CHEMICAL ENGINEERING TRANSACTIONS
VOL. 51, 2016
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.aidic.it/cet
Guest Editors: Tichun Wang, Hongyang Zhang, Lei Tian

ISBN 978-88-95608-43-3; **ISSN** 2283-9216
DOI: 10.3303/CET1651114

Performance Analysis of TCP in Long Propagation Delay Environment

Meihua Liu

Chongqing College of Electronic Engineering, Chongqing 401331, China
306110193@qq.com

Satellite-based Internet architecture has become the latest trend of development of broadband satellite communication. We construct different TCP modules and obtain the performance of TCP by OPNET simulator under different propagation delay. Simulation results show that TCP NewReno does not apply to the satellite network. However, after modification of the error control mechanism, TCP Reno exhibits better performance and applicability to long-delay satellite network.

1. Introduction

Satellite-based Internet achieves an integration of digital audio & video, satellite broadcast and the Internet. It provides an open and broadband network platform capable of broadcasting. TCP/IP architecture has four layers and is the most extensively used transmission layer protocol on the Internet. Allowing for end-to-end non-real-time data transmission, TCP/IP architecture supports basically all Internet applications (Misra et al., 2000; Mathis et al., 2001; Henderson et al., 2012). TCP has no requirements on lower layers, assuming that the lower layers only provide unreliable datagram services. The upper layers are applications and the lower layers are IP protocol. For the applications, TCP protocol can realize asynchronous data transmission (Hollot et al., 2010). To achieve connection-oriented, reliable data transmission in a not so reliable network, the priority is the control of feasibility and traffic so that multiple interfaces are provided for the applications and the data are provided for multiple applications. At the receiving end, TCP must have received buffers that preserve the received data not yet read. Moreover, TCP must solve the connection problem to be connection-oriented. Communication safety is also a concern for TCP (Jain and Dovrolis, 2003).
TCP has evolved through 3 stages, namely, original TCP protocol, TCP Tahoe and TCP Reno. TCP Tahoe uses three basic congestion control algorithms: slow start, congestion control and fast retransmission. TCP Reno is currently the most extensively used version and has a new mechanism developed on the basis of TCP Tahoe, which is fast recovery (Partridge and Shepard, 1997; Ratnam and Matta, 1998). TCP Reno uses cwnd adjustment for the control of the number of data packets to be transmitted within a RTT and the largest window to limit the maximum of cwnd. Here TCP module is constructed and its performance is simulated under different delay using OPNET. Based on the simulation, the TCP most adapted to satellite network can be invented.

2. TCP module in OPNET

OPNET simulator is used for performance analysis of TCP. A most advanced simulation tool, OPNET provides users with a series of simulation models for performance simulation of the networks.
TCP module in OPNET has the following features: (1) Connection establishment by three-way handshake; (2) Dynamic adjustment of the size of send window using the buffer resources at the receiving end, so as to achieve traffic control; (3) Rearrangement of the disordered data packets; (4) Some protocol algorithms available for choice (Ozkan et al., 2010; Jiang and Dovrolis, 2010).
OPNET's model library also provides a standard TCP module that calls two programs.
The first is TCP manager (Fig. 1), which supports continuous working of TCP. It receives all instructions from the lower layers and the applications.

Please cite this article as: Liu M.H., 2016, Performance analysis of tcp in long propagation delay environment, Chemical Engineering Transactions, 51, 679-684 DOI:10.3303/CET1651114

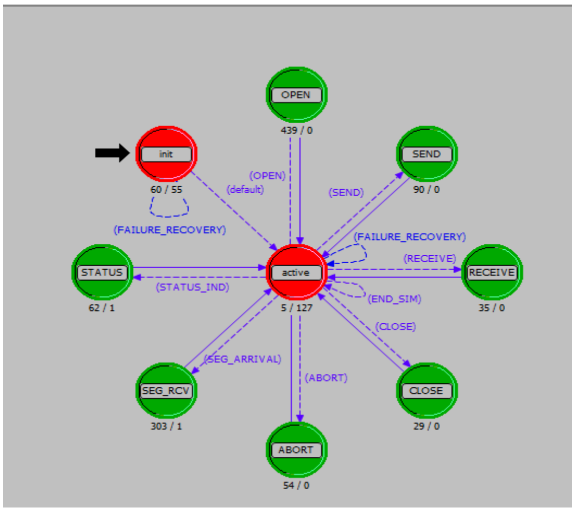

Figure 1: Working mechanism of TCP manager

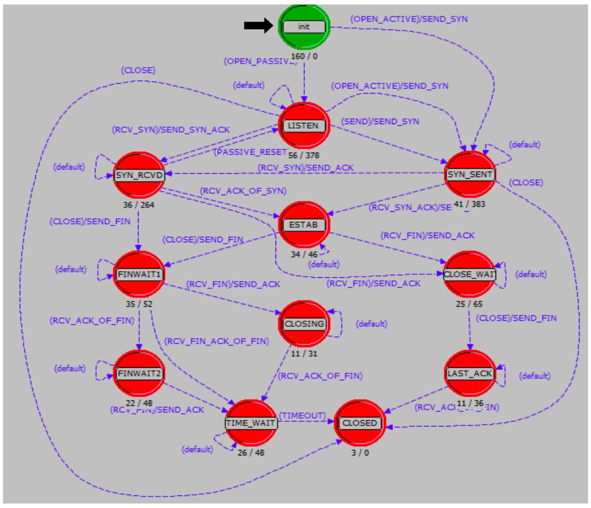

Figure 2: Working mechanism of TCP connection

The second is TCP connection, whose working principle is shown in Fig. 2. TCP connection is realized based on RFC793 and RFC1122, and it is the subprogram of TCP manager. To establish a new connection, TCP connection is called (Ludwig and Katz, 2000). The number of programs is equal to the number of connections in TCP connection. Hence every TCP module is capable of controlling multiple connections in a dynamic manner.

3. Construction of simulation platform

TCP modules are constructed using modularization design. Every TCP module consists of three functional blocks: data treatment block, congestion control block and error control block. The performance of TCP Tahoe, TCP New Reno and TCP Reno applied to the satellite network is simulated. The layered modularization design is more favorable for the extension of the system. The new TCP module implements the error control strategy and congestion control strategy under the three TCP protocols. The nodes in the wireless mesh network make a request of an FTP application to the wireless server node_0, and a file of 1K is sent to each node. The traffic sent by the wireless server node_0 and Wireless Lan throughput are estimated along with the WLAN throughput at the nodes in wireless mesh network. The simulation results are compared between the three versions of TCP (Ha et al., 2005).

3.1 Congestion control module

When the loss of data packets is confirmed, congestion control module will implement different congestion control strategies under different TCP versions.

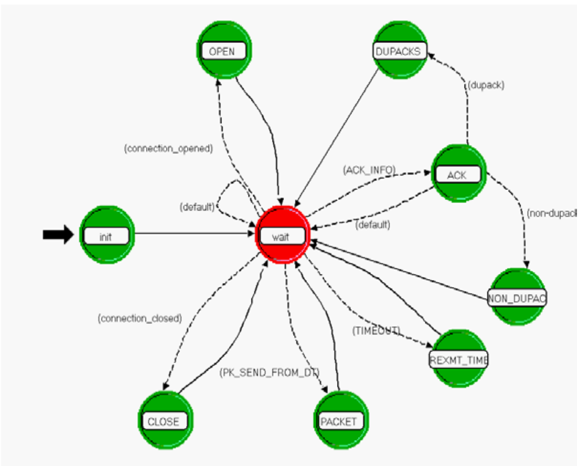

Figure 3: Working mechanism of congestion control module

3.2 Error control module

Upon receiving this information, the OPENT state in the error control module will instruct other states to start preparations. When the data treatment module sends data packets, the information packet containing the serial number of the sent data packet will be transmitted to the error control module. Upon receiving this information packet, the PACKET state will acquire the serial number of the sent data packet and write it into the variable pk _ seq _ recorded that records the serial number of the sent data packet (Almirall et al., 2004; Elina et al, 2008). The working mechanism of the error control module is shown in Fig. 4.

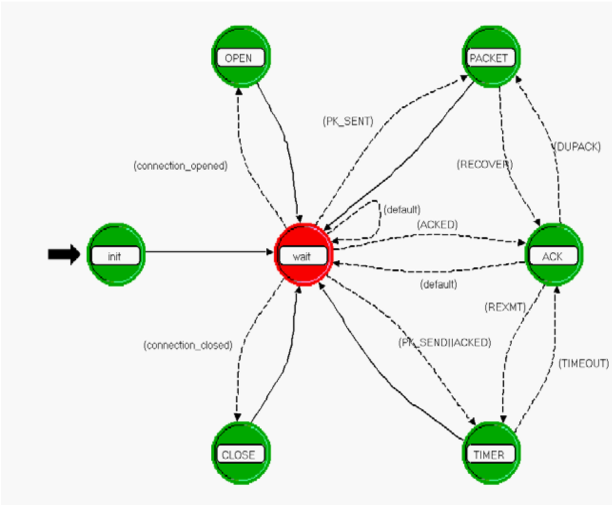

Figure 4: Working mechanism of error control module

3.3 Description of data treatment module

The working mechanism of the data treatment module is shown in Fig. 5. First some programs and variables are initialized and the connections are established by three-way handshake. After that, the error control module and congestion control module send the information that the connection is established to the data treatment module. The DATA_RCVD state in the data transmission module will receive the application data flow information from the upper protocol layer. The application data flow information is divided into data packets with fixed length by the maximum segment length. The data packets received by the data treatment module but not yet confirmed will be first stored in a queue. Only after the received data packets are confirmed by the data treatment module will the confirmed data packets be removed from the queue.

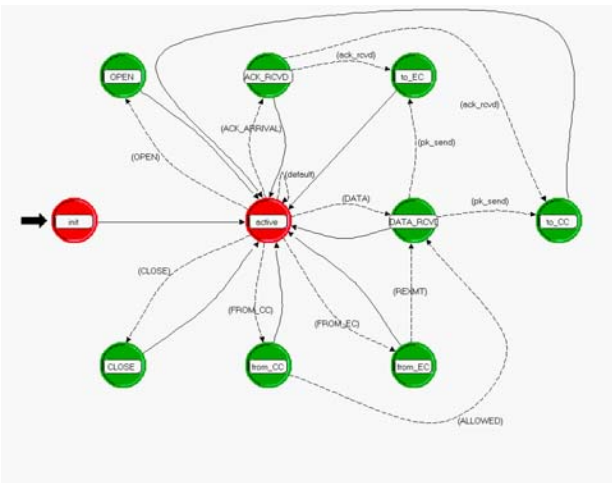

Figure 5: Working mechanism of data treatment module

4. Simulation result

OPNET simulator (version 14.5) is used for the simulation. The architecture of simulation is shown in Fig. 4-1, where node_0 is a wireless server that provides various application services, including file transmission protocol (FTP), hypertext transfer protocol (HTTP), Voice over Internet Protocol (VoIP) and video conferencing (Widmer et al., 2001). Station nodes are the nodes in wireless mesh network, which receive data from the wireless server node_0. These nodes not only receive data themselves, but also act as relays for other nodes. The entire network architecture constitutes a wireless mesh network.
Nodes in the wireless mesh network make a request of an FTP application to the wireless server node node_0, and a file of 1K is sent to each node. The traffic sent by the wireless server node_0 and Wireless Lan throughput are estimated along with the WLAN throughput at the nodes in wireless mesh network. The simulation results are compared between the three versions of TCP.

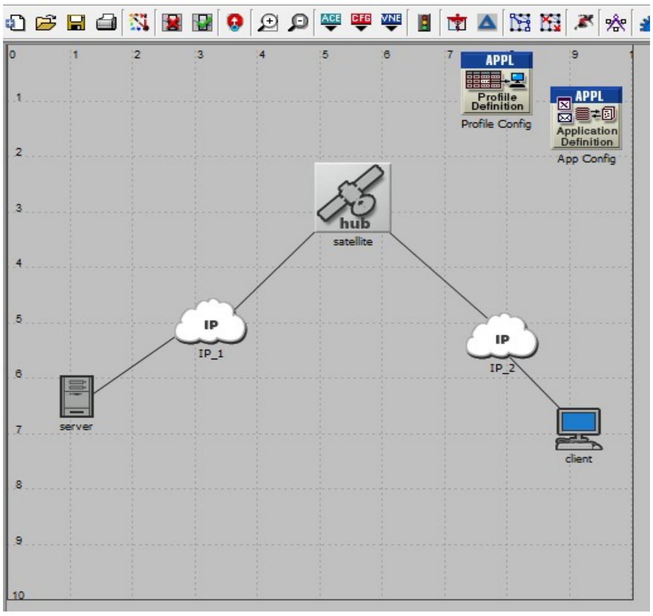

Figure 6: Architecture of the simulation model

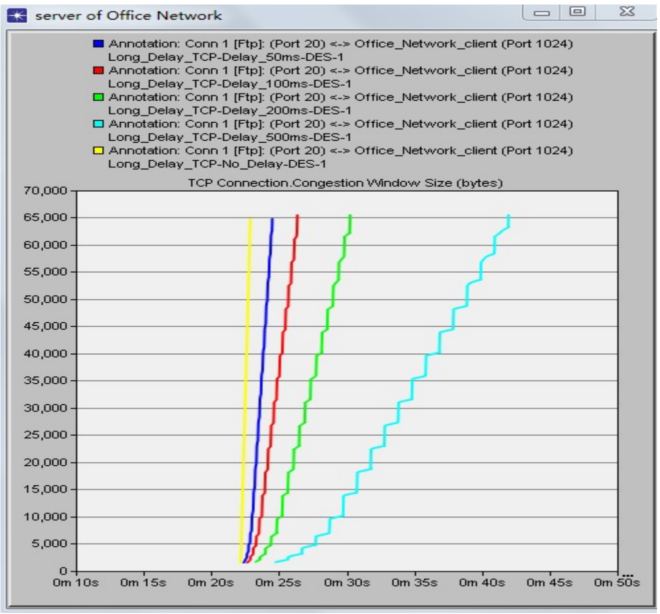

Figure 7: Size of segments sent by the congestion window under different delay

Fig. 7 shows the comparison of size of segments sent by the congestion window under no delay and under the delay of 50ms, 100ms, 200ms and 500ms. The x axis is the simulation time, and y axis throughput (unit: bits/sec). It can be seen from the figure that the time taken for sending segments of the same size is the shortest under no delay.

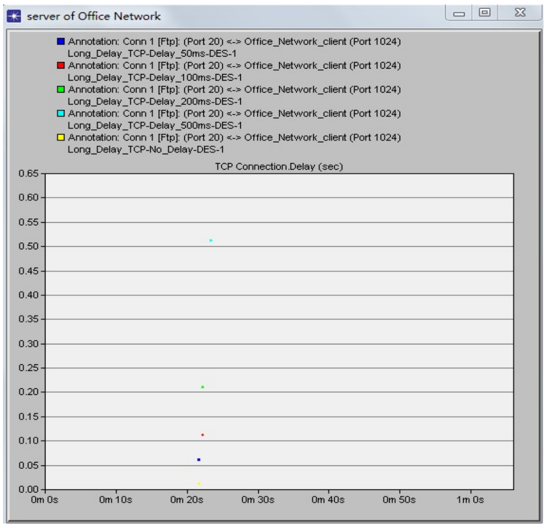

Figure 8: Comparison of delays of TCP connections

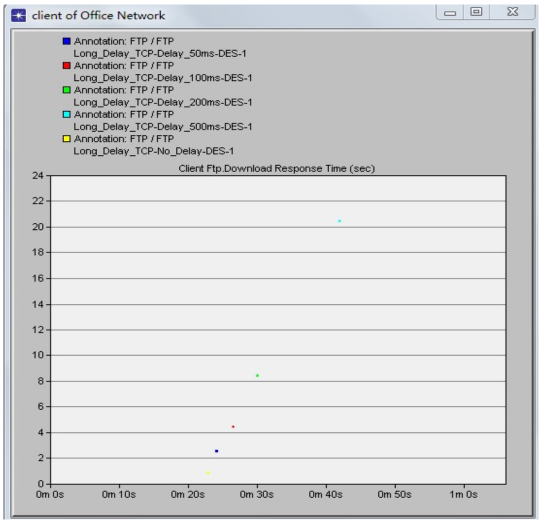

Figure 9: Comparison of FTP download response time under different propagation delay

Fig. 8 is the comparison of delays of TCP connections. X axis represents different propagation delay conditions (no delay, delay of 50ms, 100ms, 200ms and 500ms). Y axis is the TCP connection delay. The longer the propagation delay, the longer the TCP connection delay is.

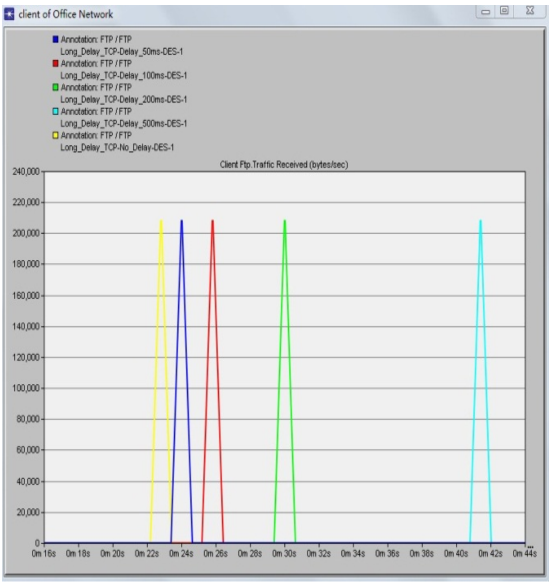

Figure 10: Comparison of FTP traffic receive at each node

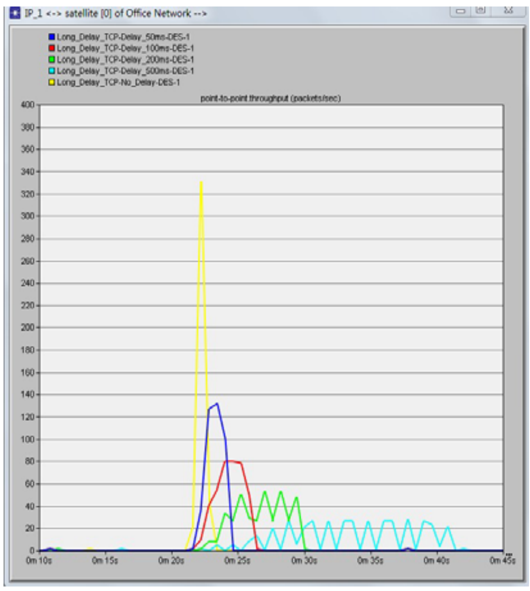

Figure 11: Comparison of throughput from IP1 to satellite uplink

Fig. 9 shows the comparison of FTP download response time under different propagation delay (no delay, delay of 50ms, 100ms, 200ms and 500ms). Y axis represents FTP download response time. The shorter the propagation delay, the shorter the FTP download response time is.

Fig. 10 is the comparison of FTP traffic received at each node. X axis represents the propagation delay (no delay, delay of 50ms, 100ms, 150ms and 200ms). Y axis is the FTP traffic received. As the propagation delay

increases, the time used for FTP download also increases.
Fig. 11 is the comparison of throughput from IP1 to satellite uplink. X axis represent the propagation delay (no delay, delay of 50ms, 100ms, 150ms and 200ms), and y axis the throughput. The longer the propagation delay, the smaller the throughput is. As shown by the simulation, TCP NewReno is not applicable to the satellite network. But with the modification of error control mechanism, TCP Reno displays better performance and higher applicability in long delay environment.

5. Conclusion

Simulation results show that TCP NewReno does not apply to the satellite network. However, after modification of the error control mechanism, TCP Reno exhibits better performance and applicability to long-delay satellite network.

Reference

Almirall A., Larrecq G., Delgado J.A., Martínez S., Planell J.A, Fabrication of low temperature macroporous hydroxyapatite scaffolds by foaming and hydrolysis of an alpha-TCP paste [J]. Biomaterials, 2004, 25(17): 3671-3680.

Elina M., Teijo P., Jonna N., Karoliina V, 2008, A Antti. Negative regulation of EGFR signalling through integrin-alpha1beta1-mediated activation of protein tyrosine phosphatase TCPTP [J]. Nature Cell Biology, 47(3): A352.

Ha S., Rhee I., Xu L, 2005, CUBIC: a new TCP-friendly high-speed TCP variant [J]. Acm Sigops Operating Systems Review, 42(5): 64-74.

Henderson T., Gurtov A., Nishida Y, Floyd S, 2012, The New Reno Modification to TCP’s Fast Recovery Algorithm [J]. Expires, 345(2): 414-418.

Hollot C.V., Misra V., Towsley D, 2010, WB Gong. Designing improved controllers for AQM routers supporting TCP flows [J]. Proceedings of IEEE Infocom, 3(6): 1726-1734.

Jain M., Dovrolis C, 2003, End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput [J]. IEEE/ACM Transactions on Networking, 11(4): 537-549. Doi: 10.1109/TNET.2003.815304

Jiang H., Dovrolis C, 2010, Passive Estimation of TCP Round-Trip Times [J]. Acm Sigcomm Computer Communication Review, 32(3): 75-88.

Ludwig R., Katz R.H, 2000, The Eifel algorithm: making TCP robust against spurious retransmissions [J]. Acm Sigcomm Computer Communication Review, 30(1): 30-36.

Mathis M., Semke J., Mahdavi J, 2001, T Ott. The macroscopic behavior of the TCP congestion avoidance algorithm [J]. Acm Sigcomm Computer Communication Review, 27(3): 67-82

Misra V., Gong W, 2000, DF Towsley. Fluid-based analysis of network of AQM routers supporting TCP flows with an application to RED [J]. Acm Sigcomm Computer Communication Review, 30(4): 151-160.

Ozkan S., Kalyon D.M., Yu X, 2010, Functionally graded β-TCP/PCL nanocomposite scaffolds: In vitro evaluation with human fetal osteoblast cells for bone tissue engineering [J]. Journal of Biomedical Materials Research Part A, 92(3): 1007-1018.

Partridge C., Shepard T.J., 1997, TCP/IP Performance over Satellite Links[J]. IEEE Network Maganize, September/October, 11(5): 44-49

Ratnam K., Matta I, 1198, WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links [D]. In: Proceedings of the 3rd, 74-78. Doi: 10.1109/ISCC.1998.702450

Widmer J., Denda R., Mauve M, 2001, A survey on TCP-friendly congestion control [J]. IEEE Network, 15(3): 28-37. Doi: 10.1109/65.923938