Selective-TCP for Wired/Wireless Networks

Rajashree Paul
rpaul2@cs.sfu.ca

Communication Networks Laboratory

Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

Motivation

- TCP is a transport protocol extensively used over the Internet: 95% of IP traffic in 2002
- TCP performance degrades in wireless networks and thus, also over mixed wired-wireless networks (cellular networks)
- Main reason for TCP's poor performance in wireless networks is TCP's inability to distinguish losses due to wireless link errors from losses due to congestion

Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Comparison of Selective-TCP with TCP packet control
- Conclusions

Background

- Three approaches:
  - end-to-end: TCP Reno, TCP NewReno
  - split connection: I-TCP, M-TCP
  - link layer level: Snoop, TCP packet-control
- End-to-end schemes are promising:
  - significant performance gain without help from intermediate nodes: though not as effective as link layer based approaches in handling wireless losses

H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *in Proc. SIGCOMM*, vol. 26, no. 4, pp. 256–269, Aug. 1996.

Related work

- Improving TCP performance in wired/wireless network by detecting the type of loss:
- End-to-end:

C. P. Fu and S.C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE J. Select. Areas Commun.,* vol. 21, no. 2, pp. 216–228, Feb. 2003.

N. K. G. Samaraweera, "Non-congestion packet loss detection for TCP error recovery using wireless links," *IEE Proceedings-Communications,* vol. 146, no. 4, pp. 222–230, Aug. 1999.

D. Barman and I. Matta, "Effectiveness of loss labeling in improving TCP performance in wired/wireless networks," in *Proc. 10th IEEE International Conference on Network Protocols*, Boston, MA, Nov. 2002, pp. 2–11 2002.

C. Parsa and J. J. Garcia-Luna-Aceves, "Differentiating congestion vs. random loss: a method for improving TCP performance over wireless links," in *Proc. of IEEE Conference on Wireless Communications and Networking*, Chicago, IL, Sept. 2000, vol. 1, pp. 90–93.

T. Kim, S. Lu, and V. Bharghavan, "Improving congestion control performance through loss differentiation," in *Proc. Eighth International Conference on Computer Communications and Networks*, Oct. 1999, pp. 412–418.

V. Tsaoussidis and C. Zhang, "TCP-Real: receiver-oriented congestion control," *Computer Networks*, vol. 40, no. 4, pp. 477–497, Nov. 2002.

Related work

- Improving TCP performance over wired-wireless network by detecting the type of loss:
- Link layer level:

F. Sun, V. O. K. Li, and S. C. Liew, "Design of SNACK mechanism for wireless TCP with new snoop," in *Proc. IEEE Wireless Communications and Networking Conference*, Atlanta, GA, Mar. 2004, vol. 2, pp. 1051–1056.

K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE J. Select. Areas Commun.,* vol. 22, no. 4, pp. 747–756, May 2004.

Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

Selective-TCP algorithm: overview

- An end-to-end solution
- Implemented in ns version 2.27
- Type of loss detected at the receiver
- Corrective measures:
  - wireless loss: Selective Negative Acknowledgement (SNACK) sent to the sender
  - congestion loss: sender's congestion window size set according to the bandwidth measured at the receiver
  - bandwidth = (*no. of lost packets* × *size of packets in bits*) / (*inter-arrival time between last in-sequence packet received and most recent packet* × 1,000) kbps

S. Biaz and N. H. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *Proc. IEEE Symposium on Application-Specific Systems and Software Engineering and Technology*, Richardson, TX, Mar. 1999, pp. 10–17.

D. Anantharaman, "Performance analysis of snack in satellite networks through simulation," M.S. Thesis, Lamar University, Lamar, TX, 2004.

Loss detection scheme

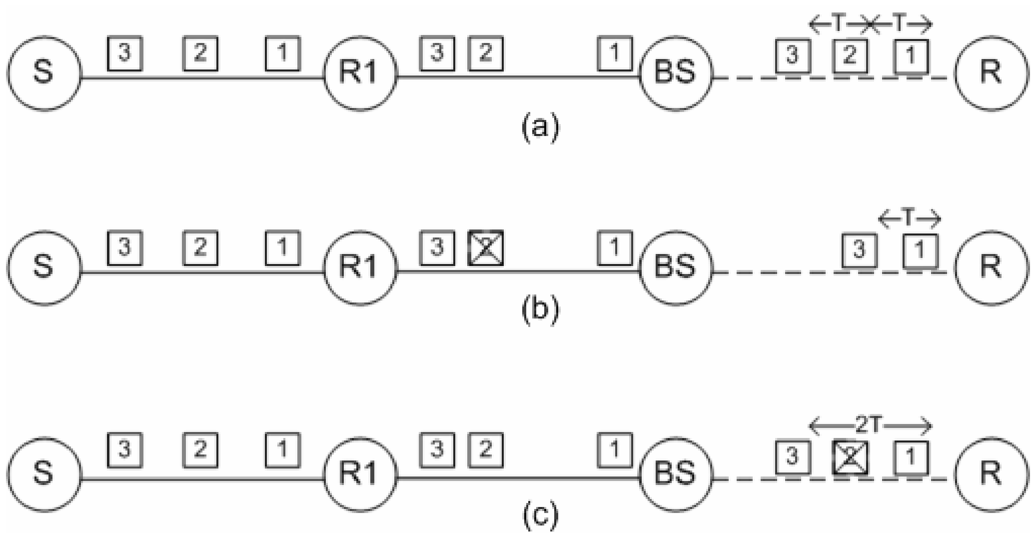
- Detects the type of loss based on the packet inter-arrival times at the receiver
- Assumes the following:
  - wireless link is the bottleneck
  - sender performs bulk data transfer
  - on the connection path, only the last link is wireless
- Given the assumptions, accuracy of detection is high

S. Biaz and N.H. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *Proc. ASSET'99*, 24-27 March 1999, pp. 10-17.

S. Cen, P.C. Cosman, and G.M. Voelker, "End-to-end differentiation of congestion and wireless losses," *Proc. of the SPIE - the International Society for Optical Engineering,* vol. 4673, pp. 1-15. 2001

Loss detection scheme

- Wireless link is the bottleneck: packets will queue at the base station



- Thus, for a single packet loss:
  If (T < inter-arrival time ≤ 2T) { wireless loss }
  Else { congestion loss }

TCP-SNACK

- An option of satellite communication protocol stack-transport protocol (SCPS-TP)
- Widely used for satellite links
- Negative: receiver informs sender about the segments not received
- Selective: can send information for multiple lost segments:
  - good for long delay networks
- When a sender receives SNACK:
  - aggressively retransmits packets indicated as lost, preventing unnecessary retransmission time-outs

Consultative Committee for Space Data Systems, *Space Communications Protocol Specification—Transport Protocol (SCPS-TP)*, Blue Book, issue 1, May 1999.

Selective-TCP: implementation

- Extensions made in:
  - tcp-newreno.cc
  - tcp-newreno.h
  - tcp-sink.cc
  - tcp.h
- Location: ns-allinone-2.27/ns-2.27/tcp

- Scenarios generated:
  - TCL script files

Pseudo-code of Selective-TCP algorithm at the receiver

```
if (out-of-order packet received) {
        // check type of loss
        if ( wireless loss) {
                if (snack_delay = 0)
                        send SNACK
                else
                        do nothing
        }
        else { // congestion loss
                1) set congestion_count = congestion_count + 1
                2) set congestion_info = current bandwidth  measured
                at the TCP receiver
                        if (congestion_count = k) {
                        1) send congestion_info to the TCP sender
                        2) reset congestion_count
                        }
                        else
                        send ACK  // as in the case of standard TCP sink
        }
 else // in-sequence packet received
        send ACK (same as standard TCP-sink)
```

Selective-TCP: module at receiver

- tcp-sink.cc:

```
void SnackSink::recv(Packet* pkt, Handler*)
{     ..................
  // code inserted by Rajashree
      prev_pkt_ts = present_pkt_ts;
      present_pkt_ts = Scheduler::instance().clock();
      tmin = present_pkt_ts - prev_pkt_ts;
      // T_min is the minimum packet inter-arrival time seen so far
      if (T_min > tmin)
              T_min = tmin;
   ................
}
```

Pseudo-code of Selective-TCP at the TCP sender

```
if (SNACK received) {
        1) retransmit lost packet(s) as indicated in SNACK
        2) reset retransmission timer
        }
else if (congestion_info ≠ 0) {
        //set size of congestion window as the bandwidth measured at receiver
        1) set cwnd_ = congestion_info/base_rtt
        //cwnd_ denotes congestion window size and
        //base_rtt is the initial round trip time
        2) reset congestion_info
        }
else //standard ACK received
        do as standard TCP NewReno sender
```

Selective-TCP: module at sender

- tcp-newreno.cc:

```
void NewRenoTcpAgent::recv(Packet *pkt, Handler*)
{
       ………….
       if (tcph->snack_option()) // Sending Snack
            processSnack(pkt);
       else if (tcph->congestion_info())  // Setting congestion window size
      {        ………
               seq_num = tcph->seqno();
               cwnd_= tcph->congestion_info()/base_rtt;
               output(t_seqno_++,0);
               Packet::free(pkt);
       }
        …………….
 }
```

Selective-TCP algorithm: description

- In case of wireless loss, the receiver sends SNACK
  - sender retransmits the lost packets immediately without waiting for retransmission timer to expire:
    - no duplicate ACKs sent
    - congestion control not invoked
  - SNACK is selective
    - helps in presence of packet reordering
  - result: increased bandwidth utilization and higher goodput

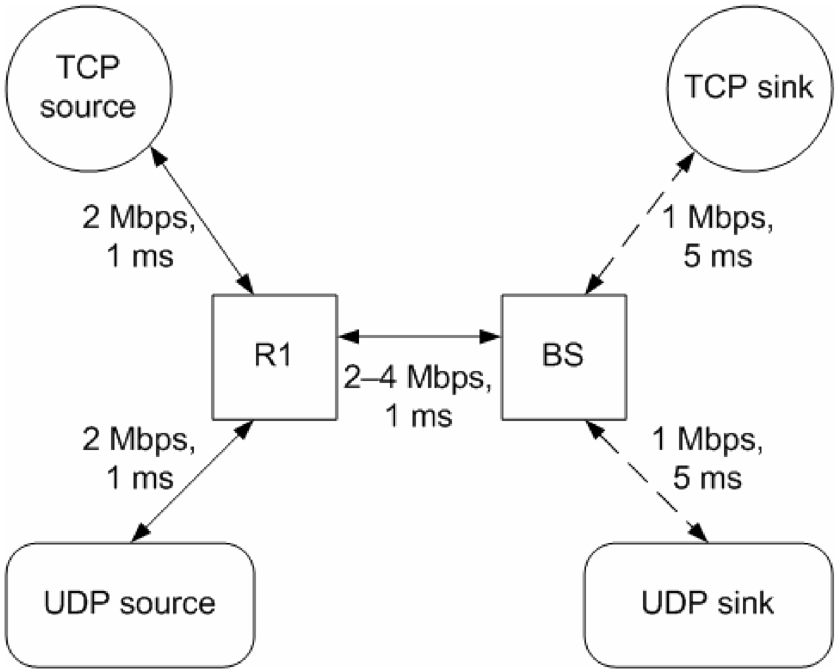
Selective-TCP algorithm: description

- In case of congestion loss:
  - sender's congestion window size is set according to the bandwidth measured at receiver
  - TCP's AIMD algorithm is prevented from setting congestion window lower than necessary
  - result: increased bandwidth utilization and higher goodput

Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

Simulation scenario: network topology



TCP source:

sending rate = 2 mbps

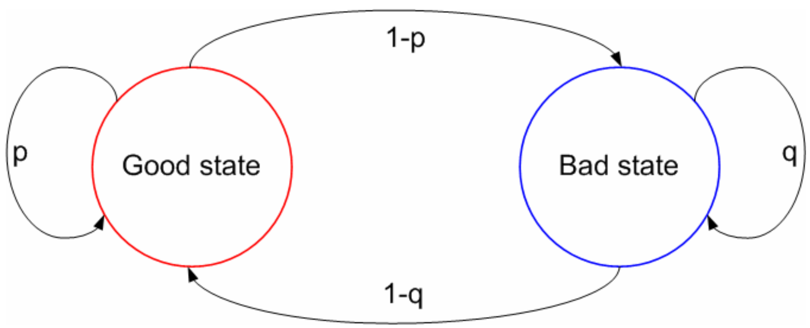
UDP source:

sending rate = 512 Kbps

- Wired link: bandwidth 2–4 Mb, propagation delay 1 ms
- Wireless link: bandwidth 1 Mb, propagation delay 5 ms

Simulation scenario: error model

- Two-state Markov model for modeling burst errors over wireless link



$$\pi = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$$

and

$$\varepsilon = \frac{1-p}{2-p-q}$$

Simulations:

good state: 0 packet loss

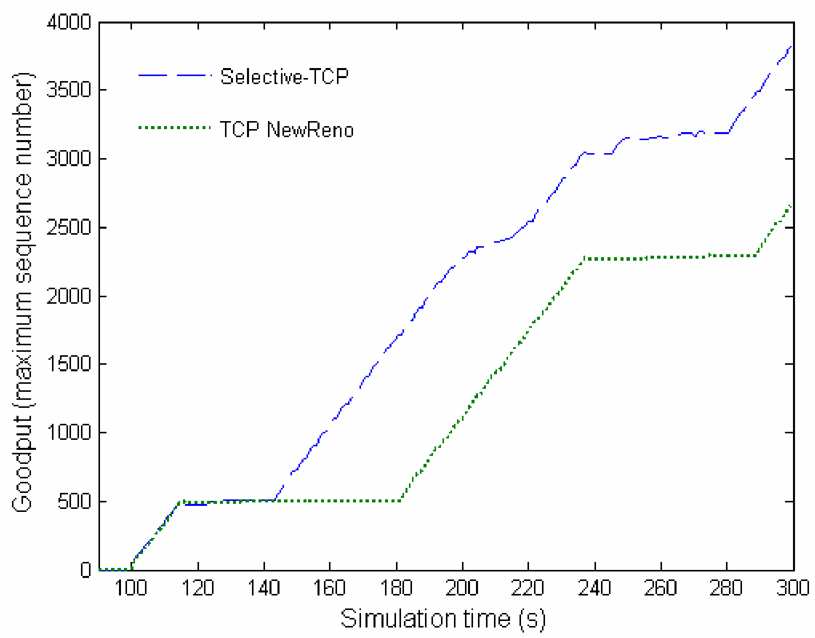
bad state: 1 packet loss

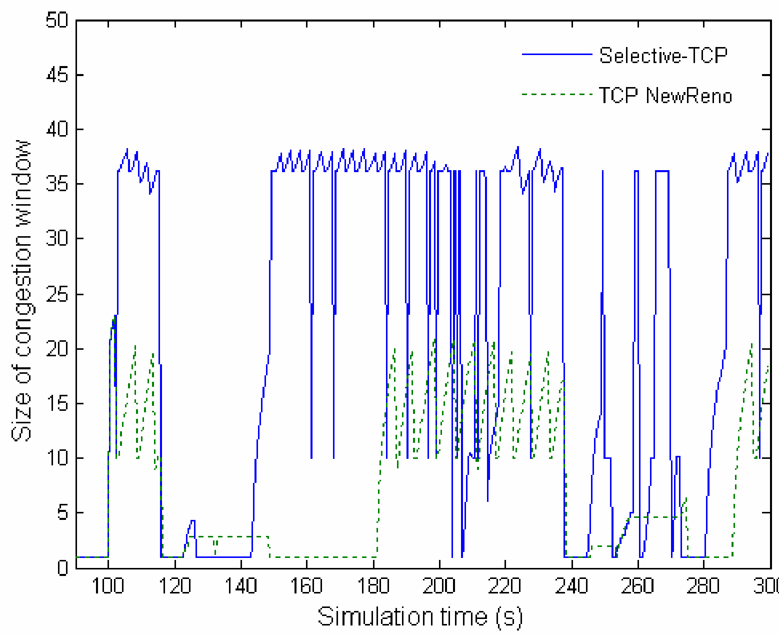
p = 0.9913 and q = 0.8509

A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-based channel model algorithm for wireless networks," *Wireless Networks,* vol. 9, no. 3, pp. 189–199, May 2003.

Simulation results:

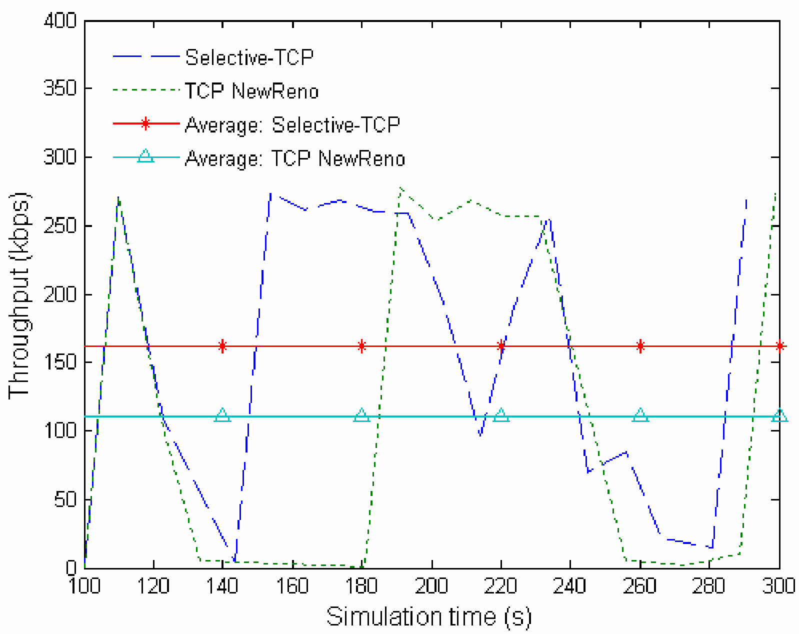
In the presence of congested link:

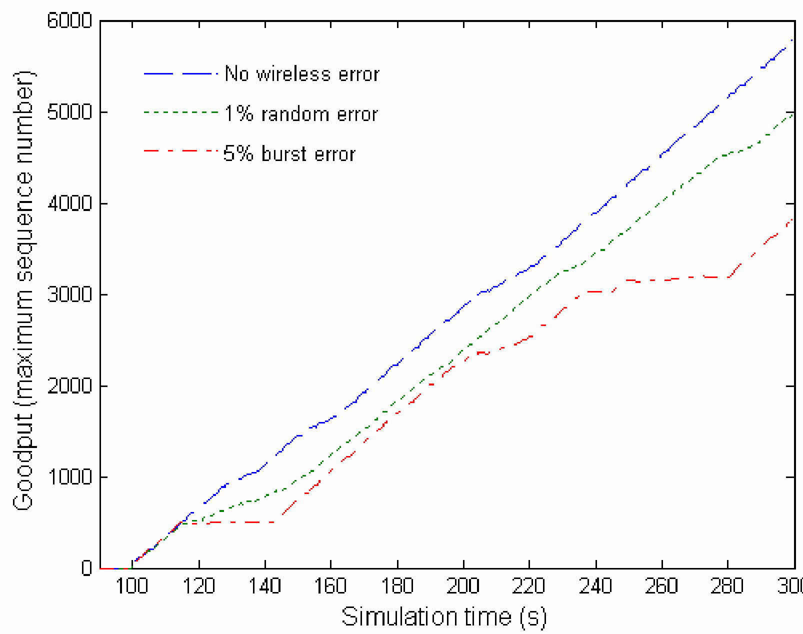




Simulation results:

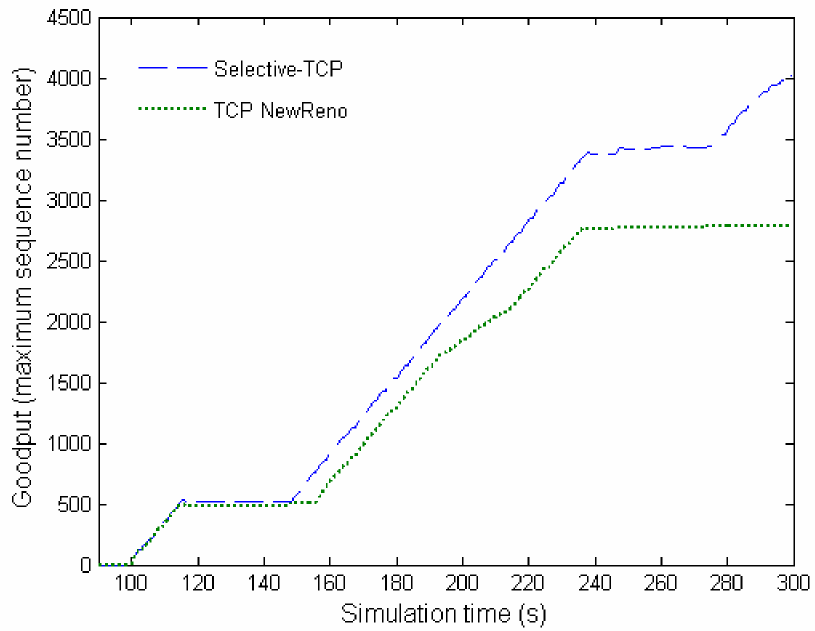
In the presence of congested link:

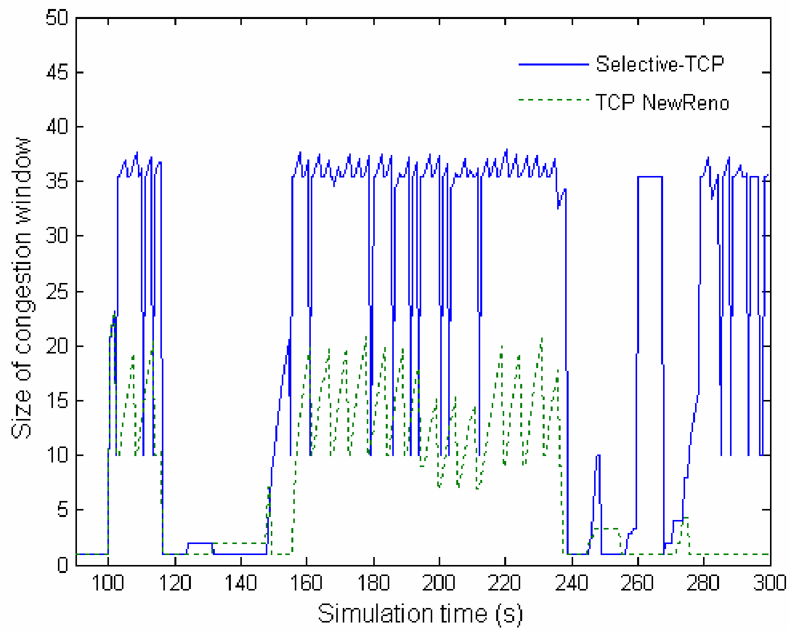




Simulation results:

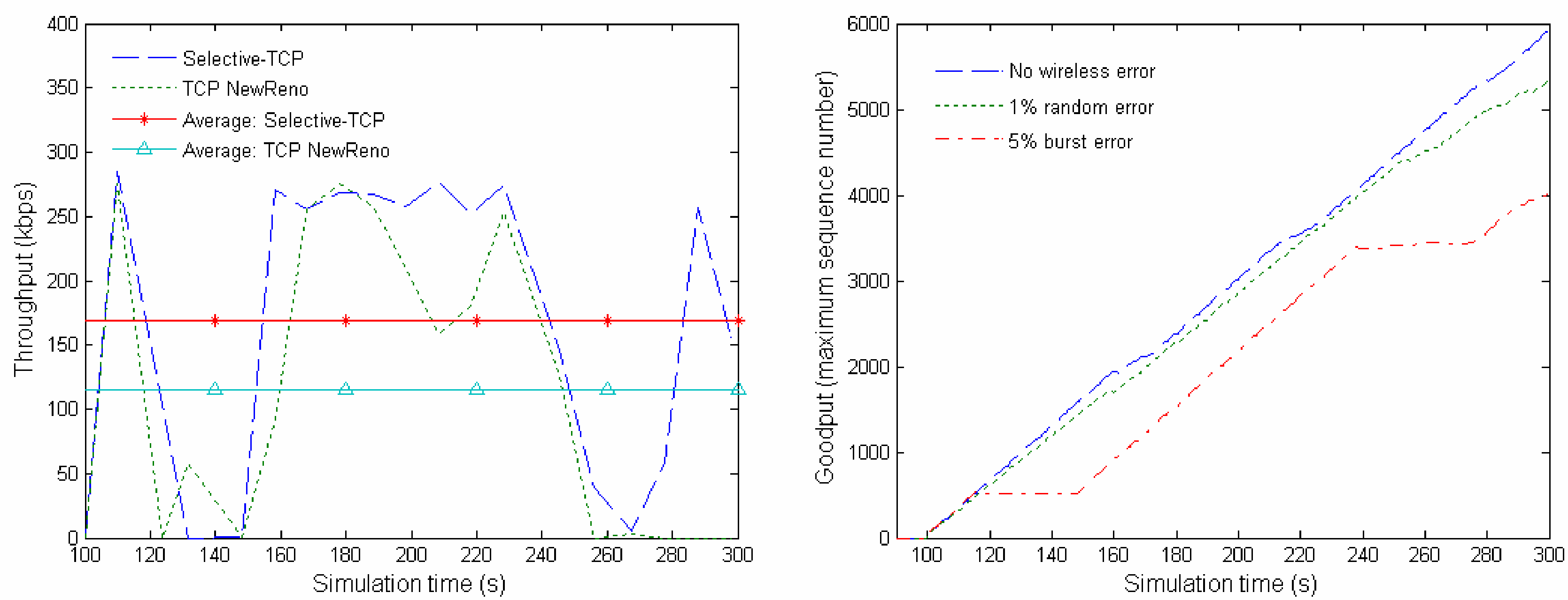
In the absence of congested link:





Simulation results:

In the absence of congested link:



Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

TCP packet control: overview

- A link layer based approach to improve TCP performance in wired/wireless networks:
  - modifications made in the intermediate node (base station): implemented in ns version 2.26
  - addresses two problems specific to wireless networks
    - delay variation (spurious fast retransmit)
    - sudden large delays (spurious fast timeout)
  - hides wireless losses from TCP sender (fixed host)
  - filtering done locally at base station
  - improves goodput up to 30% with 1% packet loss, compared with TCP-Reno

W. G. Zeng and Lj. Trajkovic, "TCP packet control for wireless networks," presented at *WiMob 2005*, Montreal, Canada, Aug. 2005.

TCP packet control: implementation

- Data and ACK filters:
  - ll-wz.cc
- in ns-allinone-2.26/ns-2.26/mac

Packet control algorithm: data filter

- Data filter: filters data segments, sent from fixed host

if (new or unacknowledged data segment)
    forward to receiver (mobile host)
else // acknowledged data segment
    drop the segment

Packet control algorithm: ACK filter

- ACK filter: filters ACKs, sent from mobile host

```
if (old ACK received)
   drop the ACK
else if  (new ACK received) {
   i) update last_received_ACK
   ii) reset number of DUP_ACKs to 0
   iii) forward the ACK to fixed host
   }
else   //duplicate ACK received {
   i) update number of DUP_ACKs
   ii) drop or forward the duplicate ACKs depending
   on user-defined DUPACK_threshold
   }
```

TCP-packet control: data filter

- void LLWz::myRecv(Packet *p, Handler *h, bool bDelayedPacket)

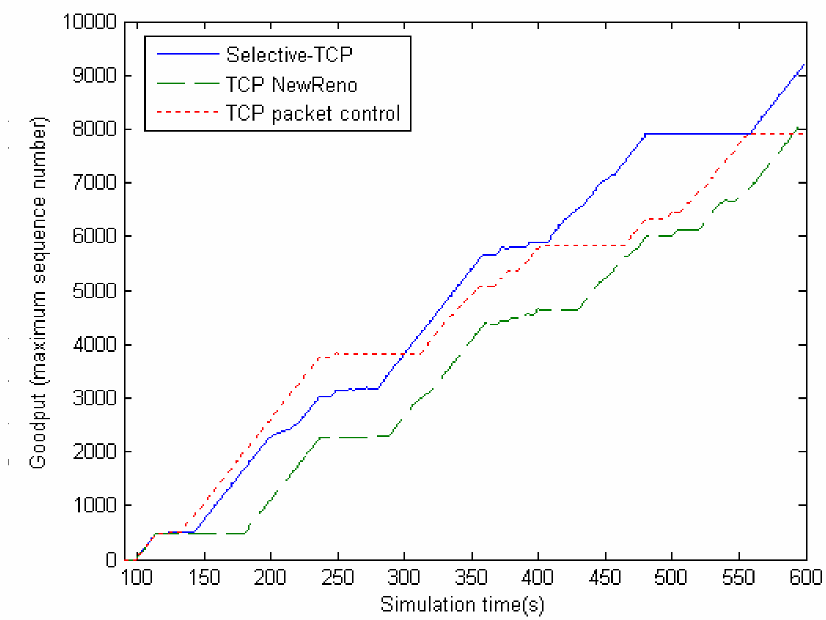
```
if(enableAckControl_ != 0)
        {
                //Data Filter
                if(tcph->seqno() <= m_iLastWirelessAck)
                {
                //we drop every second retransmitted packet.
                if(recordRetransPacket(tcph->seqno()) % 2 == 1)
                {
                        m_iNumOfWiredDataPacketDroped++;
                        Packet::free(p);
                        return;
                }
          ..........
          }
```

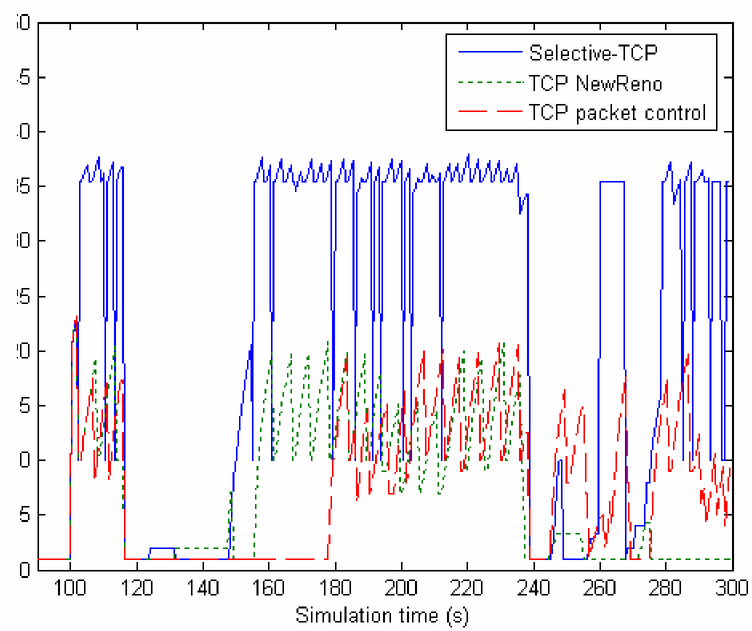
Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

Simulation results:

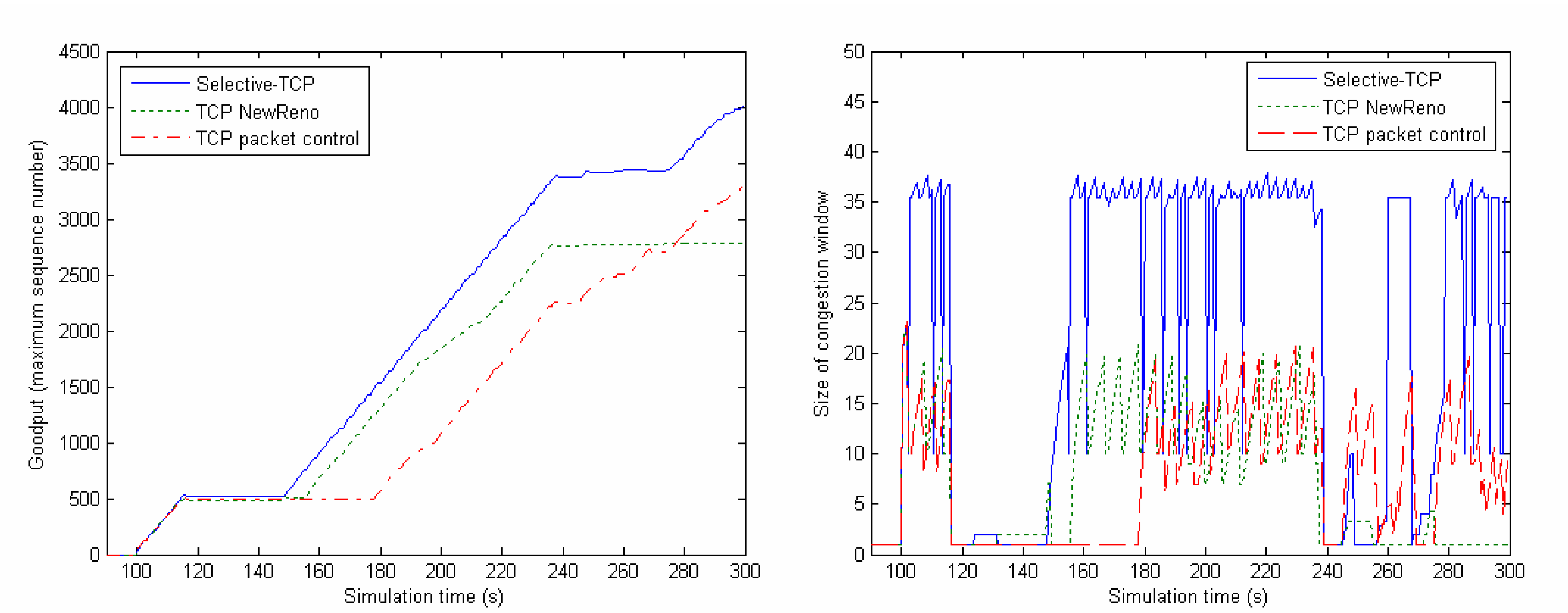
In the presence of congested link:





Simulation results:

In the absence of congested link:



Roadmap

- Motivation
- Background and related work
- Selective-TCP
  - overview
  - implementation
- Simulation results
- TCP packet control
- Performance evaluation: comparison of Selective-TCP and TCP packet control
- Conclusions

Conclusions

- Selective-TCP:
  - an end-to-end approach
  - distinguishes wireless losses from congestion losses
  - takes corrective action depending on the type of loss
  - improves goodput up to 45% in mixed wired/wireless networks in presence of 5% burst errors, compared with TCP NewReno

References

1. W. G. Zeng and Lj. Trajkovic, "TCP packet control for wireless networks," *IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob 2005)*, Montreal, Canada, Aug. 2005, pp. 196–203.
2. K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *Computer Communication Review*, vol. 27, no. 5, pp. 19–43, Oct. 1997.
3. A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," in *Proc. 15th Int. Conf. on Distributed Computing Systems*, Vancouver, Canada, May 1995, pp. 136–143.
4. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *Computer Communication Review*, vol. 26, no. 4, pp. 256–269, Aug. 1996.
5. S. Biaz and N. H. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in *Proc. IEEE Symposium on Application-Specific Systems and Software Engineering and Technology*, Richardson, TX, Mar. 1999, pp. 10–17.
6. Consultative Committee for Space Data Systems, *Space Communications Protocol Specification—Transport Protocol (SCPS-TP)*, Blue Book, issue 1, May 1999.
7. D. Anantharaman, "Performance analysis of snack in satellite networks through simulation," M.S. Thesis, Lamar University, Lamar, TX, 2004.

References

8. C. P. Fu and S. C. Liew, “TCP Veno: TCP enhancement for transmission over wireless access networks,” *IEEE J. on Select. Areas Commun.,* vol. 21, no. 2, pp. 216–228, Feb. 2003.
9. L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: new techniques for congestion detection and avoidance,” in *Proc. SIGCOMM*, London, U.K., Oct. 1994, pp. 24–35.
10. N. K. G. Samaraweera, “Non-congestion packet loss detection for TCP error recovery using wireless links,” *IEE Proc. Communications,* vol. 146, no. 4, pp. 222–230, Aug. 1999.
11. D. Barman and I. Matta, “Effectiveness of loss labeling in improving TCP performance in wired/wireless networks,” in *Proc. 10th IEEE Int. Conf. on Network Protocols*, Boston, MA, Nov. 2002, pp. 2–11.
12. C. Parsa and J. J. Garcia-Luna-Aceves, “Differentiating congestion vs. random loss: a method for improving TCP performance over wireless links,” in *Proc. IEEE Conf. on Wireless Communications and Networking*, Chicago, IL, Sept. 2000, vol. 1, pp. 90–93.
13. T. Kim, S. Lu, and V. Bharghavan, “Improving congestion control performance through loss differentiation,” in *Proc. Eighth International Conference on Computer Communications and Networks*, Boston, MA, Oct. 1999, pp. 412–418.

References

14. F. Sun, V. O. K. Li, and S. C. Liew, "Design of SNACK mechanism for wireless TCP with new snoop," in *Proc. IEEE Wireless Communications and Networking Conference*, Atlanta, GA, Mar. 2004, vol. 2, pp. 1051–1056.
15. K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE J. Select. Areas Commun.,* vol. 22, no. 4, pp. 747–756, May 2004.
16. V. Tsaoussidis and C. Zhang, "TCP-Real: receiver-oriented congestion control," *Computer Networks*, vol. 40, no. 4, pp. 477–497, Nov. 2002.
17. S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Trans. Networking,* vol. 11, no. 5, pp. 703–717, Oct. 2003.
18. ns-2 [Online]. Available: http://www.isi.edu/nsnam/ns.
19. A. Gurtov and S. Floyd, "Modeling wireless links for transport protocols," *Computer Communication Review,* vol. 34, no. 2, pp. 85–96, Apr. 2004.
20. J. McDougall and S. Miller, "Sensitivity of wireless network simulations to a two-state Markov model channel approximation," in *Proc. GLOBECOM*, San Francisco, CA, Dec. 2003, pp. 697–701.
21. A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A Markov-based channel model algorithm for wireless networks," *Wireless Networks,* vol. 9, no. 3, pp. 189–199, May 2003.