TCP Packet Control for Wireless Networks

Wan Gang Zeng and Ljiljana Trajković
{wgzeng, ljilja}@sfu.ca

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

communication networks laboratory

Roadmap

- Motivation
- TCP and wireless networks
- Wireless TCP performance
- Proposed algorithm: Packet Controller
- Simulations
- Conclusion
- References

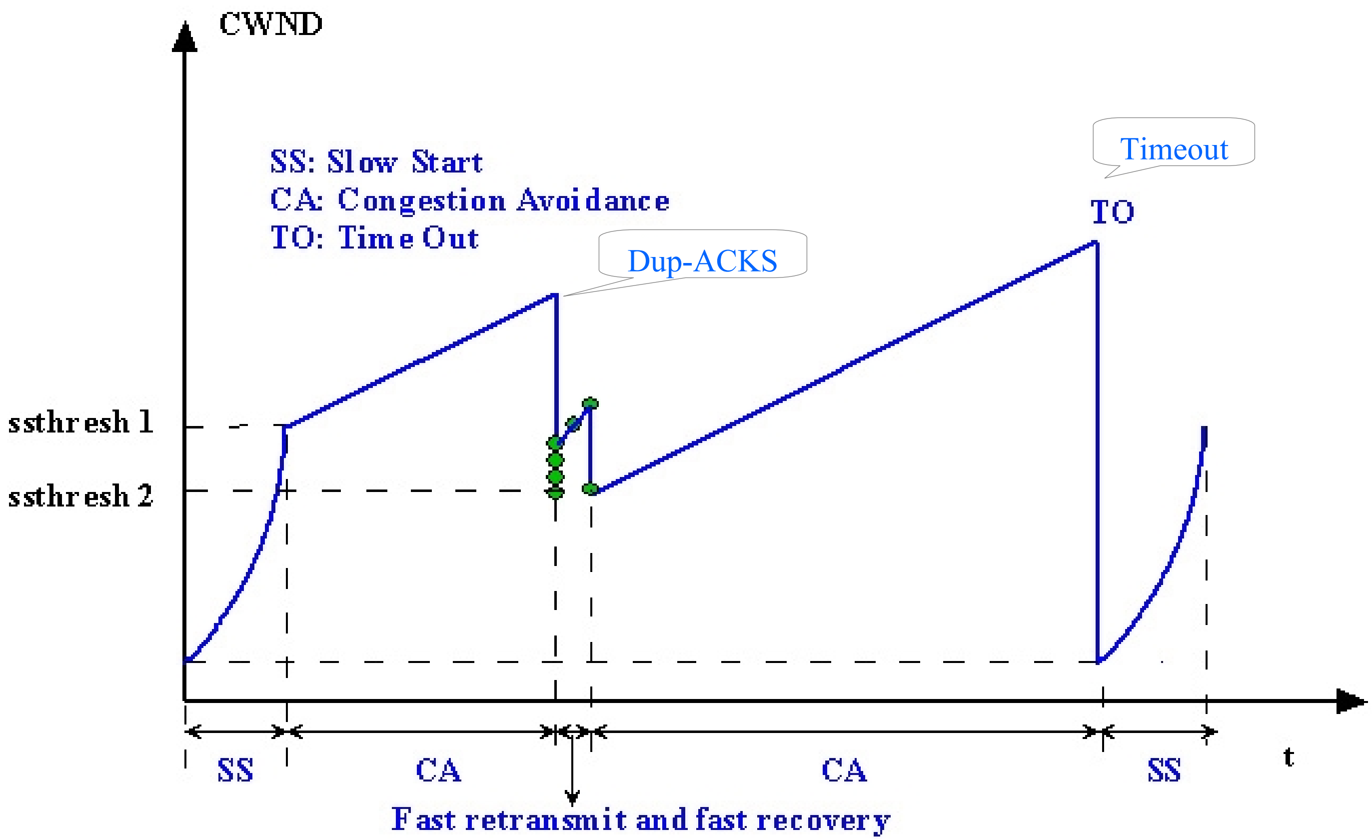
Motivation

- Transmission Control Protocol (TCP) is the most widely used transport protocol in wireline networks:
  - it carries 95% of Internet traffic
- TCP is suitable for data applications
- Important to support application over wireless and wireline links:
  - laptops, PDAs, data-capable cell phones, 3G devices
- Mobile devices require TCP features similar to wireline networks
- TCP in wireless environment demands special attention

TCP

- Transport protocol
- Typically designed for wireline networks
- Essential: ACK paced transmission, window management, and timer based retransmission
- Services:
  - reliability
  - congestion control
  - connection management
  - flow control

TCP: congestion control



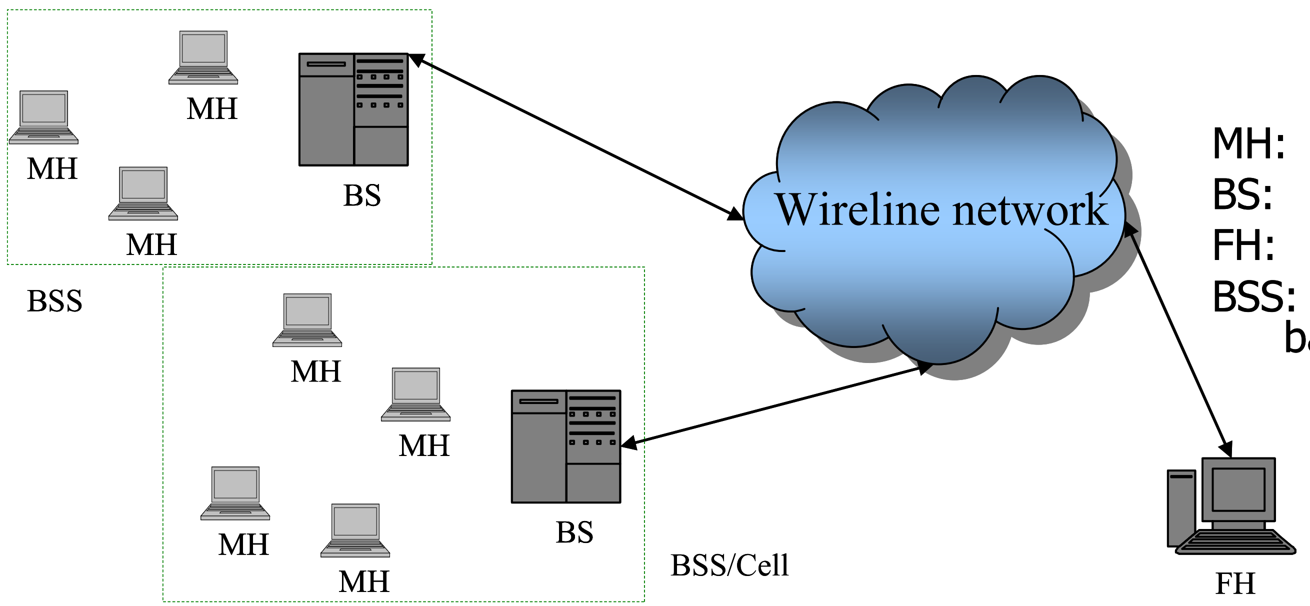
cwnd: congestion window
rwnd: receiver's advertised window

TCP: congestion control

- Monitor packet losses:
  - 3-duplicate ACKs, retransmission timeouts
  - due to congestion in network
- Congestion control:
  - congestion windows (cwnd) size decreases
  - amount of data sent into network decreases
    - sending window = min (cwnd, rwnd)
  - throughput decreases

Wireless network: architecture

- Conventional cellular networks and wireless LANs (WLANs)
- Assumption: MHs are directly connected to BS connected to a wireline Internet



MH: mobile host
BS: base station
FH: fixed host
BSS: infrastructure basic service set

Wireless network: properties

- Properties:
  - high bit-error rate (BER): random loss
  - bursty traffic: mixed voice/data, channel access asymmetry
  - disconnections: handoffs, interferences
- Impact on TCP:
  - fast retransmit, timeouts, large and varying delay

TCP performance: link errors

- Unable to differentiate packet losses caused by congestion from link errors
- Assumes that packet loss is due to congestion and resolves it by decreasing sending rate
- In wireless networks:
  - high probability of packet loss caused by transmission error or mobility
  - temporary
  - network can recover itself
  - congestion control degrades TCP performance

Solutions

- General approach:
  - hide error losses from the sender
  - inform the sender of the cause of packet loss
- Specific designs:
  - split-connection: I-TCP, M-TCP
  - link layer solution: Snoop
  - end-to-end: TCP Westwood, TCP-Jersey

TCP performance: large sudden delay and delay variation

- Large sudden delay and delay variations are caused by:
  - wireless link properties: limited bandwidth, randomness of wireless channel
  - protocols: link layer or media access control (MAC) retransmission schemes
  - queuing algorithms
  - device mobility
  - handoffs
  - different traffic priorities

Large sudden delay and delay variation: adverse effects

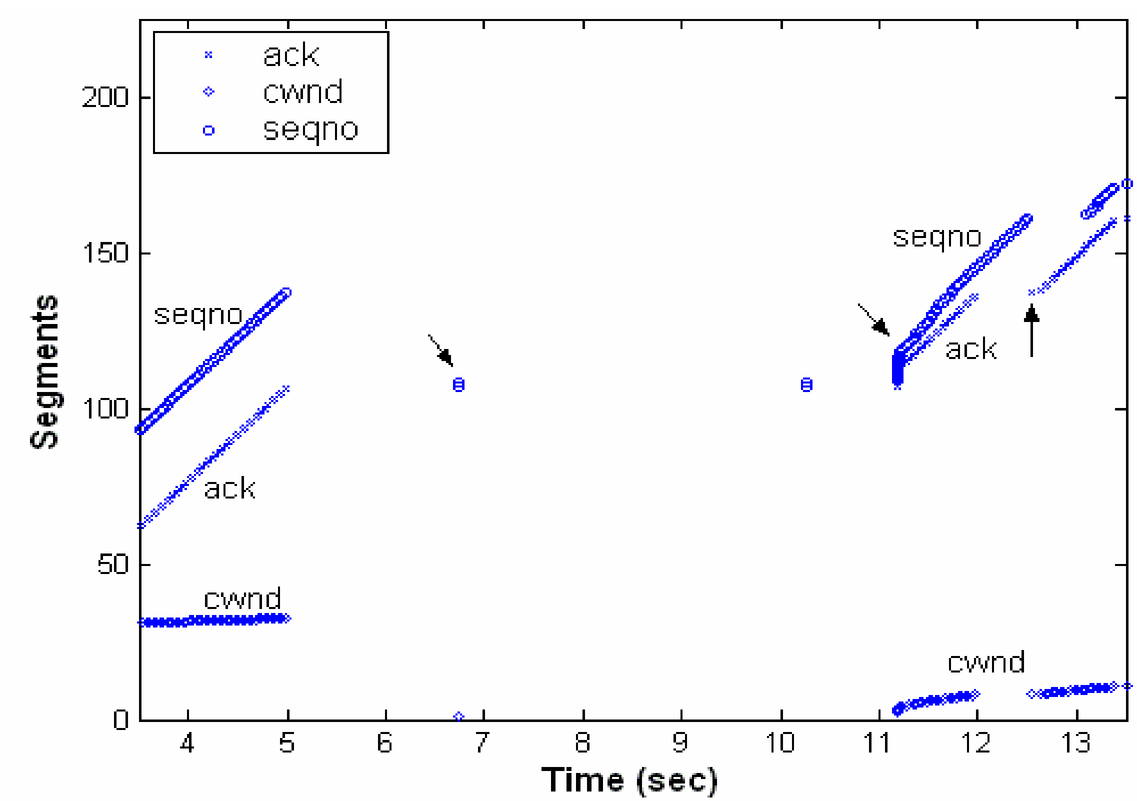
- TCP does not react well to large sudden delay and delay variations on links
- Delay variation causes:
  - spurious fast retransmit
    - caused by packet reordering
    - TCP receives 3-duplicate ACKs
  - ACK compression
    - TCP sender receives accumulated ACKs
    - increases traffic burstiness and the chances of bursty packet losses

Large sudden delay and delay variation: adverse effects

- Large sudden delay causes:
  - ACK compression
  - spurious timeouts
    - TCP's RTT prediction cannot react fast enough to the sudden delay change
  - spurious fast retransmit
    - triggered by TCP's retransmissions after timeouts

ns-2 simulations

- Large sudden delay -> spurious timeout -> retransmission -> duplicate ACKs (spurious fast retransmit) -> longer delay



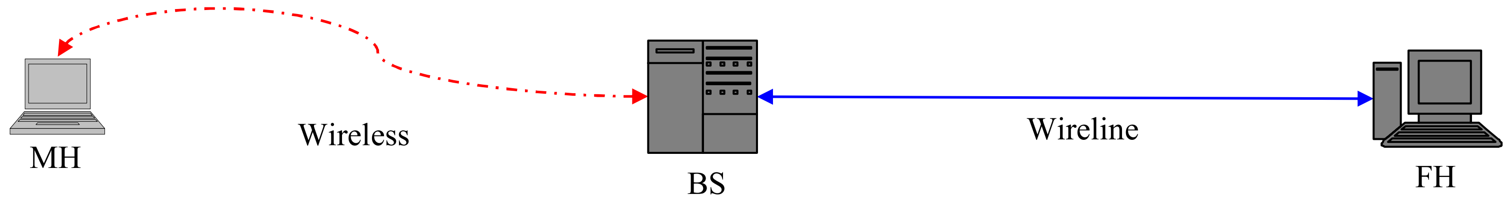
Roadmap

- Motivation
- TCP and wireless networks
- Wireless TCP performance
- Proposed algorithm: Packet Controller
- Simulations
- Conclusion
- References

Design goals

- We consider wireless link characteristics:
  - large sudden delay and delay variation
  - handoff or short disconnections
- The approach is to introduce minimum changes in TCP:
  - simple to implement
  - easy to deploy

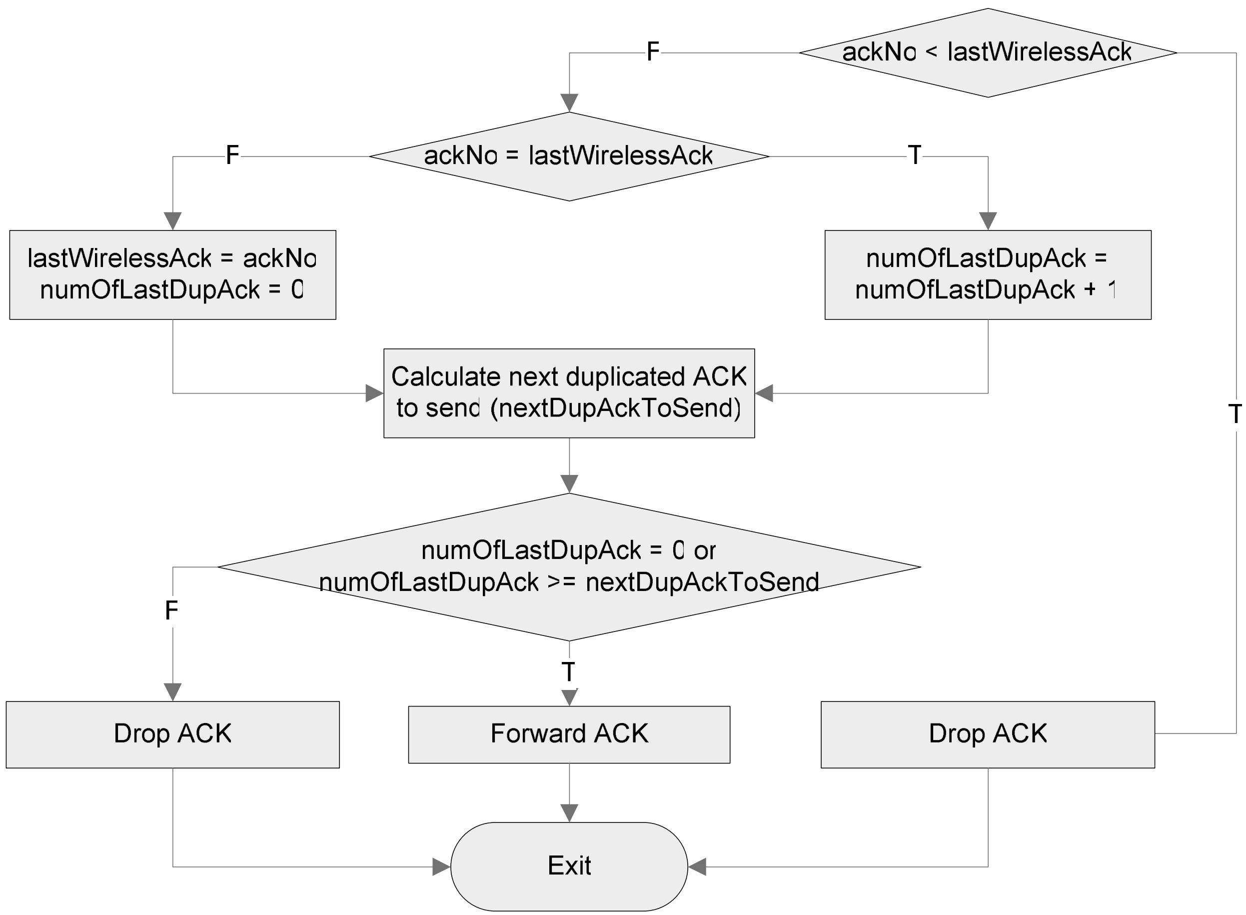
Proposed algorithm: ACK filter spurious fast retransmission



- Modify BS
- Perform packet control for delay variation:
  - duplicate ACK filter:
    - redefine “3-dup ACK” threshold to detect packet losses (dupAckThresh)
    - filters duplicate ACKs

Implementation: ACK filter

- ACK packet received from MH to FH



Proposed algorithm: Data filter spurious timeouts

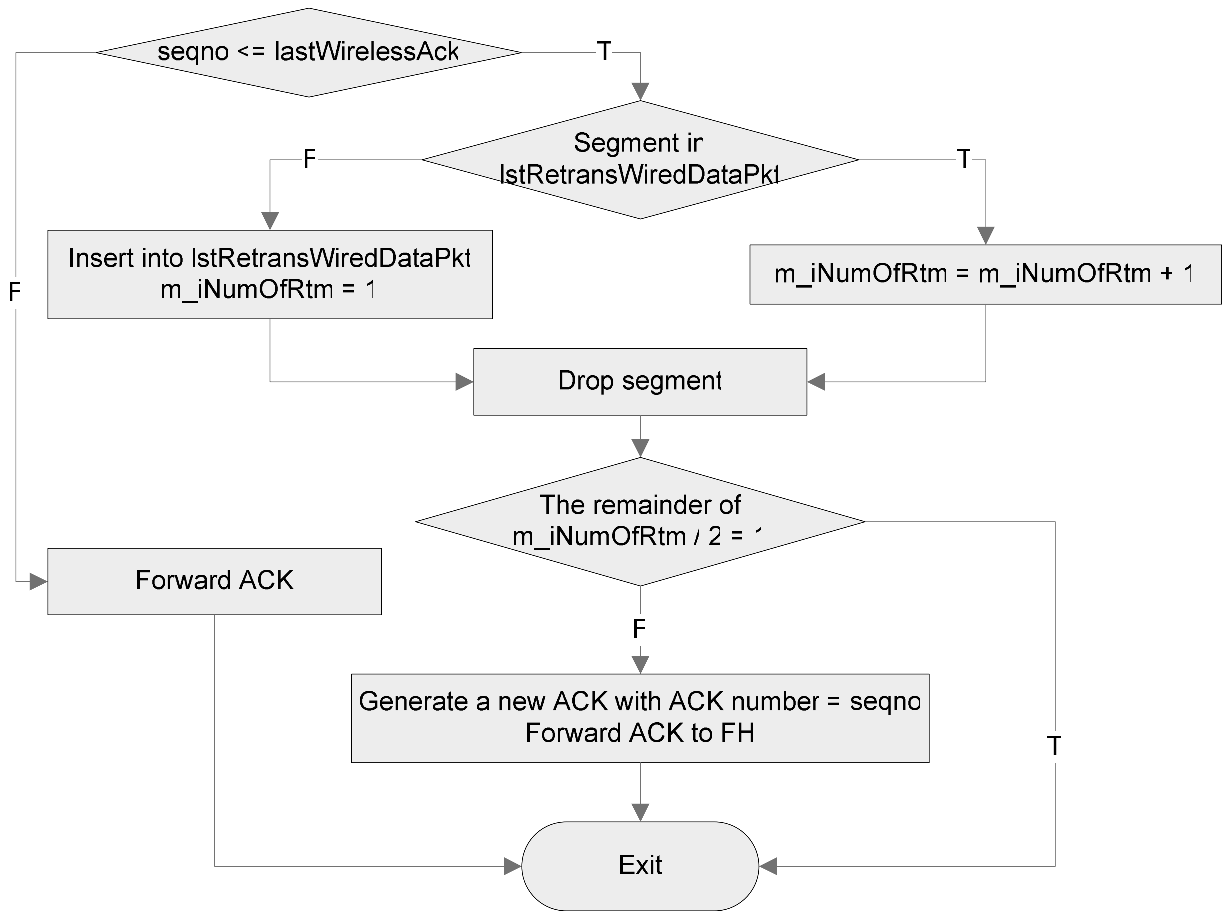
- Idea: prevent chain reaction caused by spurious timeouts
  - unnecessary packet retransmission
  - spurious fast transmission
- For every data packet received from FH:
  - Case 1: new packet
  - Case 2: packet is a retransmission,
    but has not been ACKed
    - handle it as in Case 1

Proposed algorithm: Data filter

- Case 3: packet is a retransmission and has been ACKed
  - store the last ACK sent to FH (lastAckSent)
  - compare lastAckSent with the seqno of packet received
  - if packet is ACKed, drop the packet: avoids additional DUPACKs, reduces response time and saves wireless bandwidth
  - to consider the case when ACK packet is really lost on wireline link, for every 2 retransmissions of the same segment, send an ACK from BS

Implementation: Data filter

- Data packet received from FH to MH:



Design approach

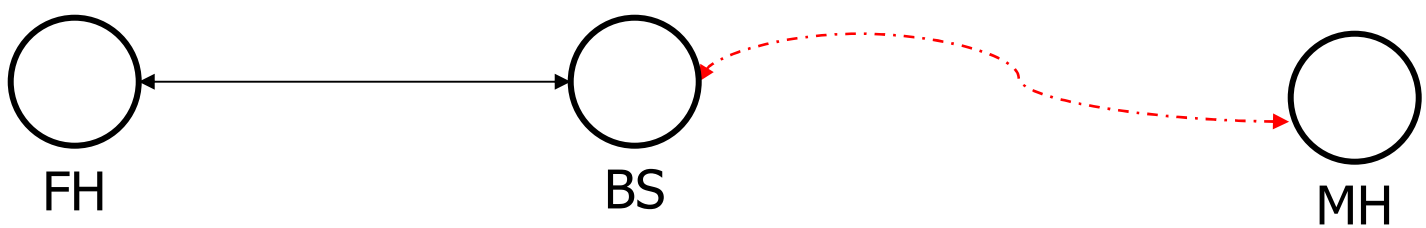
- TCP option, as opposed to end-to-end solution:
  - TCP is the most successful and tested transport protocol
  - wireless links require services similar to those provided by TCP in wireline links
  - any new protocol would require thorough validation and would face difficulties of deployment in the existing network
- Introduced ACK “queue” in BS does not require memory storage

Design approach

- Proposed design has small impact on handoff operations
- Example:
  - Snoop requires implementation of a data packet queue in BS, which needs to be transferred to the next BS during handoffs
  - this additional memory transfer increases handoff time

Simulation setup

- Simulator: ns-2.26 (old version number ns-2.1b10)
- OS: RedHat Linux 9
- Network setup

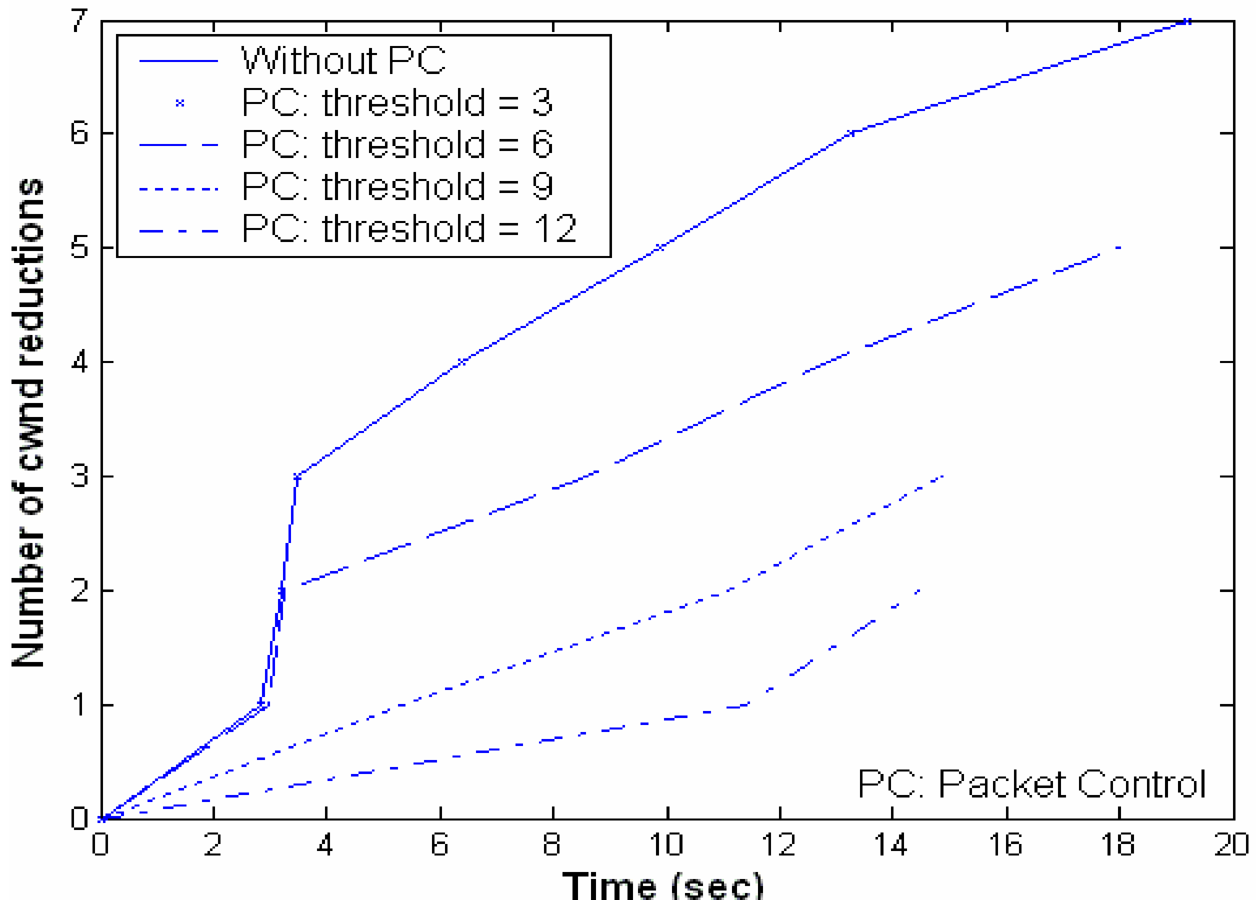


Scenario 1: delay variation

- Scenario: delay variation with small delay
  - FTP data is sent from FH to MH for 20 sec
  - TCP Reno
  - TCP data packet size: 1040 bytes (default in ns-2)
  - link FH-BS:
    - link capacity: 250 Kbps, 5 ms delay
    - queue: DropTail
  - link BS-MH:
    - link capacity: 250 Kbps, around 190 ms of variable delay, delay variation simulated since time 0.5 sec
    - queue: DropTail

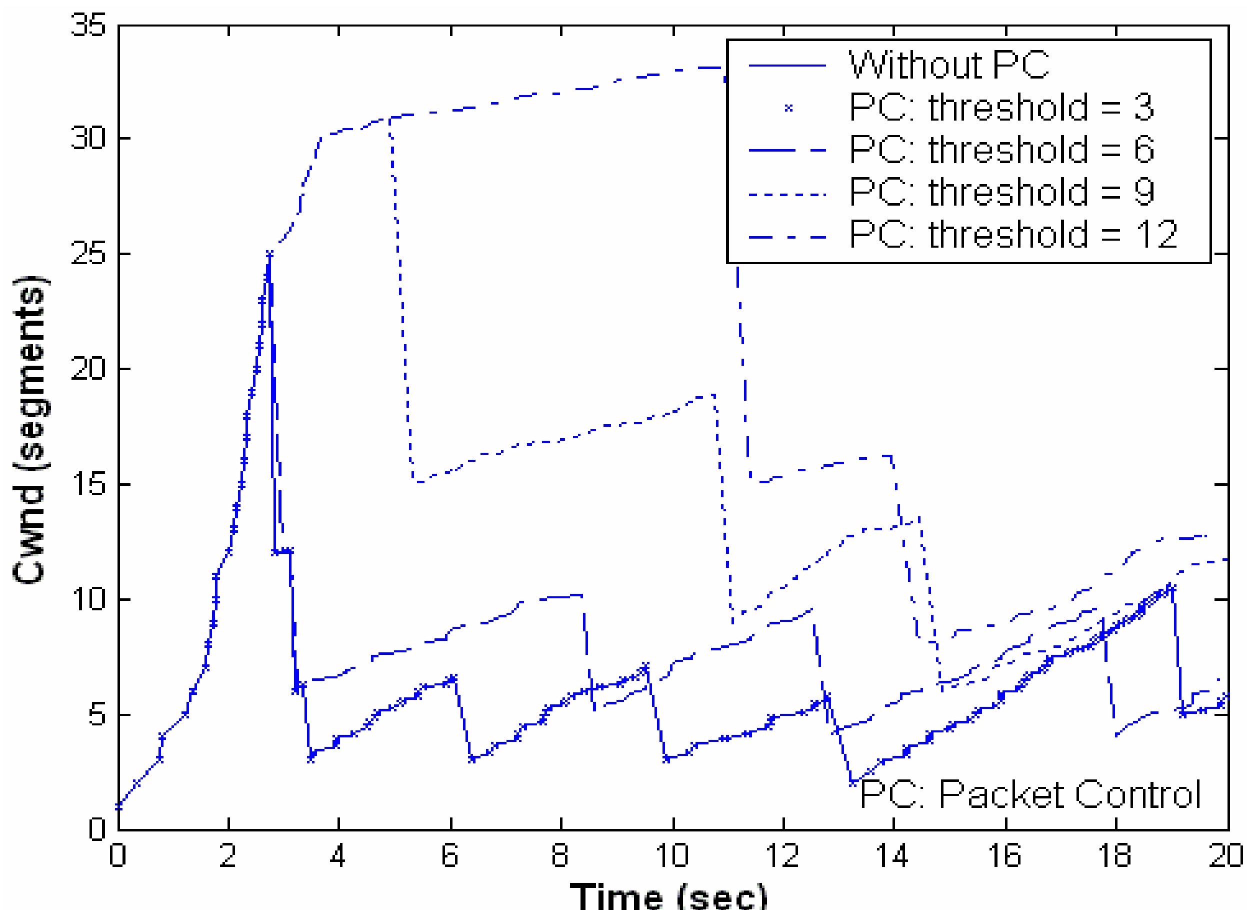
ns-2 simulations: cwnd reductions

- Larger dupAckThresh results in smaller cwnd reductions



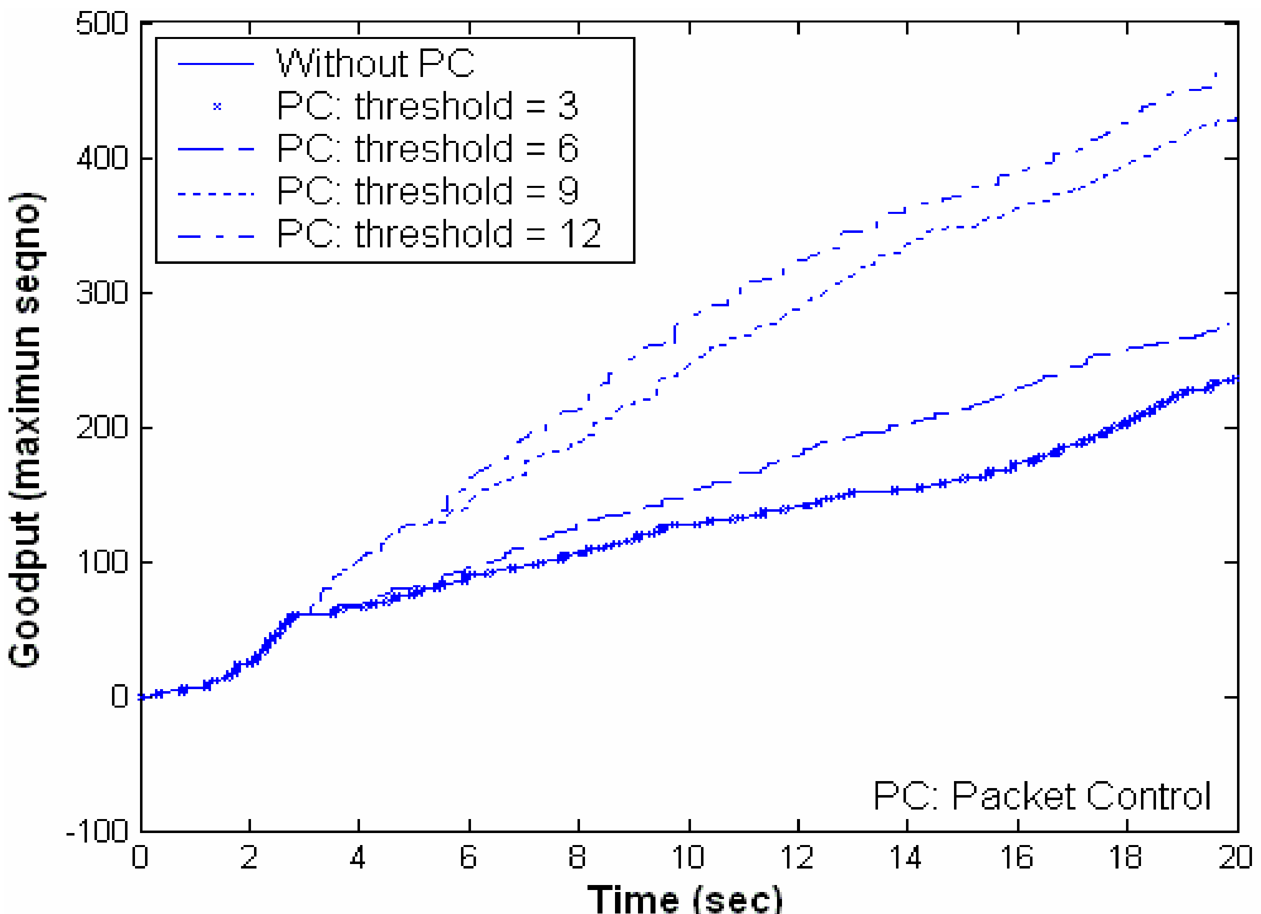
ns-2 simulations: cwnd

- Larger dupAckThresh results in higher cwnd



ns-2 simulations: goodput

- Larger dupAckThresh results in higher goodput

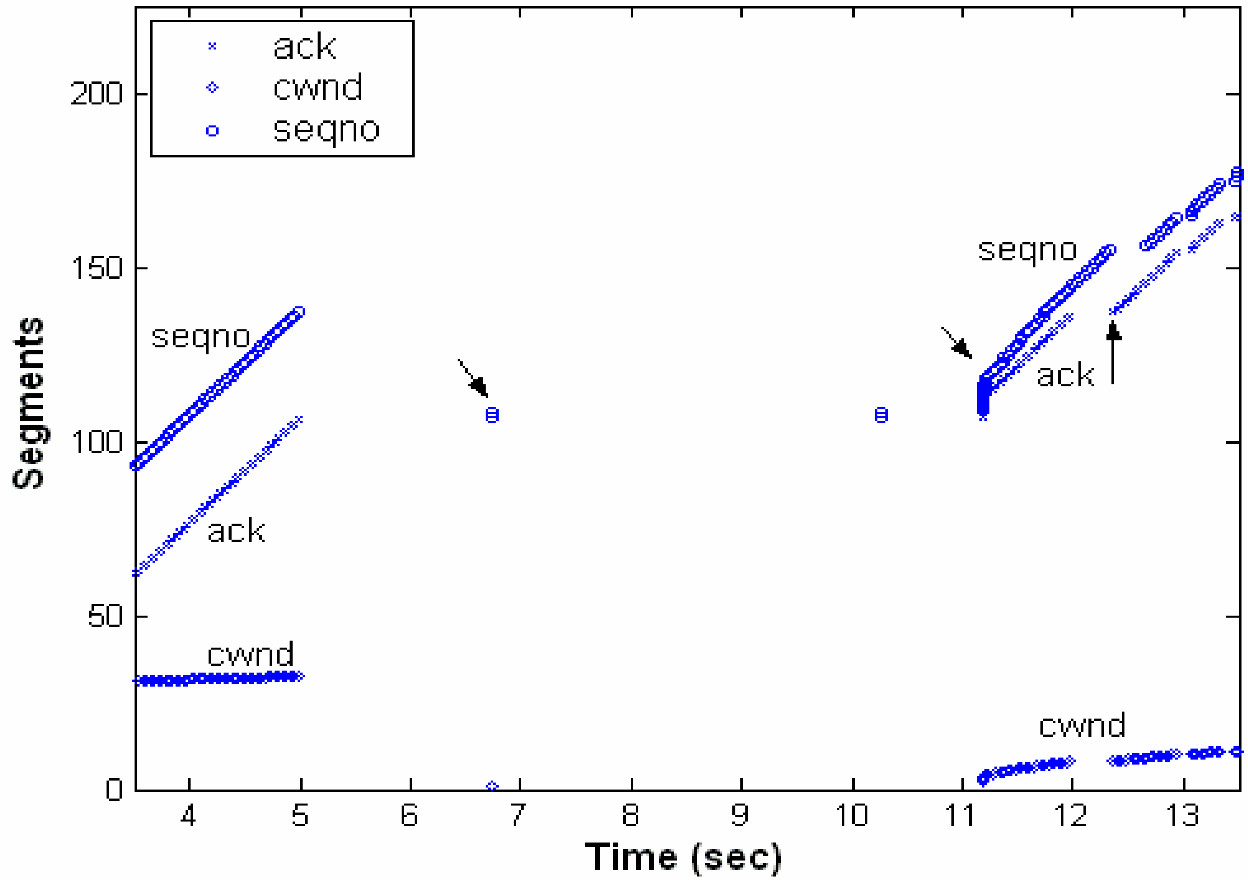


Scenario 2: spurious timeouts

- Scenario: sudden large delay
  - configurations as in Scenario 1, but with a large delay at time 5 sec. Delay lasts for 6 sec.
- Purpose: to investigate how TCP reacts to sudden large delay increase and how Packet Controller may help

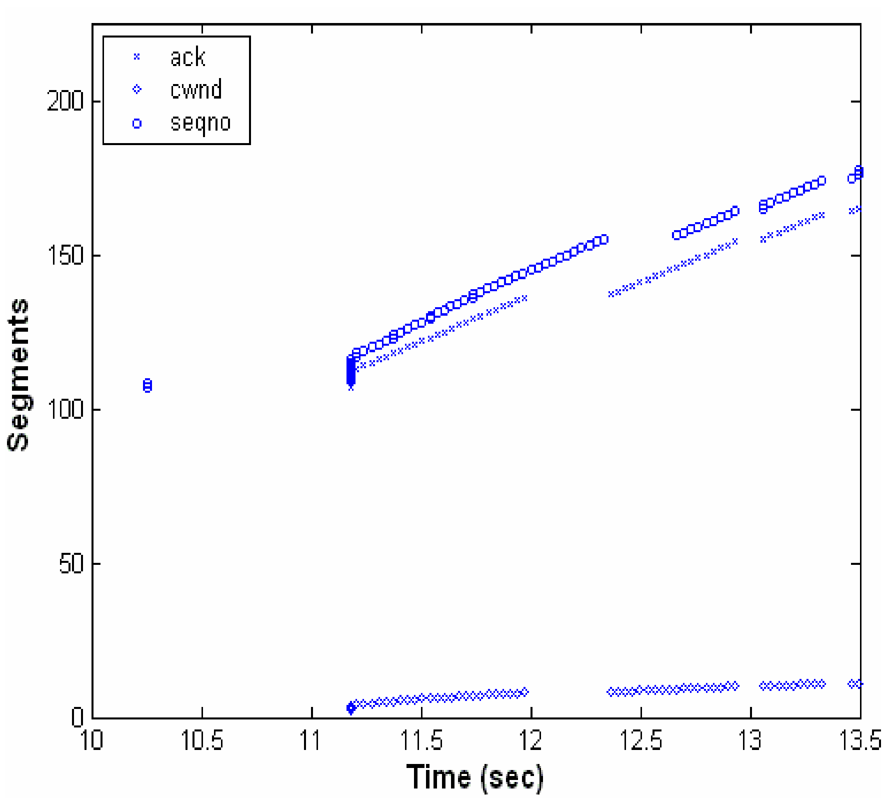
ns-2 simulations: ack, cwnd, seqno

- Large sudden delay -> Spurious timeout -> retransmission -> duplicate ACKs (spurious fast retransmit) -> longer delay

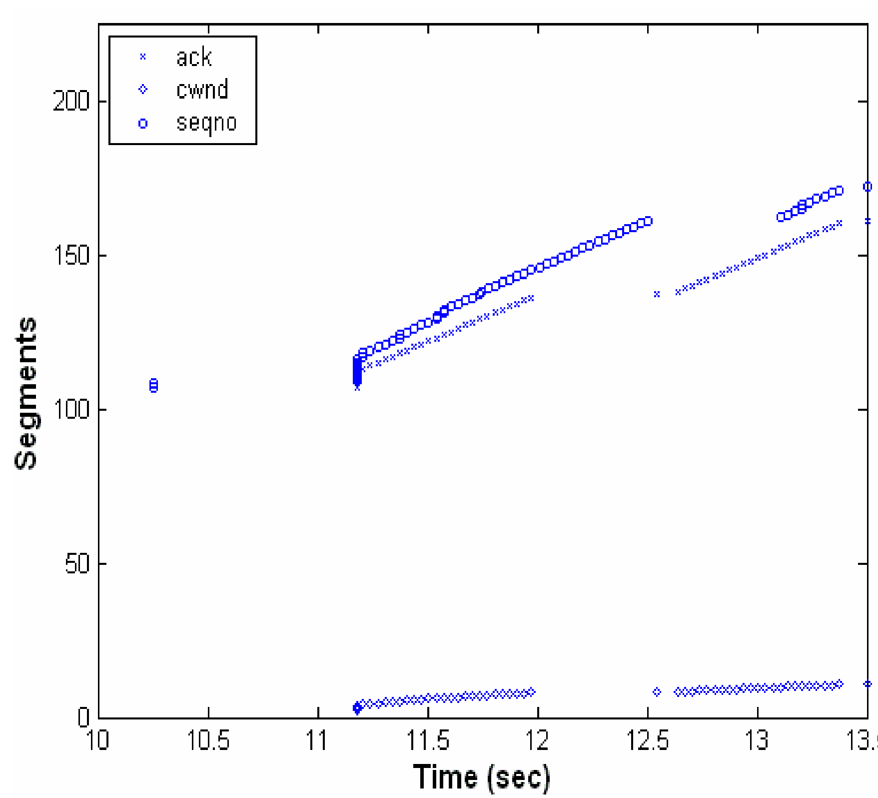


ns-2 simulations: ack, cwnd, seqno

With Packet Control

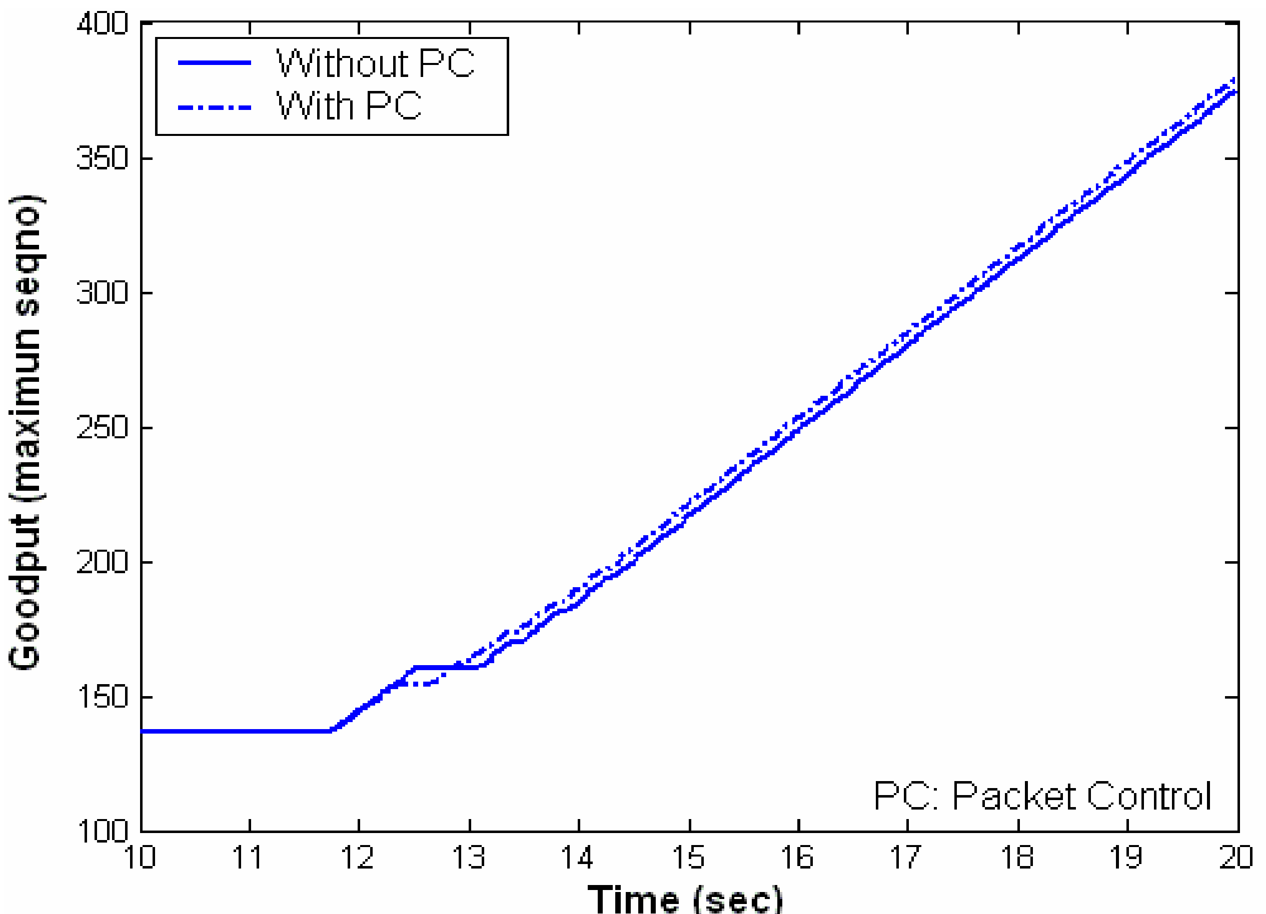


Without Packet Control



ns-2 simulations: goodput

- Higher goodput is achieved with Packet Control



Conclusions

- Contribution: proposed packet controller successfully reduces spurious fast retransmit and spurious timeouts caused by delay variation and large sudden delay in wireless link
- TCP goodput increased over 60% in the simulation scenario 1 (the exact improvement is highly depended on actual link characteristics)
- Chain reaction of spurious timeout is stopped and TCP performance is improved

Conclusions

- Proposed algorithm is easy to implement and deploy
  - modifications is required only on BS
- Proposed algorithm does not change TCP's congestion control semantics for end users
  - it should not pose restriction on future TCP evolution
- Has small impact on handoffs

References

- W. Stevens, *TCP/IP Illustrated*, Volume 1: The Protocols. Reading, MA: Addison-Wesley, Professional Computing Series, 1984.
- V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 314–329.
- M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," *IETF RFC* 2581, Apr. 1999.
- I. Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*. New York, NY: John Wiley & Sons, 2002.
- H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," *IETF RFC 3481*, Feb. 2003.
- S. Floyd, "A report on recent developments in TCP congestion control," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 84–90, Apr. 2001.
- A. Gurtov, "Effect of delays on TCP performance," in *Proc. IFIP Personal Wireless Communications* (*PWC'01*), Aug. 2001.
- R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmission," *ACM Computer Communications Review*, vol. 30, no. 1, pp. 30–36, Jan. 2000.
- S. Fu, M. Atiquzzaman, and W. Ivancic, "Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment," in *Proc. International Conference on Computer Communications and Networks*, Miami, FL, Oct. 2002, pp. 575–578.
- K. Luo and A. O. Fapojuwo, "Impact of terminal mobility on TCP congestion control performance," in *Proc. 3rd IASTED International Conference on Wireless and Optical Communications*, Banff, Canada, July 2003, pp. 360–365.
- M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proc. ACM MobiCom*, Atlanta, GA, Sept. 2002, pp. 71–82.

Thank You!

TCP: RTT, RTO, and Karn's algorithm

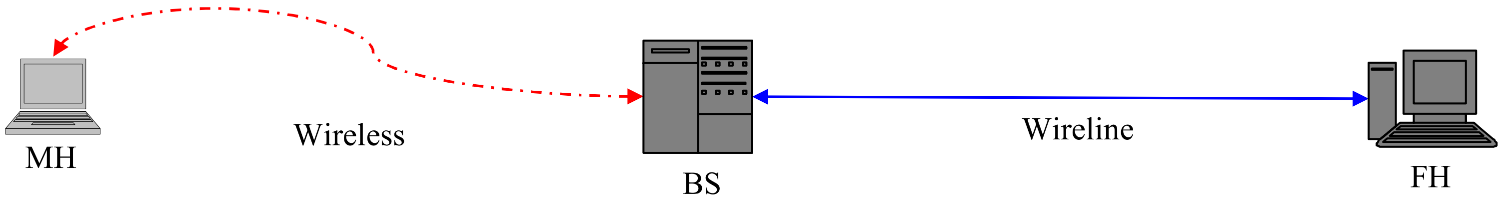
- RTO is calculated based on RTT and RTT deviation
- Karn's algorithm:
  - sample RTT: sampleRtt is measured for data packets
  - smoothed RTT:

    srtt = (1 – g) * srtt + g * sampleRTT
  - mean deviation:

    rtttvar = (1 – h) * rtttvar – h * | sampleRTT – srtt |
  - recommended values for g, h

    g = 1/8 = 0.125, h = ¼ = 0.25
  - RTO = srtt + 4 rttvar

Proposed solution: spurious fast retransmission



- Parameter dupAckThresh are:
  - global for the wireless link between BS and MH
  - are connection independent
  - they reflect properties of the link
- Possible performance improvements for multiple connections

Implementation: receiving data packet

- When data packet received from FH to MH:
  - record seqno, current time in packet queue

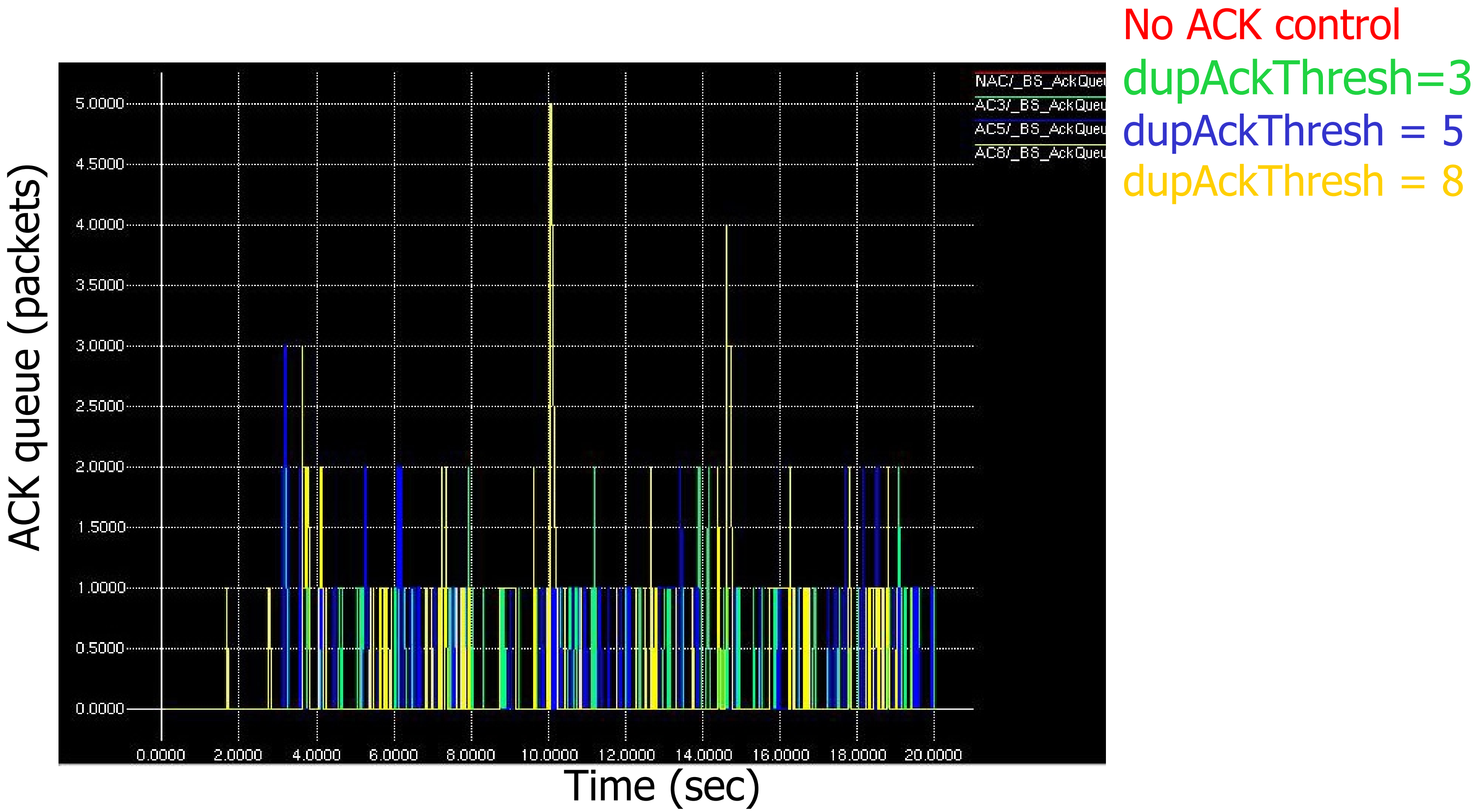
```
struct PacketInfo
{
  int m_iSeqno;
  //time packet was recieved
  double m_dSendTime;
  int m_iNumOfDupAcks;
}
```

Implementation: receiving ACK

- What it does:
  - filters duplicate ACKs
  - redefines duplicate ACK threshold to detect packet losses

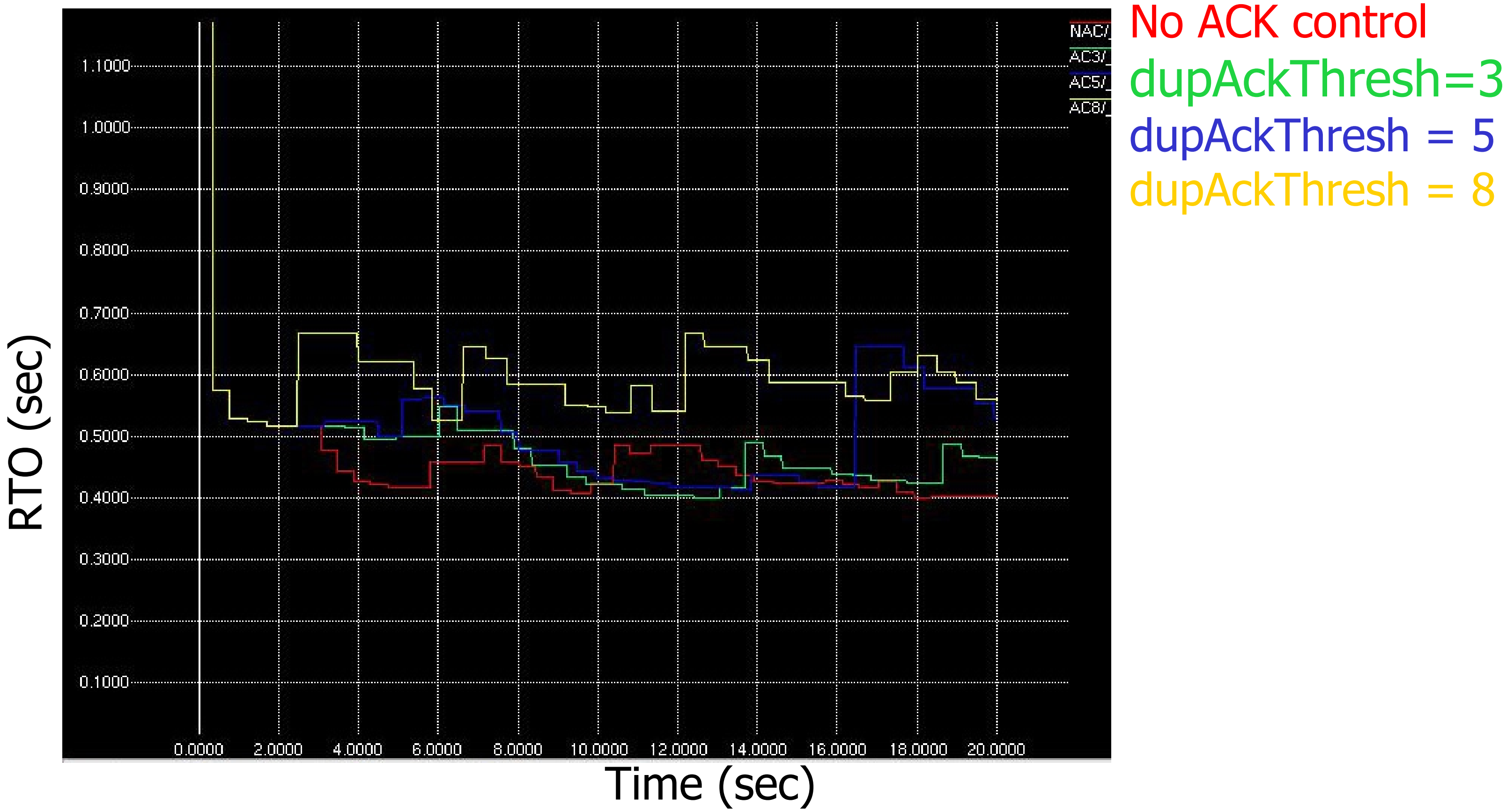
ns-2 simulations: ACK queue

- Larger dupAckThresh results in larger ACK queue



ns-2 simulations: RTO

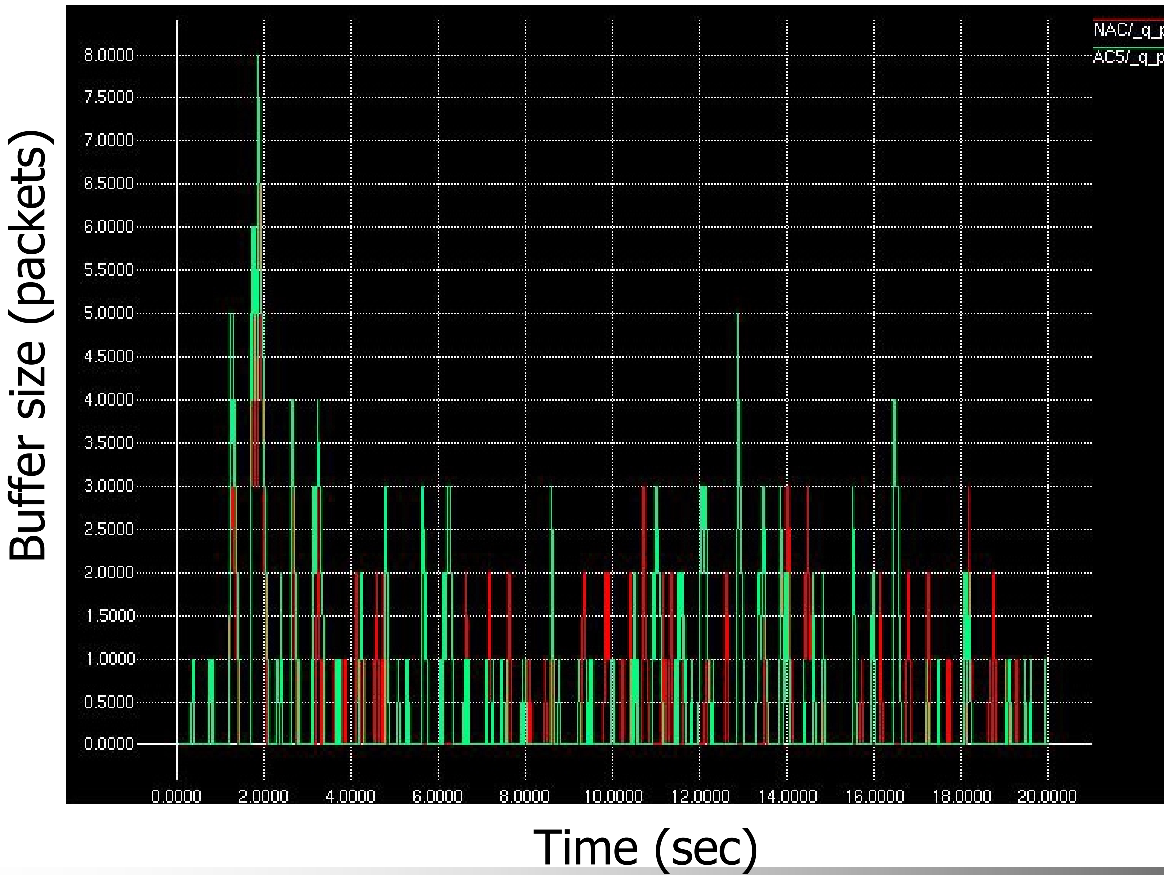
- Larger dupAckThresh -> larger RTO -> possible longer recovery time when there is timeouts



ns-2 simulations: data buffer size

- Larger dupAckThresh -> larger buffer required

No ACK control
dupAckThresh = 8



ns-2 simulations: cwnd

- With ACK Control, higher cwnd is achieved

Future work

- Implement ACK compression
- More accurate traffic delay generator in sumulations
- Consider short-connections:
  - web traffic, short messaging service
  - packet loss usually results in timeouts
  - considerable costs to recover

Future work

- Consider short-connections:
  - approach: classify short connections and use wireless link information we collected on BS to detect timeouts and use 3-dup ACKs to trigger retransmission.
    - cwnd reduction has little effect
    - retransmission of packets for short connections has little effect on network utilization

References

- W. Stevens, *TCP/IP Illustrated*, Volume 1: The Protocols. Reading, MA: Addison-Wesley, Professional Computing Series, 1984.
- Information Sciences Institute, "Transmission control protocol," *IETF RFC* 793, Sept. 1981.
- V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 314–329.
- I. Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*. New York, NY: John Wiley & Sons, 2002.
- M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," *IETF RFC* 2581, Apr. 1999.
- S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," *IETF RFC 2582*, Apr. 1999.
- S. Floyd, "A report on recent developments in TCP congestion control," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 84–90, Apr. 2001.
- H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," *IETF RFC 3481*, Feb. 2003.
- A. Gurtov, "Effect of delays on TCP performance," in *Proc. IFIP Personal Wireless Communications* (*PWC'01*), Aug. 2001.

References

- A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proc. 15th International Conference on Distributed Computing Systems* (*ICDCS*), Vancouver, BC, May 1995, pp. 136–143.
- K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 747–756, May 2004.
- H. Balakrishan, S. Seshan, E. Amir, and R. H. Katz, "Improving TCP/IP performance over wireless networks," in *Proc. ACM MobiCom*, Berkeley, CA, Nov. 1995, pp. 2–11.
- R. Ludwig and R. H. Katz, "The Eifel algorithm: making TCP robust against spurious retransmission," *ACM Computer Communications Review*, vol. 30, no. 1, pp. 30–36, Jan. 2000.
- S. Fu, M. Atiquzzaman, and W. Ivancic, "Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment," in *Proc. International Conference on Computer Communications and Networks*, Miami, FL, Oct. 2002, pp. 575–578.
- K. Luo and A. O. Fapojuwo, "Impact of terminal mobility on TCP congestion control performance," in *Proc. 3rd IASTED International Conference on Wireless and Optical Communications*, Banff, Canada, July 2003, pp. 360–365.

References

- K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 5, pp. 19–42, Oct. 1997.
- J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby "Performance enhancing proxies," Internet Draft [Online]. Available: http://community.roxen.com/developers/idocs/drafts/draft-ietf-pilc-pep-04.html.
- C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *Wireless Networks* (*WINET*), vol. 8, no. 5, pp. 467–479, Sept. 2002.
- P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bhargavan, "WTCP: a reliable transport protocol for wireless wide-area networks," in *Proc. ACM MobiCom*, Seattle, WA, Aug. 1999, pp. 231–241.
- P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocol," *ACM Transactions on Computer Systems*, vol. 9, no. 4, pp. 364–373, Nov. 1991.
- ns–2 [Online]. Available: http://www.isi.edu/nsnam/ns.
- M. C. Chan and R. Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation," in *Proc. ACM MobiCom*, Atlanta, GA, Sept. 2002, pp. 71–82.