Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver

By
Saad Biaz,Nitin H.Vaidya
Texas A&M University
IEEE Symposium ASSET'99

Presented by
Preeti Phadnis

Outline

- Introduction
- The Proposed Scheme
- Simulations
- TCP Performance using TCP-Aware
- Conclusions and Future Work

Introduction

- TCP may perform poorly over wireless links.
- Implicit assumption in TCP that all packet losses are due to congestion.
- Unnecessarily reduce throughput if packet losses happen to be due to wireless
- Modify TCP-Reno to TCP-Aware

Related Work

- Past proposals require cooperation from an intermediate host, like I-TCP.
- Previously adapt congestion avoidance schemes to enable the sender to distinguish between congestion losses and wireless losses.
- Not always good , receiver has better view of losses

Typical Scenarios

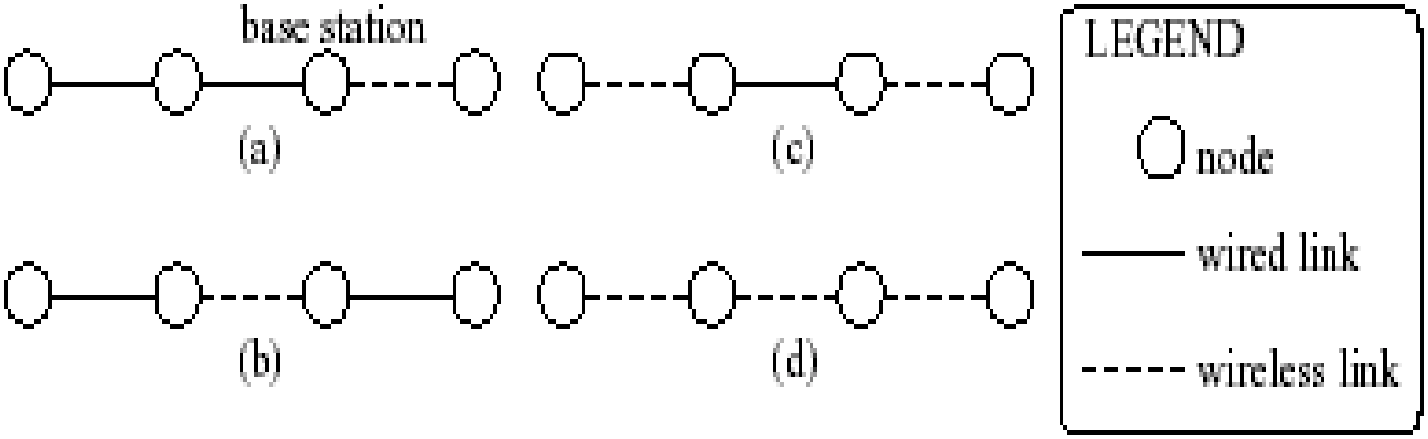


Figure 1. Typical scenarios

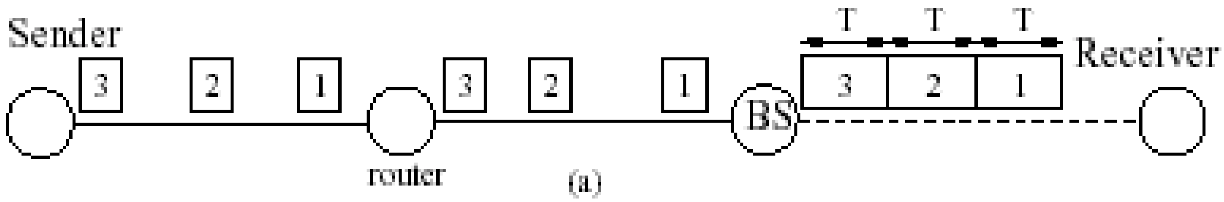
Scenario A is Used

Assumption : Wireless link is the bottleneck for the connection

Assumptions

- Only the last link is wireless
- The wireless link is the bottleneck for the connection. Packets tend to queue up at the base station. Most packets sent back to back on the wireless link
- The sender performs bulk data transfer

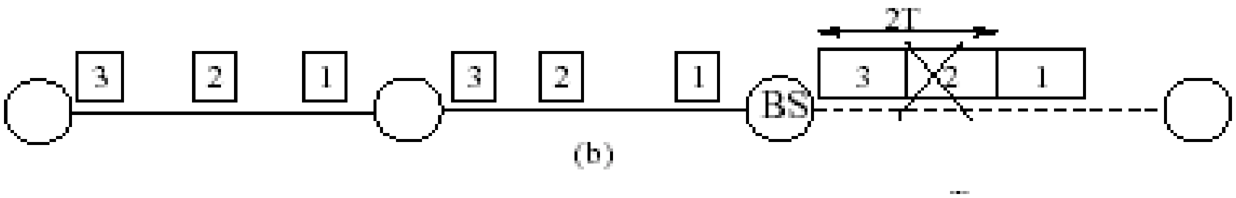
Scenario 1



No packets are lost

Packet Inter-arrival gap = time T required to transmit one packet on the wireless link

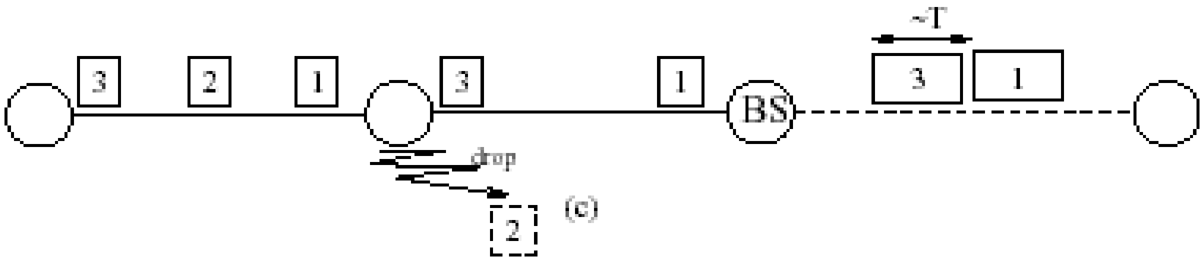
Scenario 2



Packet 2 is lost

Time between arrival of packet 1 and 3 = 2T

Scenario 3



Packet 2 is lost due to congestion

Inter-arrival gap = T (holds true if packet 3 arrives at the base station either before, or just after, the base station has transmitted packet 1)

Packet 3 is out-of order packet

The heuristic

- $T_{min} =$ Minimum Inter-Arrival Time observed so far by the receiver
- P_o : Out of order packet
- P_i : Last in-sequence packet received before P_o
- T_g : Time between arrival of packets P_o and P_i.
- n : Packets missing between P_i and P_o
- If $(n+1)T_{min} <= T_g < (n+2)T_{min}$, then n missing packets are lost due to wireless transmission errors.
- Else n missing packets are assumed to be lost due to congestion
- Scheme restrictive as better to mistake wireless loss as congestion loss

Metrics

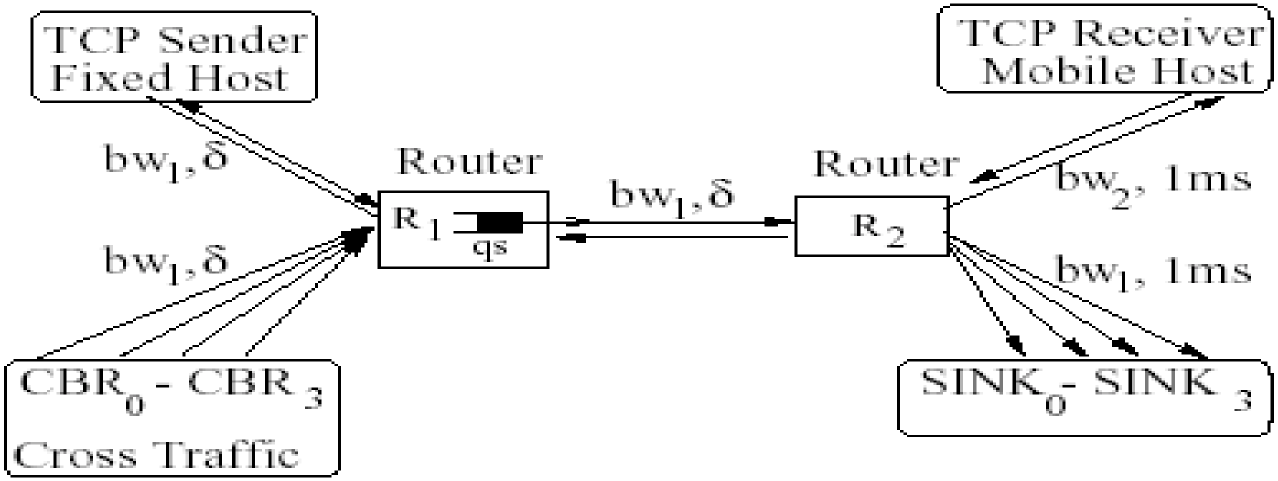
- Ac : Accuracy of congestion loss discrimination

$$= \frac{\text{Number of congestion Losses correctly identified}}{\text{Total Number of congestion Losses}}$$

- Aw : Accuracy of wireless discrimination

$$= \frac{\text{Number of Wireless Losses correctly identified}}{\text{Total Number of Wireless Losses}}$$

Simulation Model



Cross Traffic by 4 Traffic/Expoo agents – CBR UDP source with idle time and busy time exponentially distributed with mean 0.1 sec.

10 times one long-lived TCP connection (300-1000 secs)
Warm up period of 100 ms

Wireless link Transmission error rate rw = 1% to 5%

rc = congestion loss rate

δ = propagation delay – 1,8,18 ms

Round trip Propagation time – 6ms – 74ms

Simulation Parameters

- rc = 1% to 5%
- rw = 1% to 5%
- bw1 = 64kbps , 128Kbps, 256Kbps , 512Kbps ,1Mbps, 2Mbps
- bw2 = 64Kbps, 128 Kbps, 256 Kbps, 512Kbps, 1Mbps, 2Mbps
- Round trip propagation time Tp = 6ms,34ms, 74ms

Accuracy Ac and Aw

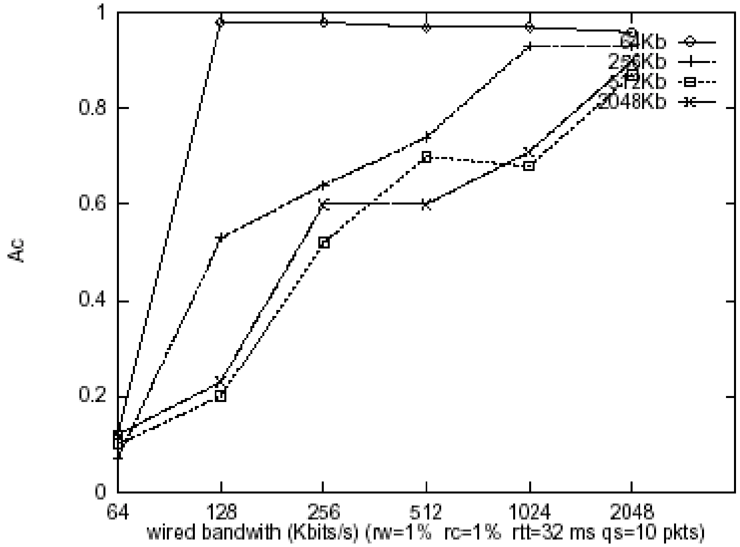


Figure 4. Accuracy A_c (r_c=1%, r_w=1%)

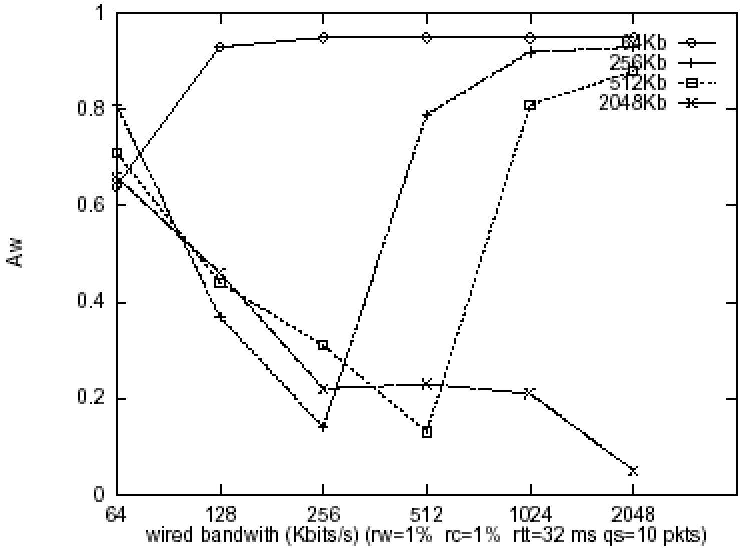


Figure 5. Accuracy A_w (r_c=1%, r_w=1%)

bw1/bw2 < 1 Aw increases, Ac decreases

Most Congestion losses mistakenly diagnosed as transmission losses

Factors determining Ac and Aw

- The Ratio bw1/bw2
- Overall loss rate rc + rw when bw1/bw2 < 1
- bw1/bw2 >1 , Aw increases with ratio
- bw1/bw2 = 1 , Aw small
- bw1/bw2 < 1 , smaller rc + rw larger will the fraction of congestion losses mistakenly diagnosed as wireless loss. Link R1 --> R2 bottleneck , Tmin = Service time at router R1, not transmission time over wireless link.

bw1/bw2 < 1

Packet 2 gets dropped

Only one cross traffic pattern between 1 and 3

Interarrival between 1 and 3 > twice transmission time

Mistaken for wireless loss

Cross traffic low rc is low

Packet 2 gets dropped

Two cross traffic packets between 1 and 3

Interarrival between 1 and 3 > thrice router service time

Correctly identified as congestion loss

Cross traffic high rc high

(rc = 5% , rw = 1%) and (rc = 1%, rw = 5%) and (rc = 5% rw = 5%)

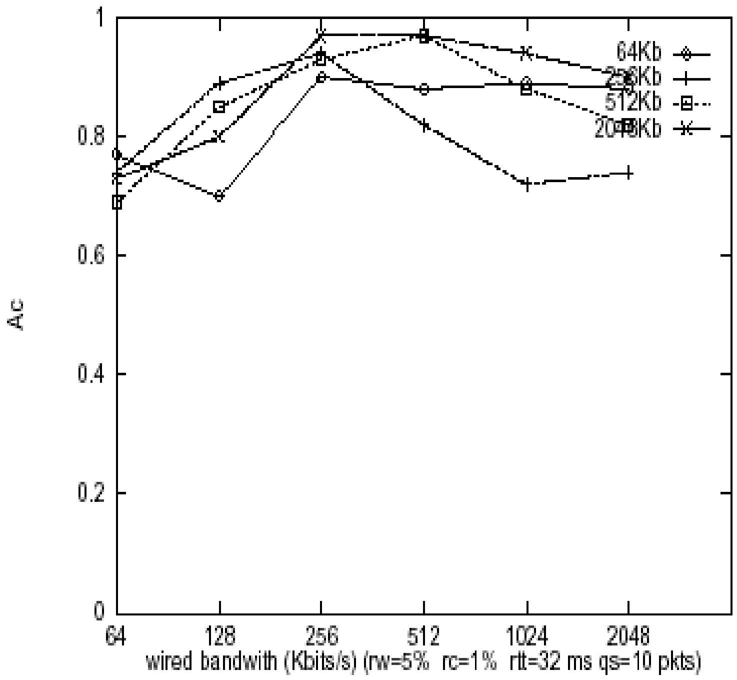


Figure 7. Accuracy A_c (r_c=5%, r_w=1%)

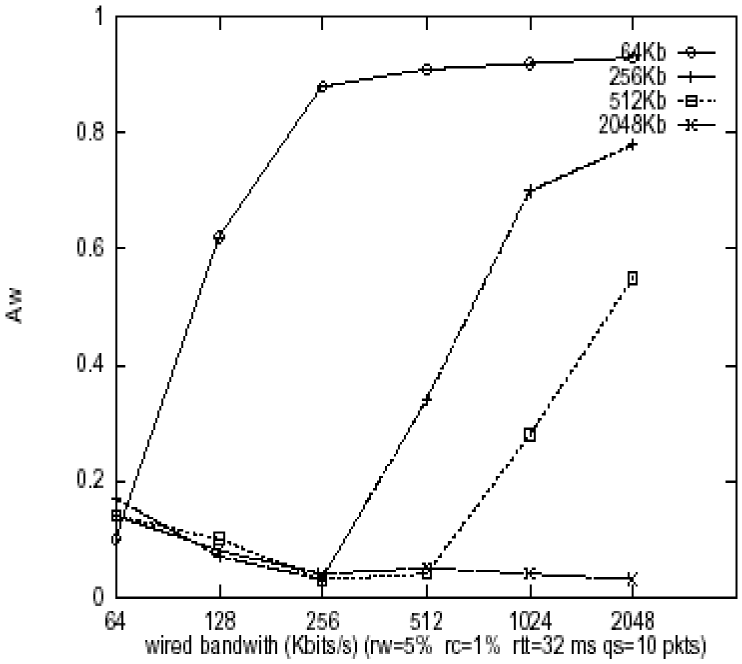


Figure 8. Accuracy A_w (r_c=5%, r_w=1%)

bw1/bw2 < 1 most losses diagnosed as congestion losses, Ac high Aw low

bw1/bw2 > 1 Aw increases with the ratio, Ac also high

TCP Aware

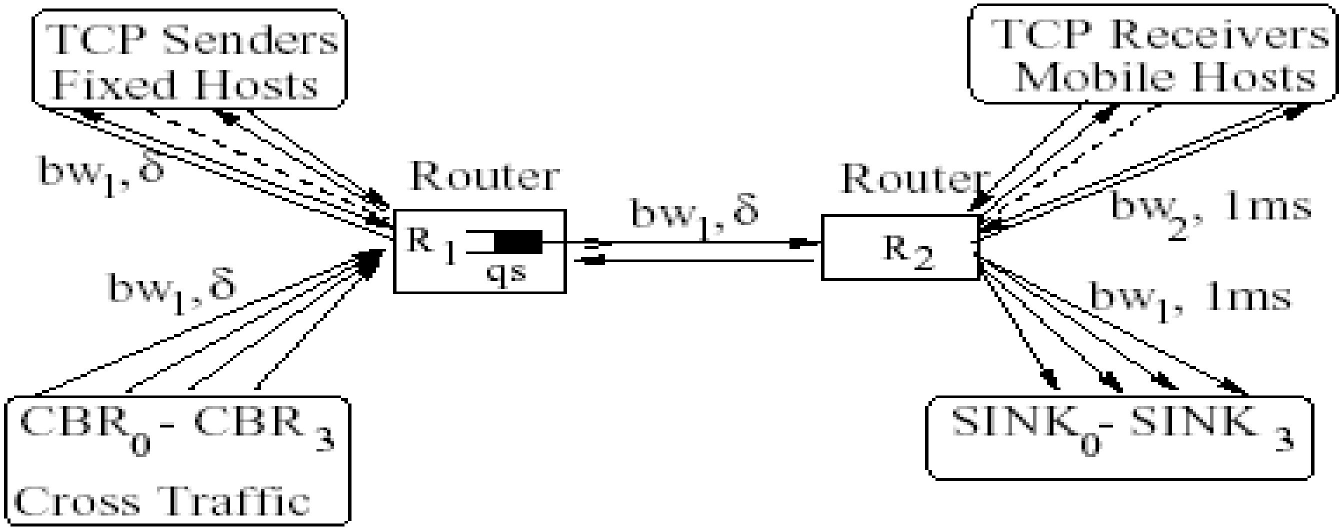


Figure 9. Simulation model

4 wireless and 4 fixed host

δ = 12 ms Round trip propagation = 50ms

TCP performance

- Ideal TCP-Reno has perfect knowledge of the cause of loss
- $$R_A = \frac{\text{average throughput using TCP-Aware}}{\text{average throughput using TCP-Reno}}$$

$$R_I = \frac{\text{avg throughput using Ideal TCP-Reno}}{\text{avg throughput using TCP-Reno}}$$

R_I represents highest improvement we can bring to TCP-Reno.

Mild Congestion – rc = 3-4%

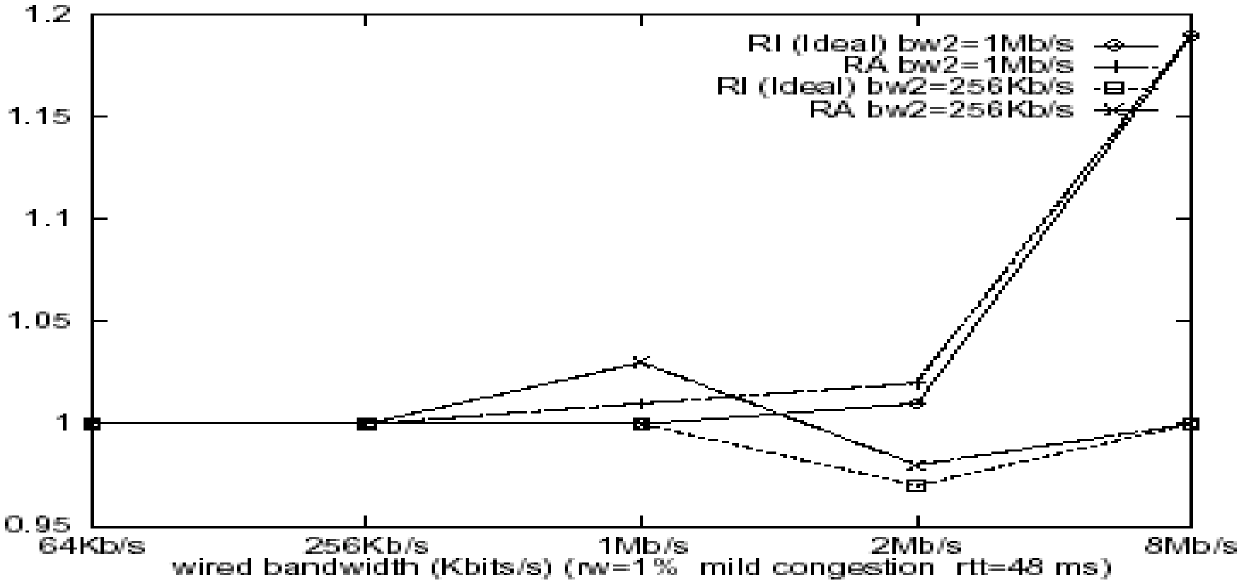


Figure 10. Performance improvement with $r_w = 1\%$ for mild congestion

Improvement marginal for both

Heavy Congestion – rc >= 5%

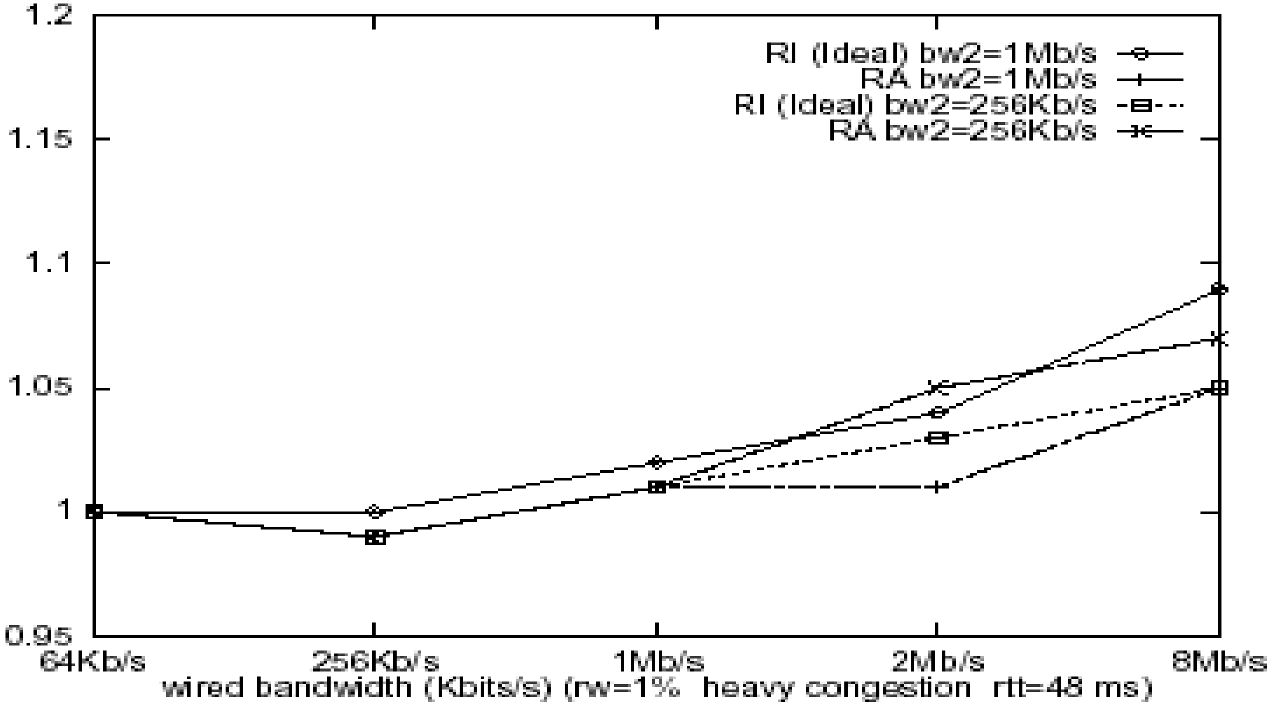


Figure 11. Performance improvement with $r_w = 1\%$ heavy congestion

No significant improvement

Ideal TCP-Reno

- Two factors determine the performance improvement
- Ratio rw/rc
- Bandwidth delay product
- Throughput $T = \frac{MSS\ C}{RTT \sqrt{p}}$

$$T_{tcp} = \frac{MSS\ C}{RTT \sqrt{rc+rw}}, \quad T_{ideal} = \frac{MSS\ C}{RTT \sqrt{rc}}$$

$$\text{ratio of improvement} = \frac{T_{ideal}}{T_{tcp}} = \sqrt{(rc+rw)/rc}$$

Ideal TCP Reno

- Bandwidth delay-product : when small cant achieve significant performance improvement.
- When bw2 = 256Kbps , RTT = 48ms

  Bandwidth delay product = 1536 bytes so improvement low

  (1.5 pkts with pkt size 1000 bytes)

  When bw2 = 1Mbps then improvement high

TCP Aware vs. Ideal TCP Reno

- TCP aware performs better
- Statistical variations
- TCP-Aware does not backoff for all congestion losses when Ac < 1

Conclusions

The heuristic works best when

- Bandwidth of wireless link smaller than bandwidth of the wired link
- Last hop for connection is wireless
- Overall packet rate is small
- TCP Aware performs similar to Ideal TCP Reno except when wireless loss is high and congestion is not heavy.