A Comparison of the TCP Variants Performance over different Routing Protocols on Mobile Ad Hoc Networks

S. R. Biradar [1], Subir Kumar Sarkar[2] , Puttamadappa C[3]
[1]Sikkim Manipal Institute of Technology, Majitar, Rangpo, East Sikkim -737 132, SIKKIM
[2] Electronics and Telecommunication Enginnering Department,
Jadavpur University, Kolkata- 700 032 WEST BENGAL
[3] Electronics and Communications Department,
SJBIT, Bangalore, KARNATAKA

Abstract: We describe a variant of TCP (Tahoe, Vegas), TCP is most widely used transport protocol in both wired and wireless networks. In mobile ad hoc networks, the topology changes frequently due to mobile nodes, this leads to significant packet losses and network throughput degradation. This is due to the fact that TCP fails to distinguish the path failure and network congestion. In this paper, the performances of TCP over different routing (DSR, AODV and DSDV) protocols in ad hoc networks was studied by simulation experiments and results are reported.

1. INTRODUCTION

Recent advances in wireless communication and portable devices have resulted in the rapid growth of mobile wireless networks leads to the exponential growth of the cellular network, which is based on the combination of wired and wireless technologies. Wireless[1] networks are fast becoming popular as they allow users to remain connected when they are moving. The wireless network can be either be infrastructure based or infrastructure less (ad hoc) networks.

A Mobile Ad Hoc Network (MANET) is considered an autonomous collection of wireless mobile nodes that are capable of communicating with each other without the use of a network infrastructure or any centralized administration[5]. Due to the host mobility, the network topology may change rapidly and unpredictably over time. The network is decentralized no administrator is required to manage network. It is self organizing and enables communication in situations where there is no time to set up the necessary infrastructure or situations where the need for a communication network is temporally required. MANETs have a wide range of applications form military to search and rescue operations during disaster. The interest of the scientific and industrial community in the area of telecommunications has recently shifted to more challenging scenarios in which a group of mobile units equipped with radio transceivers communicate without any fixed infrastructure.

I. **Transmission Control Protocol** (TCP)

Transmission Control Protocol[11,2,3] is the Internet's most widely used transport control protocol. TCP's strength lies in the adaptive nature of its congestion avoidance and control algorithm and its retransmission mechanism, first proposed by V. Jacobson[6,7] as a part of TCP Tahoe. It was further refined in Reno and New Reno versions of TCP. TCP Vegas[9], proposed by Brakmo et al. proposes a fundamentally different congestion avoidance scheme from that of TCP Tahoe. The major control mechanisms of TCP are its congestion avoidance and congestion control mechanism. They are discussed in detail below:

Slow Start: Before TCP can send data at a fast rate, it needs to estimate the bandwidth available. If this is not done, the throughput of the TCP connection will drastically decrease, as the intermediate routers would have to queue or drop the packets form buffer, if buffer is full. The slow start mechanism adds a new parameter that controls the rate at which packets are sent, congestion window denoted by cwnd. When a new TCP connection is established, the initial value of cwnd is set to a value less than or equal to 2*Maximum Segment Size (MSS), but not greater than two segments. Every time an ACK segment is received, the cwnd is increased by one segment. Thus, when an ACK arrives that acknowledges the first packet, the cwnd is increased to two and two data segments are sent. When ACKs for these two segments arrives, the cwnd is increased to four. It provides an exponential increase to the cwnd parameter. The TCP sender can send up to the minimum of receiver's advertised window and its own value of cwnd. TCP remains in this exponentially increasing slow start phase as long as cwnd value is less than or equal to slow start threshold (ssthresh)

Congestion Avoidance: Congestion avoidance is the algorithm used by TCP to avoid losing packets, if packets are lost. TCP performs congestion avoidance[4,8,12] when cwnd is greater than ssthresh. In the congestion avoidance phase, the cwnd is increased by 1 full-sized segment every round-trip time (RTT). Congestion avoidance continues until congestion is detected. Congestion can be detected in two ways:

1) Receipt of duplicate acknowledgment
2) Due to time timeout.

If the detection is done using the retransmission timer

timeout, the value of ssthresh is updated as follows:
ssthresh = max (Flight Size / 2, 2*MSS)

Flight Size is the amount of outstanding data in the network. Then cwnd is set to 1 segment. After the packet has been retransmitted (whose timer had expired), TCP starts again in the slow start mode using the above mentioned algorithm to raise the cwnd from 1 to the new value of ssthresh, after which, congestion avoidance again.

However, instead of the retransmission timer expiring, if three duplicate ACKs (three duplicate ACK is equal to four identical ACKs without the arrival of any other packet in between) were received at the sender, TCP assumes that it is an indication of packet loss. It then uses what is called the "fast retransmit" and "fast recovery" algorithm to recover the packet and fill the network again. The

TCP sender does the following on the arrival of the third duplicate ACK:

Set ssthresh to the value given in the equation above.

Retransmit the lost packet and set cwnd to ssthresh + 3*MSS (fast retransmit).

For each additional duplicate ACK that is received, cwnd is incremented by one MSS.

If this new value of cwnd and receiver window allows, transmit new segment(s).

When the ACK that acknowledges the receipt of a new segment is received, set cwnd to ssthresh (fast recovery).

TCP Vegas

TCP Vegas was proposed by Brakmo et al. It has a very different congestion control algorithm compared to New Tahoe. TCP Vegas[10] in general controls its segment flow rate based on its estimate of the available network bandwidth. Among the many new features implemented in TCP Vegas, the most important difference between it and TCP Tahoe lies in its bandwidth estimation scheme. Studies on TCP Vegas have shown that Vegas achieves higher efficiency than Tahoe, causes fewer packet retransmissions.

TCP Vegas dynamically varies its sending window size based on fine-grained measurement of RTTs, whereas TCP Tahoe continues to increase its window size until packet loss is detected. While TCP Tahoe views packet losses as a sign of network congestions, TCP Vegas uses a sophisticated bandwidth estimation scheme, wherein it uses the difference between the expected and achieved flow rates to estimate the available bandwidth in the network. The idea is that when the expected and actual throughputs are almost equal, the network is not congested. In other words, in a congested scenario, the actual throughput will be much smaller than the expected. Thus, based on this estimate of network congestion, TCP Vegas updates the congestion window from the following equations:

1) expected_rate = cwnd(t)/base_rtt
   Where cwnd(t) is the current congestion window size and base_rtt is the minimum RTT of that connection.
2) actual_rate = cwnd(t)/rtt
   Where rtt is the present round-trip time
3) diff = expected_rate – actual_rate
   The source estimates the backlog in the router queue from the difference
4) Using this value of diff, the congestion window value (cwnd) is adjusted as:
   cwnd +1 if diff < α
   cwnd –1 if diff > β
   cwnd = cwnd otherwise

When expected and the actual throughput are close to each other, the connection may not be utilizing the available network bandwidth, and hence should increase the flow rate. On the other hand, when the actual throughput is much less than the expected throughput, the network is probably experiencing congestion and hence the connection should reduce the flow rate.

Impact of mobility on TCP performance:

TCP suffers from throughput degradation because of mobility in MANETs. Route failure is one of the key characteristics of a mobile ad hoc, whenever the route fails, a TCP connection that is using the route can potentially reduce congestion window.

Each TCP variant has its own advantages or disadvantages. Which TCP variant provides is efficient in MANETs scenarios? The performance comparison of TCP variants on the same set of test scenarios should be investigated. The results are helpful for congestion control protocol designer.

Performance Metrics:

Packet Delivery Ratio: The ratio between the number of packets originated[5] by the application layer FTP sources and the number of packets received by the FTP sink at the final destination.

Routing Overhead: The total number of routing packets transmitted during the simulation. Each hop wise transmission if a control message by a node is counted as one transmission.

End-to-End Throughput: Average successful transmission rate measure of the number of packets successfully transmitted to their final destination per unit time.

Average Delay: The average time a packet takes to reach its destination.(delay between the time when a data packet is given to IP layer at the source node and the time when the packet arrives at the IP layer of the destination). This can only computed for packets that are successfully delivered.

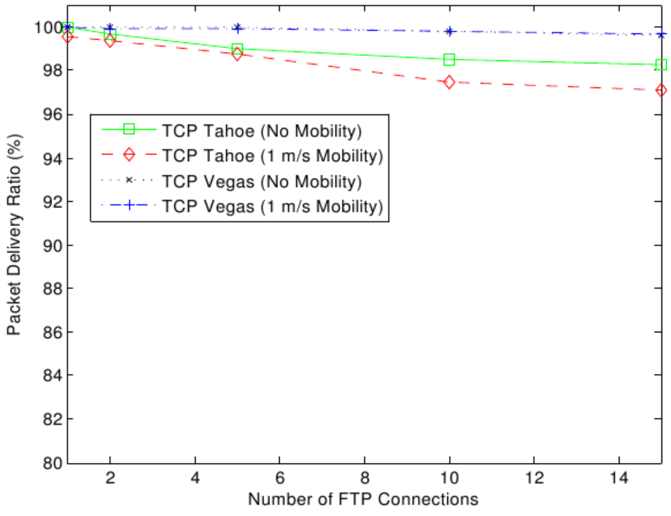
MAC Overhead: The number of routing , ARP, and control packets (RTS, CTS and ACK) transmitted by the MAC layer essentially it considers the both routing and MAC control overhead. It is similar routing overhead, this metric is also accounts for transmission at every hop. Large routing and MAC control overhead impacts both throughput and delay, it also causes network congestion.

Simulation Environment:

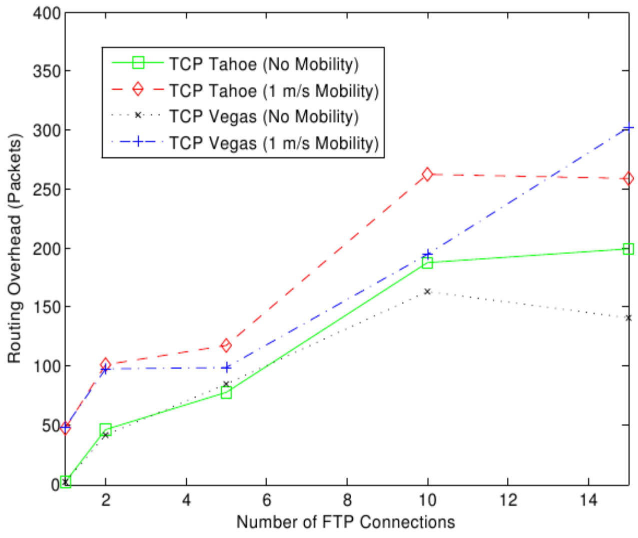
The experiments were conducting using the ns-2 network simulator[13], developed at the University of California at Berkeley, with the wireless extensions provided by the CMU Monarch Project. The radio model is based on the Lucent Technologies WaveLAN 802.11 product, providing a 2 Mbps transmission rate and transmission range of 250 m. The link layer modeled is the Distributed Coordination Function (DCF) of the IEEE 802.11 wireless LAN standard. In this simulation 25 mobile nodes moved in an area of 500m x 500m for a period of 500 seconds. Random Waypoint (RW) mobility pattern was generated using the setdest tool which is a part of the ns-2 distribution. The maximum speed Vmax was set to 0, 1, 2, 5 and 10 m/sec to generate different movement patterns for same mobility model. The traffic pattern was generated by the cbrgen tool that is part of the ns-2 distribution. The traffic consisted of 1, 2, 5, 10 and 15 FTP connections. The source destination pairs were chosen at random. The data rate used was 8 packets/sec, window size 32 and the packet size was 512 bytes. The three thresholds α, β and γ are set to be 1, 3 and 1, and parameter *p* in slow start phase is set to be 1/8 as default.

Presentation and Results:

PDR and routing overhead graphs are achieved under DSR, AODV and DSDV routing protocol are shown in figure 1, 2 and 3 respectively.



a) Packet Delivery Ratio



b)Routing Overhead

Figure 1(DSR)

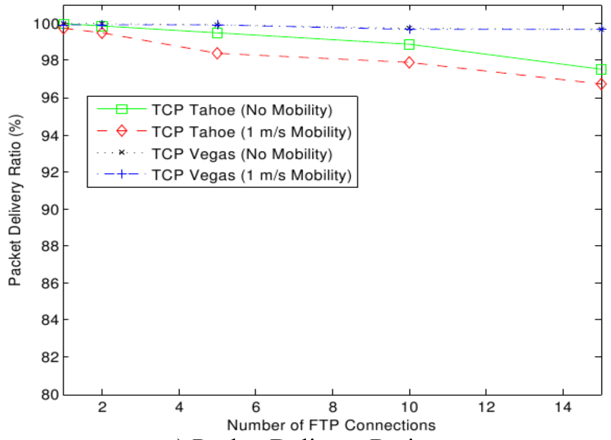
Figure 1a shows the PDR a function of the number of FTP sources for the two different values of mobility (no mobility, and 1m/s mobility).

PDR is maximum for TCP Vegas compared to TCP Tahoe.

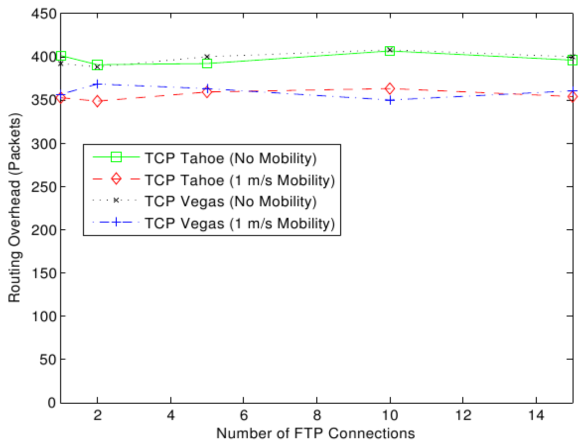
FTP Traffic has little impact on TCP Vegas, where as TCP Tahoe get affected by both mobility as well as traffic.

Routing overhead shown in Figure 1b. It shows the increase of the traffic correspond to increase in routing overhead. However the overhead without mobility lower in both the cases.

It shows the increase of the mobility to increase in packet drop approximately 50 packets. Control overhead is higher in Tahoe than Vegas.

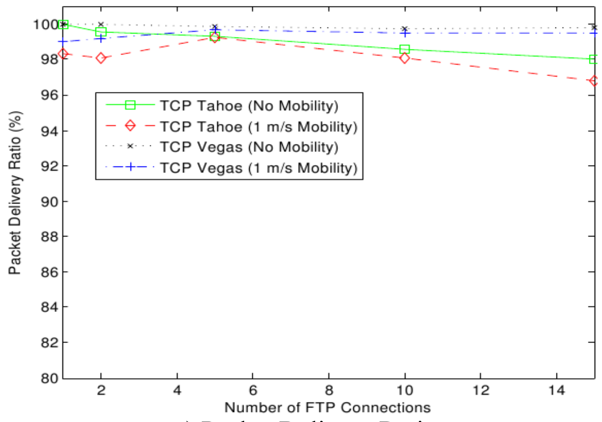


a) Packet Delivery Ratio

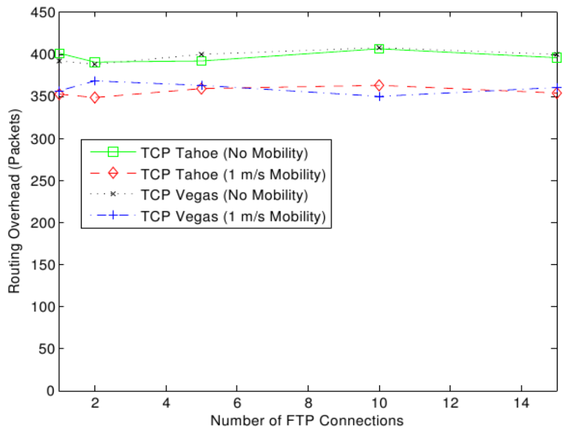


b) Routing Overhead
Figure 2 (AODV)

TCP Vegas PDR was higher than TCP Tahoe. The figure 2a shows that the ability of the protocol to deliver the packet to their destination degrades as the traffic increase in Tahoe.
Vegas packet drop increase slower then Tahoe. In vegas mobility model has little impact on packet drop. Routing overhead increase with respect to number of FTP connection.



a) Packet Delivery Ratio



b) Routing Overhead
Figure 3 (DSDV)

The PDR graph in Figure 3a show that constant PDR. The RTR graph shows the increasing in mobility increase the traffic approximately 8-10%. The routing overhead in Vegas and Tahoe on different mobility is very similar in both the cases (no mobility and 1m/s mobility).

CONCLUSION

Reactive routing protocol overhead increase as FTP connections increase, in case of proactive routing protocol overhead is almost constant. PDR decrease as number of FTP connection increase in reactive protocol. TCP do not affect the proactive routing protocol. Overall Vegas perform compared to Tahoe.
Coparsion of Tahoe vs Vegas shown in table 1.

Table 1

	Tahoe	Vegas
Slow Start, *cwnd* updated with every ACK as	*cwnd*+ 1	Increase every other RTT
Congestion avoidance, *cwnd* update with every ACK as	cwnd+ 1/cwnd	Linear increase if Diff< α Linear decrease if Diff> β
Change from slow start to congestion avoidance when	cwnd= ssthresh	Diff< γ
Fast recovery	none	Retransmit with ACK if RTT>timeout
ACK format required	ACK	ACK

References

[1] A. Bon, C. Caini, T. De Cola, R. Firrincieli, D. Lacamera, M. Marchese, An integrated testbed for wireless advanced transport protocols and architectures, in: Proc. IEEE TridentCom, Barcelona, Spain, Mar. 2006, pp. 522-525.
[2] Allman, M., Floyd, S., and Partridge, C., "Increasing TCP's Initial Window," RFC 2414, September 1998.
[3] Allman, M., Paxson, V., and Stevens, W., "TCP Congestion Control," RFC 2581, April 1999.
[4] Floyd, S., "TCP and Explicit Congestion Notification," ACM Computer Communication Review , Vol. 24, No. 5, October 1994.
[5] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva.. A performance comparison of multi-hop wireless ad hoc network routing protocols. In
Proceedings of the International Conference on Mobile Computing and Networking (MobiCom'98), Oct. 1998
[6] Jacobson V., "Congestion Avoidance and Control," ACM Computer Communication Review , Vol. 18, No. 4, August 1988.
[7] Jacobson, V., Braden, R., and Borman, D., "TCP Extensions for High Performance," RFC 1323, May 1992.
[8] L. Xu, K. Harfoush, I. Rhee, Binary increase congestion control for fast long distance networks, in: Proc. IEEE INFOCOM, Hong Kong, March 2004, vol. 4, pp. 2514–2524.
[9] L.S. Brakmo and L.L. Peterson, TCP Vegas: End to end congestion avoidance on a global internet, IEEE J. Select. Areas Commun. 13 (1995), pp. 1465–1480.
[10] L.A. Grieco and S. Mascolo, Performance evaluation and

comparison of Westwood+, New Reno, and Vegas TCP congestion control, ACM Computer Commun. Review 34 (2004), pp. 25–38.
[11] Postel, J., "Transmission Control Protocol," RFC 793, September 1981.
[12] S. Floyd, T. Henderson, A. Gurtov, The NewReno modification to TCP's fast recovery algorithm, IETF RFC, 3782, 2004.
[13] The ns Manual. The VINT Project. A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.