as coordinator in the network. The APNs can be located on different positions. An APN of MANET has information about the nodes, and these nodes are assigned the IPs locally through Dynamic Host Configuration Protocol (DHCP) Server. An APN that is part of MANET is said to be MANET Anchor Point Node (MAPN) similarly, the node that is located at the area where wireless range becomes weak is called Infrastructure Based Anchor Point Node (IBAPN). Both APNs can play a role as coordinators and formulate possible communication for rest of nodes in fixed and MANET segment of network given in figure 3.4. We use Random Waypoint (RW) mobility model at different mobility ratios and speeds in this network. The MANET network is routed with Dynamic Source Routing Protocol (DSR), which is described in section 6.

Figure 6. Architecture Design of Hybrid Network
6. Overview of Random Waypoint Mobility Model

A detailed survey of mobility models can be found in (Camp et al., 2002) but the mobility models surveyed are mostly theoretical in nature and major variations in mobility patterns found in real world scenarios. The new mobility models were also introduced in (Bai et al., 2003), such as free Mobility models (FM), Manhattan Mobility model ((MM) and Reference Point Group Mobility model (RPGM). The Social Network Mobility model (SNM) is discussed in (Musolesi et al., 2004). The research community mostly uses the Random Waypoint (RW) Mobility model in Mobile Ad-hoc networks. RW Mobility model is also incorporated in ns2 and detail of this model is given in following Para:

The nodes move to random destination with given velocity by using normal or uniform distribution [Velocity minimum, Velocity maximum] when nodes reach the destination, they stop for the time given by the “pause” time. The pause time can be constant value or uniform distribution [0, time pause maximum]. After completion of pause time, mobile nodes decide the destination and direction randomly and this process continues till the simulation time ends.

7. Simulation Setup

Ns2.28 on Red Hat 8 is used for simulation. The Random Waypoint mobile scenario is generated. The simulator gives a proper model for signal propagation and transmission range is 250 meter. The sensing and interference range is 550 meter. TCP New Reno, Reno, Tahoe, SACK, Westwood and Vegas are simulated and investigated on the same network so as to ensure fairness and behavior of the TCP Variants. The length of packet is 1040 bytes including 40 bytes are overhead. In this simulation, 40 mobile nodes both in wireless and MANET segment of network are placed. As we check the mobility of MANET-nodes, which move within rectangular field of 600 *1200 meters. RW generates mobile scenario and start location of nodes. Constant values for pause time have been set, which are 10 seconds after each 50 seconds. Total simulation time is 300 seconds. The minimum speed of the node \( V_{\text{min}} \) is 0 m/sec and maximum speed \( V_{\text{max}} \) are 10 m/sec respectively.

The moving speed of node is randomly obtained through uniform division \([V_{\text{min}}, V_{\text{max}}]\). We run simulations, which cover combination of the pause time and moving speed of nodes (Tracy camp. September 2002). The percentage of mobility means how many mobile nodes move and resulting how many links break in the MANET. Hence 50% mobility shows 20 nodes move out of 40 nodes. We also find out the broken links. When simulation time is completed, the performance of simulated TCP Variants is calculated and underlying protocol is Dynamic Source Routing (DSR), which is reactive routing protocol which gives better performance for routing in multi-hop Mobile Ad-hoc Network (MANET) and produces minimum routing overhead. It has also capability to deliver approximately all originated data packets, even with perpetual, rapid movement of all nodes in the network. The major cause for better performance is that DSR functions completely on demand with no periodic motion of any type are required at any stage in the network (Xin yu et al., December, 2004).

8. Simulation Result

In this subsection, we discuss the results of simulated scenario.

8.1. Throughput for TCP Variants at different mobility ratios

We have simulated Hybrid network Scenario with network simulator-2 and analyzed the performance of TCP Variants (Reno, Vegas, New Reno, Tahoe, Westwood and SACK). We have collected acknowledged packets for each TCP Variants and analyzed the throughput performance. The figure 7 shows the throughput performance for each TCP Variants at the maximum speed of 5 m/sec with Random Waypoint Mobility model. The performance gradually decreases to each TCP Variants from 5% mobility to 50% mobility. The reasons for decreasing rate of delivered packet are mobility. since 2 nodes move in the network that less links breaks and takes less time to recover whereas 20 mobility nodes cause more time to recover the broken links.

The network topology of MANET segment of network is part of Hybrid network, the topology of MANET remains mostly dynamic and major affecting factors are radio channel fading and mobility of nodes. The mobility also degrades the performance of TCP Variants because mobility causes the change of routing information in network, which causes long RTT and repeated timeouts resulting takes long time in retaining. Due to mobility, the receiver gets out of order segments and resulting, the receiver generates Acknowledgements (ACK) only for highest in-order packets. This causes the duplicate ACKs and fast retransmission algorithm starts and congestion window reduces (Andereas Hafslund et al., 2004).

Therefore \( \text{ssthresh} = \max \left( \text{unacknowledged data}/2, 2 \_\text{MSS} \right) \) & \( \text{cwnd} = \text{ssthresh} + 3\text{MSS} \)

TCP Vegas delivers more packets and TCP Tahoe and Reno relatively send same amount of Packets. The reason of delivering the
more packets for TCP Vegas is that TCP Vegas retransmits the lost packets after receiving the 2 duplicate acknowledgements whereas other TCP Tahoe, Reno, New Reno, SACK and Westwood retransmit the segments after receiving the three duplicate acknowledgments but in some cases third (dupack) takes either long time or does not receive third dupack and timeout expires. It is advantage of Vegas over above TCP variants because TCP Vegas mostly retransmits the lost segments before Retransmission time out (RTO). The other reason is that TCP Vegas does not wait for loss to trigger congestion window (cwnd) reduction.

Vegas possesses interesting approach regarding the congestion because it estimates the level of congestion before it occurs rather try to avoid it. The level of congestion is measured on basis of sample RTT and size of sending window that is also the reason; the Sender estimates the current throughput against every RTT (Vassilis Tsaoussidis et al., 2002). TCP Tahoe and Reno have delivered fewer packets than rest of TCP Variants. TCP Tahoe faces the problem due to repeat of slow start phases on each dropped segment, particularly when error is transient and not constant. In this case congestion window shrinks and bandwidth cannot fully be utilized (Vassilis Tsaoussidis et al., 2002).

Fast Recovery algorithm for TCP New Reno can degrade the performance due to multiple losses of packets during single window because Fast Recovery algorithm can manage only single loss per RTT (Vassilis Tsaoussidis et al., 2002). Some TCP Variants have more difficulty to differentiate between loss and congestion in wireless environment such as IEEE 802.11. The performance degrading factor for TCP Reno and Tahoe is also size of congestion window because these variants cannot send data during the timeout period, if mainly packets loss occurs. The findings of our simulation were also validated by researchers in different papers.

TCP Reno and Tahoe avoid time outs in case of multiple consecutive losses occur. The major factor of degrading the performance for TCP Tahoe has no support of fast recovery algorithm. This algorithm causes to recover the lost segments frequently. Figure 8 shows the throughput performance of TCP Variants at the maximum speed of 10 m/sec. the performance is affected by increasing the speed and TCP New Reno is greatly affected, the trivial drops of packets have been analyzed; the reason of weak performance for TCP New Reno is also aggressive behavior of fast retransmission algorithm whenever duplicate acknowledgement (DupAck) are received and high mobility of nodes is available.

Due to aggressive behavior of fast retransmission algorithm, it is difficult to deliver the packets even partial Acknowledgements (ACKs) are received to sender. Multiple losses due to high mobility make the weak performance of TCP New Reno because multiple losses cannot be handled properly and network becomes more congested and packets start to drop quickly. Another performance degrading factor relates to TCP New Reno is to take one RTT to perceive each packet loss.
8.2. RTT Fairness of TCP Variants

We show fairness by sharing the bandwidth among different TCP variant according Round Trip Time (RTT) given in figure 9. There are numerous reasons for RTT fairness as one reason is to attain the equal bandwidth allocation where the different competing flows may allocate similar bottleneck. Variation of RTT utilizes more resources, which causes the discouraging throughput (Sangtae Ha et al., 2006). TCP Vegas gets higher throughputs than other TCP Variants because slow start and congestion recovery algorithms mostly influence the throughput.

Hence slow start and congestion recovery mechanism work in different way for each TCP Variants because TCP Vegas depends on difference of expected and actual throughput. The multiple losses can be retained by avoiding timeouts because the TCP Vegas retransmit the lost segments after receiving 2 (DupAck), as that is reason before timeouts, dropped segments are retransmitted and better throughput is obtained. Original feature for TCP Vegas is its congestion detection mechanism because it shows the problems concerning to fairness. In congestion avoidance, the congestion detection algorithm of TCP Vegas verifies each RTT that is benefit of TCP Vegas over rest of TCP Variants. Moreover TCP Tahoe, TCP Reno, TCP New Reno and SACK reduce the congestion windows more than once during the single RTT that is also reason for unfairness and producing minimum throughputs where as RTT of Vegas reduces only once during the RTT (Sonia Fahmy and Tapan Prem Karwa September 2001).
8.3. Time for In-order delivery of data for each TCP Variants

We have evaluated the performance for each TCP Variants given in figures 10 and 11 show the time, which spent particularly for in-order delivery of data at 5 m/sec and 10 m/sec. During the total simulation, different overheads consume the time; these overheads are control packets and routing overheads. The remaining time is spent for In-order delivery of data. The TCP Vegas has delivered more packets and resulting more time than other TCP Variants consumed for In-order delivery of data at the speed of 5 m/sec and 10 m/sec. TCP Tahoe and New Reno have spent minimum time for In-order delivery of data at the maximum speed of 5m/sec and 10m/sec respectively. There are different reasons behind this assumption. If more than one TCP flows get same wireless link resulting the performance of TCP is degraded.

Multihoming of Adhoc network creates the delay due to rerouting but it does not create problem for TCP (Andereas Hafslund et al., 2004). TCP Variants can have challenges by using IEEE 802.11b protocol in Hybrid network. If TCP sends the data through satellite links that major challenge is propagation time because propagation time for IEEE 802.11b is in few microseconds whereas satellite links have 20-30 ms in better case but it can be extended up to 150 ms in worse case. Due to long propagation delay, TCP timeouts occur and unnecessary retransmission continues (Adam Kaplan et al., 2002). Above are general reasons for loss of data but some particular reasons for TCP Vegas for better In-order delivery of data is its proactive approach because it determines to encounter the congestion that is much more resourceful than reactive approach.

The TCP Vegas does not depend only on packet loss as indication of congestion but it perceives the congestion before the packet losses occur. TCP Vegas makes three major changes, which are new Re-transmission mechanism, congestion avoidance and modified slow start. TCP Tahoe and New Reno degrade the performance due to using of Additive-Increase/Multiplicative-Decrease (AIMD) pattern because this paradigm increases the congestion window (cwnd) which gets the existing bandwidth and unexpectedly decreases the cwnd. The minimum time for In-order delivery of data for Tahoe is that it gets whole timeout interval to identify the losses of packets.

Figure 10: Time spent for in-order delivery of data for each TCP Variants at the speed of 5 m/sec.
8.4. Effect of Mobility at Goodput

We show an effect of mobility on each TCP Variants in Figure 12 & 13 from mobility ratio of 5% to 50% at the maximum speed of 5m/sec and 10m/sec. Table 2 shows the total loss in Goodput at different mobility ratios. An effect of mobility can be calculated as follows: Firstly we calculate the successfully delivered and acknowledged segments then on the basis of delivered segments (bytes); Goodput can be obtained as: \( \text{Goodput} = \frac{\text{Number of delivered segments (bytes)}}{\text{Total number of generated segments (bytes)}} \times 100 \). Hence, we calculate the Goodput for each mobility ratio and from first mobility ratio; the next mobility ratio is subtracted that is said to be as an Effect of mobility. Similarly an effect of mobility is calculated for each TCP Variants at different ratios. The graphs show the fluctuating trends for each TCP Variants. The minimum effect on Goodput has been analyzed for TCP Westwood at both speeds of 5m/sec and 10m/sec. TCP Westwood has delivered and acknowledged fewer amounts of bytes than TCP Vegas and TCP SACK but from other side, less effect of bytes has been calculated for TCP Westwood. The minimum effect for TCP Westwood is End-to-End estimation of the bandwidth. The window is adaptively set, if network congestion is experienced. Congestion window (cwnd) and slow start threshold (ssthresh) are set identical to the estimated bandwidth (BWE) (Luigi A. Grieco and Saverio Mascolo., 2004). It is the reason that TCP Westwood drops minimum bytes than rest of simulated TCP Variants. The other reason for generating the minimum bytes is its modified version like New Reno.

The fast retransmission algorithm of Westwood works like New Reno algorithm, if segment is lost that even congestion window is increased and lost segments are retransmitted, in case of high congestion, the segments are started to loss quickly. TCP Westwood waits for acknowledgment for retransmitted packets. Moreover TCP Westwood depends on base Round trip time (RTT) and this is also reason of decreasing the performance of TCP Westwood from segments point of view. TCP Tahoe from mobility ratio 40% to 50% at the maximum speed of 5 m/sec has been affected more than rest of TCP Variants.

The reason is that TCP Tahoe has no support for fast recovery algorithm because whenever more links are broken that network is congested and it takes more time to retransmit the lost segments. TCP New Reno becomes worse at the maximum speed of 10 m/sec and Goodput is highly affected from 30% to 50%. TCP New Reno is minor change of TCP Reno. It identifies multiple losses but in case of high congestion, TCP new Reno is unable to control the losses and resulting moves to slow start and bandwidth is not fully utilized. The aggressive behavior of New Reno is also another factor for affecting the Goodput.
Figure 12: Effect of Mobility on Goodput at the maximum speed of 5 m/sec with RW Model

Figure 13: Effect of Mobility at Goodput at the maximum speed of 10 m/sec with RW Model

Table 2: Total Effect of mobility at Goodput for TCP Variants with different speeds

<table>
<thead>
<tr>
<th>Variants</th>
<th>5m/sec</th>
<th>10 m/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reno</td>
<td>7.74761</td>
<td>8.834109</td>
</tr>
<tr>
<td>Vegas</td>
<td>6.627126</td>
<td>7.367635</td>
</tr>
<tr>
<td>New Reno</td>
<td>6.945812</td>
<td>9.078856</td>
</tr>
<tr>
<td>SACK</td>
<td>6.786934</td>
<td>7.573525</td>
</tr>
<tr>
<td>Tahoe</td>
<td>7.684417</td>
<td>9.0116</td>
</tr>
<tr>
<td>Westwood</td>
<td>6.016598</td>
<td>6.599608</td>
</tr>
</tbody>
</table>

8.5. End-to-End delay for each TCP Variants at speed of 5 m/sec and 10 m/sec

End-to-End delay is an average elapsed time for delivery of individual data packets. All possible delays are included and caused by routing discovery, transmission at the MAC layer and queuing at the interface queue, etc but successfully delivered packets are calculated. We show trend for each TCP Variants in figure 14 & 15. Vegas has minimum End-to-End delay at the speed of 5m/sec and 10 m/sec whereas TCP Reno and TCP Tahoe have almost similar maximum End-to-End delay at 5 m/sec. As at the speed of 10 m/sec, maximum delay has been analyzed for TCP New Reno. The reason for maximum End-to-End delay for TCP Reno and TCP Tahoe at...
the speed of 5 m/sec is weakness of fast retransmission algorithms. Since TCP Tahoe does not send instant (ACKs) and depends on commutative (ACKs). Therefore when packet is lost then it waits for timeout or pipeline is emptied. This causes high bandwidth delay. TCP Reno behaves like TCP Tahoe whenever multiple losses occur, consequently multiple losses are identified as single segment loss (Kevin Fall et al., 1996).

Another problem occurs with TCP Reno when the size of window is small; numbers of duplicates (ACKs) are not detected for fast retransmission and have to stay for coarse grained timeout. From other side, TCP New Reno performs weak by increasing the mobility and takes long End-to-End delay. Reason is Limitation of retransmitting single lost segment against per RTT, consequences large delay occurs in retransmitting the later lost packets in the window. From other side, if the sender is restricted by the receiver’s advertised window during recovery time, then the sender is unable to utilize the existing bandwidth successfully and takes long End-to-End delay. Minimum End-to-End delay for TCP Vegas is fairness of retransmission algorithm when segment is lost that TCP Vegas waits for 2 DupAck and retransmit the lost segments before expiry of timeouts and RTT of Vegas shrinks only once during the RTT.

![Figure 14: Average End-to-End delay for each TCP Variants at the maximum Speed of 5 m/sec with RW Model](image)

![Figure 15: Average End-to-End delay for each TCP Variants at the maximum Speed of 10 m/sec with RW Model](image)

8.6. Control overhead at different speeds with RW Model

“This is the ratio between the total numbers of control packets generated to the total number of data packets received during the simulation time” (Asad Amir Pirzada et al., 2006). Control overhead contains control packets, which are used to set up a path to the destination, maintaining and repairing the routes. Control packets are Route Request (RREQ), Route Response (RREP) and Route Error (RERR). Table 3 & Figure 16 show the trend of control overhead at different mobility ratio.

All Reactive protocols illustrates good performance in control packet overhead especially DSR demonstrates significant performance
in control overhead than AODV (Jong_Mu and Young_Baeko 2004).

It is very hard to discover a functional route to destination when speed increases. Contention and congestion due to the overflowing behavior of DSR protocol dominate the effect of the speed. Whenever speed increases that extra routes are needed in DSR. The overhead of control packets increases significantly as speed increases. Hence more route request segments and route error are transmitted at the higher speeds (Tracy Camp et al., 2003). Whenever mobility ratio and speed increase that more links break, resulting many control packets are required for route discovery. Due to increasing of mobility ratio and speed that more segments travel over non-optimal routes with larger hop counts, which may be accumulated in a route cache. As a result, these segments will experience longer End-to-End delay and causes the creation of many overhead (control packets) (David oliver jorg et al., 2003).

DSR also creates higher control overhead packets because it often uses corroded routes due to the large route cache, which causes frequent segments retransmission and very high delay times (Farhat Anwar et al., August 2008). DSR is appropriate for networks in which mobile nodes travel at reasonable speed but not higher. If speed of nodes is increased, resulting more control overhead (control packets) is produced (Samba Sesay et al., 2004). The behavior of routing protocol, increase in mobility ratio and speed of mobile nodes are three factors, which creates more control overhead (control packets). The byte overhead in each packet also affects the total performance of network.

Table 3: Control Overhead (Number of control Packets) at different speeds with RW Model

<table>
<thead>
<tr>
<th>Mobility</th>
<th>5m/sec</th>
<th>10 m/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>445</td>
<td>661</td>
</tr>
<tr>
<td>5%</td>
<td>669</td>
<td>992</td>
</tr>
<tr>
<td>10%</td>
<td>1003</td>
<td>1488</td>
</tr>
<tr>
<td>15%</td>
<td>1223</td>
<td>2976</td>
</tr>
<tr>
<td>20%</td>
<td>1468</td>
<td>3574</td>
</tr>
<tr>
<td>25%</td>
<td>2201</td>
<td>4286</td>
</tr>
<tr>
<td>30%</td>
<td>3074</td>
<td>5185</td>
</tr>
<tr>
<td>35%</td>
<td>3688</td>
<td>6741</td>
</tr>
<tr>
<td>40%</td>
<td>4796</td>
<td>8089</td>
</tr>
<tr>
<td>45%</td>
<td>5855</td>
<td>10515</td>
</tr>
<tr>
<td>50%</td>
<td>8632</td>
<td>14722</td>
</tr>
</tbody>
</table>

Figure 16: Control Overhead (Number of control Packets) at different speeds With RW Model
8.7. Number of Broken links due to different mobility rates

We show an average broken links for all TCP Variants at the speed of 5 m/sec and 10 m/sec given in Table 4 & Figure 17. These broken links are calculated at the 50% mobility. When MANET nodes want to establish the sessions to obtain internet services from wired segment of network then routing protocols start route discovery process. Route Request packet (RREQ) is broadcasted into network to obtain any single appropriate route to destination.

When route request packet is reached to destination, in response route reply packet (RREP) is sent to originator RREQ. If a link is broken due to mobility and speed of middle nodes, a route error packet is sent to the destination. Meanwhile destination finds another route. The process is repeated until the reply reaches the target. Therefore, destination finds another route if any error occurs in current route. This process causes delay in packet delivery. The high mobility and speed makes more broken links due to discovery of route. The high mobility and speed continuously change the direction of node, inconsequence more links break. Due to increase of speed, topology changes rapidly and more links are broken particularity in DSR when more connections are established between the nodes.

Table 4: broken links at different speeds with RW Model

<table>
<thead>
<tr>
<th>Mobility</th>
<th>Broken links at 5m/sec</th>
<th>Broken links at 10 m/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>10%</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>15%</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>20%</td>
<td>17</td>
<td>26</td>
</tr>
<tr>
<td>25%</td>
<td>25</td>
<td>34</td>
</tr>
<tr>
<td>30%</td>
<td>31</td>
<td>41</td>
</tr>
<tr>
<td>35%</td>
<td>38</td>
<td>46</td>
</tr>
<tr>
<td>40%</td>
<td>42</td>
<td>51</td>
</tr>
<tr>
<td>45%</td>
<td>47</td>
<td>55</td>
</tr>
<tr>
<td>50%</td>
<td>52</td>
<td>61</td>
</tr>
</tbody>
</table>

Figure 17: Broken links at different speeds with RW Model

9. Discussion of Results.

We have simulated TCP Variants in ns-2 over Hybrid network, which consists of MANET, Wired & Wireless segments of network. We have increased mobility in MANET segment of network by using DSR routing protocol and analyzed that by increasing the mobility and speed the performance of each TCP Variants gradually decreases. DSR incurs several problems relating to cache usage. Multiple routes obviously give benefits but creates disadvantage due to high mobility. In larger networks, the source-routing principle can also generate a trouble. It has been observed that TCP Vegas performed better than other TCP Variants. It produces healthy throughput, better in-order delivery of data, minimum End-to-End delay, and good RTT at different mobility ratios and speeds. The
major reasons for degraded performance of TCP Reno, Tahoe and New Reno are timeouts, as during the timeouts period, Variants cannot resend the lost segments whereas TCP Vegas does not wait for loss to trigger cwnd decrease and calculate approximately the current throughput during each RTT. Westwood greatly miscalculates the existing bandwidth, which is potentially troublesome for fairness and can lead to starvation of simultaneous connections Performance evaluation of TCP Westwood+ (Mascolo.s et al., April 2003). That is the reason to produce lesser throughput than TCP Vegas. Hence Losses in wired network are due to overflow of buffer at routers.

TCP Reno, Tahoe and New Reno have been designed particularly for wired network and meet the requirement of IEEE802.11 but their performance become weak in Hybrid network especially satellite link is involved. Minimum mobility ratios create less control overhead, which causes the better performance for each TCP Variants, which is also proved in our simulation. TCP experiences most losses in multi hop wireless networks, which are caused by packet drop at wireless link layer IEEE 802.11. To improve the performance, new congestion control variants have been introduced and various schemes are incorporated. Explicit congestion Notification (ECN) has been incorporated to improve the congestion control. If congestion occurs in network then that intermediate routers will mark the congestion experience (CE) code point in header of TCP. This message will inform the end hosts that network is congested resulting unnecessary packet drops can be prevented (Andres Hafslund et al., 2004).

10. Conclusion & Future Work

Mobile Adhoc networks (MANET) can be deployed to many locations without the use of infrastructure support. In military environment, disaster situation, scattered educational institutions need such networks to route data packets through dynamically mobile nodes. MANET is better choice for these extremely mobile and dynamic applications, which are not supported by centralized administration. If internet services are required than MANET provides better solution with integration with wired network to construct a Hybrid network in order to obtain an Internet facility. To investigate the performance of different transmission control protocols, we have done simulation in ns-2 by using Random Waypoint Mobility model and analyzed different metrics such as throughput, RTT fairness, In-order delivery of data, an effect of mobility on Goodput, End-to-End delay and control overhead. We have particularly focused on MANET & wired portions of network to investigate the performance of TCP Variants. The minimum effect of mobility has been analyzed on TCP Westwood and reasons are already discussed in detail but it delivers lesser segments than TCP Vegas and SACK whereas TCP Vegas has better throughput, minimum End-to-End delay, better In-order-delivery of data and improved RTT. TCP SACK also performs better and does not loss many segments because sender is informed which segment has been received. TCP SACK uses SACK blocks at receiver side to indicate the contiguous block of data successfully received. The sender can find out through SACK blocks which segments are lost, as this is the reason to control the loss of segments frequently. TCP Reno, TCP New Reno and Tahoe degrade the throughput in high mobility ratios and take more End-to-End delay time as compare to other TCP variants. In future, we will analyze and evaluate TCP Variants in Hybrid network with respect to different mobility models including social network model, Random Walk Mobility Model, Random Direction Mobility Model, City Section Mobility Model etc. We would study underutilized and congested network conditions by maximizing traffic flows. We would also analyze multi-homing issues in future. Finally we suggest if the features of TCP Vegas and TCP Westwood are combined that new Variant could be better from mobility point of view in MANET and mixed environments.

REFERENCES


Antonio Annese, Nedo Celandroni, Franco Davoli, Erina Ferro, Alberto Gotta.,TCP performance measured over heterogeneous wireless networks with high delay-bandwidth product, Work funded by the European Commission in the framework of the (Sat NEx) NoE (contract no. 507052,Italy,2004.


Allman. et.al. TCP congestion control, (RFC 2581),1999.


(CISCO Network Academy., 1995).


Vassilis Tsaoussidis and Ibrahim Matta., Open Issues on TCP for Mobile Computing, NSF grant Career ANI-0096045, USA, 2002.
