Incorporation of TCP Proxy Service for improving TCP throughput

G.P. Bhole and S.A. Patekar

***Abstract*--TCP slow start algorithm works well for short distance between sending and receiving host. However as the distance in two host increases, the TCP throughput decreases. In this paper, an attempt has been made to improve the TCP throughput by splitting the TCP connection in two parts and incorporating a predeployed service at the intermediate node as Proxy Transport Service (PTS) node. Simulation for the said scheme and the vis-à-vis standard TCP has been done. Influence of the three key parameters namely: PTS node location, PTS service time, and PTS buffer capacity are examined for normal as well as congested conditions. The results show that there is a substantial improvement in throughput of the order of 30 to 50 percent which is not achieved with conventional methods such as network cache.**

***Index terms*--Active network, congestion window, proxy service, PTS, rtt, TCP, throughput**

I. INTRODUCTION

TCP (Transmission Control Protocol) is reliable, has error and flow control mechanism and uses sequencing for reordering. Congestion window (cwnd) size is adjusted according to the rate of arrival of acknowledgment. TCP is congestion aware protocol. The slow start feature of TCP which is used for congestion control causes performance degradation on high delay links. TCP works well for short connection paths but for the longer distance hosts, the TCP throughput decreases.

One alternative to improve throughput is network caches. Network caches [1] have been use to reduce the response time to user request and improve the throughput. But this works under the presumption that the cached copy of the requested data is found available on network node and hence quantification of explicit improvement is obscured.

[1]Performance Enhancement Proxies (PEP) [1] uses transport proxies in networks to improve TCP throughput. PEP is a transparent service offered at predetermined location in the network. PEP monitors all TCP flow going in and out of network and manipulates their TCP ACK packets to improve TCP throughput. However PEP, transport proxies are to be deployed on the edges on TCP unfriendly networks.

Proxy Transport Service (PTS) [2] is active networks based service, which improves the TCP bulk data throughput for TCP connections with large rtt. PTS splits the TCP connection between sender and receiver in two separate TCP connections: *PTS connection-1*, between sender and PTS node and *PTS connection-2* between PTS and receiver. PTS node is an active node located in the path between the source and destination host such that it is approximately equidistant in terms of delay from each of the hosts. PTS node acknowledges the sender's TCP segments even before they have reached the receiver, thus reducing the rtt on *PTS connection-1*, thereby increasing the sender's transmission rate.

PTS does not reduce the end-to-end delay for a TCP segment but it improves the rate at which the sender's congestion window increases. PTS is not a replacement for TCP; on the contrary it complements TCP to improve its performance for long delay connections. It is possible to deploy the services in side the network. Current research in the areas of security, active node architecture, active packet formats and active network applications will provide the essential components required to deploy PTS [3] [4] 5]

In this paper Proxy Transport Service (PTS) is used. Simulation is carried out on a considered network. Results show that there is a substantial improvement in throughput.

.

II. PROPOSED SCHEME

In PTS two separate connections are established, one between the source host and a designated intermediate node (the PTS node), and other between the PTS node and the destination host. All the TCP segments from the sending hosts are acknowledged to the source host, and forwarded to the destination host by the PTS node as shown in fig 1 below.

During the setup of PTS session, a requested amount of buffer space is provided for that session inside the PTS node. For every TCP segment received by the PTS node, a copy of the segment is stored in the buffer and an acknowledgment is sent to the source host. PTS node in turn transmits accepted TCP segment to the destination host on other PTS connection. The copy of the TCP segment is preserved in the buffer until an acknowledgement is received from the destination host. The PTS node also takes care of retransmission of unacknowledged segment on other PTS connection when timeout occurs or duplicate ACK's are received.

G.P. Bhole, Veermata Jijabai Technological Institute, Mumbai (email: gpbhole@vjti.org.in)
S.A. Patekar, Veermata Jijabai Technological Institute, Mumbai (email: sapatekar@vjti.org.in)

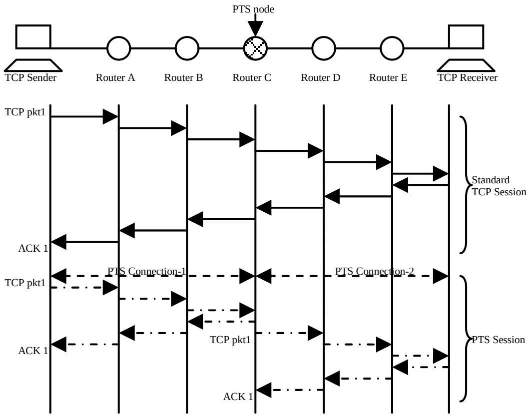

Fig. 1. Proxy Transport Service

When a new TCP connection is established the sender initializes its cwnd to 1, and increases cwnd by one whenever an acknowledgement is received. The bulk data throughput of the TCP connection is affected by flow control and slow start algorithm.

The slow starts period lasts longer for TCP than PTS connection. This is due to reduced rtt for each of the PTS connections. Also rate of increase of cwnd during congestion avoidance is longer in case of PTS because the ACK's arrive faster. Therefore more number of packets can be send using PTS with in the same time frame due to reduced rtt for each PTS connection. The extra delay that a packet suffers with in the PTS node is offset with the transmission of previously processed packets at the node thereby not affecting throughput.

A. *Setup for simulation*

A topology used for simulation is as shown in fig 2. Network Simulator [6] was used for simulating PTS and standard TCP for performance testing.

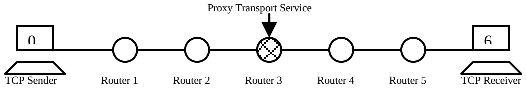

Fig. 2. Simulation Topology

The two hosts are separated by the variable number of hops. All the links between the two hosts are of equal capacity and propagation delay. A Tahoe TCP agent is attached on the two hosts. The traffic between the two hosts is generated by running ftp application. PTS functionality is provided in the form of an agent, which extends functionalities of the sender side and receiver side TCP agent. Any node in between the sender and the receiver can become a PTS node by attaching a PTS agent to that node. PTS node uses a single FIFO queue for every session. To represent a more realistic scenario, the service time of the PTS node is exponentially distributed over mean service time T_s. The ftp application sends unlimited packets from the sending host to the receiving host. Fixed sized packets (1000 bytes) are used throughout the simulation.

In the simulation example following parameters are considered. All the links between the two hosts have the same capacity (Lc) and propagation delay (Ld). PTS node location (X), PTS service Time (Ts) and PTS buffer capacity per session (Cmax). The topology consists of two hops separated by variable number of hops (Nhops). The location of the PTS node is indicated by X, which refers to the number of hops between the PTS node and the source host. TCP

Congestion window (*cwnd*) and transmitted sequence number (*seqno*) are the two parameters of the sender side TCP stack which give a good idea of the transmission characteristics of the TCP session. *cwnd* and *seqno* are used to compare standard TCP connection with the two separate TCP connections established using PTS. Both *cwnd* and *seqno* are measured in terms of packets. In all the figures TCP refers to standard TCP connections without PTS. When PTS is used, there will be two separate TCP connections. The TCP connection between the sender and the PTS node will be referred to as *PTS connection-1* and the TCP connection between the PTS node and the receiver will be referred to as *PTS connection-2*. TCP throughput (number of segments received from the receiving host per unit time) is used to measure the performance.

III. COMPARISON OF TCP AND PTS

Case I: Congestion window and throughput comparison without congestion for TCP and PTS

Following parameters are considered.

$L_c = 10Mb$, $L_d = 10ms$, $T_s = 100\mu s$,

1) $N_{hops} = 8, X = 4$

Graph shows congestion window v/s time, for TCP and PTS

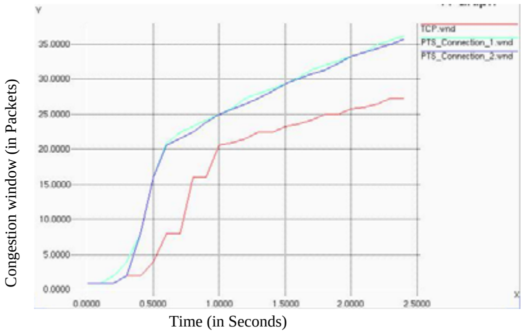

Fig. 3. Congestion window comparison for TCP and PTS

TABLE 1
THROUGHPUT COMPARISON FOR TCP AND PTS WITHOUT CONGESTION

	Number of packets transmitted in 2.5 seconds	Throughput (Mb/s)
TCP	230	0.736
PTS	510	1.632

Remark**:** It can be observed from the graph that there is substantial improvement in congestion window and hence in throughput. From Table 1, increase in throughput = 54.9 %

Case II: Congestion window and throughput comparison of TCP and PTS in presence of congestion in 1st half.
Following parameters are considered.

a) $L_C = 10$Mb, $L_d = 10$ms, $T_S = 100\mu$s,

b) $N_{hops} = 8, X = 4$

Graph shows congestion window in packets v/s time for above parameters with congestion in first half.

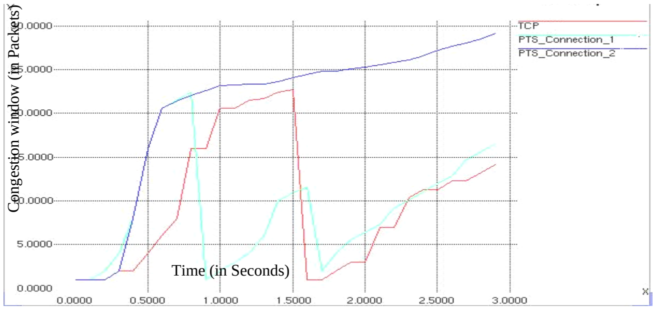

Fig. 4. Congestion window comparison of TCP and PTS in presence of congestion in 1st half

TABLE 2
THROUGHPUT COMPARISON FOR TCP (END TO END) AND PTS IN PRESENCE OF CONGESTION IN 1ST HALF

	Number of packets transmitted in 3 seconds	Throughput (Mb/s)
TCP	173	0.461
PTS	287	0.765

Remark: fig 4 graph shows performance of PTS and TCP sessions when timeout occurs or 3 duplicate ACKs are received indicating packet loss. Packet loss occurs in between sender and the PTS node. When packet loss is detected by sender it sets cwnd to 1, the cwnd of PTS and receiver is not affected. PTS node keeps transmitting to the receiving hosts as long as it has unsent segment in its buffer. From Table 2 it can be seen that increase in throughput = 39. 74 %.

Case III: Congestion window and throughput comparison of TCP and PTS in presence of congestion in 2nd t half. Parameters considered are

$L_C = 10Mb, L_d = 10ms, T_S = 100\mu s,$

$N_{hops} = 8, X = 4$

Graph shows congestion window in packets v/s time for above parameters with congestion in second half.

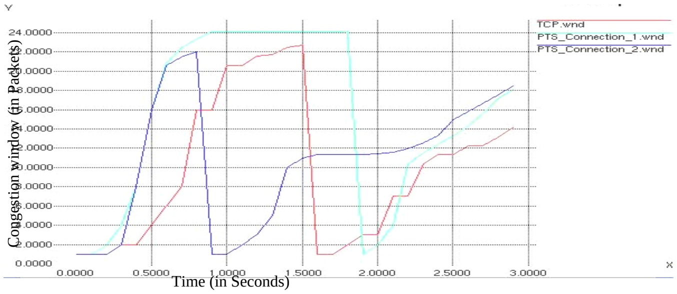

Fig. 5. Congestion window comparison of TCP and PTS in presence of congestion in 2nd half

TABLE 3
THROUGHPUT COMPARISON FOR TCP AND PTS IN PRESENCE OF CONGESTION IN 2ND HALF

	Number of packets transmitted in 3 seconds	Throughput (Mb/s)
TCP	173	0.461
PTS	252	0.672

Remark: for momentary congestion PTS node will recover its steady state transmission rate for the session much faster than standard TCP. From the table 3, the increase in throughput is 31.4 %

Case IV: Effect of PTS node location

Location of the PTS node along the path between the sending and receiving hosts determines the end-to-end performance of the PTS session.

X stands for distance of the PTS node from the sending host in terms of hops. X=0 represents standards TCP session. PTS node location is proportional to the rtt.
Following parameters are considered.

c) $L_C = 10$Mb, $L_d = 5$ms, $T_S = 100\mu$s,

d) $N_{hops} = 8$, $C_{max} = 1024$

As rtt is proportional to the distance, rtt1:rt2 gives proportional distance of PTS connection 1 and PTS connection 2. Thus ratio rtt1:rtt2 is appropriate for the location of PTS. Graph show throughput at receiver for various PTS node position.

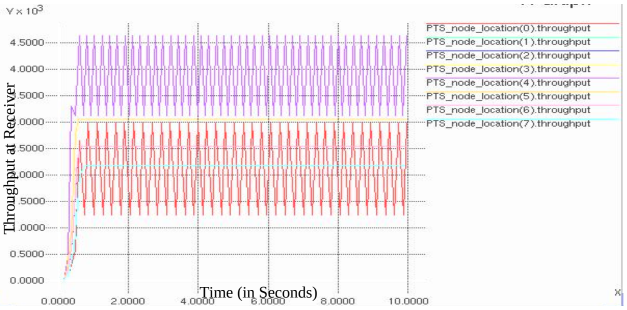

Fig. 6. Effect of PTS node location on TCP throughput:

TABLE 4
EFFECT OF PTS NODE LOCATION ON TCP THROUGHPUT

rtt1: rtt2	Number of packets transmitted in 10 seconds	Throughput (Mb/s)	% Increase
Standard TCP	2210	1.768	
1 : 7	2530	2.024	12.65
2 : 6	2970	2.376	25.59
3 : 5	3570	2.856	56.56
1 : 1	4470	3.576	50.56
5 : 3	3570	2.856	56.56
6 : 2	2970	2.376	25.59
7 : 1	2530	2.024	12.65

Remark: PTS node near center gives maximum throughput.

Case V: Effect of PTS delay (service Time)

Every TCP segment will be delayed by bounded amount of the time inside the PTS node i.e. PTS service time (Ts). This time includes waiting and service time to modify the TCP header.

Following parameters are considered.

L_c = 10*Mb*, L_d = 8*ms*, N_{hops} = 8, X=4, C_{max}=400*pkts*

Graph shows throughput at receiver for different PTS delay

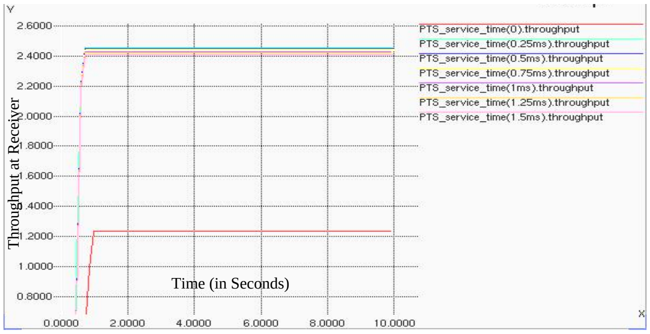

Fig. 7. Effect of PTS delay on TCP throughput

TABLE 5
EFFECT OF PTS DELAY ON TCP THROUGHPUT

PTS service time	Number of packets transmitted in 10 seconds	Throughput (Mb/s)	% increase in throughput
Standard TCP	1410	1.128	
0.25ms	2850	2.280	50.53
0.5 ms	2844	2.275	50.42
0.75ms	2830	2.264	50.18
1 ms	2817	3.254	65.33
1.25ms	2810	2.248	49.82
1.5 ms	2790	2.232	49.46

Remark: PTS can provide good performance when the mean service time is in the range of milliseconds.

Case VI: Effect of PTS buffer size

The amount of buffer space provided per session (Cmax) by the PTS node can determine the transmission rate of the sender. The effect of the buffer capacity is examined and results are as shown below.

Following parameters are considered.

L_c = 10*Mb*, L_d = 8*ms*, T_s = 100*μs*, N_{hops} = 8, C_{max}=100*pkts*

Graph shows throughput at receiver for different PTS buffer size in terms of packets

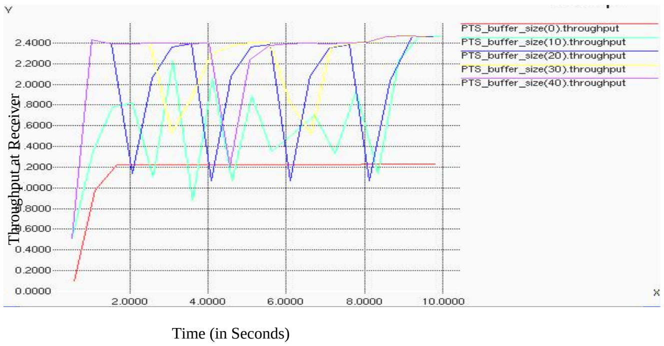

Fig. 8. Effect of PTS buffer size on TCP throughput

TABLE 6
EFFECT OF PTS BUFFER SIZE ON TCP THROUGHPUT

Buffer size (in Packets)	Number of packets transmitted in 10 seconds	Throughput (Mb/s)	% increase in throughput
Standard TCP	1390	1.112	
10	1948	1.558	28.63
20	2361	1.888	41.10
30	2602	2.081	46.56
40	2708	2.166	48.66

Remark: A small buffer size causes a PTS session to overflow quite frequently which ultimately reduces the transmission rate of the sender. With the appropriate buffer size, throughput can increase almost to double.

Case VII: Effect of multiple PTS nodes.

Following parameters are considered.

L_c = 5Mb, L_d = 2ms, T_s = *1ms*, N_{hops} = *14*

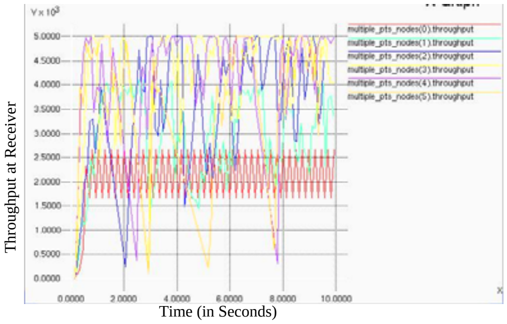

Fig. 9. Effect of multiple PTS nodes on TCP throughput

Graph shows throughput at receiver for different PTS nodes with sender and receiver 14 hops apart.

Remark: For one, two and three PTS node performance improves.

IV. CONCLUSION

In this paper it is demonstrated through simulation example that the Proxy transport service (PTS), can improve the TCP throughput of large *rtt* connections by thirty to fifty percent, which is a significant improvement over conventional techniques such as network cache and PEP even under congested network conditions.

PTS service is not a very resource intensive and does not affects the best effort traffic significantly.

Performance has been examined under the influence of key parameter namely: PTS node location (rtt_x), PTS service time (T_s) and PTS buffer capacity (C_{max}) and increase in throughput is not impaired.

V. REFERENCES

[1] Debojyoti Dutta and Youngguang Zhang. An active proxy based architecture for tcp in heterogeneous variable bandwidth networks. IEEE GLOBECOM, 2001.

[2] Siddharth Patil and Mohan Kumar, IWAN 2003, LNCS 2982 TCP Enhancement using Active Network Based Proxy Transport Service

[3] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden. A survey of active network research. IEEE communications Magazine, 35(1):80-86, 1997.

[4] David L. Tennenhouse and David J. Wetherall . Towards an active network architecture. Computer Communication Review, 26(2), 1996.

[5] K. Psounis. Active networks: Application, security, safety, and architectures. IEEE communications Surveys, pages 1-16, First Quarter 1999.

[6] Lee Breslau, Deborah, Estrin, Kevin Fall, Sally Floyd, John, Heidemann, Ahmed, Helmy, Polly Huang, Steven, McCanne, Kannan, Varadhan, Ya Xu, Haobo Yu. Advances in Network simulation. IEEE Computer, 33(5):59-67, May 2000.