Simulation-based Analysis of TCP Behavior over Hybrid Wireless & Wired Networks

Qiang Ni, Thierry Turletti, Wei Fu
Planete Group
INRIA Sophia Antipolis

Outline

- Motivation
- TCP Congestion Control
- RTT-based Discrimination Scheme
- Simulation Analysis
- Conclusions

Motivation

- Problems:
  - TCP originally designed for wired and slow IP networks: hybrid network (wired/wireless) affects TCP performance
  - TCP reduces its congestion window in response to each loss and unnecessarily degrades the throughput
- Here, we focus on:
  - How the main different TCP versions behave under hybrid IP networks
  - How well RTT-based discrimination scheme work

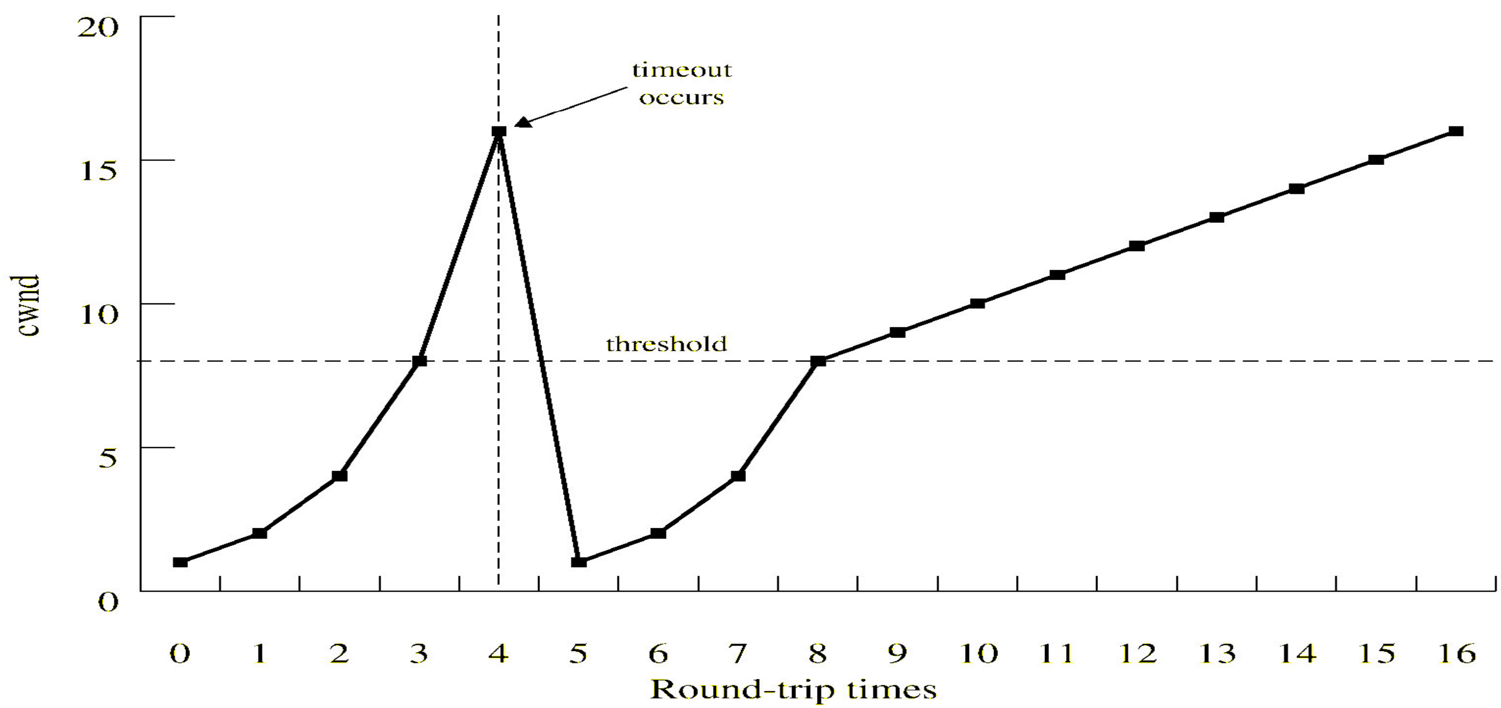
Related works

- Several comparisons of TCP versions performance available:
  K.Fall and S.Floyd (ACM CCR 96)
  H.Balakrishnan, etc. (IEEE/ACM Transaction on Networking 97)
  V.Tsaoussidis and I.Matta (J. Wireless communication & mobile computing 02)

- Schemes improving performance of TCP have been designed:
  Pure end-to-end, (C.Zhang WWIC02), (S.Chen 02), (S. Biaz 99/98), (Y.Tobe 00), (Samaraweera 99), (Vegas 94)
  Split-connection, (Bakre 95/97), (Yavatkar 94), (Hass 97 ICC), (Wang 98 Infocom)
  Explicit notification scheme, (S. Floyd ECN 94), (H.Balakrishnan 98), (Bakshi 97)
  Link-layer scheme, (H.Balakrishnan Snoop 95), (K.Ratnam WTCP 98), (Vaidya Delayed Dupacks 99)

TCP Congestion Control(1)

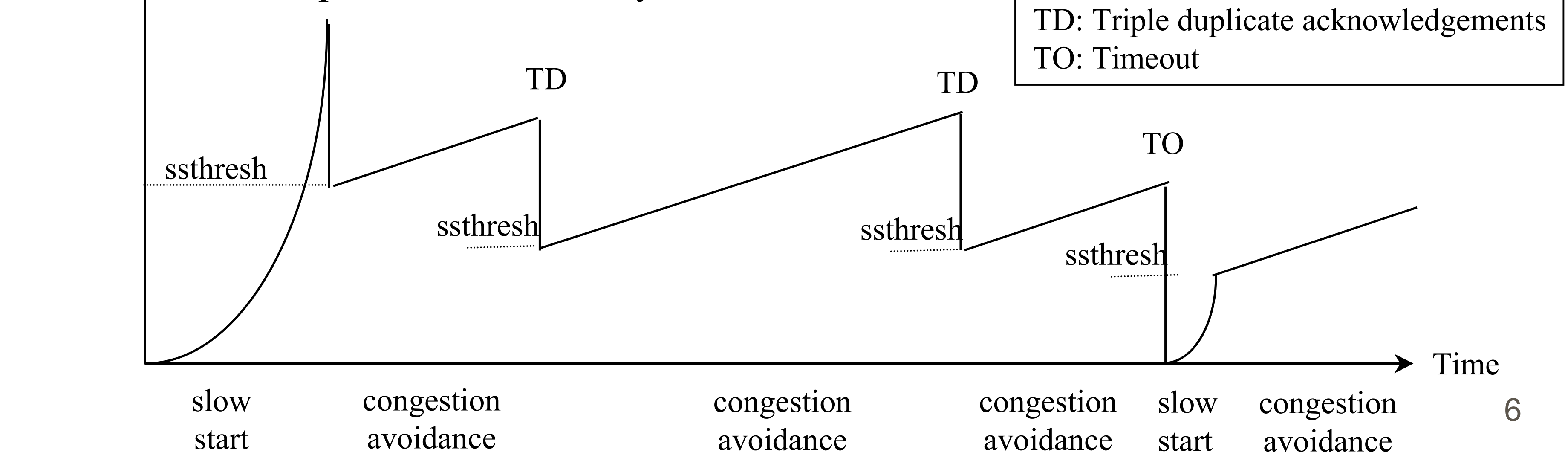
- TCP Tahoe
  - Slow start
  - Additive increase, multiplicative decrease
  - Fast retransmit



TCP Congestion Control(2)

- TCP Reno
  - Skip the slow start
  - Fast recovery
  - Directly reduce to half the last successful congestion window

Problem: It is not optimized for multiple packet drops from a single window, and this could negatively impact performance. Especially in a hybrid wired and wireless environment, frequent two or more packet losses in the same window will degrade Reno's performance heavily



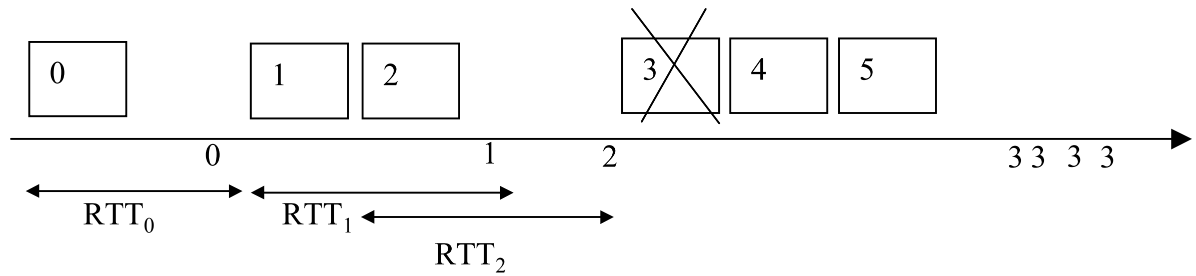
TCP Congestion Control(3)

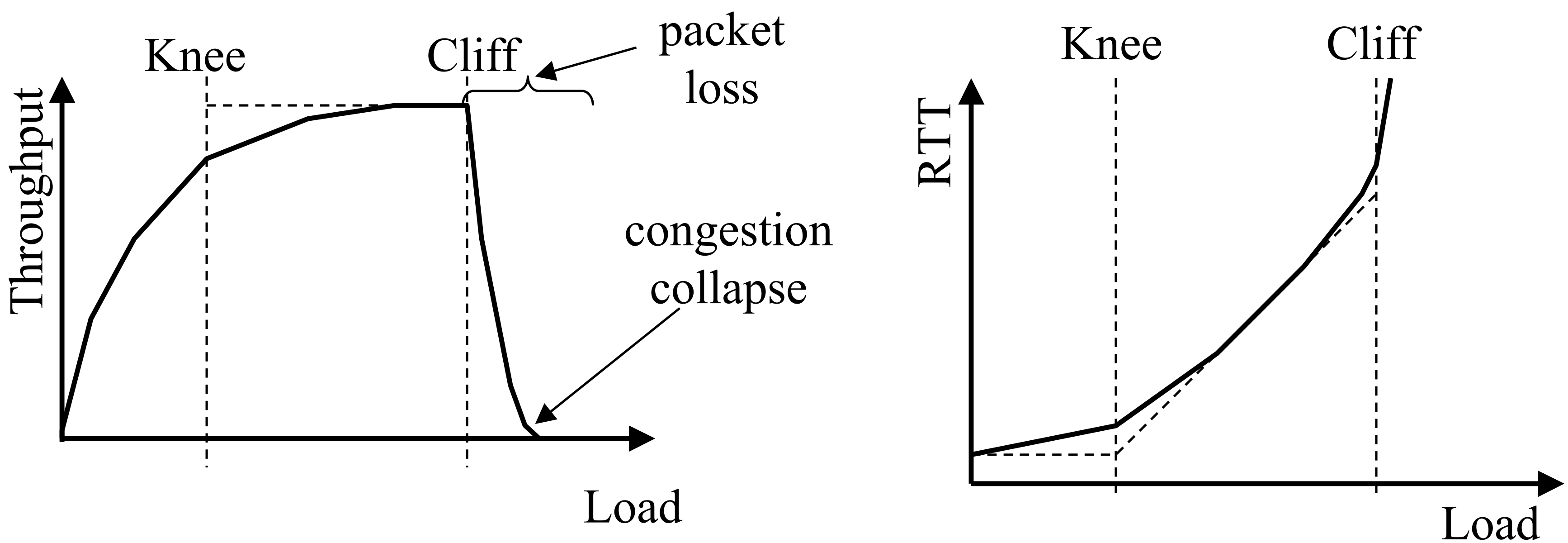
- TCP NewReno
  - Partial ACKs will not take NewReno out of fast recovery
  - NewReno eliminates Reno's wait for a retransmit timer when multiple packets are lost from a window
  - NewReno remains in fast recovery period until all of the data outstanding when fast recovery was initiated has been acknowledged
  - When multiple packets are lost from a single window of data, NewReno can recover without a retransmission timeout, retransmitting one lost packet per round-trip time until all of the lost packets from that window have been retransmitted

TCP Congestion Control(4)

- TCP SACK
  - Based on selective acknowledgement and selective retransmission
  - The sender has a better idea of which packets have been successfully delivered
  - Sender can avoid unnecessary delays and retransmissions, resulting in improved throughput, especially in the behavior when multiple packets are dropped from one window of data (high loss, bursty wireless channel)

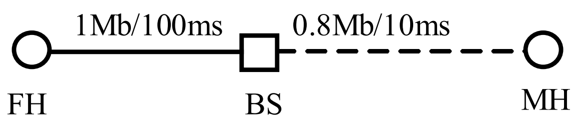
RTT-based Discrimination Scheme



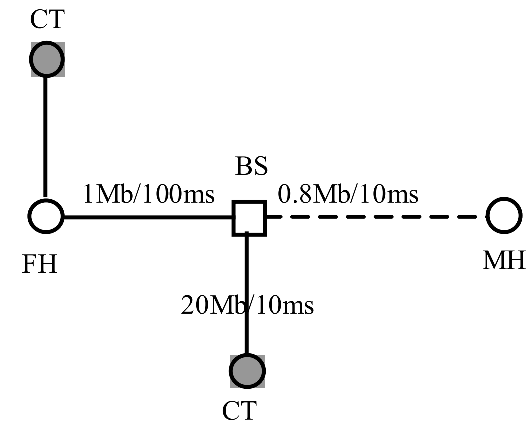


Throughput and RTT vs network load

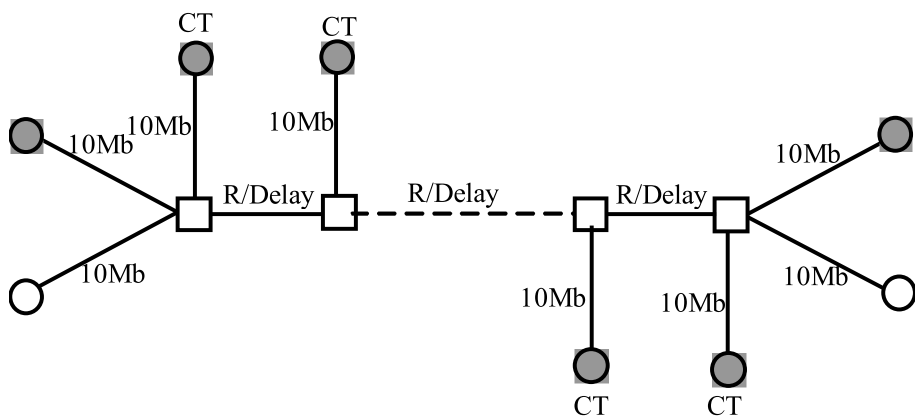
Simulation Analysis(1)



(a) Wireless lasthop



(b) Wireless lasthop + one wired bottleneck



(c) Wireless backbone+multiple wired bottleneck

Simulation Topologies

Simulation Analysis(2)

Simulation parameters

Ns version	ns-2.1b8a
Link type	Duplex-link
Router type	Droptail
Queue size	20 pkts
Small queue size	3-5 pkts
Packet size	1000 bytes
Application type	FTP
FTP size	2Mbytes, 100Kbytes
Pareto Background	On 1s, Off 2s
Loss model	Uniform, Gilbert
Fading effect	No
Simulation time	400 second

Simulation Analysis(3)

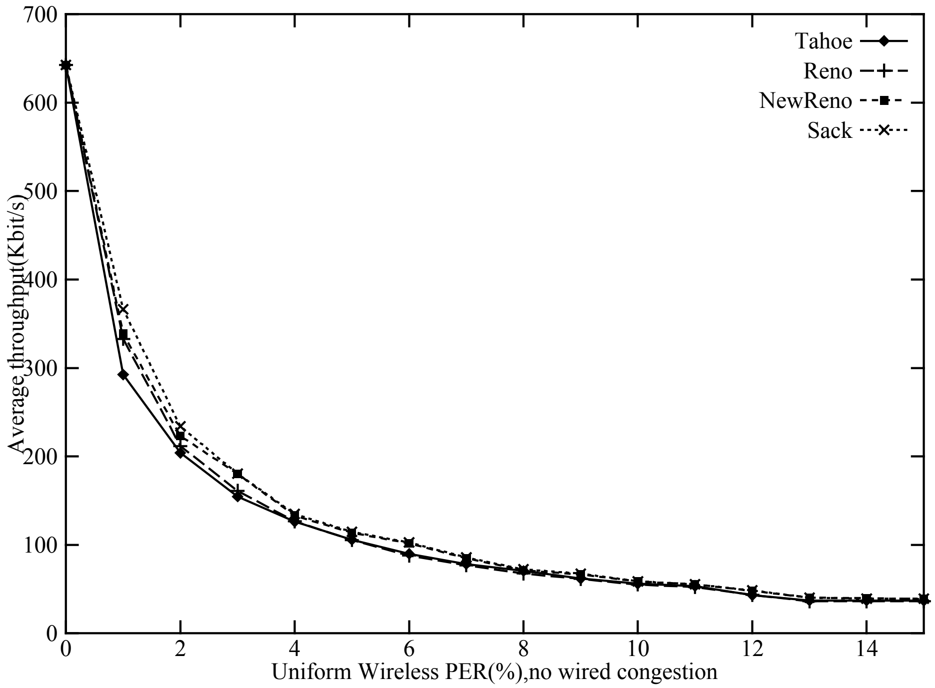
- For each scenario, we choose different random seeds and perform 5 sets of experiments. In each set, PER is randomly generated based on uniform error model or two-state Gilbert model. Fading effect is not considered. Throughput results are averaged over 5 simulations
- To show the effect of wireless backbone buffer length, we use 20pkts as long buffer size and 3-5pkts as small buffer size
- Several Pareto distributed ON/OFF sources are used for short TCP background traffics. During the On period, each source sends at 300Kbps, 160Kbps and 200Kbps. The shape parameter of Pareto distribution is 1.5 and the number of ON/OFF sources is 5

Simulation Analysis(4)

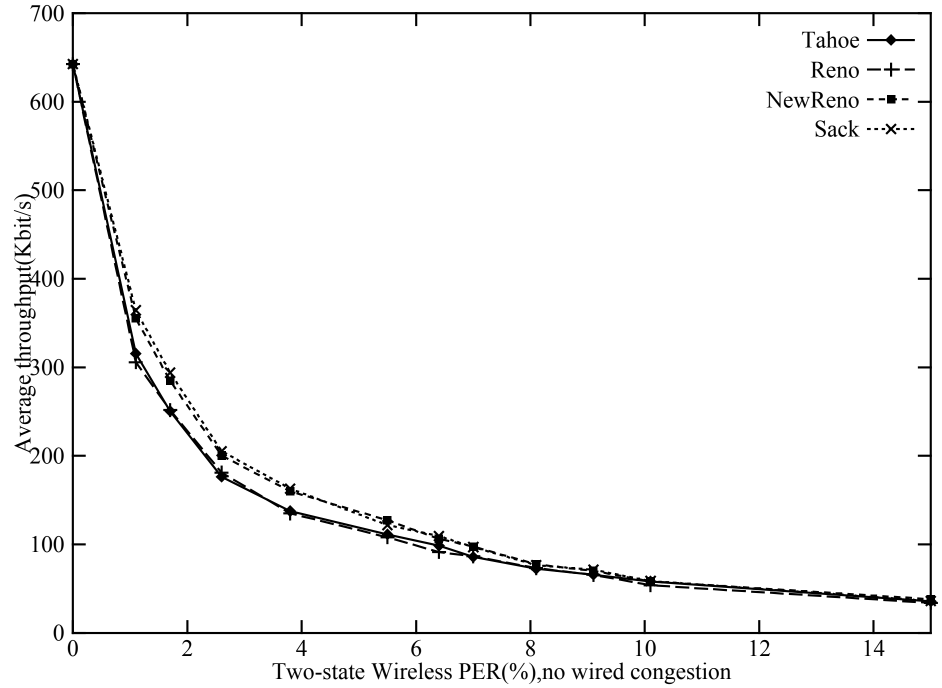
p	q	Loss rate(%)
0.007	0.693	1.1
0.011	0.665	1.7
0.015	0.735	2.6
0.024	0.731	3.8
0.035	0.665	5.5
0.045	0.705	6.4
0.055	0.728	7.0
0.070	0.810	8.1
0.075	0.758	9.1
0.078	0.692	10.1
0.114	0.653	15.0
0.121	0.673	15.3

Two-state Gilbert wireless channel loss model

Simulation Analysis(5)



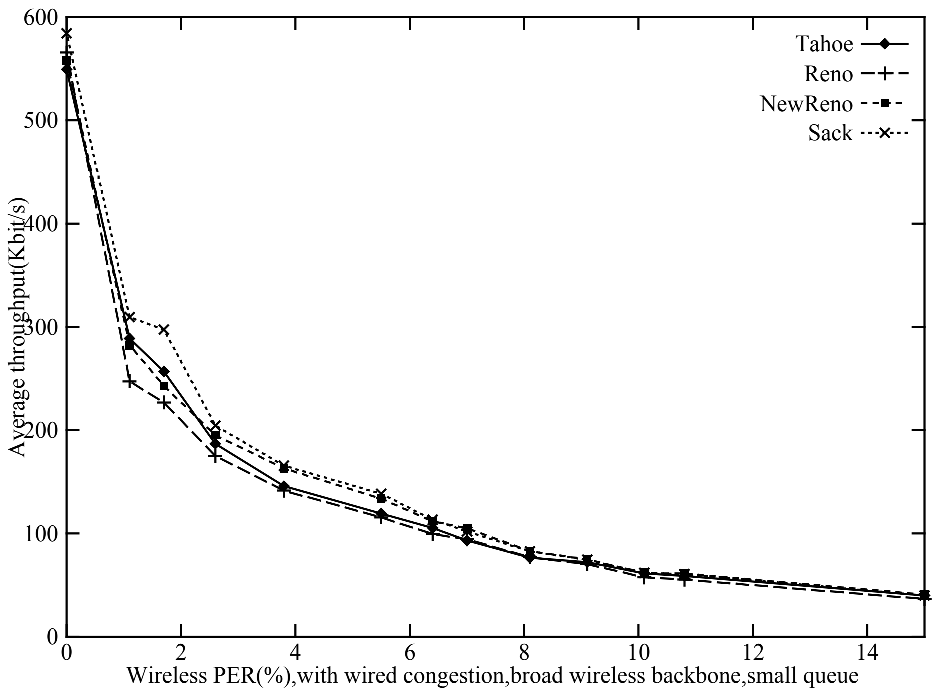
Uniform loss model



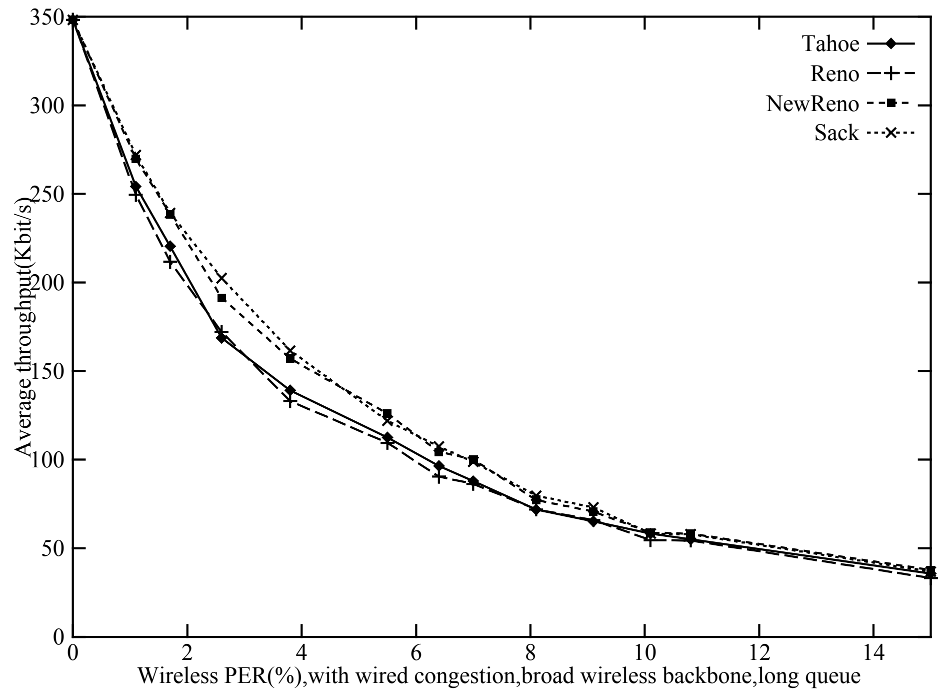
Two-state loss model

- **SACK and NewReno are slightly better than Tahoe and Reno**
- **Especially better when applying a two-state Gilbert channel loss model**

Simulation Analysis(6)



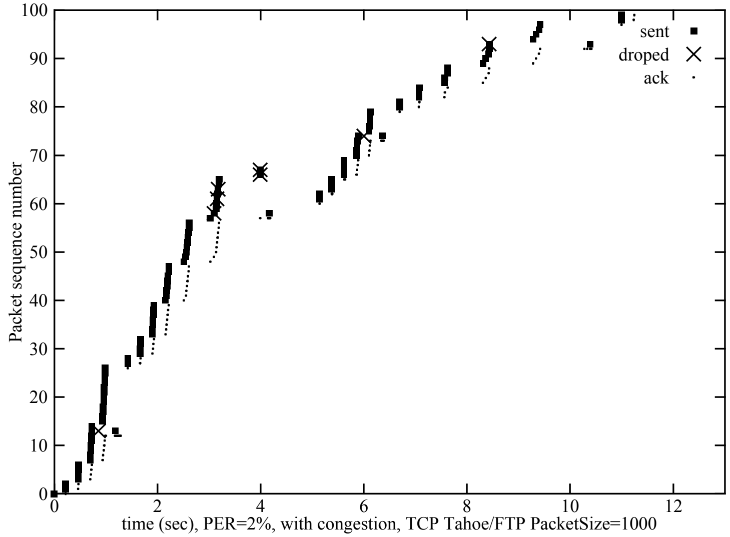
Wireless backbone (small queue)

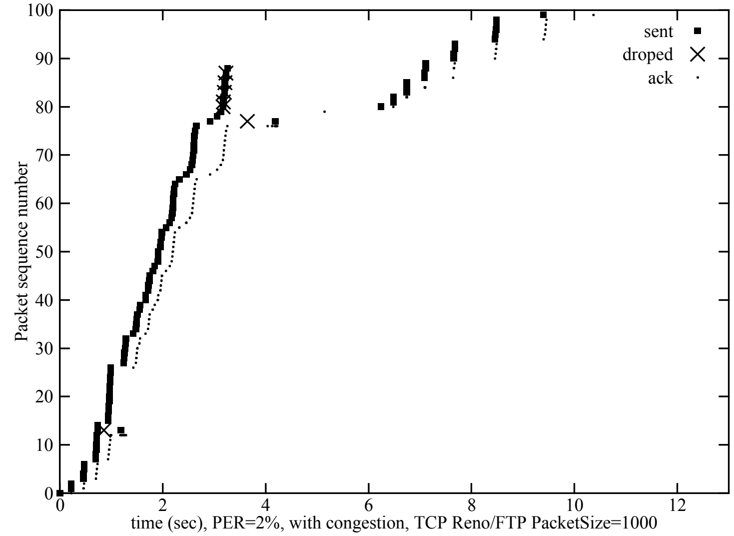


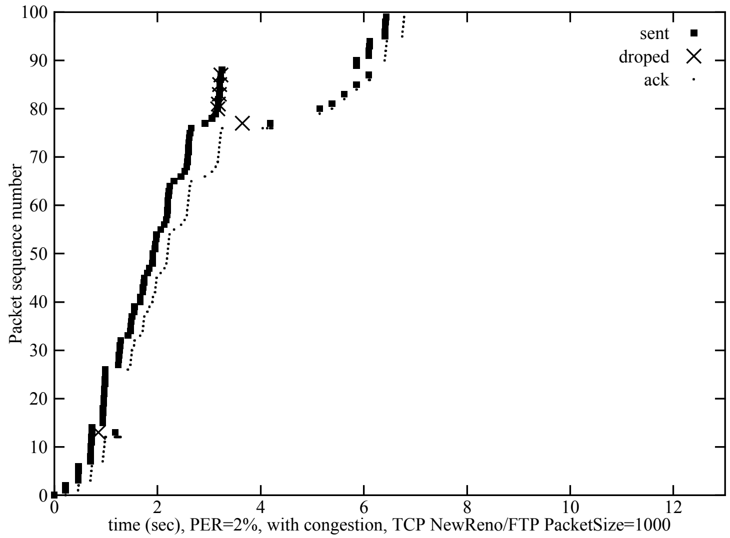
Wireless backbone (long queue)

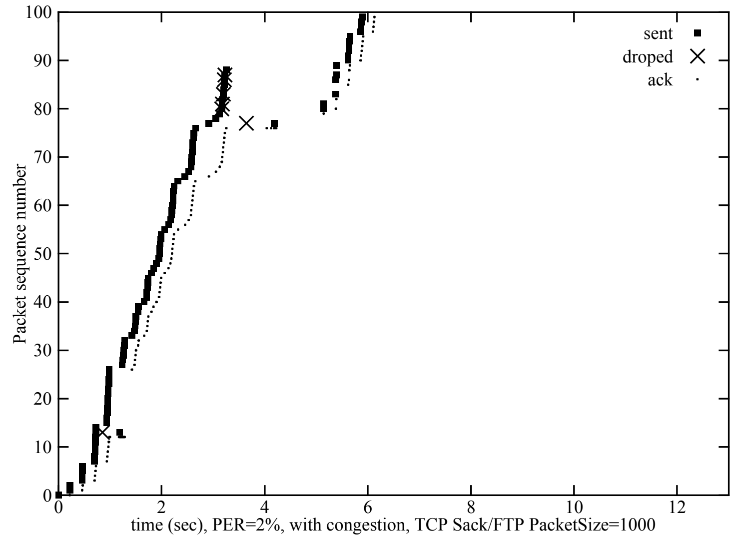
- **SACK still performs best in these two cases**
- **NewReno degrades the performance in small queue case**
- **Reno performs worst in two cases**

Simulation Analysis(7)

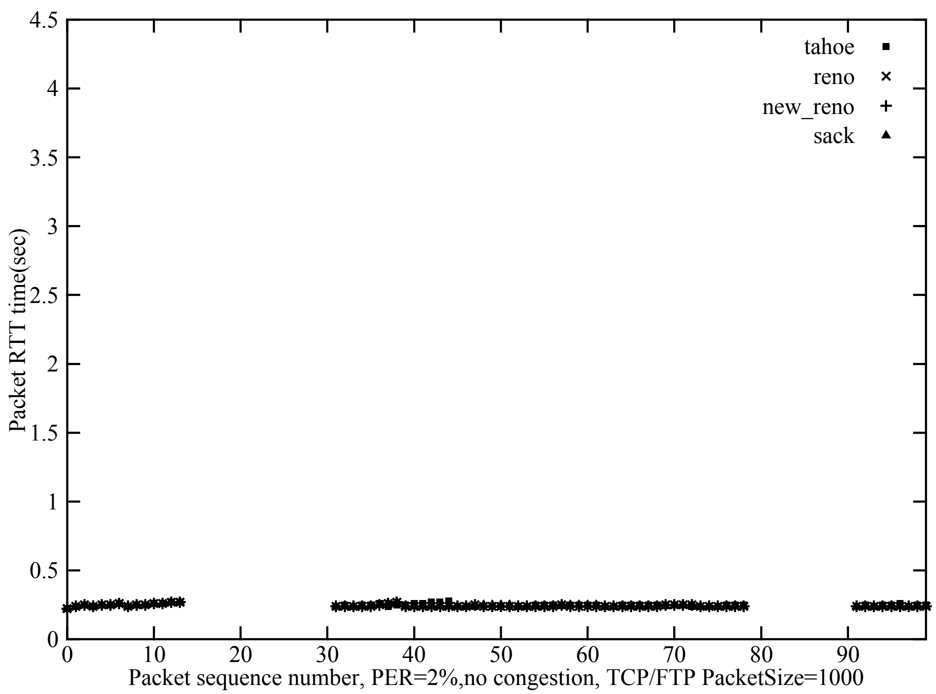


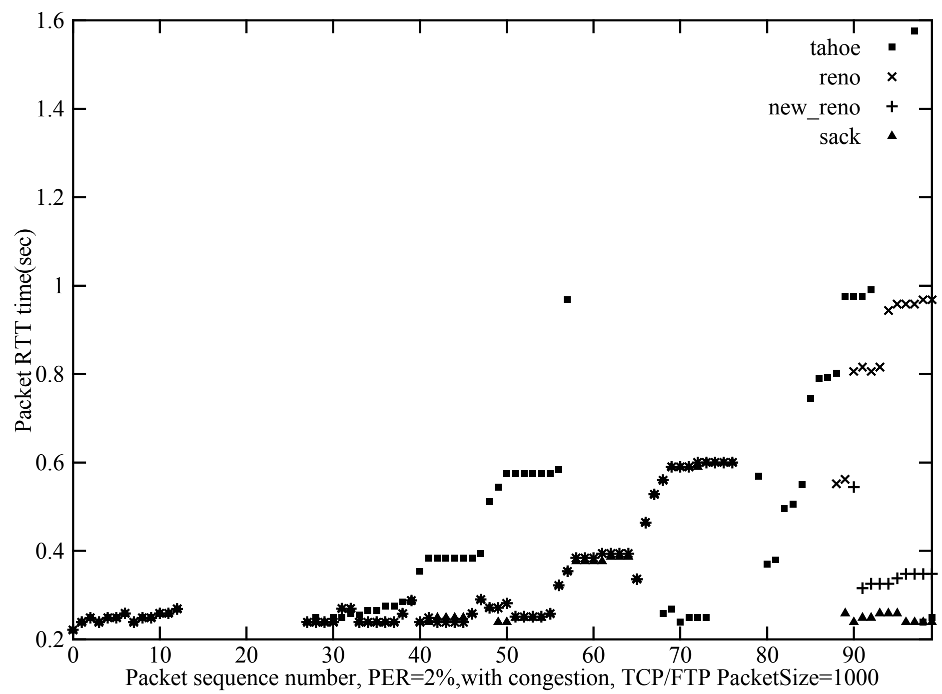






Simulation Analysis(8)



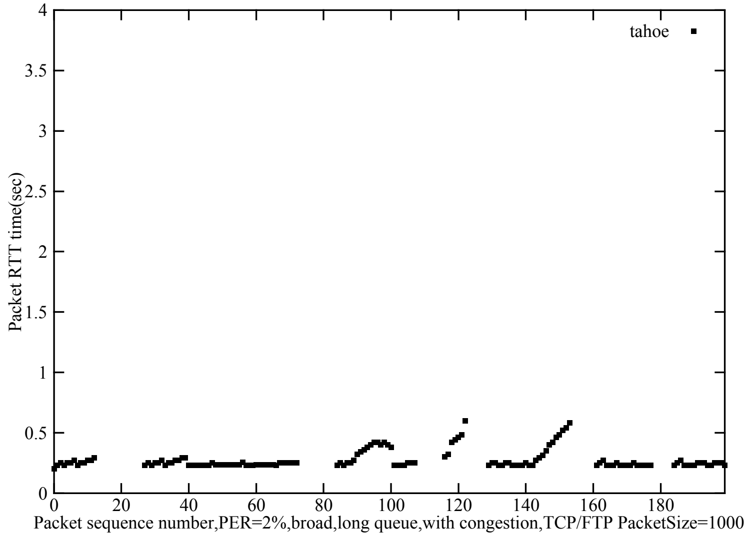


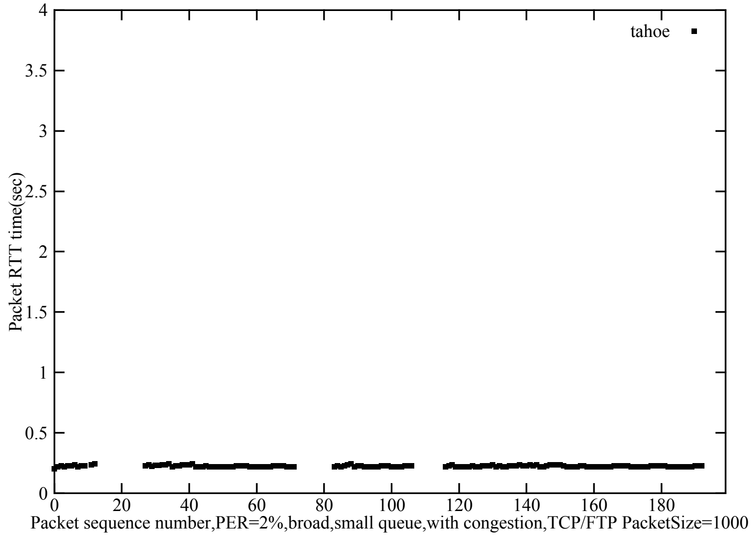
Only wireless loss, no congestion

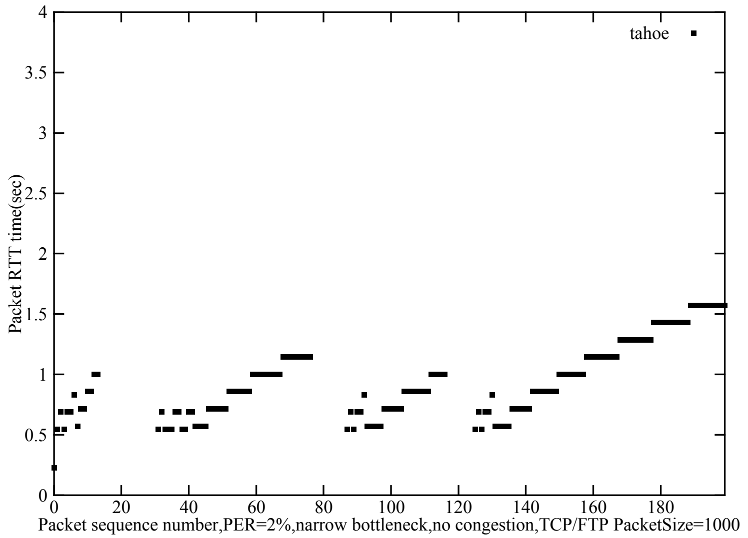
Wireless loss + wired congestion

RTT variation can distinguish wireless loss from congestion loss in such case

Simulation Analysis(9)







RTT variation can not distinguish wireless loss from congestion loss in these cases

Conclusions

Results

☞ Since fading model is not used, the average throughput of SACK and NewReno are only slightly higher than those of Tahoe and Reno, especially better when two-state Gilbert wireless channel loss model is used.

☞ Simple RTT variation schemes can not always predict congestion from wireless link error well. Moreover they do not work well under all kinds of simulation topologies and background load environments

Future works

☞ Fading channel model and more statistical analysis will be performed

☞ Design a robust end-to-end discrimination scheme

☞ Selective reject ARQ and FEC scheme at wireless link layer

Q&A?

Thanks!

Web: http://www-sop.inria.fr/planete/qni/

E-mail: Qiang.Ni@sophia.inria.fr