Performance Study of TULIP Protocol over 802.11 Wireless Networks

Kanwalvir Singh Dhindsa
Dept. of CSE & IT
BBSBEC, FGS, Punjab, India

Parminder Singh
Dept. of IT
CEC, Landran, India

Gurpal Singh
Dept. of CSE & IT
BBSBEC, FGS, Punjab, India

ABSTRACT

The TCP performance degrades over the wireless networks due to non congestion related problems but performs well in case of wired networks. The current network scenarios consist of combined wired and wireless links which demands for investigating the performance in such heterogeneous scenarios. The non congestion losses on the wireless link are interpreted by TCP as congestion losses and eventually congestion control algorithms are invoked. This leads to high end to end delay and decreased application throughput. This paper investigates the TCP performance improvement using TULIP protocol. In the proposed algorithm, the TULIP communicates with MAC protocol which provides efficient and reliable frame exchange over the shared wireless channels. TULIP also uses RTS-CTS strategy for decreasing collision over the wireless links. The protocol also uses fast recovery and retransmit algorithm to recover the lost packets for achieving higher TCP performance. The proposed algorithm has been implemented using NS2 simulator and result obtained thereof has been presented in the form of important TCP metric. Results showed that the proposed algorithm results in performance improvement over the existing Split connection and End to End technique.

General Terms

Wireless Networks, TULIP, Network security.

Keywords

Slow start, congestion avoidance, RTT (Round Trip Time), E2E (End to End).

1. INTRODUCTION

IEEE 802.11 based wireless networks are plagued with the problem of packet losses due to collisions, interference and the random access mechanism. When TCP traffic is deployed over such networks; severe performance degradation is experienced as all the link layer losses may manifest as congestion at the transport layer. Therefore, the previous algorithms are not effective enough to prevent these problems in the TCP over wireless link environment. One possible solution to such problems is to develop solution based on cross layer approach where local link layer losses are taken care of separately so that the performance at TCP layer remains unaffected. TULIP is a transport unaware link improvement protocol that improves the performance of TCP in wireless networks. The protocol works on the link layer which provides a reliable service for TCP protocol and unreliable for UDP protocol and is also called unaware protocol. The ability of unaware protocol is to maintain local recovery of all lost packets at the wireless link in order to prevent the unnecessary and delayed retransmission of packet over the end to end path and a subsequent reduction in TCP's congestion window. This is similar to the aware ability which hides the link layer losses from TCP. The TULIP protocol improves the overall performance of wireless networks and is effectively and reliable in case of TCP but not UDP. TULIP uses delayed ack strategy to improve TCP performance. TULIP uses RTS-CTS strategy in 802.11 MAC so that data packets would suffer less collisions resulting in higher throughput. The table 1 shows different Algorithm approach used by the TULIP protocol.

Table 1: TULIP Algorithmic approach

Protocol	*Algorithm Approach*			
	Slow Start	*Delayed Acknowledgment*	*Fast Retransmit and Recovery*	*RTS-CTS Strategy*
TULIP	Y	Y	Y	Y

2. LITERATURE SURVEY

A brief literature survey of the area related to this work has been conducted in this section. In H. Balakrishnan et al.(1995), authors have classified the schemes into three categories namely End to End protocols where the sender is aware of the wireless link, Link layer protocols that provide local reliability and split connection approach that breaks the end to end connection into two parts at the base station.

Work in H. Balakrishnan et al. (1995), conducts experimental study of the approach in both LAN and WAN environments using throughput and goodput as metrics for comparisons. Results show that an unaware link layer protocol with some knowledge of TCP provides good performance.

Christina Parsa and J. J. Garcia-Luna-Aceves (2000), authors discuss the performance of TCP in heterogeneous Network and improving TCP congestion control. This work also also surveys the different possible metric characteristics of the connection path. It also discuss the TCP performance such as packet loss, round trip time, non congestion losses, bandwidth delay and give proposed solution.

Congestion can occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. The Congestion Avoidance algorithm assumes that the packet loss caused by damage is very small (much less than 1%). Author described in this RFC to avoid the congestion using congestion avoidance algorithm (Zhenghua Fuxiaoqiao Meng et al., 2005). Authors have described various mechanisms to get high throughput for any version of TCP sender using a SACK based receiver (H. Balakrishnan et al., 1997).The base station delays the sending of the duplicate ACKs to the TCP sender by an amount of time d. At the same time it retransmits the lost segment locally. Further duplicate ACKs are also delayed. If an ACK arrives indicating that the retransmitted packet has been received, then the duplicate acknowledgments are not sent on to the TCP sender. Author uses delayed duplicate acknowledgment in unaware approach, a different sequence numbers for its packets than does TCP for its segments (Biaz S. et al., 1997).

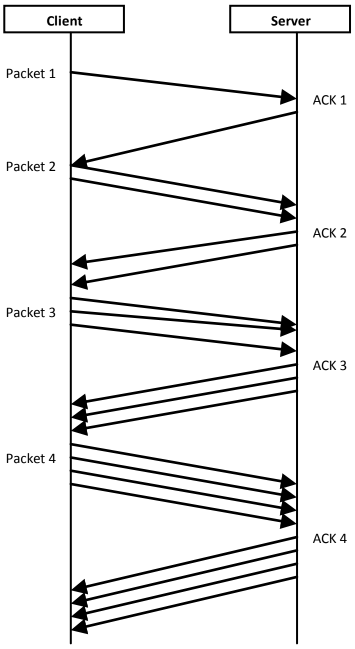
3. WORKING OF TULIP PROTOCOL

The TULIP uses slow start and congestion avoidance algorithms to improve the performance of TCP. TULIP protocol works on the link layer and provides a reliable service for TCP protocol and is transport unaware protocol. The ability of unaware protocol is to maintain local recovery of all lost packets at the wireless link in order to prevent the unnecessary and delayed retransmission of packet over the end to end path and a subsequent reduction in TCP's congestion window. This is similar to the aware ability which hides the link layer losses from TCP. TULIP uses delayed acknowledgement strategy to improve TCP performance. The protocol uses local retransmission in the sender to receiver station for effective buffering at both the ends. This further adds an RTS-CTS strategy to avoid any collision in between the channels. The advantage of RTS-CTS strategy is to find the hidden terminals in the network. The detailed description of the algorithms is in the following section.

3.1 Slow Start Algorithm

This algorithm is used to control the data transmission. It works when the amount of data is being injected in to the network and increases the congestion window exponentially, rather than linearly (V. Jacobson, 1988). The source starts out by setting CongestionWindow (cwnd) to one packet. When the ack for this packet arrives, TCP adds 1 to CongestionWindow and then sends two packets. Upon receiving the corresponding two ACKs, TCP increments CongestionWindow by 2 one for each ACK and next sends four packets. The end result is that TCP effectively doubles the number of packets it can transit every RTT. Figure 1 shows, the growth in the number of packets in transit during slow start.

Figure 1: Slow Start Algorithm



3.2 Congestion avoidance Algorithm

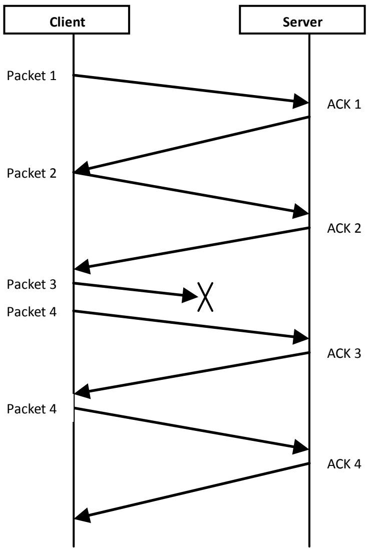
Congestion can occur when data arrives on a big pipe and gets sent out through a smaller pipe (Christina Parsa and J. J. Garcia-Luna-Aceves, 1999). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets. The assumption of the algorithm is that packet loss caused by damage is very small (much less than 1%), therefore the loss of a packet signals congestion is somewhere in the network between the source and destination.

3.3 Proposed Algorithm

In the proposed approach, TULIP protocol alongwith the slow start and congestion avoidance algorithms has been used to improve the performance of TCP over the wireless networks. This approach provides more throughput on the path and prevents the packet retransmissions. The client sends the request to the receiver where as receiver sends the acknowledgment along with the data and vice-versa. The figure 2 shows the packet transmission through delayed acknowledgment. When cwnd is less than ssthresh; then it increments the window by one and acknowledgment is received but when cwnd increases ssthresh then cwnd should be half.

```
if (l_window > cwnd){ cwnd = cwnd + 1/cwnd;
return}Else if (finwin – cwnd < max)
cwnd += (finwin-cwnd)/cwnd;
else cwnd += Smax/cwnd;
if (finwin > cwnd){ minwin = cwnd;
target_win =(max_win+min_win)/2;
}
```

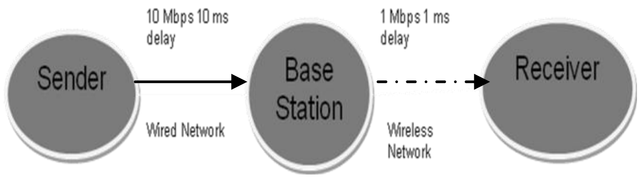
Figure 2: Fast retransmit based on duplicate acknowledgment



4. EXPERIMENT SETUP

This Scenario has been implemented in wired cum wireless networks as shown in figure 3 using NS-2 simulator. In this scenario, TULIP provides local recovery using link level buffers at the base station from where the packets move from wired to wireless portion of the networks. A TULIP agent is deployed at the base station for loss detection and loss recovery via retransmission.The TULIP agent supresses duplicate acknowledgments from the receiver to the sender and performs retransmission of the lost packet. This is achieved by detecting the duplicate acknowledgment to identify packet loss and performs local retransmission as soon as this loss is detected.The new packet arriving from the sender side are buffered and fowarded in the regular manner by the TULIP agent.

Figure 3 : Scenario of TULIP Protocol



This scenario has been implemented using otcl and c++ language using NS2allinone 2.33 simulator. TULIP agent has been implemented on Base station (BS). A TCP connection has been established between sender and MH (mobile host or receiver node). We are using TCP Reno to implement it because it gives reasonably good performance with the wired networks (H. Balakrishnan et al., 1995). The platform used in this experiment is Fedora Linux version 9. Table 2 shows the simulated parameters for the TULIP protocol.

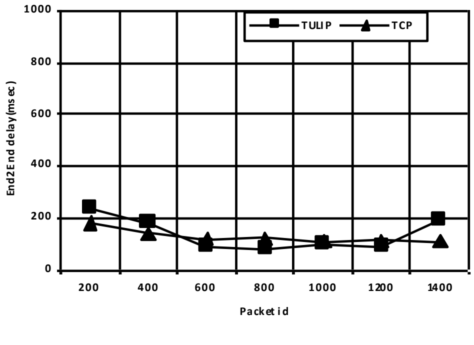
Table 2 : Summary of simulation parameters

Simulation Parameters	*Value*
Simulation length	9.897seconds
Packet Size	1460 bytes
Buffer Size	8192 bytes
Forward transfer rate	79.38 Bps
Time of each hop	42.6 ms
No. of Nodes	5

5. RESULTS AND ANALYSIS

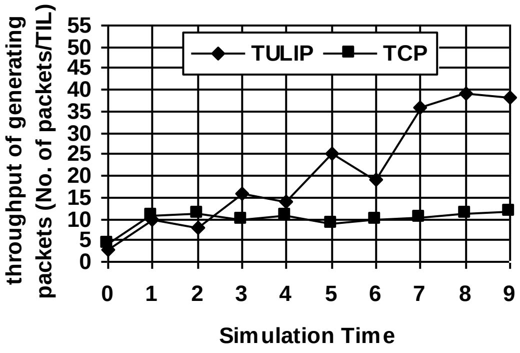
This section discusses the results obtained during simulation of the scenario. Figure 4 shows the end to end delay suffered by the packets for both TCP and TULIP protocols. It shows end2end delays between current node (sender) and other node (receiver) for simulated time. The result shows that E2E delay incurred using TULIP is almost comparable to TCP. In TULIP the packet processing time is higher as compared to the base TCP protocol and it contributes to the overall end to end delay.

Figure 4: Simulation of Packet Id vs. End to End Delay



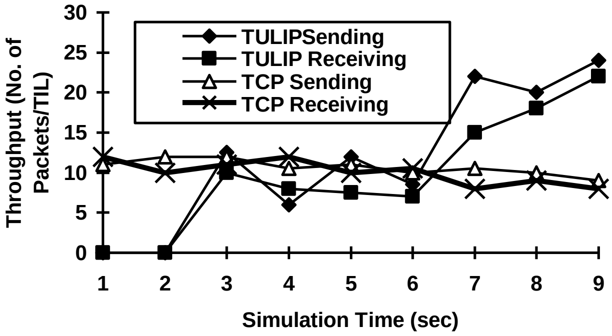
The figure 5 shows the results for Throughput. The results indicate that throughput is higher in case of TULIP as compared to the TCP protocol. This higher throughput is due to hiding the losses on wireless network from TCP by the TULIP protocol. The losses at the wireless network and taken care of by the TULIP protocol and TCP doesnot require invoking the congestion avoidance algorithm for losses occuring over the wireless link. The appreciable gain of increased throughput using TULIP protocol is clearly indicated by the results in figure 5.

Figure 5 : Throughput of generated packets



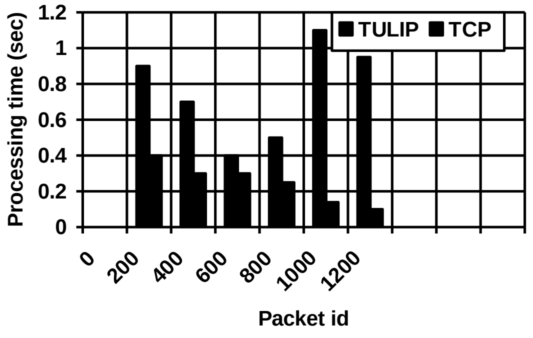
The Figure 6 shows, the packets sent and received the wireless network by TULIP and TCP protocols respectively. Results shows improvement in throughput in both the packets sent and received using TULIP protocol in wireless networks.

Figure 6: Throughput of sent and received packets



The figure 7 shows the processing time for each packet sent to the current node. In the figure, the processing time for TULIP protocol is slightly higher as compaerd to the base TCP protocol.

Figure 7 : Packet id vs. processing



6. CONCLUSION

TCP protocol was originally designed for reliable wired networks and shows performance degradation in presence of non-congestion losses of wireless origin. This work studies the usage of TULIP protocol for TCP performance improvement in wireless networks. It can be concluded that implementation of TULIP protocol in wired cum wireless networks results in increased throughput. TCP Reno flavor has been used on both ends and acknowledgment delay related problems ahave been discussed. In the future work, usage of TULIP Protocol in multihop wireless networks and mesh networks can also be investigated.

7. REFERENCES

[1] A. DeSimone, M.C. Chuah, and O.C. Yue (2003) ‘Throughput performance of transport-layer protocols over wireless LANs’, IEEE GLOBECOM, pp. 542–549.

[2] Anurag Kumar, D.Manjunath and Joy Kuri (2008) Wireless Networking ,Morgan Kaufman,USA

[3] Azzedine Boukerche (2006) Handbook of Algorithms for Wireless Networking and mobile computing, London: Chapman and Hall/CRC.

[4] B. Braden, ed.(1989), ‘Requirements for Internet Hosts Communication Layers’, RFC 1122, pp.81-85.

[5] Biaz S. and Vaidya N (1997) Technical Report for TCP over Wireless Networks using Multiple Acknowledgements, Texas A&M University.

[6] Christina Parsa and J. J. Garcia-Luna-Aceves (2000), ‘Improving TCP congestion over Internets with heterogeneous transmission media’, In Proceedings of the 7th IEEE ICNP, pp.213-21.

[7] Christina Parsa and J. J. Garcia-Luna-Aceves (1999), ‘TULIP: A Link-Level Protocol for Improving TCP over Wireless Links’, Proc. of the Wireless Communications and Networking Conf. New Orleans: IEEE, pp. 1253-1257.

[8] D. Clark, M. L. Lambert, and L. Zhang (1988) ‘A High Throughput Transport Protocol’, ACM SIGCOMM, pp.35-49.

[9] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz (1995) ‘Improving TCP/IP Performance over Wireless Networks’, ACM/IEEE Transaction on Networking, pp.756-769.

[10] H. Balakrishnan, S. Seshan, M. Stemm, and R.H. Katz (1997) ‘Analyzing Stability in Wide-Area Network Performance’, ACM SIGMETRICS’, pp.2-12

[11] H. Balakrishnan, V. Padmanabhan, and R. Katz (1999) ‘The Effect of Asymmetry on TCP Performance”, Mobile Networks and Applications (MONET)’, Vol. 4, No. 3, October, pp. 219-241

[12] H. Balakrishnan, Srinivasan Seshan, Randy H. Katz (1997) ‘A comparison of mechanisms for improving TCP performance over wireless links’, IEEE/ACM, pp. 756-69.

[13] J. B. Postel and J. Reynolds (1985) ‘ File Transfer Protocol (FTP)’ , RFC-821,Information Sciences Institute, Marina del Rey, CA.

[14] V. Jacobson (1988) ‘Congestion Avoidance and Control’, Computer Communication Review, Vol. 18 No. 4, August, pp.314-329.

[15] Zhenghua Fuxiaoqiao Meng, Songwu Lu, Lixia Zhang, Mario Gerla (2005) ‘The Impact of Multihop Wireless Channel on TCP performance’, IEEE 2005, pp.1300-15.