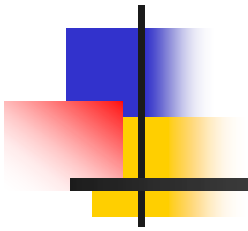
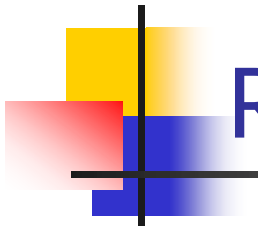


OPNET M-TCP model

Modupe Omueti
momueti@cs.sfu.ca

Communication Networks Laboratory
http://www.ensc.sfu.ca/research/cnl
School of Engineering Science
Simon Fraser University



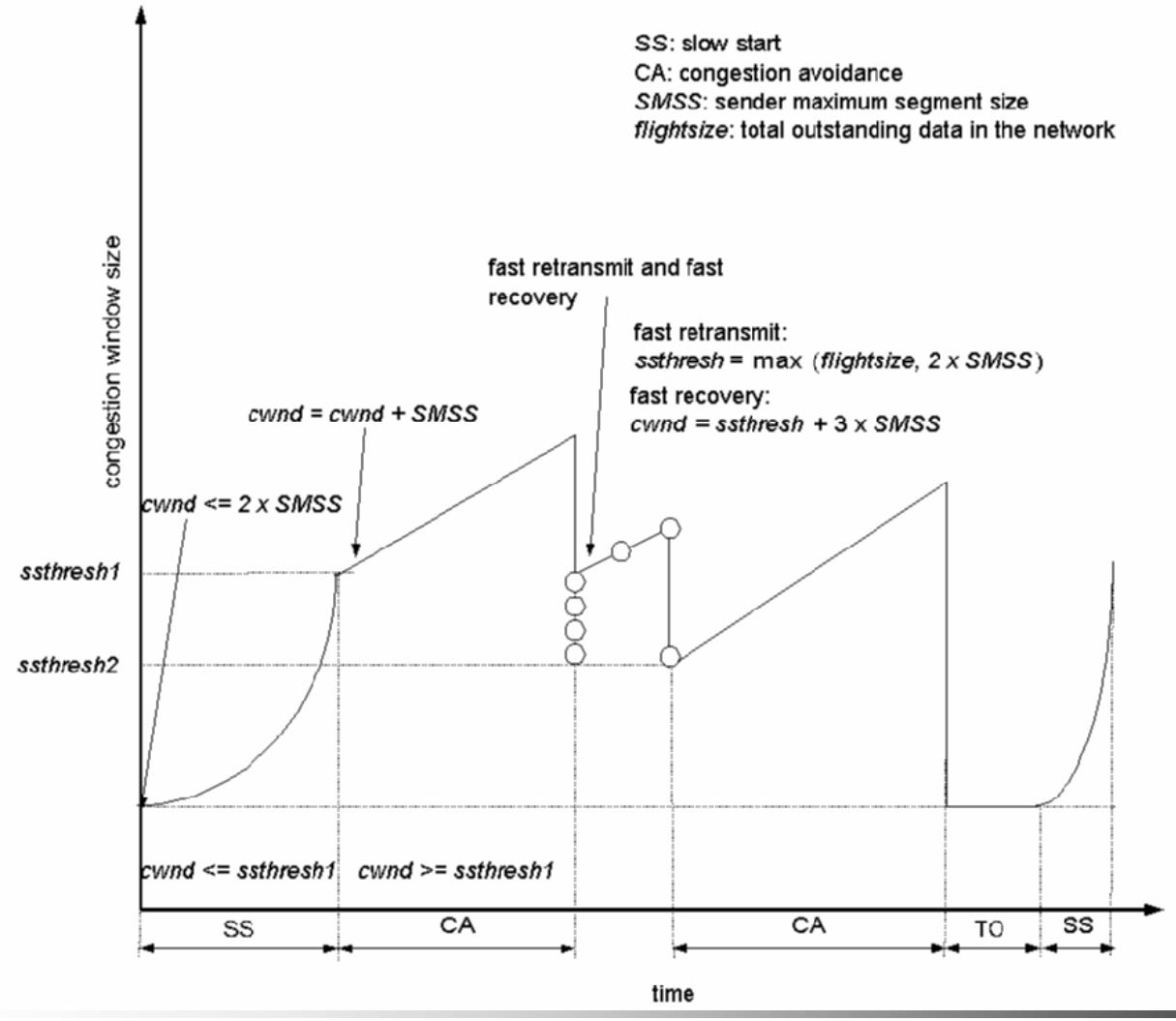
Roadmap

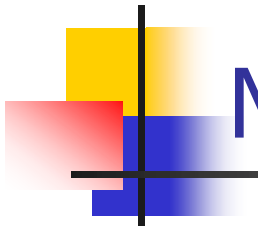
- Introduction
- Motivation for M-TCP
- Design of M-TCP
- OPNET M-TCP implementation
- Simulation scenarios
- Simulation results
- Conclusions

Introduction

- Transmission control protocol:
  - connection management, in-order delivery, flow control, and reliability
  - congestion control algorithms:
    - slow start, congestion avoidance, fast retransmit, and fast recovery
  - state variables:
    - congestion window: *cwnd*
    - receiver window: *rwnd*
    - slow start threshold: *ssthresh*

TCP congestion control algorithms





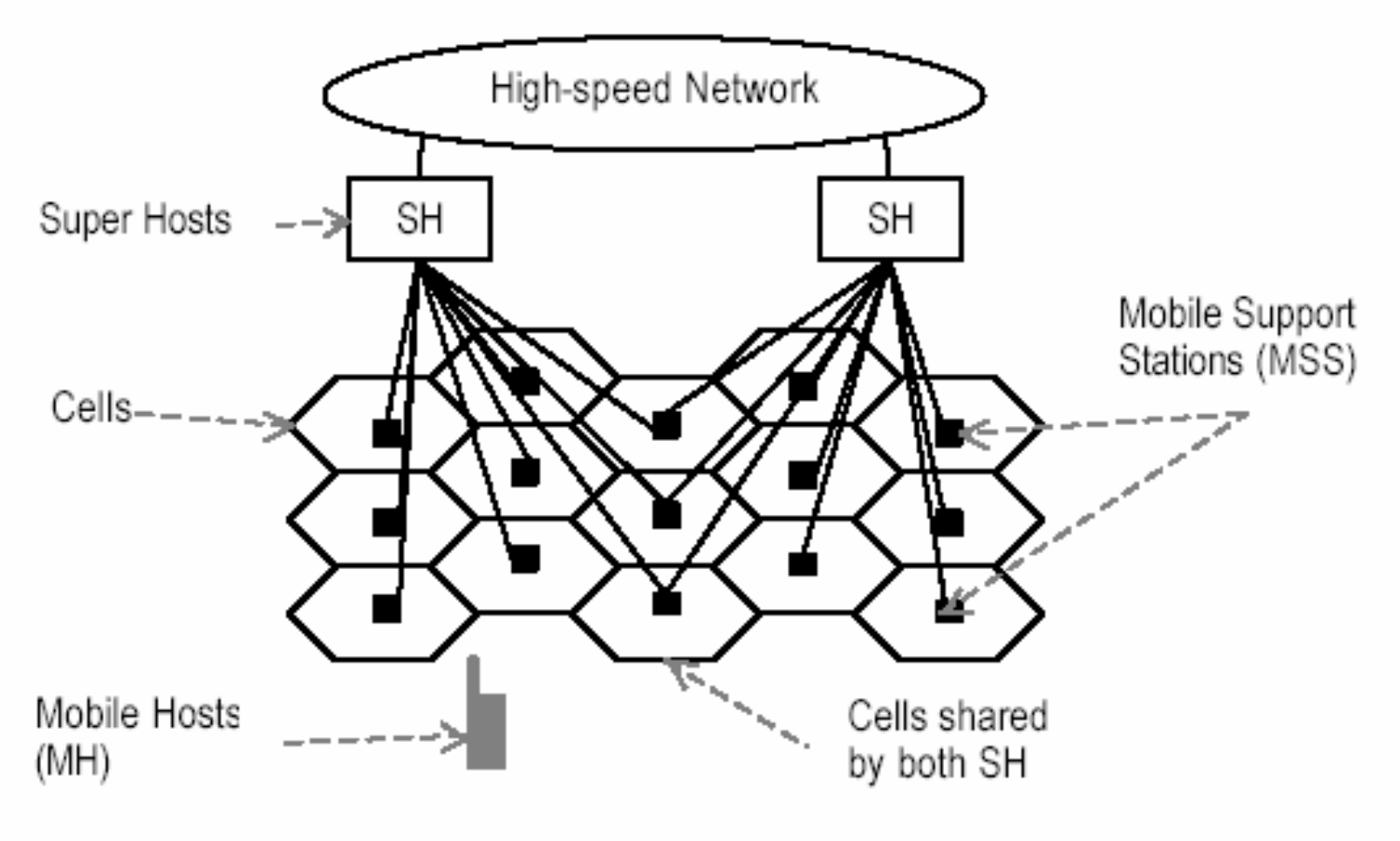
Motivation for M-TCP

- Characteristics of cellular links:
  - handoff and high bit error rate cause packet losses
  - congestion window should not be reduced in such scenarios
- TCP in wireless networks:
  - misinterprets packet losses as an indication of congestion
  - reduces congestion window size and, consequently, throughput
  - induces serial timeouts from disconnections due to handoff

M-TCP network design

- Proposed network design:
  - mobile support station (MSS) in cells for in-building pico-cellular network
  - cells with small sizes to provide higher bandwidth to users
  - cells are controlled by supervisor host (SH), which allocates available bandwidth to mobile hosts (MHs)
  - SH is connected to a high-speed wired network
- Small cell size results in:
  - lower cell latencies
  - more frequent handoffs
  - more frequent and lengthy disconnections

M-TCP network architecture



M-TCP protocol characteristics

- M-TCP:
  - maintains end-to-end semantics of TCP
  - improves TCP throughput for mobile clients
  - mitigates the effects of frequent and lengthy disconnections
  - adapts to bandwidth changes in cellular wireless links
  - ensures efficient handoffs of mobile clients
  - employs split connection approach

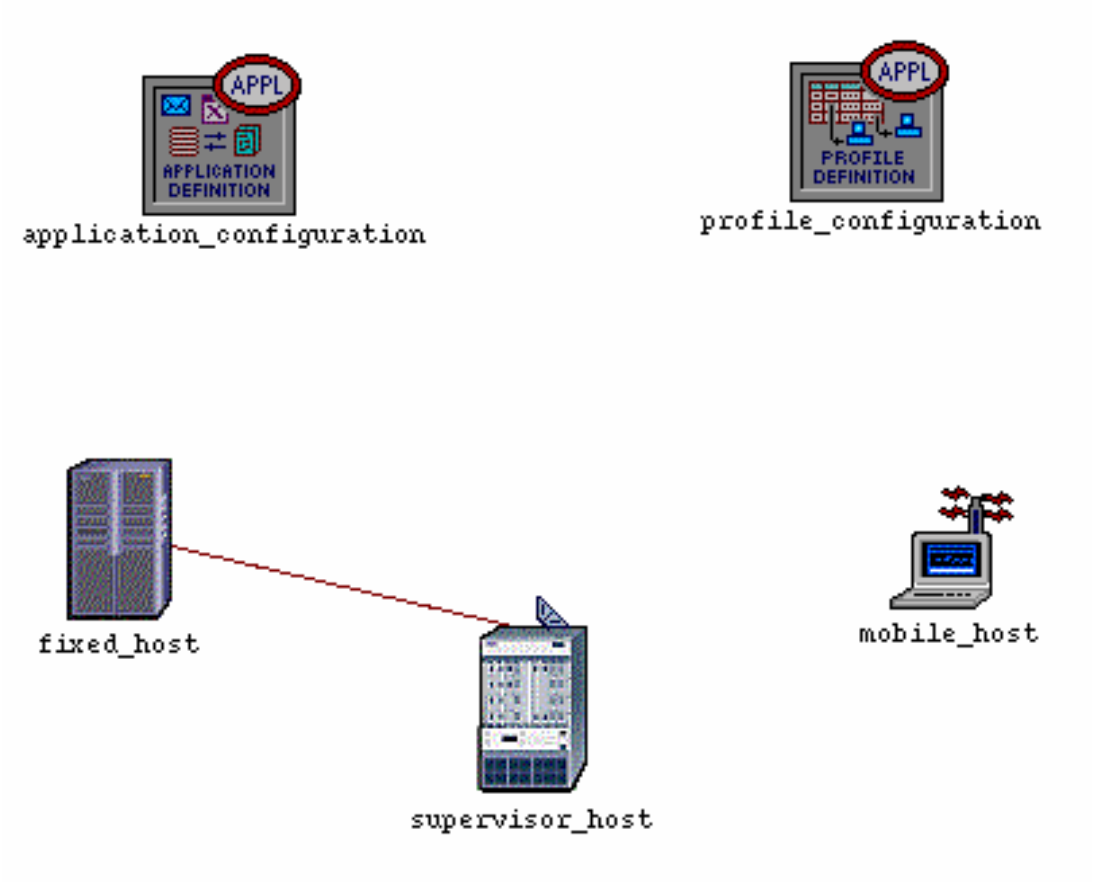
M-TCP algorithm: supervisor host

- M-TCP at SH:
  - stores a copy of FH sent packet until it receives new ACK from MH
  - receives new MH ACK of data with ACK number *w*
    - *w* ≤ maximum sent sequence number *snd_max*
  - forwards ACK number *w* - 1 to FH
  - keeps the last byte with ACK number *w* until disconnection occurs

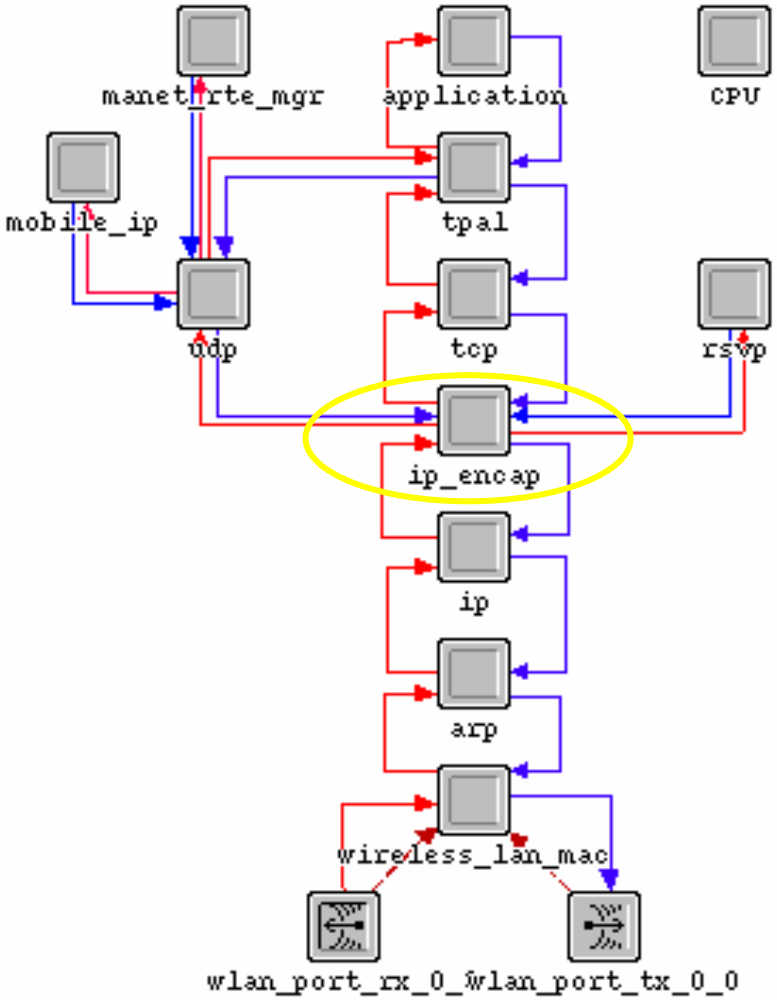
M-TCP algorithm: supervisor host

- When disconnection occurs, M-TCP at SH:
  - sends the last MH ACK with ACK number *w* and a receive window size *rcv_wnd* set to zero
  - assumes TCP sender will enter persist state with new ACK *w* and:
    - if *w* < *snd_max* TCP sender will not enter persist state even with *rcv_wnd* set to zero (TCP delayed ACK behavior)

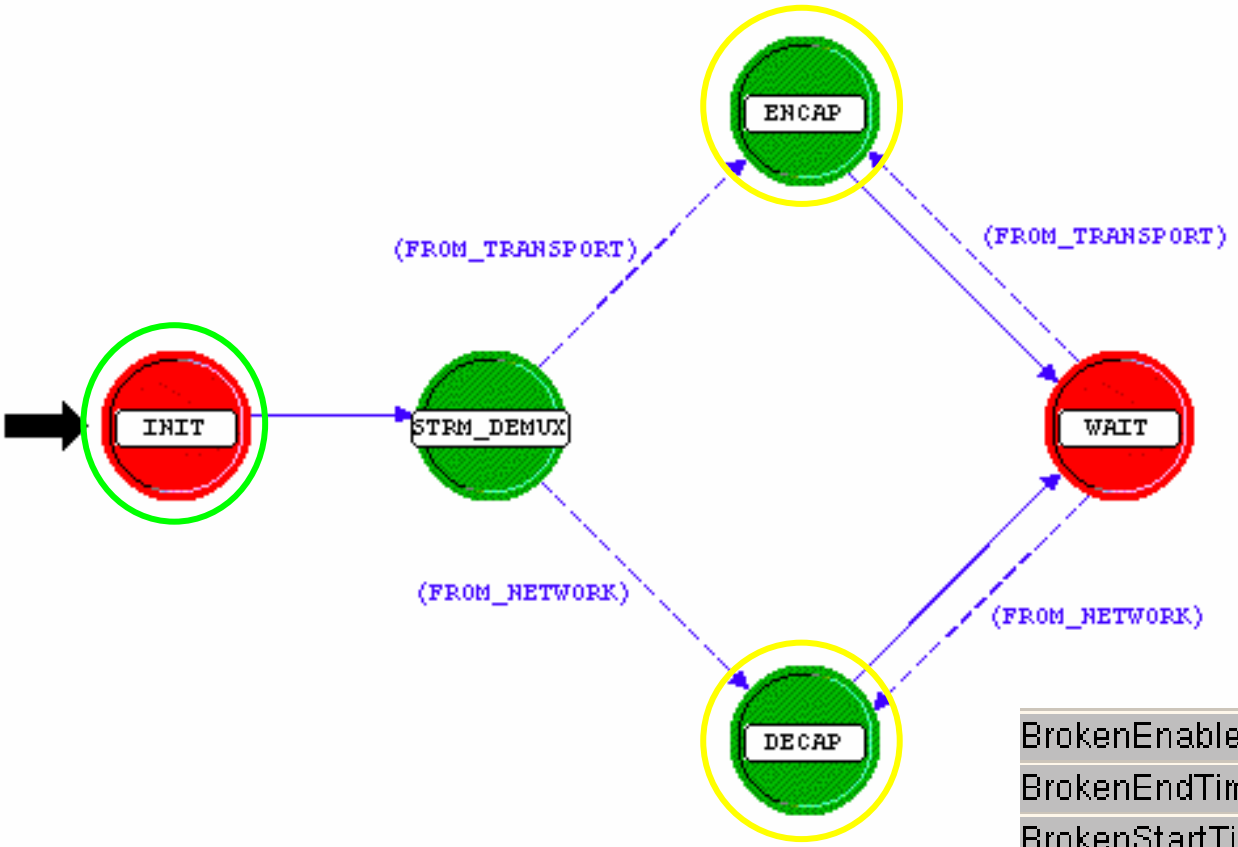
OPNET M-TCP network architecture



Mobile host: node model



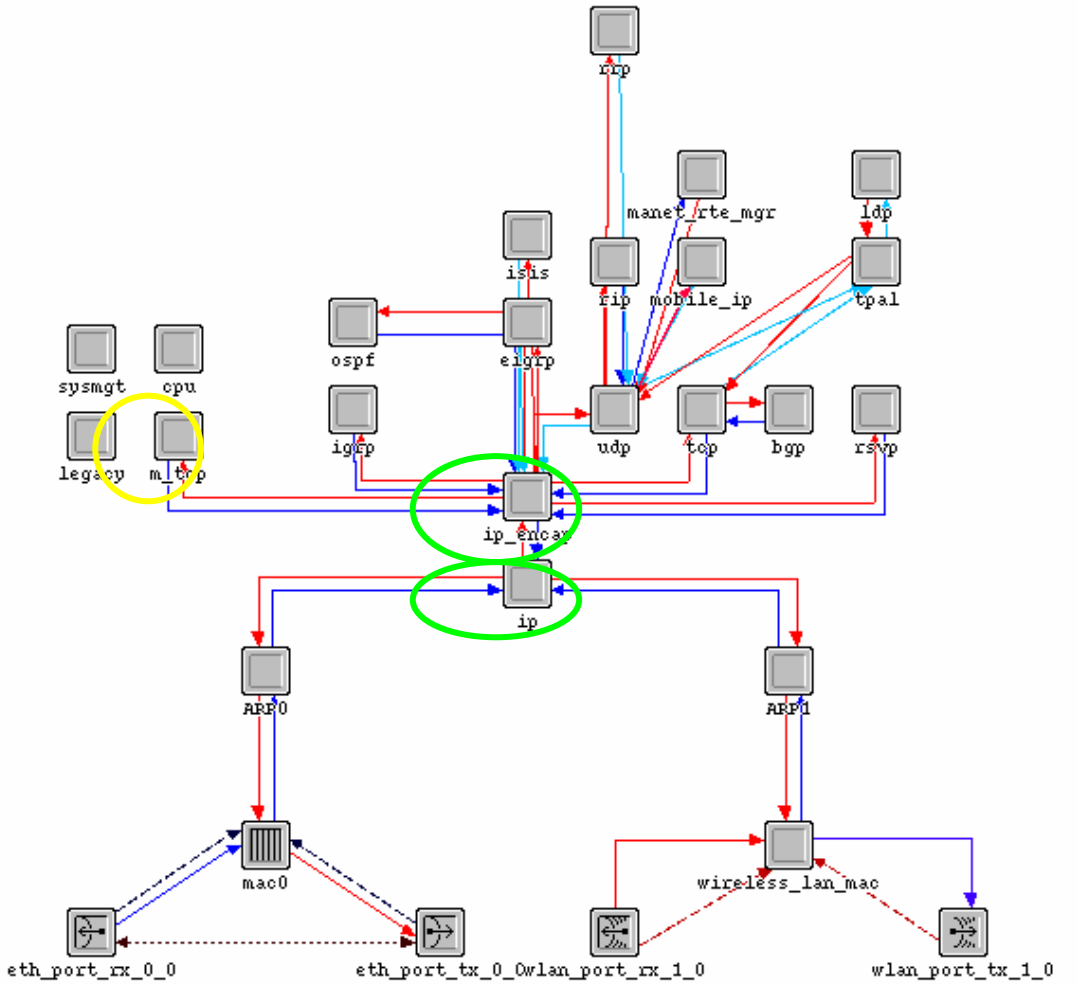
ip_encap: process model



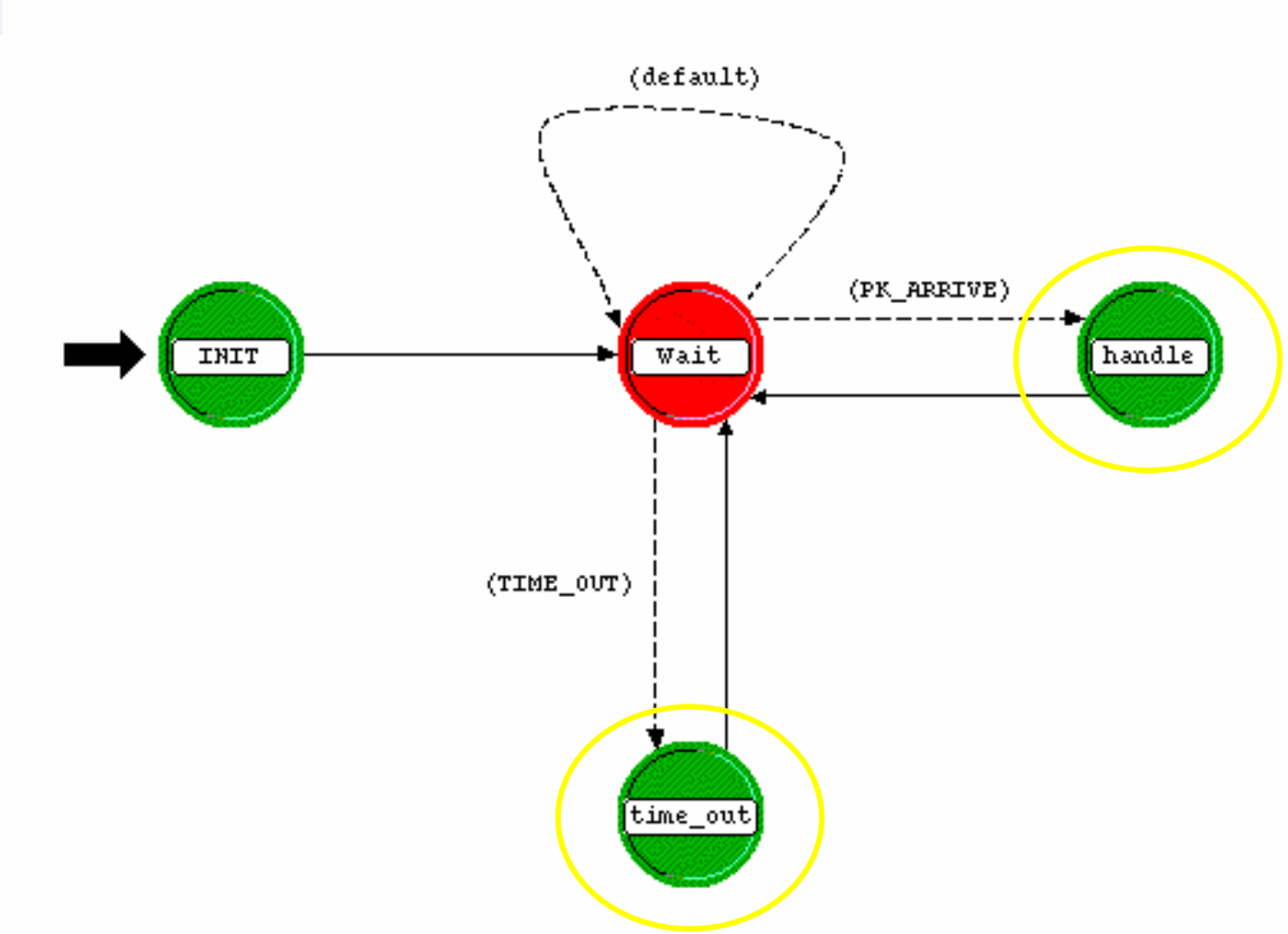
BrokenEnable	promoted
BrokenEndTime	promoted
BrokenStartTime	promoted
CycleTime	promoted

ip_encap process attributes

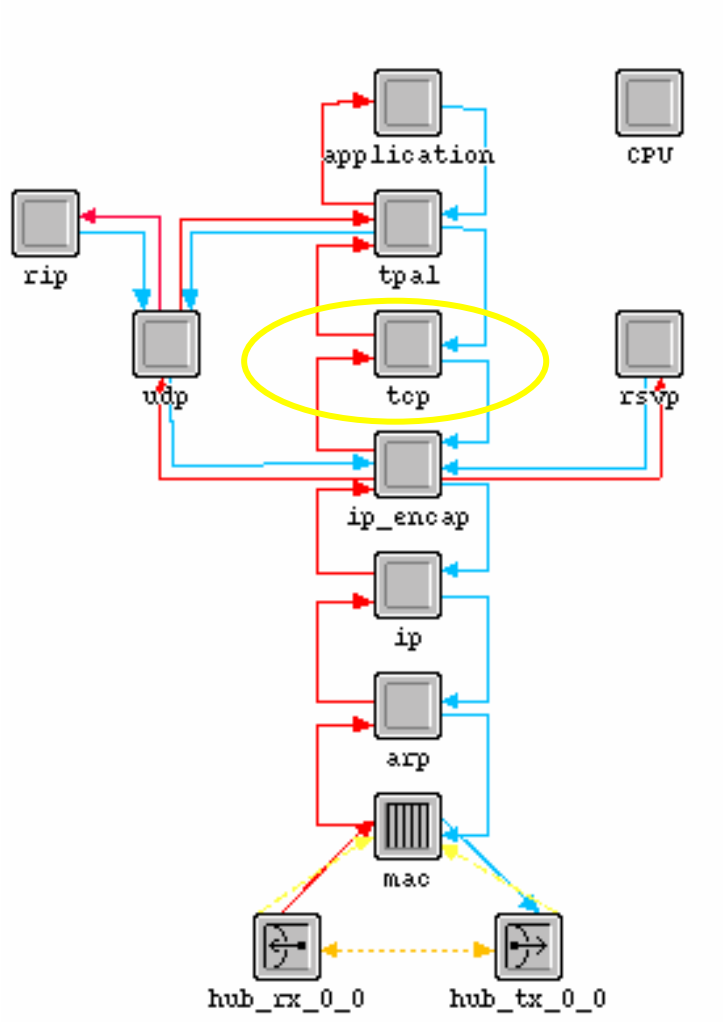
Supervisor host: node model



m_tcp: process model



Fixed host: node model



M-TCP OPNET model: supervisor host

- OPNET M-TCP at SH:
  - stores a copy of FH packet sent until it receives new ACK from MH
  - receives new MH ACK of data with ACK number *w*
    - *w* ≤ maximum sent sequence number *snd_max*
  - forwards ACK number *w* – 1 MSS to FH
  - keeps the last ACK segment with ACK number *w* until disconnection occurs

MSS: maximum segment size

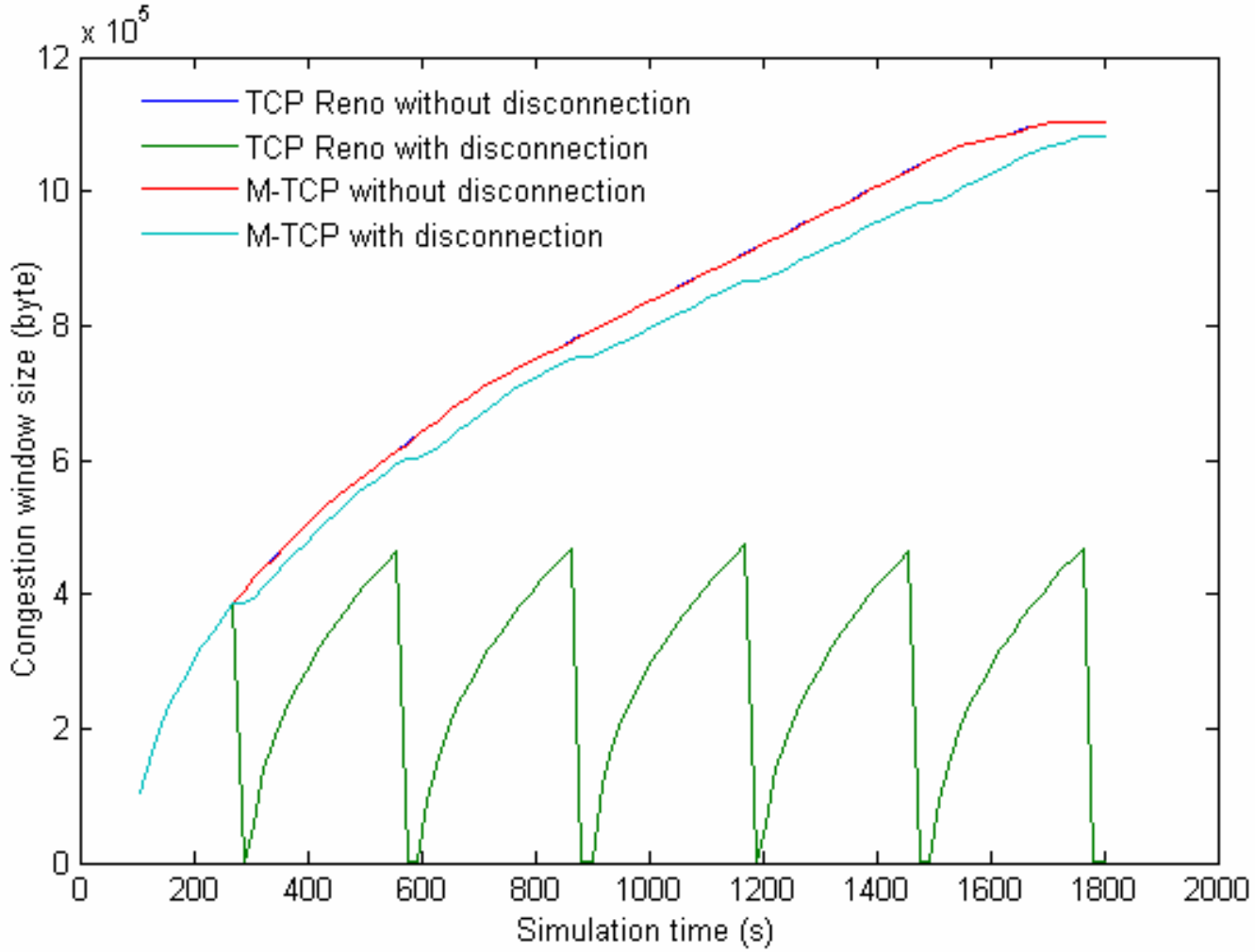
M-TCP OPNET model: supervisor host

- When disconnection occurs, OPNET M-TCP at SH:
  - sends the last MH ACK with ACK number *w* and a receive window size *rcv_wnd* set to zero
  - if *w* < *snd_max*
    - receives data segment with sequence number = *snd_max*
    - acknowledges data segment with sequence number = *snd_max*
    - forwards MH ACK with ACK number *snd_max* to FH and *rcv_wnd* set to zero
    - forces TCP sender into persist state

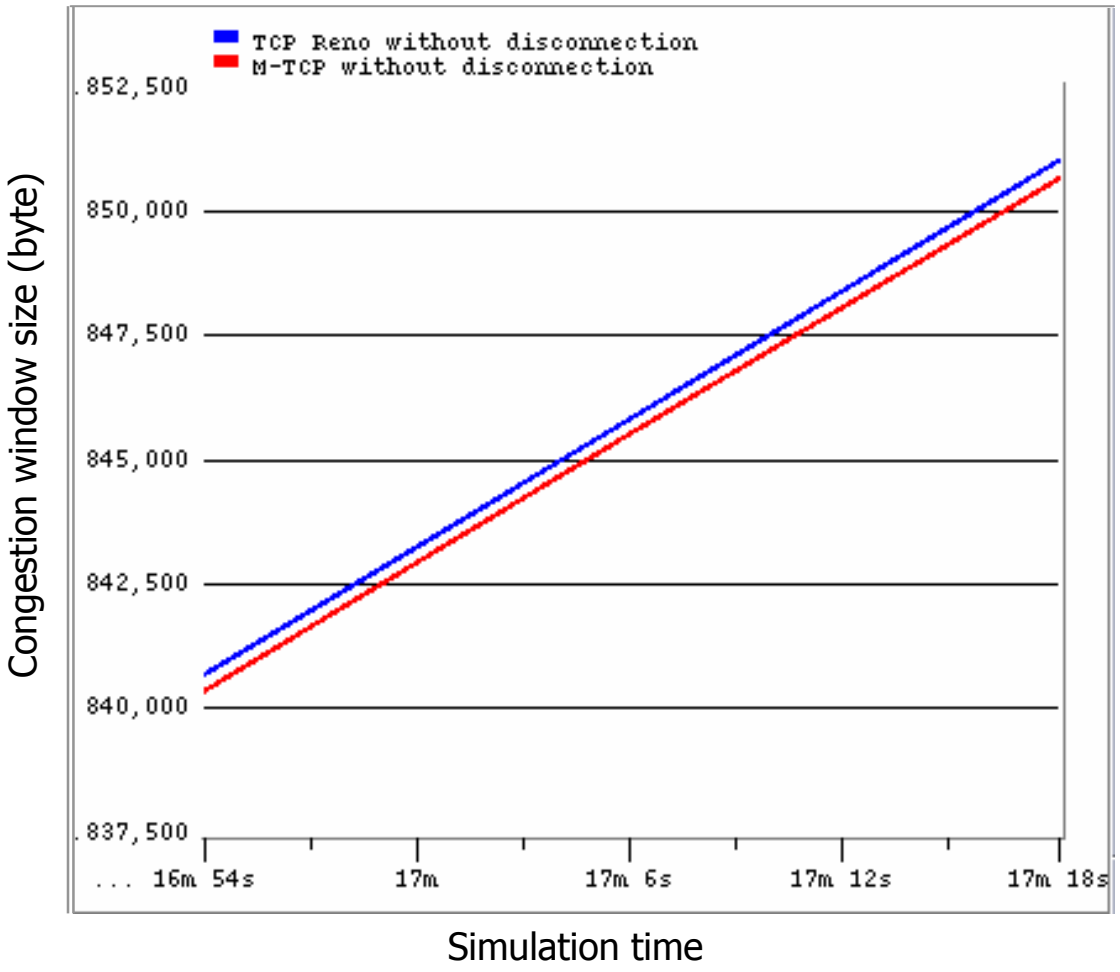
Simulation scenarios and parameters

- Scenarios:
  - TCP Reno with no disconnection is used as the baseline scenario for comparison
  - TCP Reno with disconnection
  - M-TCP without disconnection
  - M-TCP with disconnection
- Parameters:
  - file size: 1Gbyte
  - simulation time: 30 minutes
  - TCP parameters: standard OPNET TCP parameters with delayed ACK option

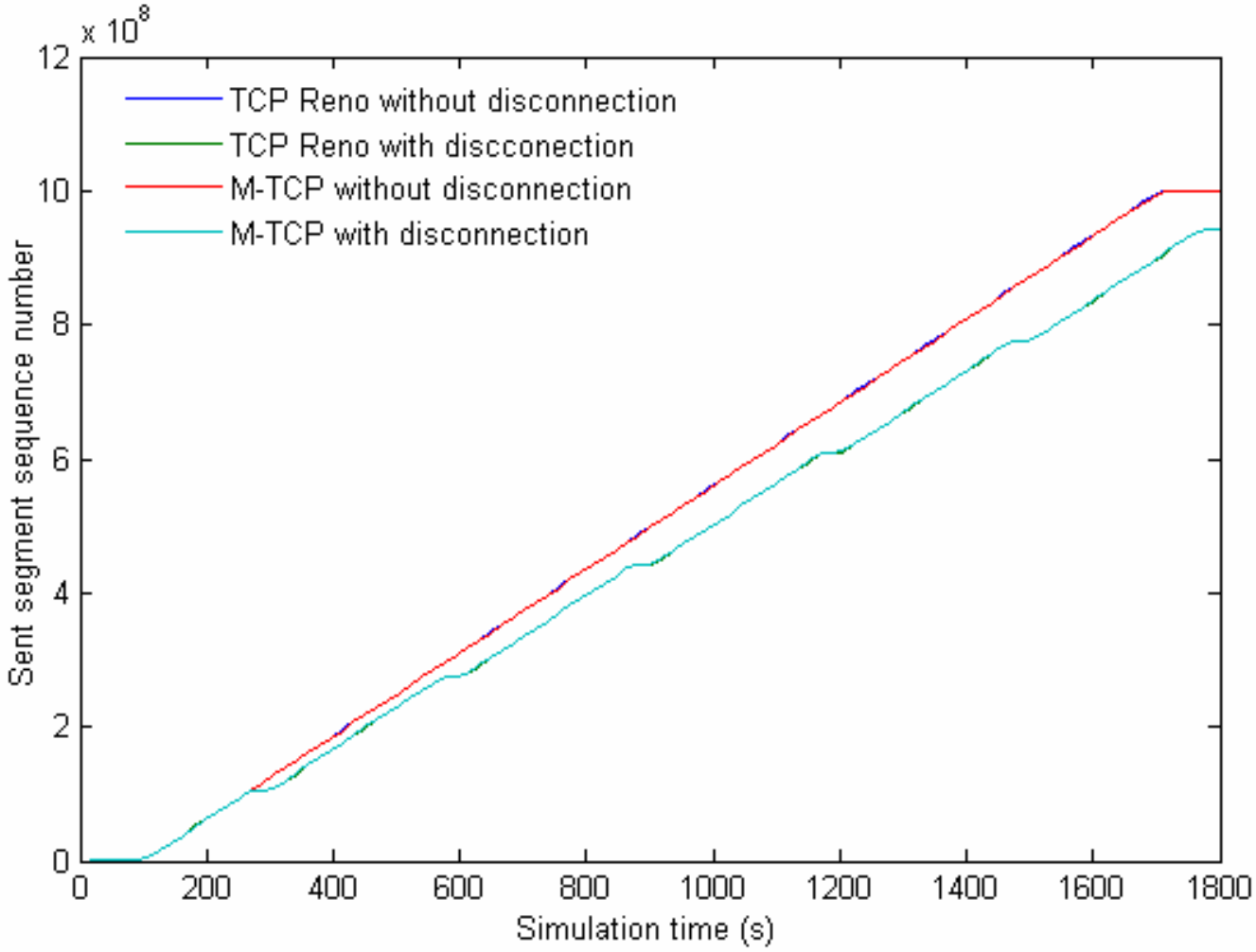
Simulation results: congestion window size



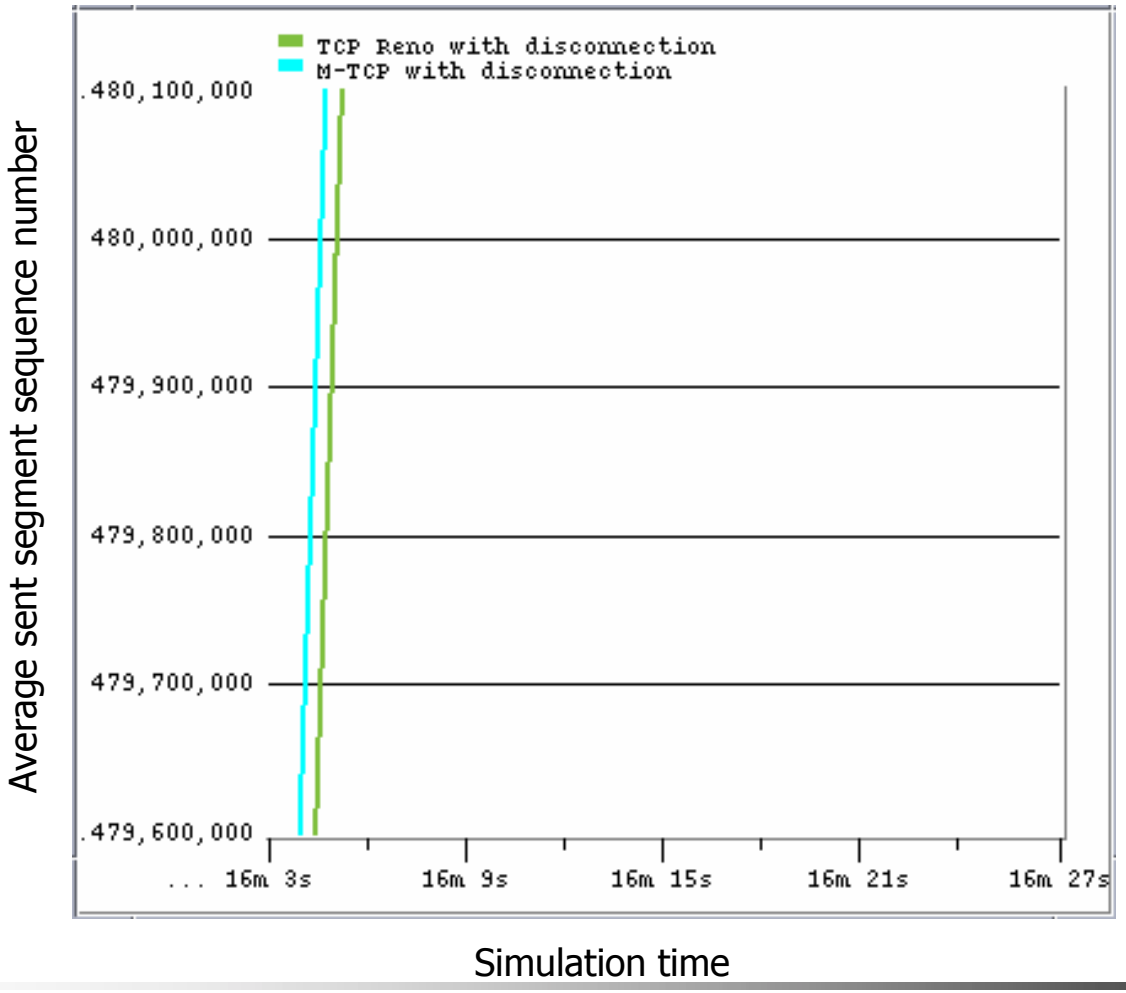
Simulation results: congestion window size



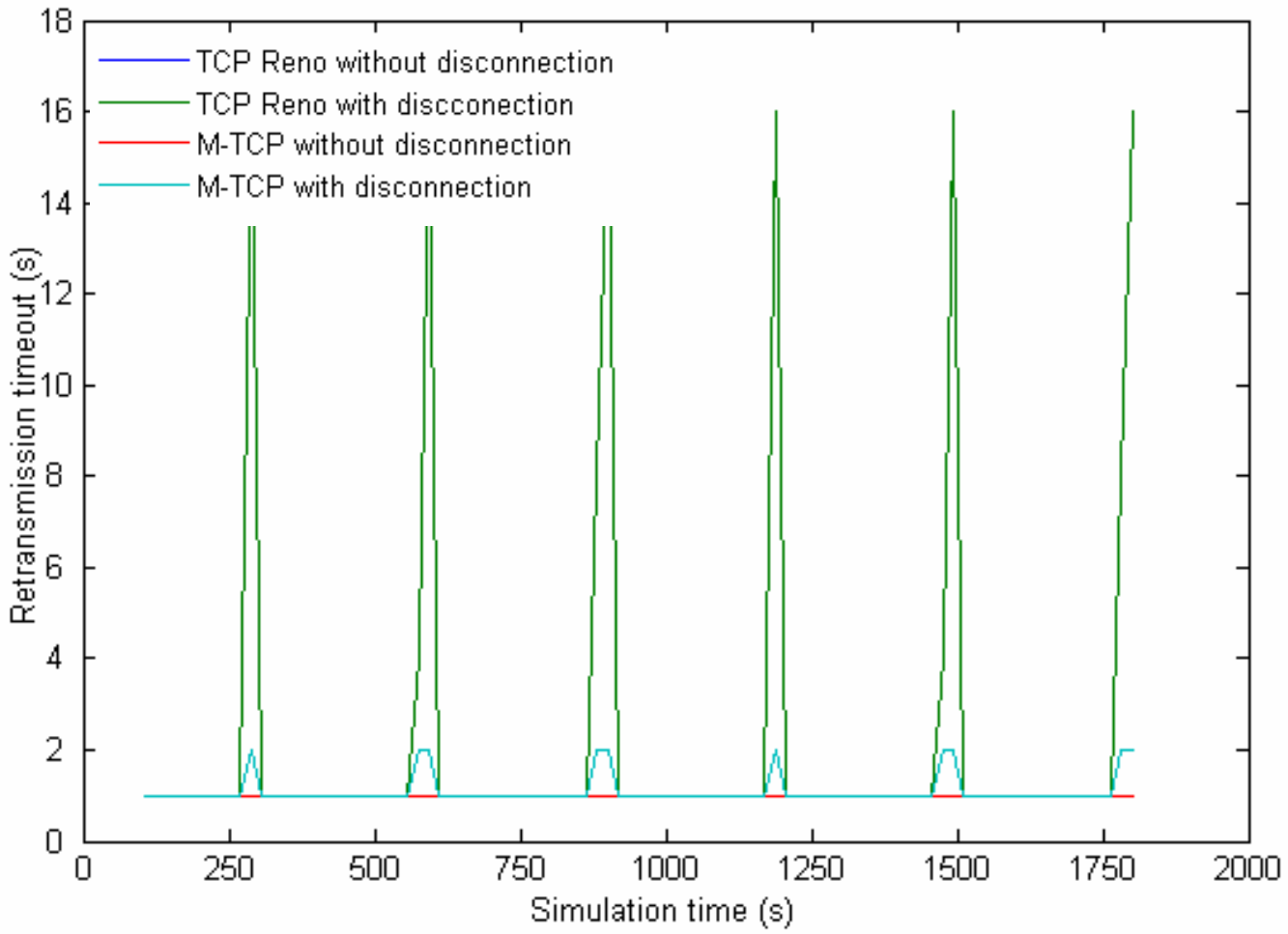
Simulation results: sent segment sequence number



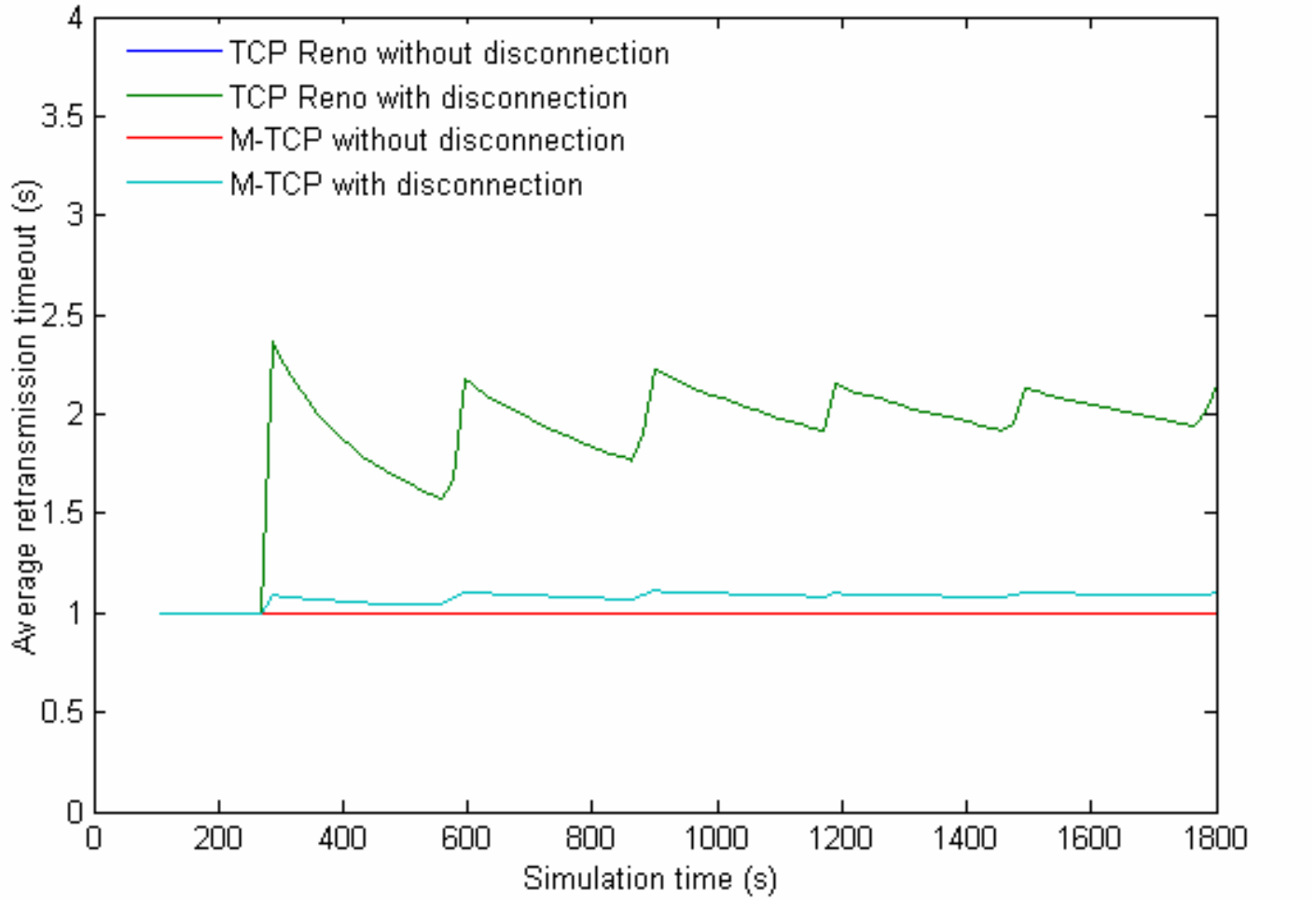
Simulation results: average sent segment sequence number

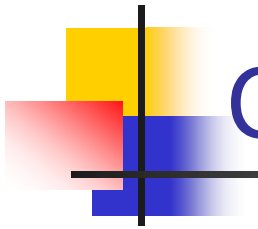


Simulation results: retransmission timeout



Simulation results: average retransmission timeout





Conclusions

- M-TCP performs better than TCP Reno in the presence of disconnection:
  - larger congestion window size
  - higher sent segment sequence number (indicator of goodput)
  - lower retransmission timeout
- M-TCP performs comparably to TCP Reno when there are no disconnections

References

- W. Stevens. *TCP Illustrated*, Volume. 1. Reading, MA: Addison-Wesley, Professional Computing Series, 1984.
- V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM '88 Symposium on Communications Architectures and Protocols*, Stanford, CA, USA, Aug. 1988, pp. 314–329.
- K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communications Review*, vol. 27, no. 5, pp. 19–42, Oct. 1997.
- W. G. Zeng, M. Zhan, Z. Lin, and Lj. Trajković, "Performance evaluation of M-TCP over wireless links with periodic disconnections," *OPNETWORK 2003*, Washington, DC, Aug. 2003.
- A. Bakre and B. R. Badrinath, "Implementation and performance evaluation of Indirect TCP," *IEEE Trans. Computers*, vol. 45, no. 3, pp. 260–278, Mar. 1997.
- OPNET Modeler software [Online]. Available: http://www.opnet.com/products/modeler/home.html.