Performance Evaluation of TCP Tahoe, Reno, Reno with SACK, and NewReno Using OPNET Modeler

Laxmi Subedi, Mohamadreza Najiminaini, and Ljiljana Trajković
{lsa38, mna28, ljilja}@cs.sfu.ca
Communication Networks Laboratory
http://www.ensc.sfu.ca/research/cnl
Simon Fraser University

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc.
Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc.   Used with permission of the Author.

Introduction

- Transmission Control Protocol (TCP):
  - predominant Internet protocol
  - carries approximately 90% of the Internet traffic
  - connection oriented transport layer protocol that provides reliable packet delivery over unreliable links
  - independent on the underlying network layers
- TCP congestion control algorithms:
  - may not perform well in heterogeneous (wireless and wired) networks
  - originally proposed based on the assumption that congestion is the only cause for packet loss

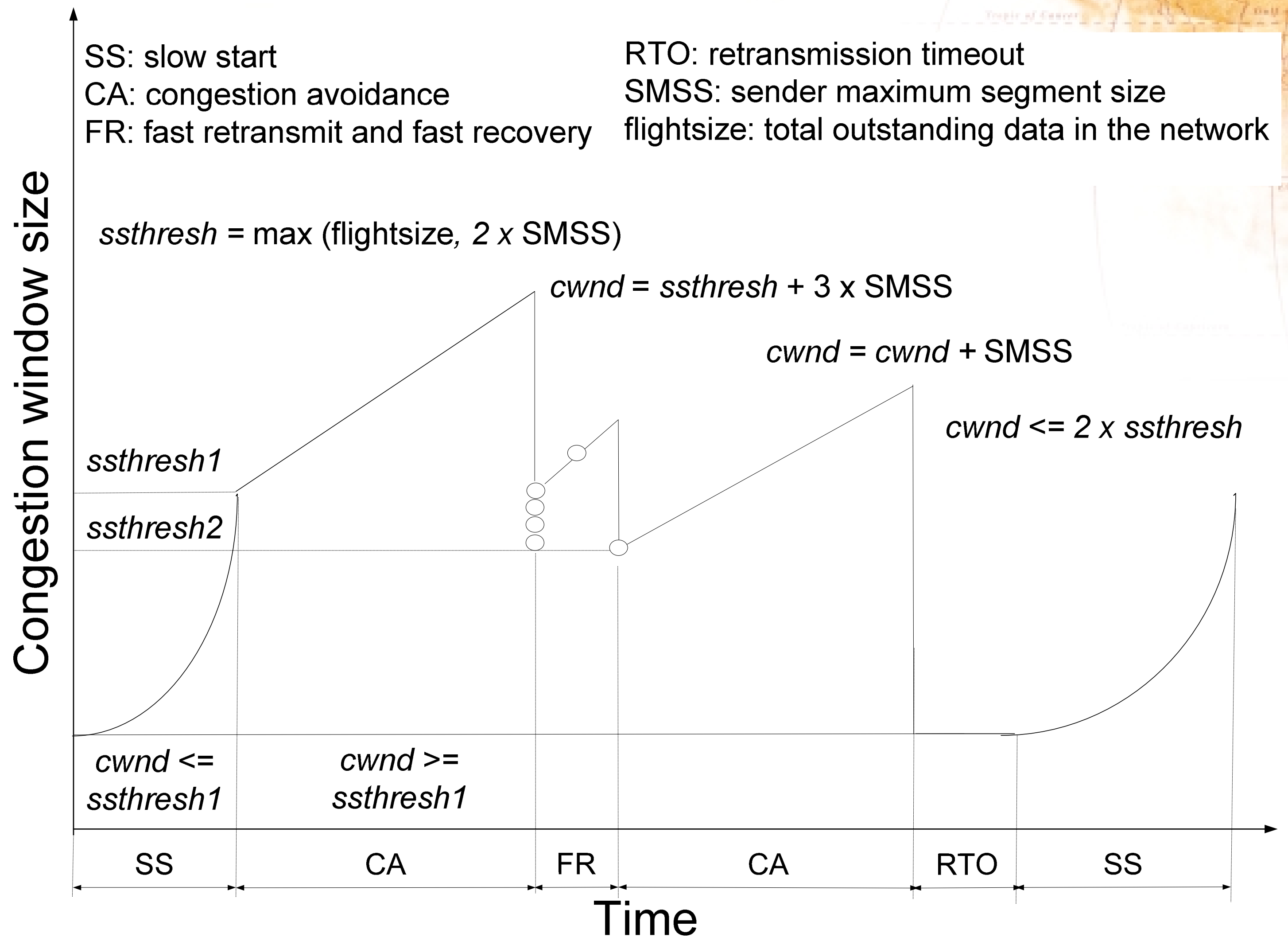
Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

TCP congestion control mechanisms


SS: slow start
CA: congestion avoidance
FR: fast retransmit and fast recovery
RTO: retransmission timeout
SMSS: sender maximum segment size
flightsize: total outstanding data in the network
ssthresh = max (flightsize, *2 x* SMSS)
cwnd = *ssthresh* + 3 x SMSS
cwnd = *cwnd* + SMSS
cwnd <= *2 x ssthresh*
ssthresh1
ssthresh2
cwnd <= *ssthresh1*
cwnd >= *ssthresh1*
Congestion window size
SS
CA
FR
CA
RTO
SS
Time

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Related work

- TCP Reno with SACK:
  - shows lower bandwidth utilization in case of congested links
  - has lower goodput than TCP Reno and TCP NewReno *
- TCP Reno, Reno with SACK, and NewReno:
  - was compared using the ns-2 simulator **
- The behavior of the various TCP algorithms over wireless links with correlated packet losses indicated that:
  - TCP NewReno often performs worse than TCP Tahoe because of the inefficient fast recovery algorithm ***

* R. Paul and Lj. Trajkovic, "Selective-TCP for wired/wireless networks," *Proc. SPECTS 2006*, Calgary, AL, Canada, Aug. 2006, pp. 339–346.

** H. Lee, S. Lee, and Y. Choi, "The influence of the large bandwidth-delay product on TCP Reno, NewReno, and SACK," *Proc. Information Networking Conference*, Oita, Japan, Feb. 2001, pp. 327–334.

*** F. Anjum and L. Tassiulas, "Comparative study of various TCP versions over a wireless link with correlated losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 370–383, June 2003.

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Transmission Control Protocol

- TCP Tahoe:
  - original TCP algorithm
  - three transmission phases:
    - slow start
    - congestion avoidance
    - fast retransmit
- TCP Reno
- TCP Reno with Selective Acknowledgment (SACK)
- TCP NewReno

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

TCP Reno

- Most widely adopted Internet TCP protocol
- TCP Reno transmission phases:
  - slow start
  - congestion avoidance
  - fast retransmit and fast recovery
- After duplicate acknowledgments (ACKs):
  - fast retransmit and fast recovery
  - fast recovery avoids slow start
- After timeout:
  - fast retransmit
  - slow start
- TCP Reno performs better than TCP Tahoe when a single packet is dropped within a round-trip time

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

TCP Reno with Selective Acknowledgment (SACK)

- SACK algorithm allows a TCP receiver to acknowledge out-of-order segments selectively rather than cumulatively by acknowledging the last correctly received in order segment
- TCP Reno with SACK helps improve performance in case of multiple packet losses
- Multiple blocks can be transmitted in a single segment
- TCP Reno with SACK:
    - initiate fast recovery upon 3 duplicate ACKs
    - sender keeps records of SACKs and understands if segments are lost
    - sender retransmits the subsequent segment from the list of the lost segments

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

TCP NewReno

- Modification of TCP Reno
- Improved retransmission process during the fast recovery phase
- Can recover multiple packet losses within a window of data
- Partial ACK:
  - occurs if multiple packets are lost
  - does not acknowledge all packets that are outstanding at the beginning of a fast recovery (this causes sender to leave fast recovery)
  - sender must wait until timeout occurs

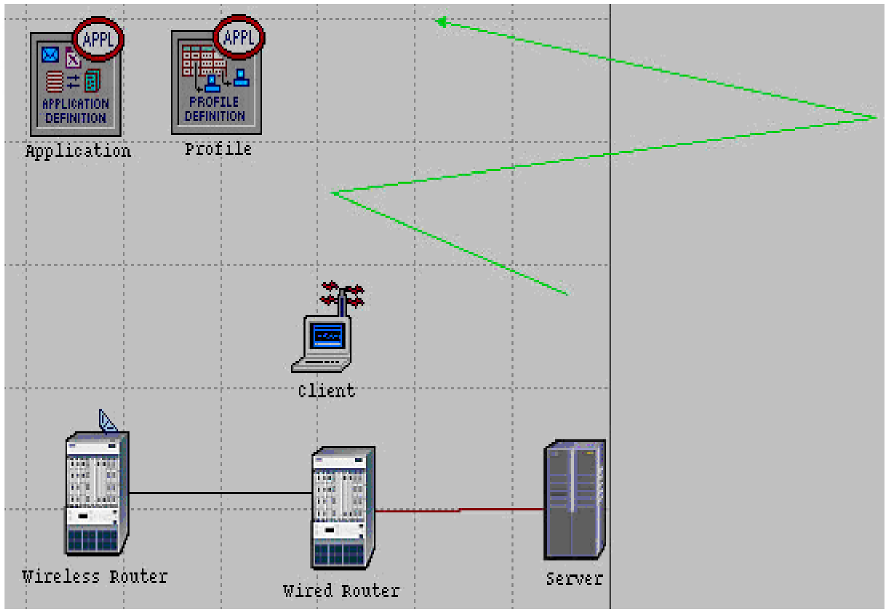
Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

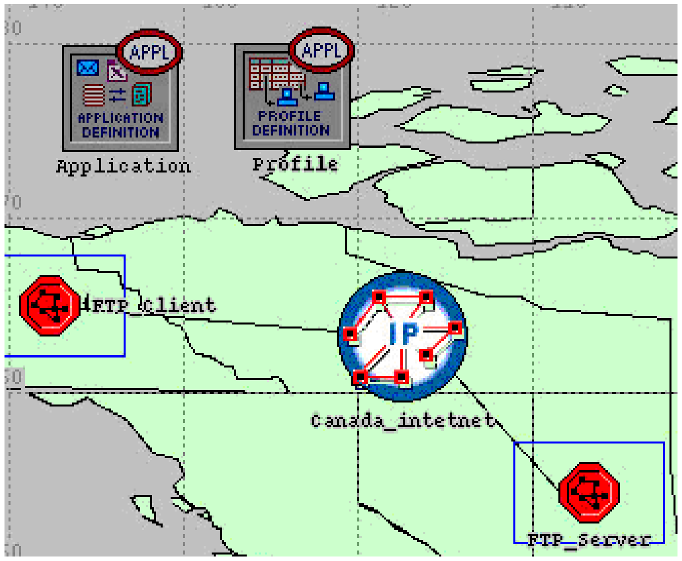
Wireless topology scenario

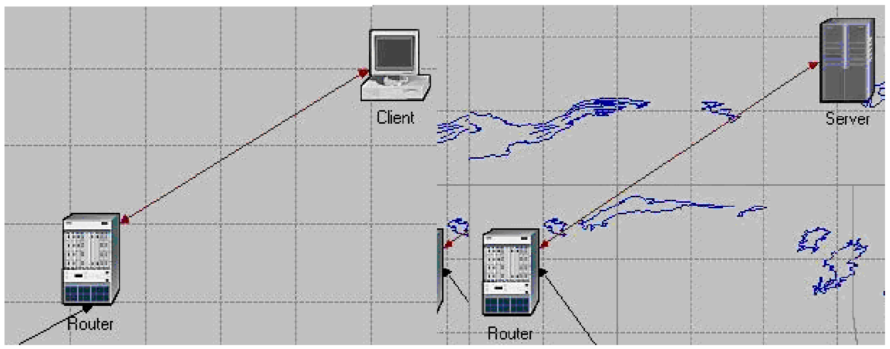


- OPNET modeler v. 11.0.A
- A file transfer protocol (FTP) server connected with wired router by a 10 Mbps link
- A wireless router (base station for a mobile client) connected to a wired router by a 1.5 Mbps link
- File size: 50 MB
- Packet losses occur due to signal attenuation

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Wired topology scenario





- OPNET modeler v. 11.0.A
- Network consists of an Ethernet server and a client connected to the corresponding wired routers via 10 Mbps wired links
- Two routers are connected to the Internet cloud via 1.5 Mbps wired link
- FTP application is used to transfer 10 MB file from the server to the client

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

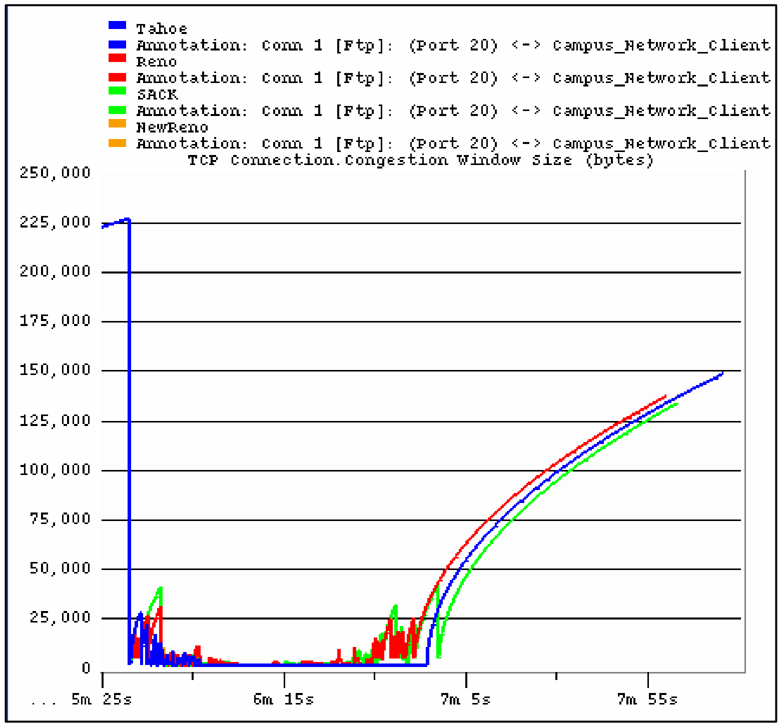
OPNET model: wireless client and server

TCP parameters	Value
Slow start initial count (MSS)	1
Receive buffer size (bytes)	65,535
Maximum ACK segment	2
Initial RTO (sec)	1.0
Minimum RTO (sec)	0.5
Maximum RTO (sec)	64
RTT gain	0.125
Deviation gain	0.25
RTT deviation coefficient	4.0
Duplicate ACK threshold	3

- TCP versions:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- Performance parameters:
  - congestion window size
  - file download response time
  - throughput
  - goodput
- TCP parameters

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Wireless network scenario: congestion window size and file download response time

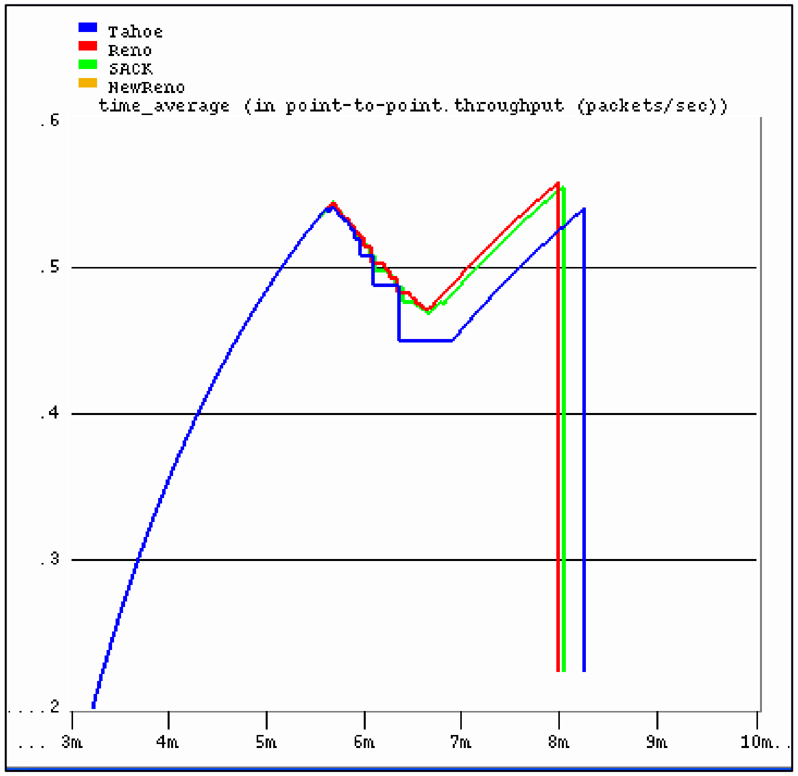


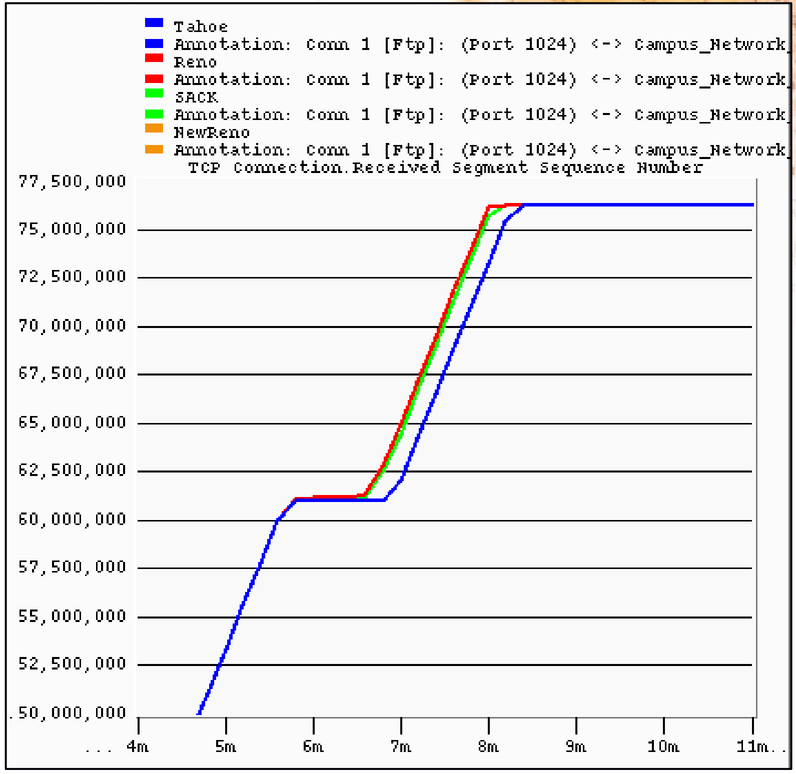
TCP variants	Download response time (s)
Tahoe	491
Reno	480
Reno with SACK	483
NewReno	496

- TCP Reno outperforms the remaining three algorithms
- TCP Reno with SACK has larger congestion window size than TCP Tahoe and TCP NewReno
- TCP Tahoe and TCP NewReno have comparable congestion window size
- TCP Reno has shorter file download response time followed by TCP Reno with SACK while TCP NewReno has the longest file download response time

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Wireless network scenario: throughput and goodput





- TCP Reno has the highest throughput and goodput compared to the remaining three algorithms
- TCP Reno with SACK has higher throughput and goodput than TCP NewReno and Tahoe

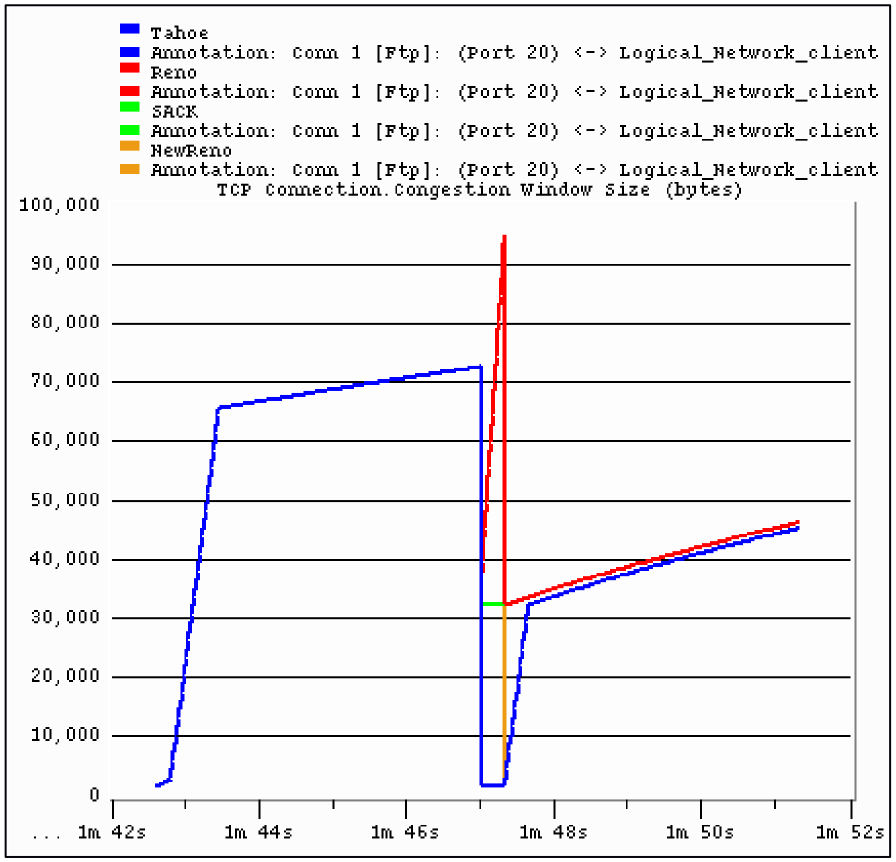
Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

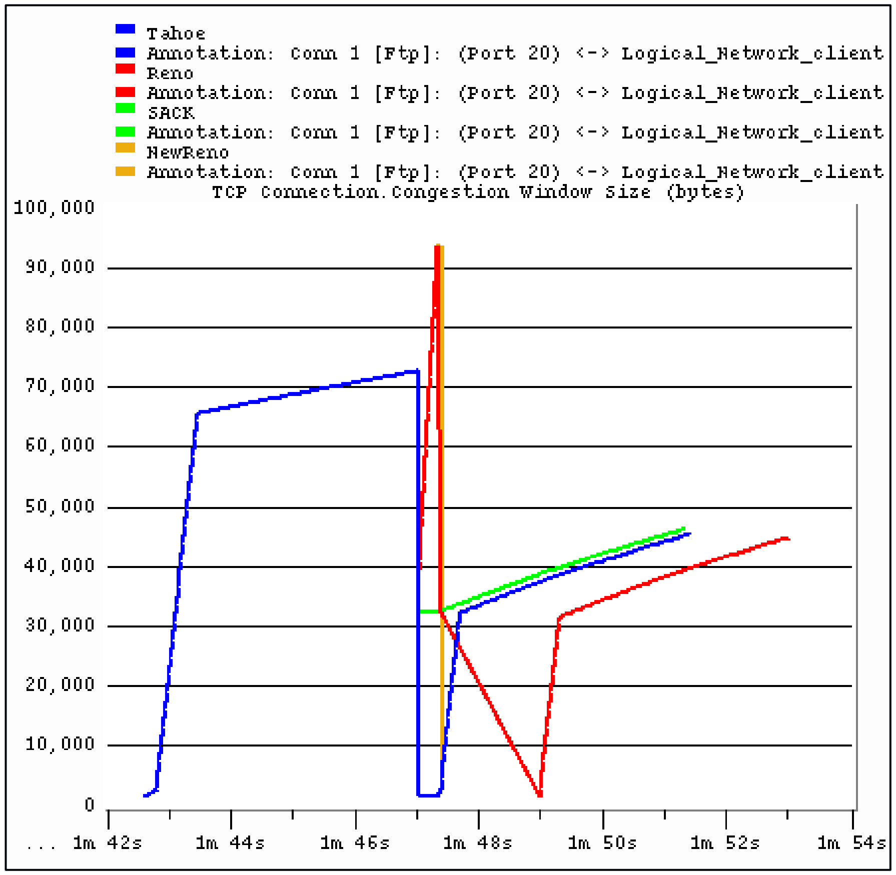
Single packet loss scenario: congestion window size



- Evolution of the congestion window size in a simulated client server network
- Single packet loss:
  - TCP Tahoe performs the slow start phase
  - other algorithm perform individual fast recovery and fast retransmit
- The four algorithms have the same file download response time

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

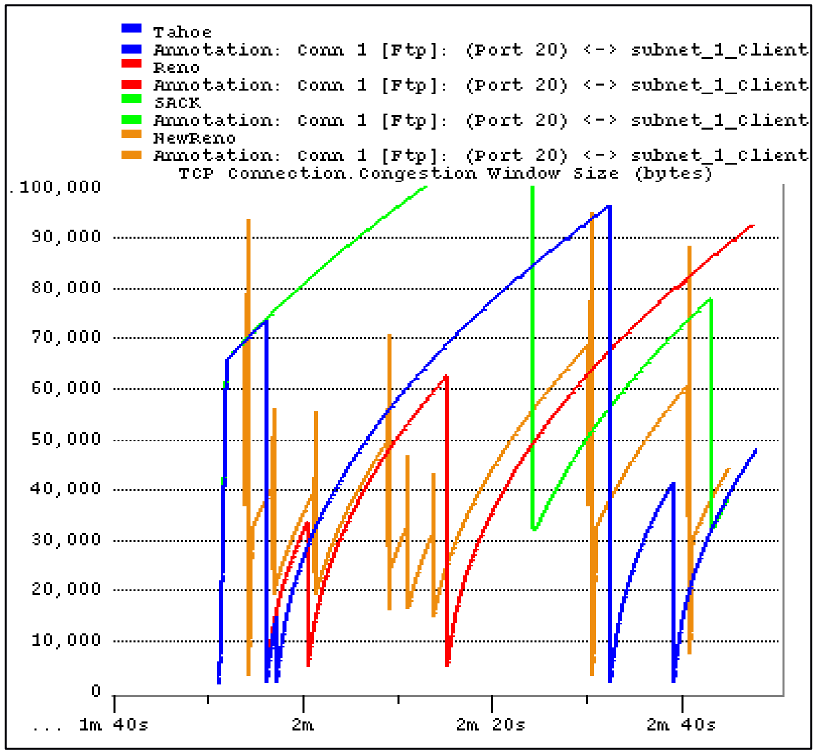
Two packet losses scenario: congestion window size



- Evolution of the congestion window size in case of two packet losses in a window of data
- TCP Reno with SACK has a larger congestion window size and shorter file download response time compared to the remaining three algorithms
- TCP NewReno and TCP Tahoe have similar congestion window size
- TCP Reno has the worst performance

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

0.05% packet loss and 0.001 sec packet latency scenario

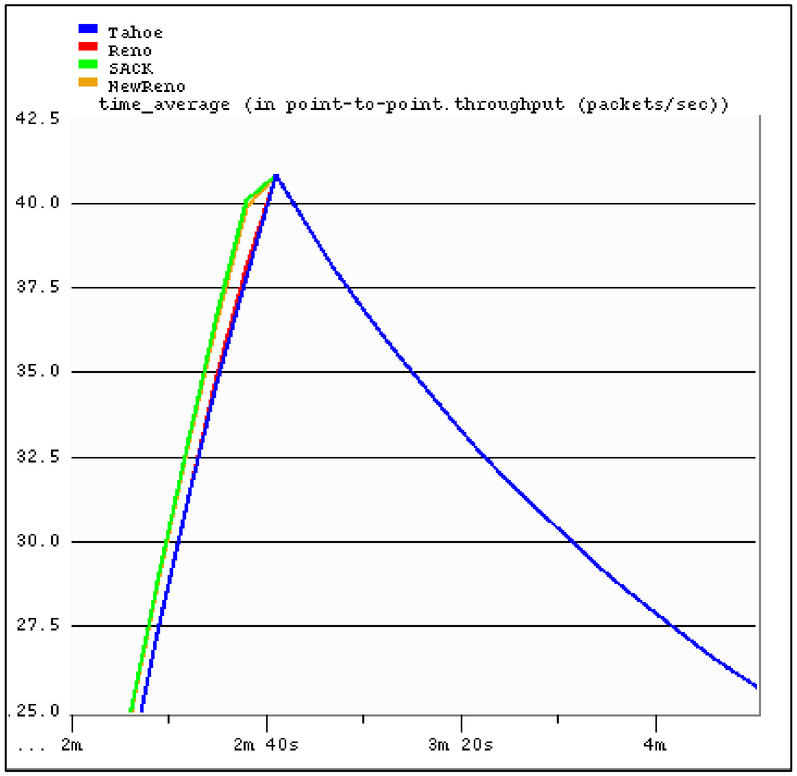


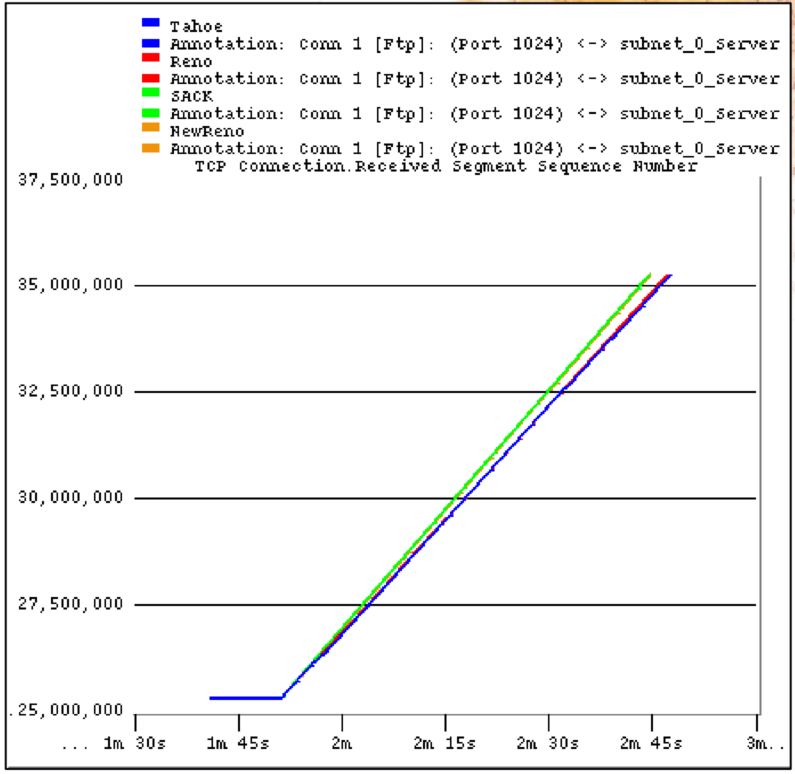
TCP variants	Download response time (s)
Tahoe	168
Reno	167
Reno with SACK	164
NewReno	165

- Evolution of the congestion window size
- File download response time:
  - TCP Reno has larger file download response time than TCP Reno with SACK and NewReno
  - TCP Reno with SACK and NewReno have approximately the same file download response time
  - TCP Tahoe performs the worst compared to the remaining three algorithms

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Wired network scenario: throughput and goodput





- TCP Reno with SACK shows higher throughput and goodput than the remaining three algorithms

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Roadmap

- Introduction
- Background and related work
- Transmission Control Protocol (TCP) algorithms:
  - Tahoe
  - Reno
  - SACK
  - NewReno
- OPNET models and scenarios:
  - network topologies
- Simulation results:
  - wireless client and server
  - wired client and server
- Conclusions and references

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

Conclusions

- We simulated and compared performance of various TCP algorithms in wireless and wired networks
- Simulation results indicated that in wireless networks with signal attenuation, fading, and multipath, TCP Reno outperforms other congestion control algorithms in terms of congestion window size and file download response time
- Throughput and goodput in the case of TCP Reno is also higher than in the remaining three algorithms
- In wired networks, TCP Reno shows significant performance degradation in case of multiple packet losses
- The overall performance of TCP Reno with SACK and TCP NewReno is comparable in terms of file download response time even though TCP Reno with SACK does not exhibit sharp decrease in congestion window as TCP NewReno when a packet loss occurs
- TCP Reno with SACK shows higher throughput and goodput than the remaining three algorithms

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

References

- A. Gurtov and S. Floyd, “Modeling wireless links for transport protocols,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 85–96, Apr. 2004.
- F. Anjum and L. Tassiulas, “Comparative study of various TCP versions over a wireless link with correlated losses,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 370–383, June 2003.
- S. Floyd and K. Fall, “Simulation based comparisons of Tahoe, Reno, and SACK TCP,” *ACM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, July 1996.
- M. N. Akhtar, M. A. O. Barry, and H. S. Al-Raweshidy, “Modified Tahoe TCP for wireless networks using OPNET simulator,” *Proc of the London Communications Symposium (LCS2003),* London, UK, Sept. 2003.
- M. Omueti and Lj. Trajkovic, “M-TCP+: using disconnection feedback to improve performance of TCP in wired/wireless networks,” *Proc. SPECTS 2007*, San Diego, CA, USA, July 2007, pp. 443–450.
- R. Paul and Lj. Trajkovic, “Selective-TCP for wired/wireless networks,” *Proc. SPECTS 2006*, Calgary, AL, Canada, Aug. 2006, pp. 339–346.
- H. Lee, S. Lee, and Y. Choi, “The influence of the large bandwidth-delay product on TCP Reno, NewReno, and SACK,” *Proc. Information Networking Conference*, Oita, Japan, Feb. 2001, pp. 327–334.

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

References

- S. Floyd and T. Henderson, “The NewReno modification to TCP’s fast recovery algorithm,” *RFC 2582*, Apr. 1999.
- W. Stevens, “TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms,” *RFC 2001*, Jan. 1997.
- M. Mathis, J. Mahdavi, S. Floyd, and A. Romanov, “TCP selective acknowledgement options,” *RFC 2018*, Oct. 1996.
- J. Postel, “Transmission control protocol,” RFC 793, Sept. 1981.
- M. Allman, V. Paxson, and W. Stevens, “TCP congestion control,” *RFC 2581*, Apr. 1999.
- I. Khalifa and Lj. Trajkovic, “An overview and comparison of analytical TCP models,” *Proc. IEEE Int. Symp. Circuits and Systems*, Vancouver, BC, Canada, May 2004, vol. V, pp. 469–472.
- Y. Shang and M. Hadjitheodosiou, “TCP splitting protocol for broadband and aeronautical satellite network,” in *Proc. 23rd IEEE Digital Avionics Syst. Conf.*, Salt Lake City, UT, Oct. 2004, vol. 2, pp. 11.C.3-1–11.C.3-9.
- J. Zhu, S. Roy, and J. H. Kim, “Performance modeling of TCP enhancements in terrestrial-satellite hybrid networks,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 4, pp. 753–766, Aug. 2006.
- Q. Shao and Lj. Trajkovic, “Measurement and analysis of traffic in a hybrid satellite-terrestrial network,” *Proc. SPECTS 2004*, San Jose, CA, July 2004, pp. 329–336.

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.

References

- V. Jacobson, "Congestion avoidance and control," *Proc. ACM SIGCOMM, Symp. on Commun. Archit. and Protocols*, Stanford, CA, USA, Aug. 1988, pp. 314–329.
- D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," *J. of Comput. Netw. ISDN Syst.*, vol. 17, no. 1, pp. 1–14, June 1989.
- C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *Wireless Netw.*, vol. 8, no. 5, pp. 467–479, Sept. 2002.
- G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *Proc. ACM/IEEE Int. Conf. on Mobile Computing*, Seattle, WA, USA, Sept. 1999, pp. 219–230.
- W. G. Zeng and Lj. Trajkovic, "TCP packet control for wireless networks," *Proc. IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications* (*WiMob 2005*), Montreal, Canada, Aug. 2005, vol. 2, pp. 196–203.
- OPNET Modeler software [Online]. Available: http://www.opnet.com/university_program/teaching_with_opnet.

Copyright © 2008 OPNET Technologies, Inc.
CONFIDENTIAL - RESTRICTED ACCESS: This information may not be disclosed, copied, or transmitted in any format without the prior written consent of OPNET Technologies, Inc. Used with permission of the Author.