SAIMD: A Novel False Congestion Detection Scheme in TCP over OBS Networks

Basem Shihada and Pin-Han Ho

University of Waterloo, Canada

4rd Workshop on Optimization of Optical Networks (OON) 2007, May 2-3 2007

Outline

- Introduction
- TCP over OBS
- Statistical AIMD
  - RTT Statistics
  - Autocorrelation
  - Congestion control using confidence
- Simulation and analytical result comparison
- Verification of the proposed scheme
- Conclusion

Introduction

- TCP
  - Many Internet applications depend on TCP for reliable data transfer
  - TCP congestion control (AIMD)
  - TCP types are:
    - Loss-based TCP (e.g. Reno, Sack)
    - delay-based TCP (e.g. Vegas)
    - Mix of dropping and delay, e.g. FAST TCP
    - Explicit notification based, e.g. ELN, ECN
    - Rate based, e.g. TCP Westwood, TCP Real

OBS

- Burst contention occurs even at low traffic loads
  - Bufferless
  - One-way resource reservation signaling
- Contention resolution schemes
  - Burst deflection
  - Burst retransmission
  - Reduce burst loss probability

=>Introduce additional delay

TCP Congestion Control

- AIMD window-based
  - Additively increase sending rate (1 segment per *RTT*) to probe the available bandwidth.
  - Multiplicatively halving the *cwnd* at the occurrence of segment loss.
- Packet dropping is taken as an important indication for network congestion
- Different transmission conditions cause severe throughput degradation
  - Segment losses could be due to reasons other than congestion (wireless, OBS, satellite, etc)
  - False congestion is a serious problem

False Congestion Detection in TCP over OBS

- When a burst containing packets from different TCP sources is dropped, multiple TCP sources will throttle back their transmission
- When a burst containing multiple segments from a single TCP flow is dropped, it can fatally affect that TCP flow transmission
- Multiple simultaneous burst drops in a congested network can cause a network-wide loss in throughput (global synchronization problem)

  - May address a profound malicious impact on the long-lasting high-bandwidth TCP flows

  - Need to develop an effective approach for achieving false congestion detection

Previous Methodology

- Link-Layer solution

  - Mechanisms undergoing in the OBS domain

  - May not be sufficient; not adaptive to the TCP dynamics
- Congestion Detection <u>with</u> Explicit notification

  - Can effectively solve the false congestion problem

  - May cause significantly more signalling efforts and nodal processing
- Congestion Detection <u>without</u> Explicit notification

  - TCP senders estimate/evaluate the OBS congestion status

Statistical AIMD (SAIMD)

- Introducing a new mechanism at TCP senders for false congestion detection without any explicit notification
- Adopts the framework of Generalized AIMD
- A histogram curve is derived by the statistics of the long-term measured RTT

  - The long-term RTT statistics represent the overall effect on the TCP sender due to network topology, routing strategy, and long-term traffic distribution

  - In the modeling, we assume the spectrum to follow a Normal distribution

Statistical AIMD (SAIMD)

- Once there is a TCP segment drop

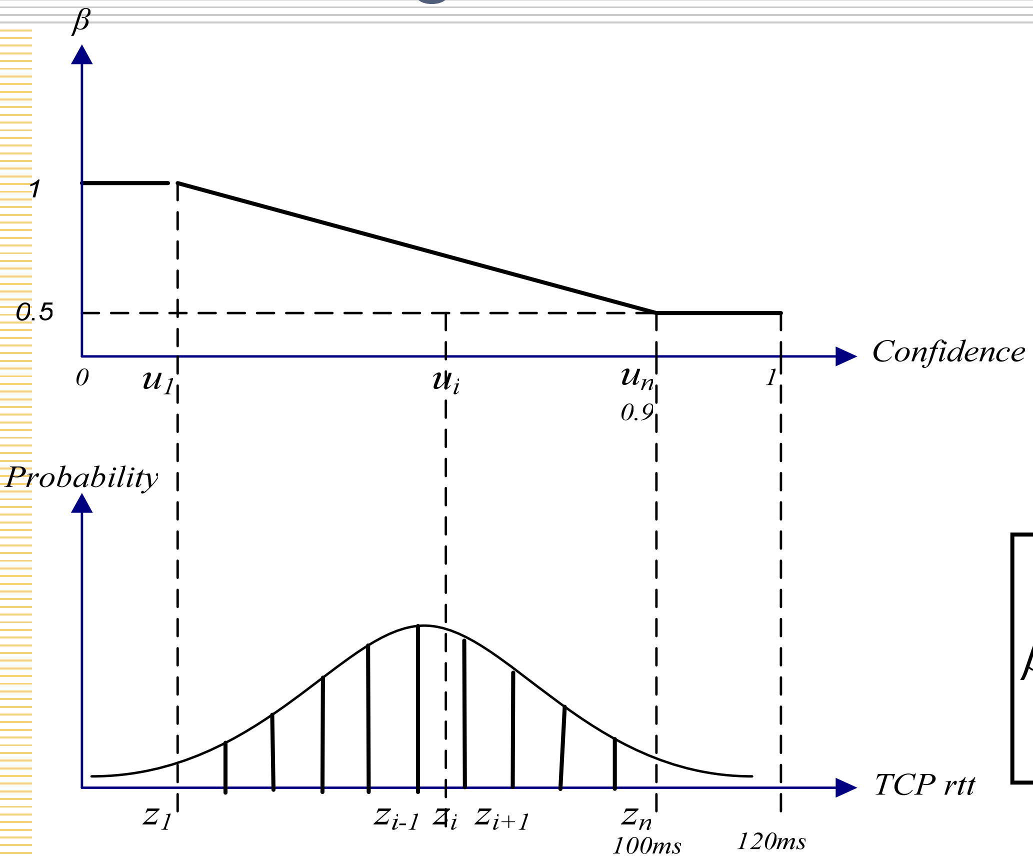
  - Derive avg_RTT_N as the short-term average RTT, and position it in the spectrum of RTT obtained from the long-term statistics

  - Obtain the confidence that the current network is in congestion

  - Use the confidence to come up with a beta value for cwnd adjustment corresponding to the current segment drop

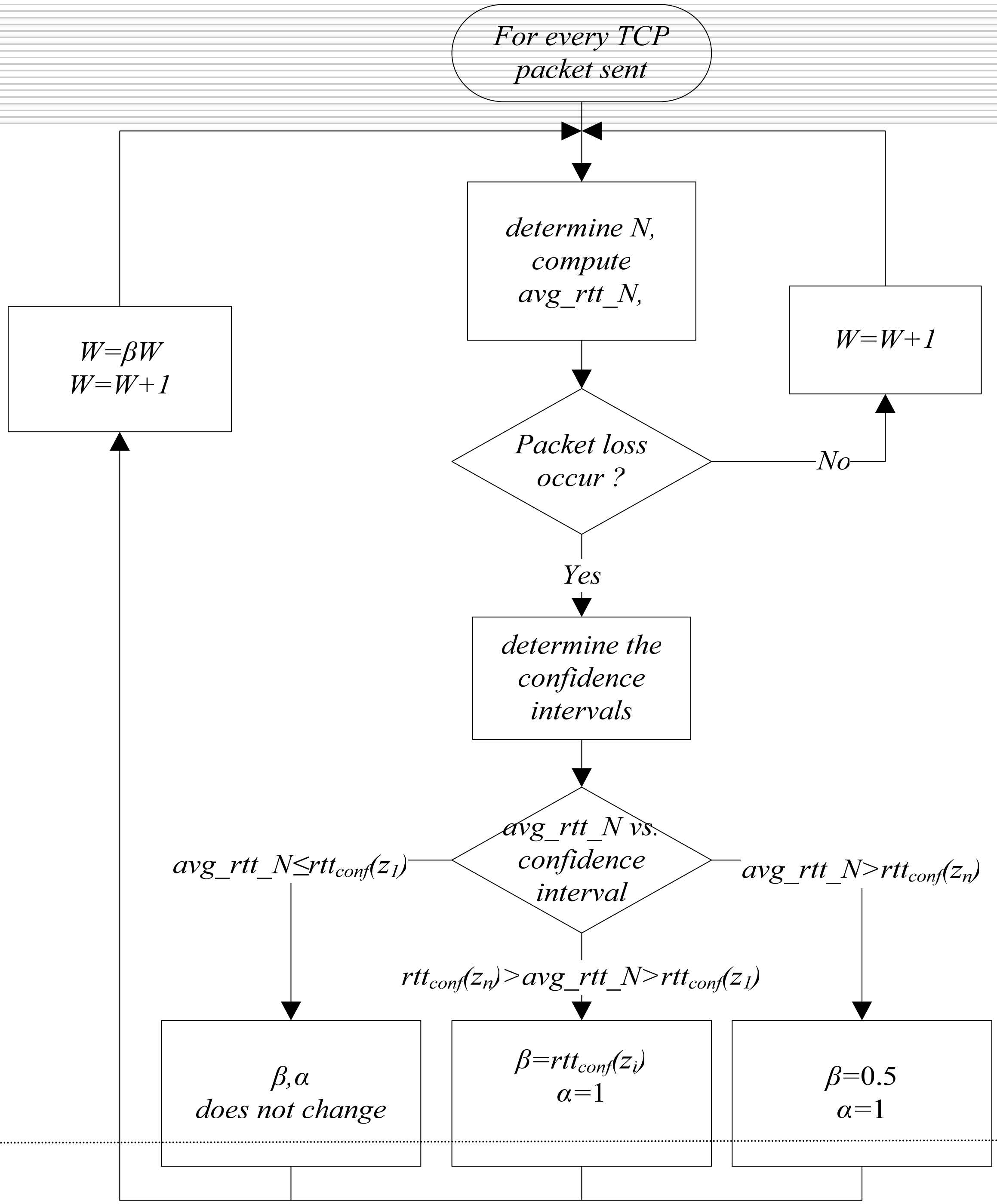
- If the short-term RTT is similar to or even less than the long-term RTT, the segment loss event is more likely caused by random contention => *cwnd* is slightly cut

SAIMD Congestion Control using Confidence



$$\beta = \begin{cases} 0.5 & avg\_rtt\_N > rtt_{conf}(u_n) \\ f(u_i) & rtt_{conf}(u_n) > avg\_rtt\_N > rtt_{conf}(u_1) \\ 1 & avg\_rtt\_N \le rtt_{conf}(u_1) \end{cases}$$

- Before z_1, low confidence of congestion, $\beta = 1$
- After z_n, high confidence of congestion, $\beta = 0.5$
- $z_1 \le z_i \le z_n$, potential for congestion, $\beta = f(u_i) = 1 - \frac{u_i - u_1}{2(u_n - u_1)}$


For every TCP packet sent
determine N, compute avg_rtt_N,
Packet loss occur ?
No
W=W+1
Yes
W=βW
W=W+1
determine the confidence intervals
avg_rtt_N vs. confidence interval
avg_rtt_N≤rtt_conf(z_1)
avg_rtt_N>rtt_conf(z_n)
rtt_conf(z_n)>avg_rtt_N>rtt_conf(z_1)
β,α does not change
β=rtt_conf(z_i)
α=1
β=0.5
α=1

Performance Modeling

Goal: to understand the behavior of SAIMD over OBS networks

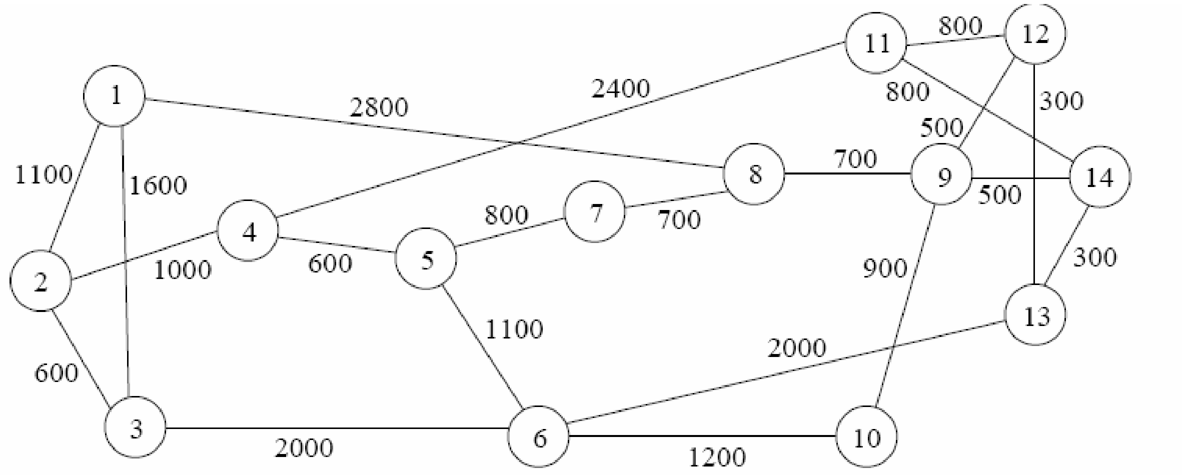
Assumption:

- TCP packets in a single *cwnd* are assembled to a single burst (i.e., TCP medium to fast flow)
- Burst contention probability in OBS networks is given.

Obtained:

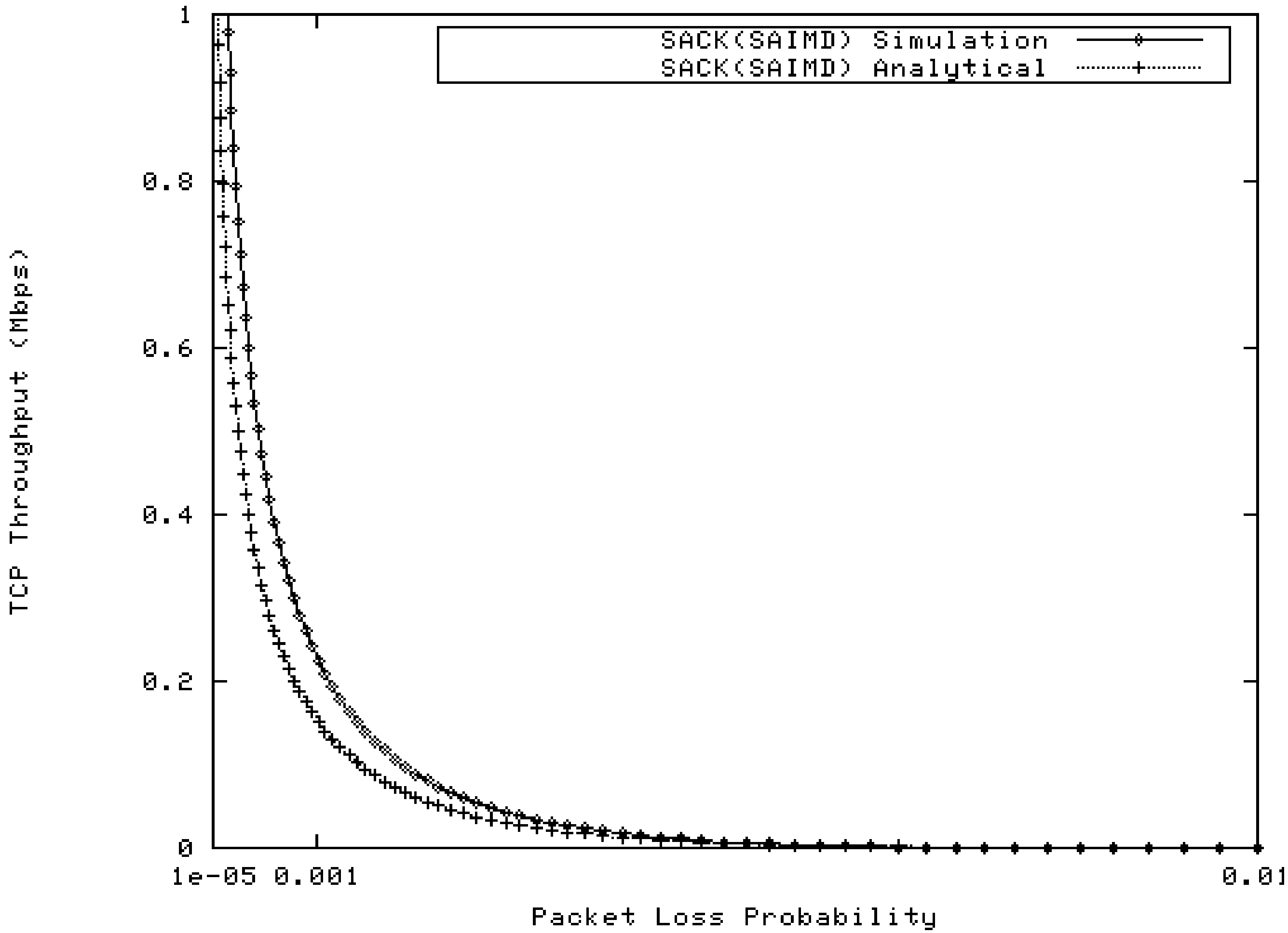
- SAIMD performance model with low and high burst contention probability
- SAIMD behavior while collecting TDs.
- SAIMD behavior while triggering TOs.

Simulation and Analytical Results

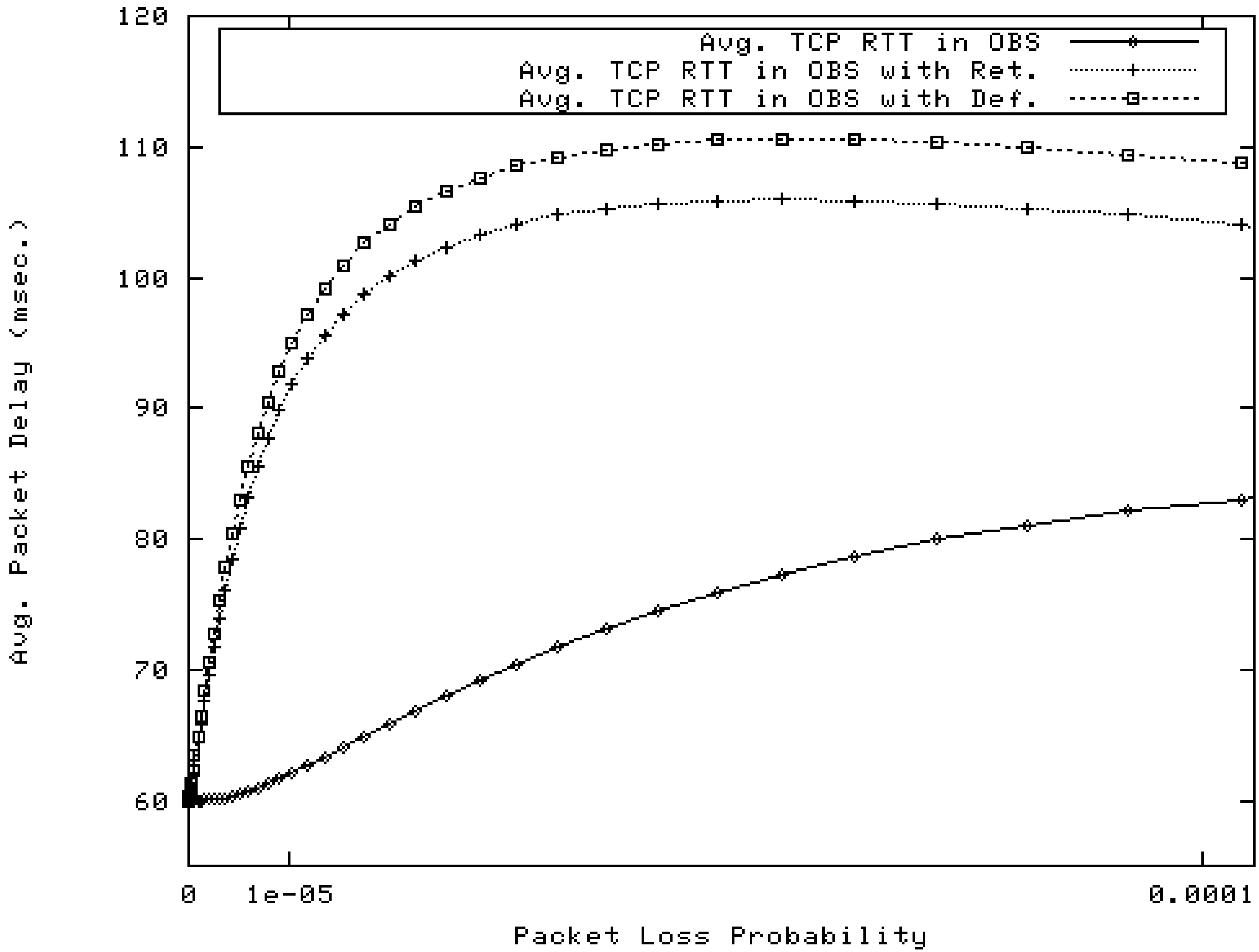


- NSF network topology
- 8 wavelengths, each operates at 10Gb/s
- Burst timeout is 5*m*sec, burst length 50KB
- TCP packet size is 1KB
- Burst offset time is 6*μ*sec.
- LAUC-VF is implemented.
- Burst retransmission and deflection routing were implemented
- Simulation time 10^4 seconds

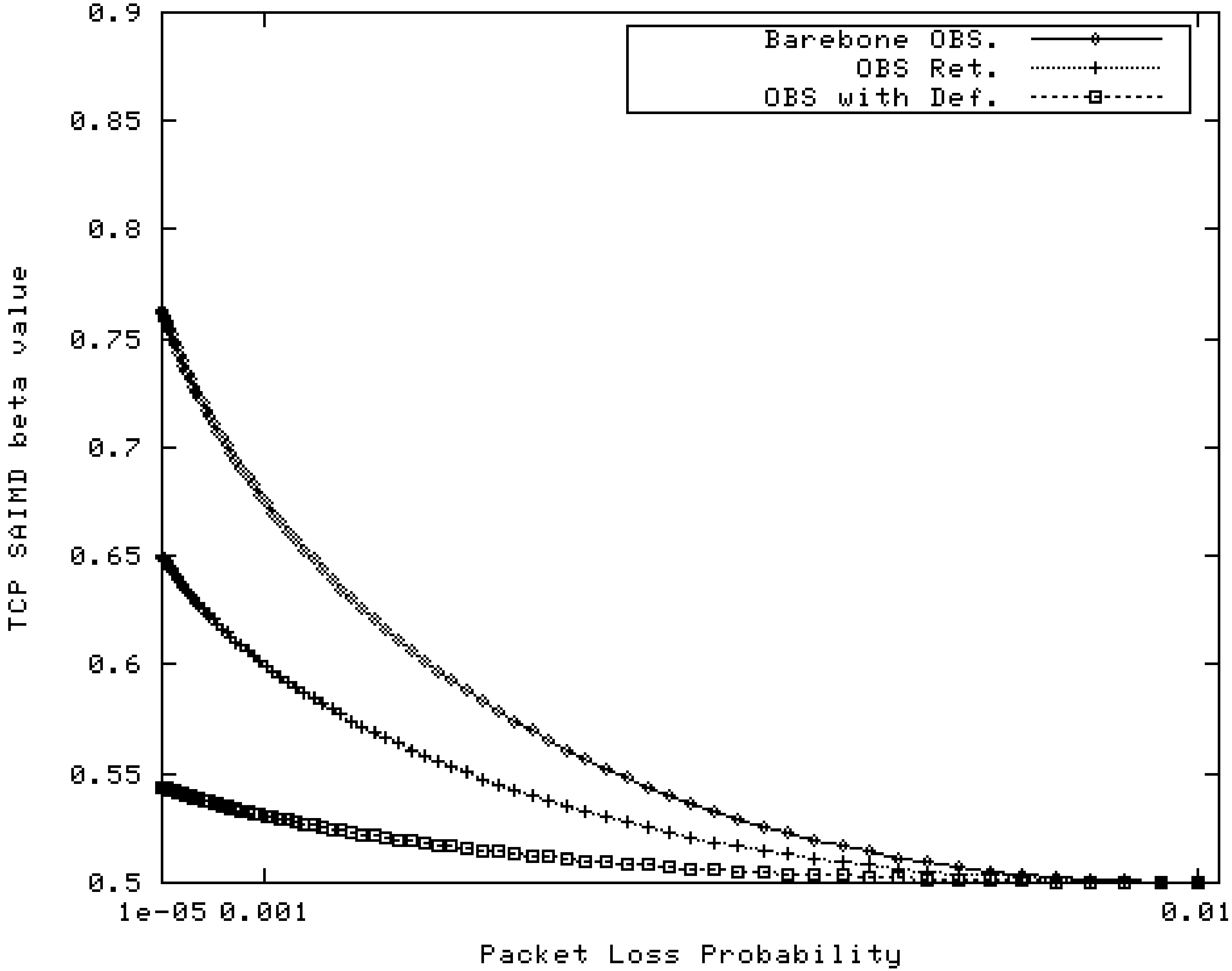
Verification of the Throughput Model



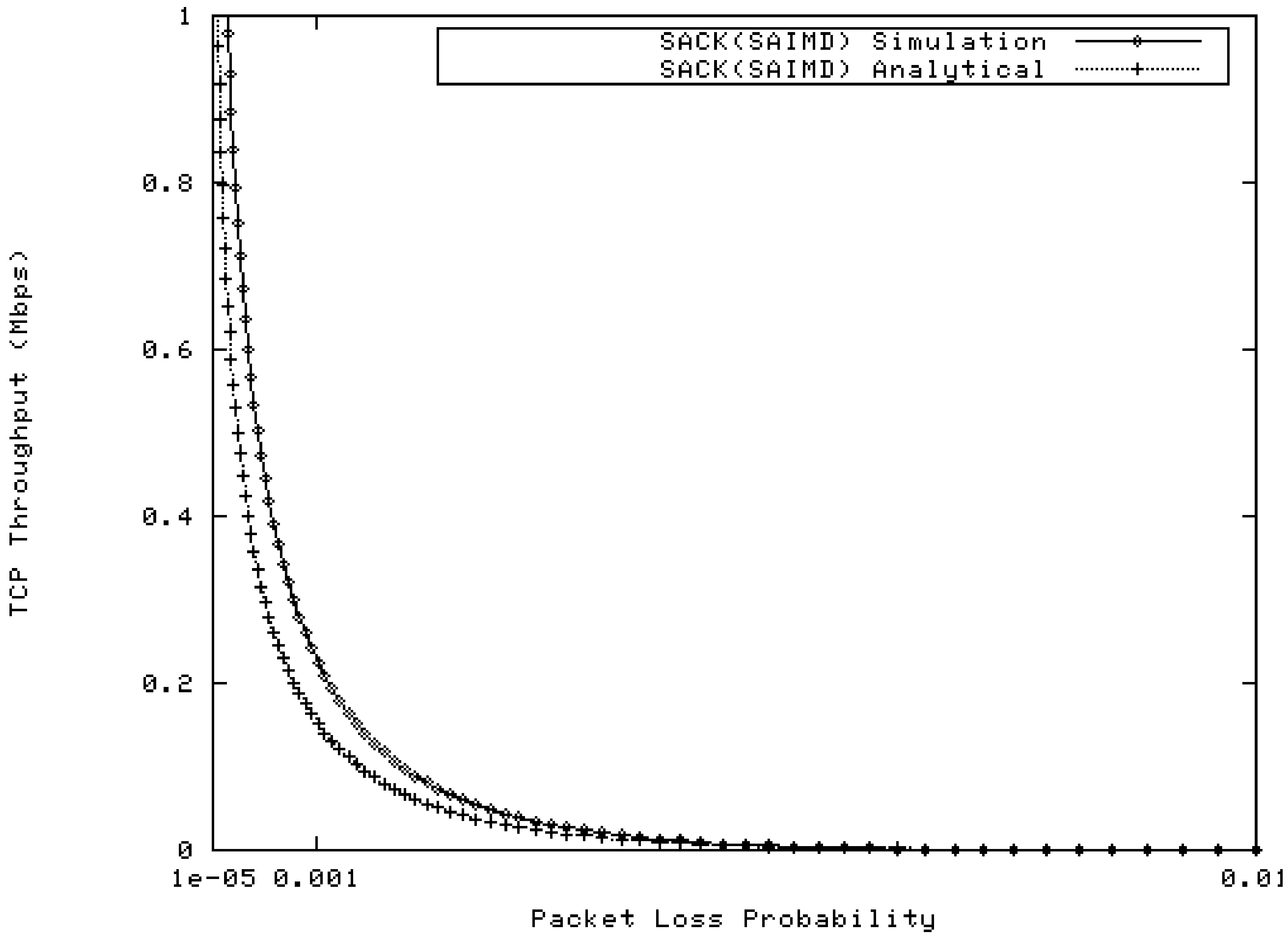
TCP RTT vs. packet loss probability



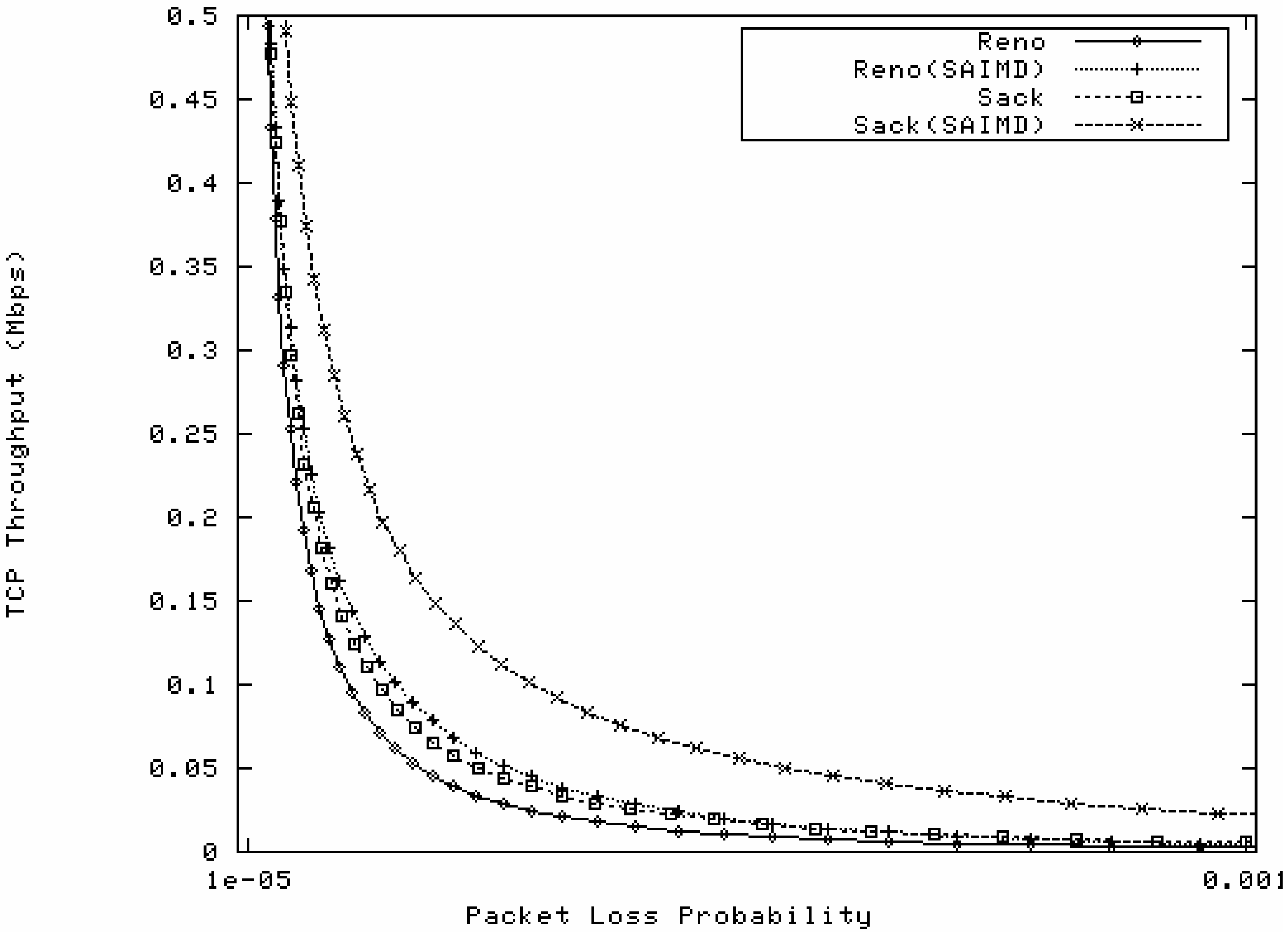
β value vs. packet loss probability



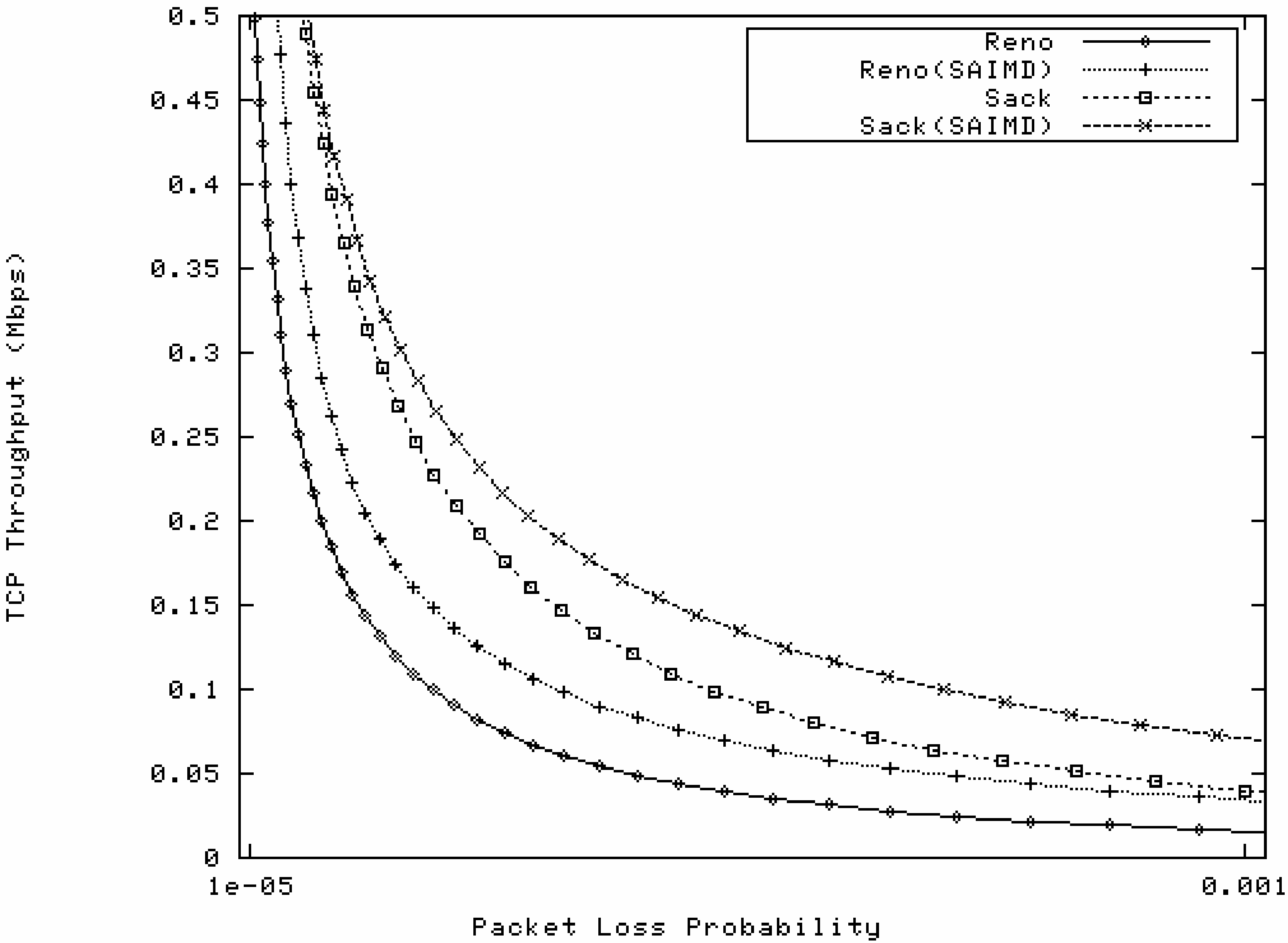
Analytical vs. Simulation Results



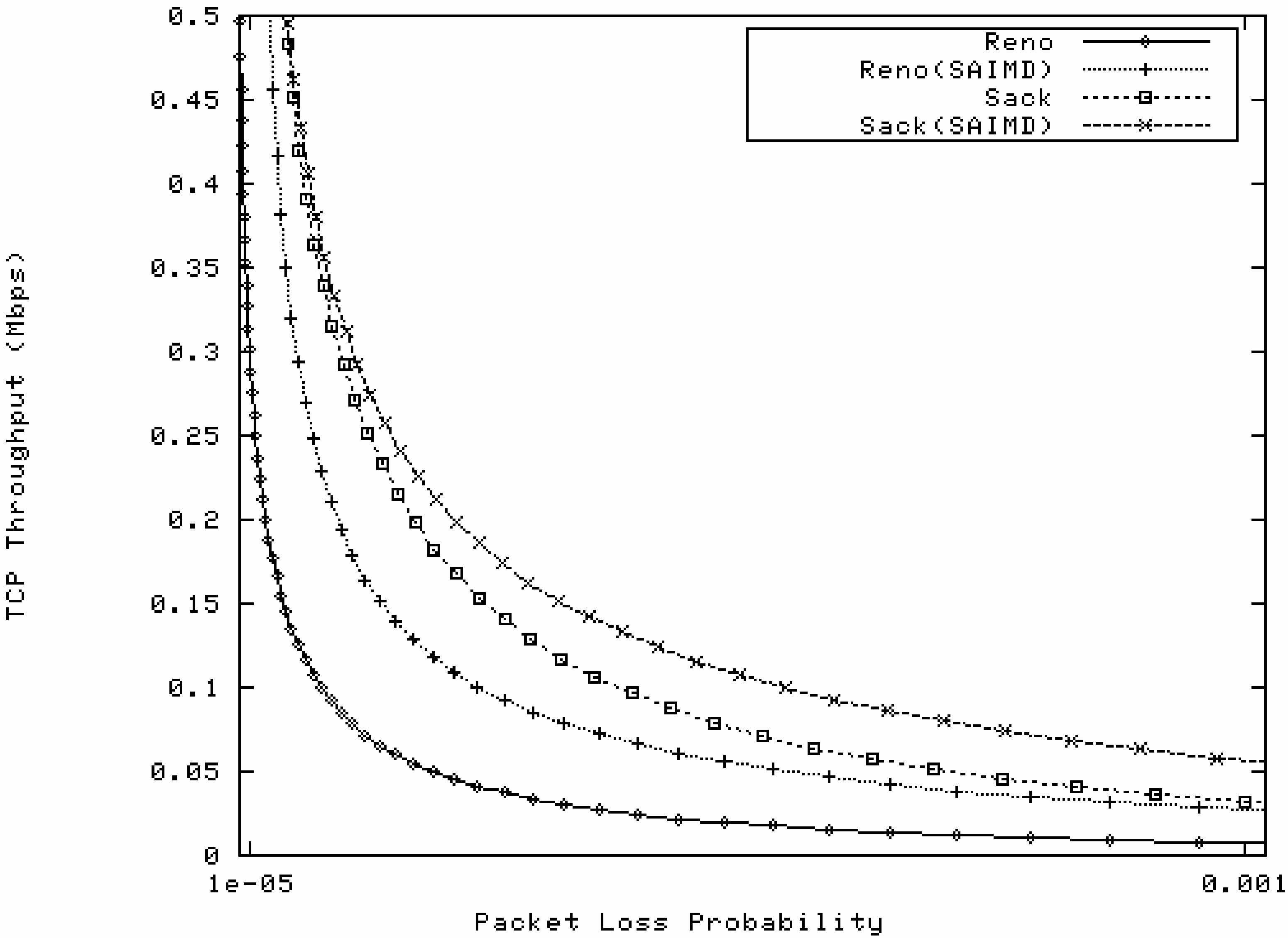
SAIMD Throughput over Barebone OBS



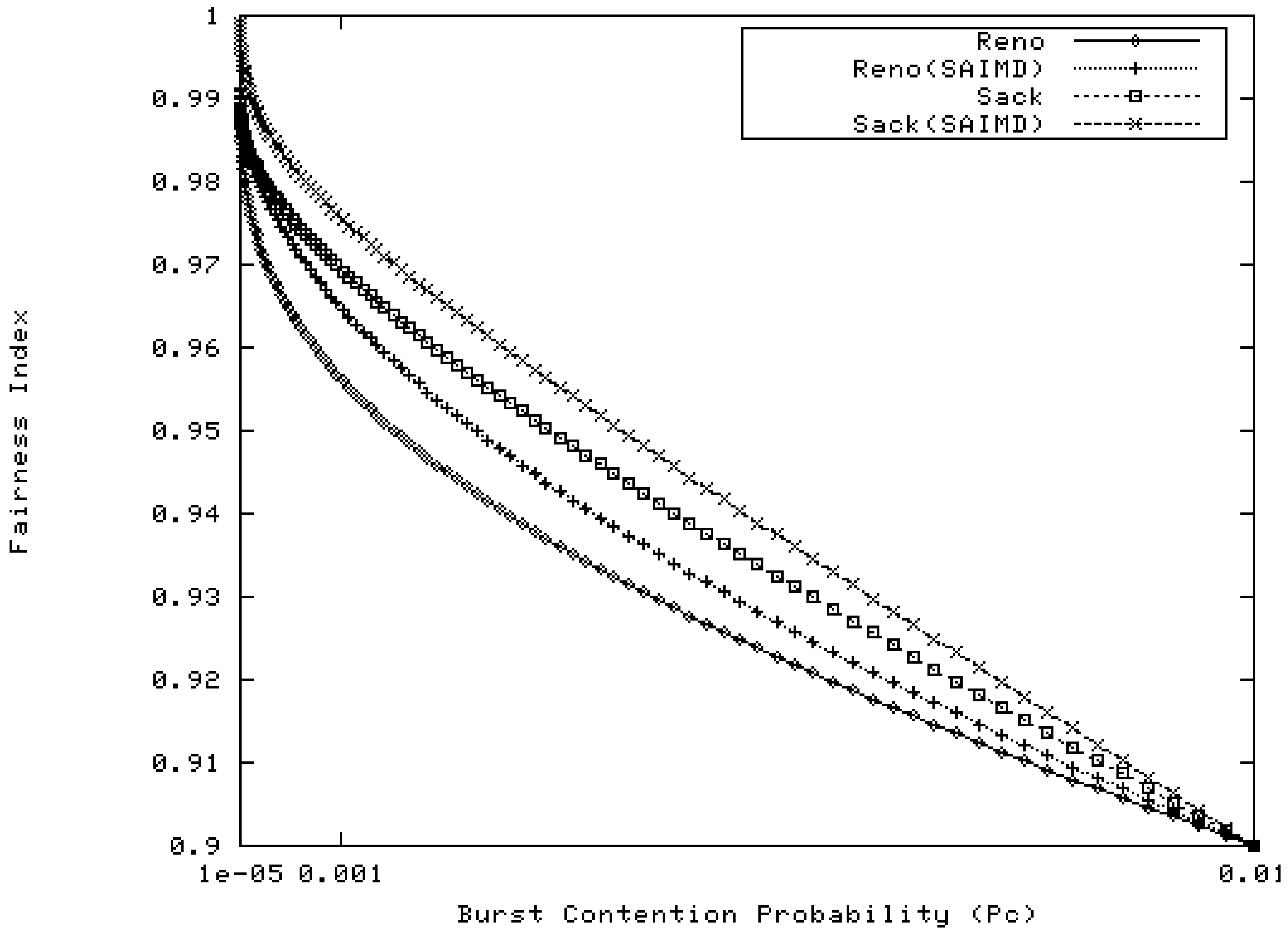
SAIMD Throughput over OBS with Burst Retransmission



SAIMD Throughput over OBS with Burst Deflection



SAIMD Fairness over OBS with Burst Retransmission



Conclusion

- Propose SAIMD, which is a new TCP implementation over OBS

  - Based on GAIMD framework => (α, β) instead of (1,0.5)

  - Dynamically determine beta value on each TCP segment drop event by evaluate the confidence on the network status.

  - Position the short-term RTT statistics on the long-term RTT statistics

  - Supports solving TCP false congestion detection problem for high-bandwidth and long-lasting TCP flows, such as applications in GRID

  - Enhances TCP throughput over OBS networks

  - Maintains Fairness with co-existing flows
- A throughput model for SAIMD
- Examined SAIMD over a number of OBS environments