# Improving TCP/IP Performance over Third Generation Wireless Networks:

M C Chan and Ramchandran Ramjee
Bell Labs, Lucent Technologies

Student Name
Yatin Manjrekar

# Agenda

- Introduction
- Windows Regulator
- Performance of long-lived TCP flows
- Short-flow differentiation
- Conclusion

# Introduction

| 2G wireless | 2.5G wireless | 3G wireless |
|---|---|---|
| Phone calls<br>Voice mail<br>Simple emails | Phone /fax<br>Voice mail<br>Large emails<br>Web browse<br>Navigation<br>New updates | Phone/fax<br>Global roaming<br>Large emails<br>High speed web<br>Video-conferencing |
| Speed: 10 kb/sec | Speed 64-144 Kb/sec | Speed:144 kb-2mb/sec |
| Download 3 min mp3<br>31-41 min | Download 3 min mp3<br>6-9 min | Download 3 min mp3<br>11 sec-1.5 min |

# Introduction cont.

- **TCP is most widely used**
- **Windows regulator that Maximizes throughput for all buffer sizes (important metric for long flows)**
- **Scheduling algorithm to reduce transfer latency for short flow**
- **Exploiting user diversity is important**

# Simplified 3G network

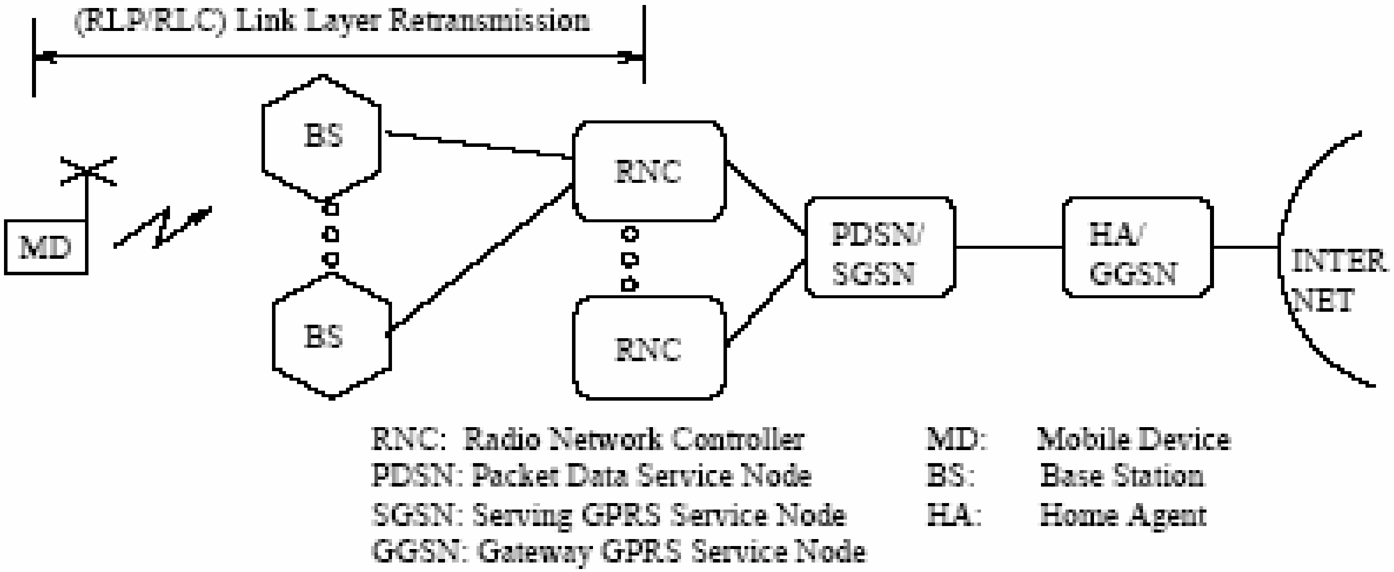
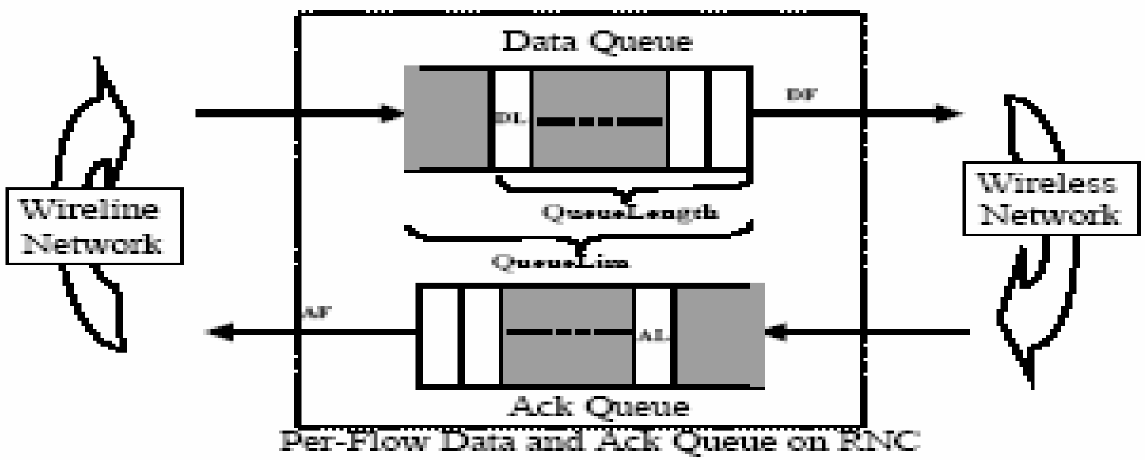



**Figure 1: 3G network architecture**

A simplified architecture of a 3G wireless network is shown

# Ack Regulator




DataSeqNoLast (DL): Largest Sequence # of Last Data Packet Received
DataSeqNoFirst (DF): Largest Sequence # of Last Data Packet Sent
AckSeqNoLast (AL): Largest Sequence # of Last Ack Packet Received
AckSeqNoFirst (AF): Largest Sequence # of Last Ack Packet Sent

Figure 6: Ack Regulator Implementation

each time an ack is sent back towards the TCP source, there is buffer space for at least two data packets from the source.

# Windows Regulator

- Wi = Ni+Yi+1 ------------------- (1)
- Yi+1 < Wi ---- ---------------------(2)
- Yi+B >= Wi ------------------------ (3)

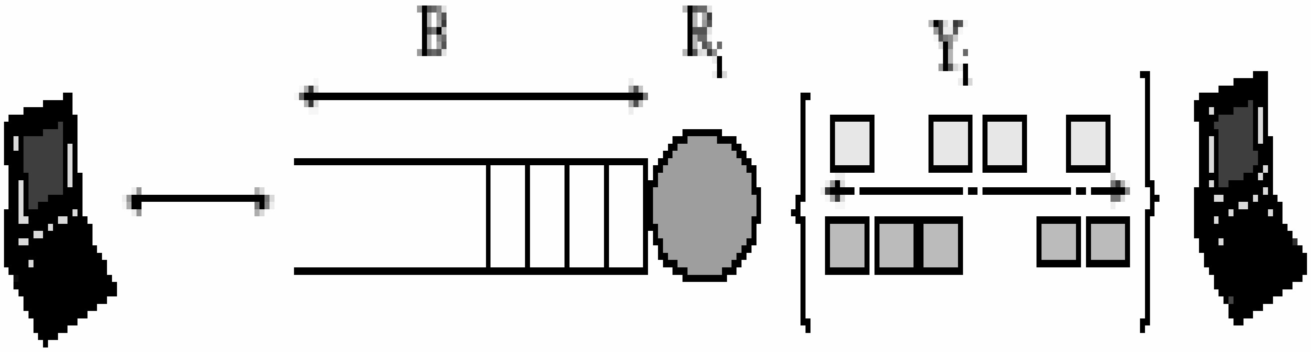



Fig. 2. Simple model of the wireless network

# Windows Regulator-static (WRS)

**Wi = B and Yi >0 fulfills equation 3**

**If B <= Yi+1 the queue could be idle**

On Enque of each Ack

set $W^r = B$

transmit the Ack to the source

Fig. 3. Window Regulator-Static

# Windows RegulatorDynamic(WRD)

**Wr = Y + B ------ fulfills Equation 3**

**If B=0, it underflows & violets Eqn 2**

**As W(wrd) >=W(wrs), throughput W/RTT is better for wrd than wrs**

On Deque of each Data packet

$Y = Y + 1$

On Enque of each Ack

$Y = Y - 1$

set $Wr = Y + B$

transmit the Ack to the source

Fig. 4. Window Regulator-Dynamic

# Windows Regulator with ack Buffer(WRB)

- **In WRD, if Y(i+1)-y(i) > B then no acks to TCP source.**
- **Ba is ack buffer in reverse direction**

On Deque of each Data packet
  $Y = Y + 1$
  if there is an Ack stored in the ack buffer, then
    $W^r$ = Y + B + $B_a$
    transmit Ack to the source
  endif
On Enque of each Ack, set
  $Y = Y - 1$
  $W^r$ = Y + B + $B_a$
  if (new Ack AND ($W^r <$ last transmitted value of $W^r$))
    store Ack in the Ack Buffer
    $B_a = B_a + 1$
  else
    transmit Ack to the source
  endif

Fig. 5. Window Regulator with ack Buffering

# Performance of Long lived TCP flow (ns-2 simulation setup)

- **L=100 mpbs, D=1ms or varied**
- **RR =64 kbps RD=uni dist mean 125ms/var 15**
- **FR = variable FD=uni dist mean 75 ms/var=30**
- **TCPsack 1000 sec long,pkt=1KB, Que=20 pkt**
- **Wmax=500kb ensure tcp never window limited**

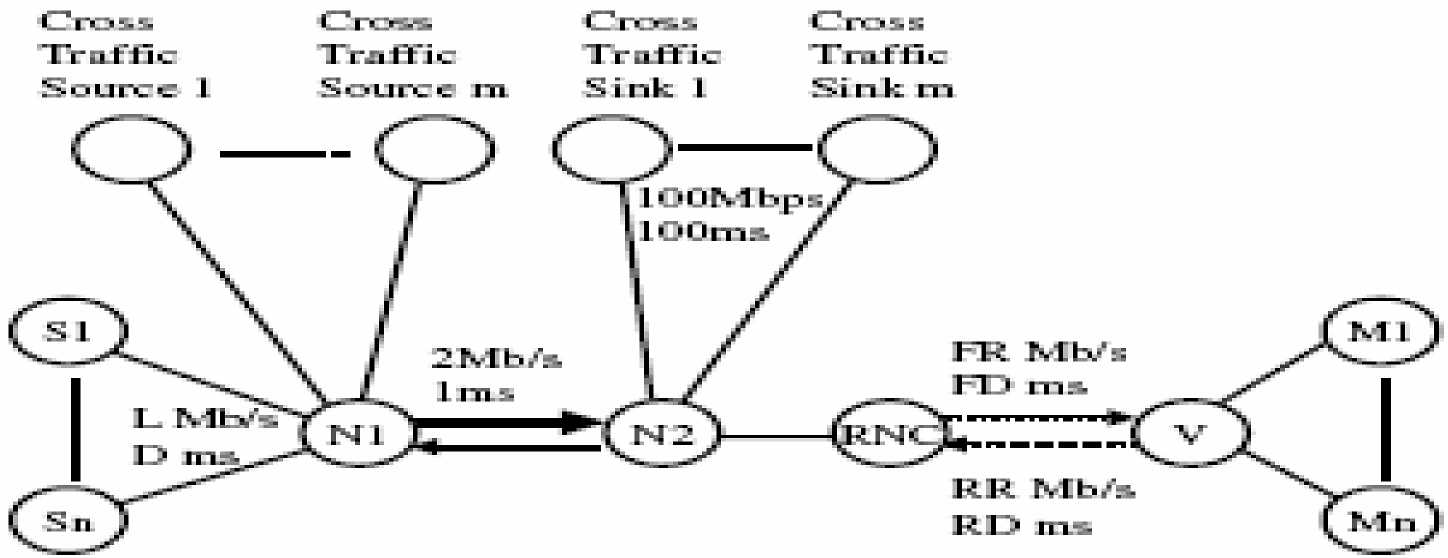



Fig. 6. Simulation Topology

# Throughput vs. buffer size(single user)

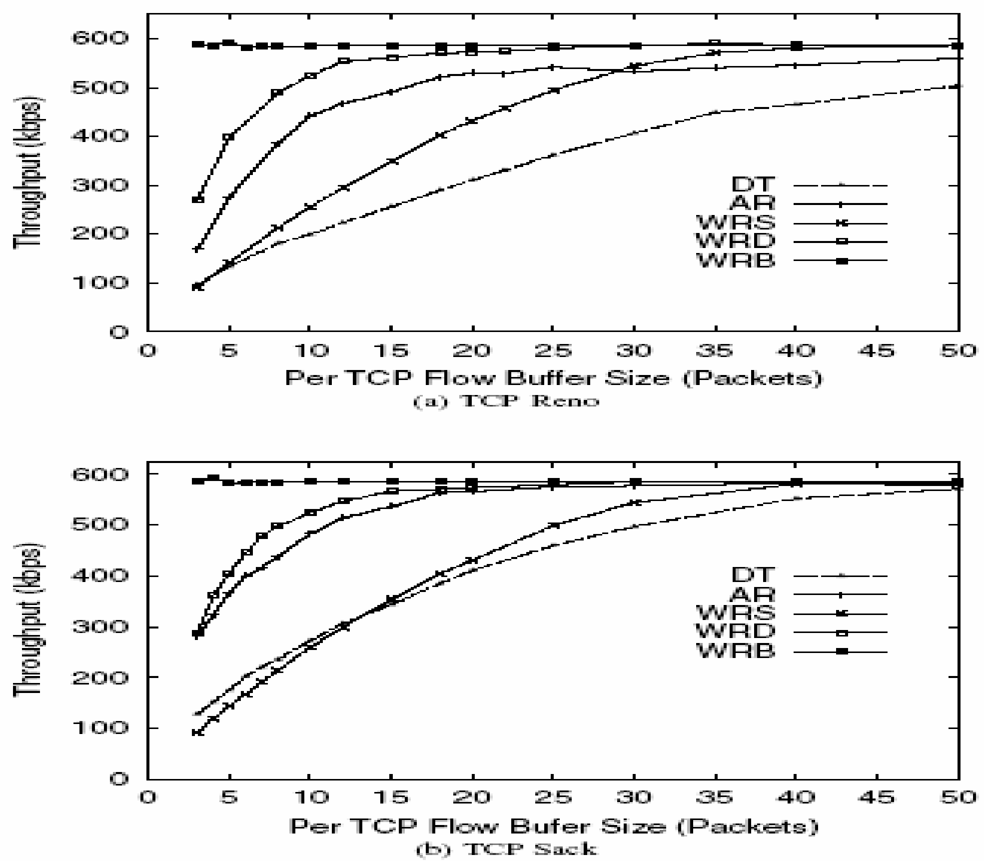



Fig. 7. Throughput vs. Queue Length for a single user/flow (TCP Reno and Sack)

- TCP sack slightly better than TCP Reno

TCP Sack:

- AR performs better than WRS and DT
- WRD is close to max with Buffer >15
- WRB outperforms all others

# Throughput vs. Buffers Multiple users

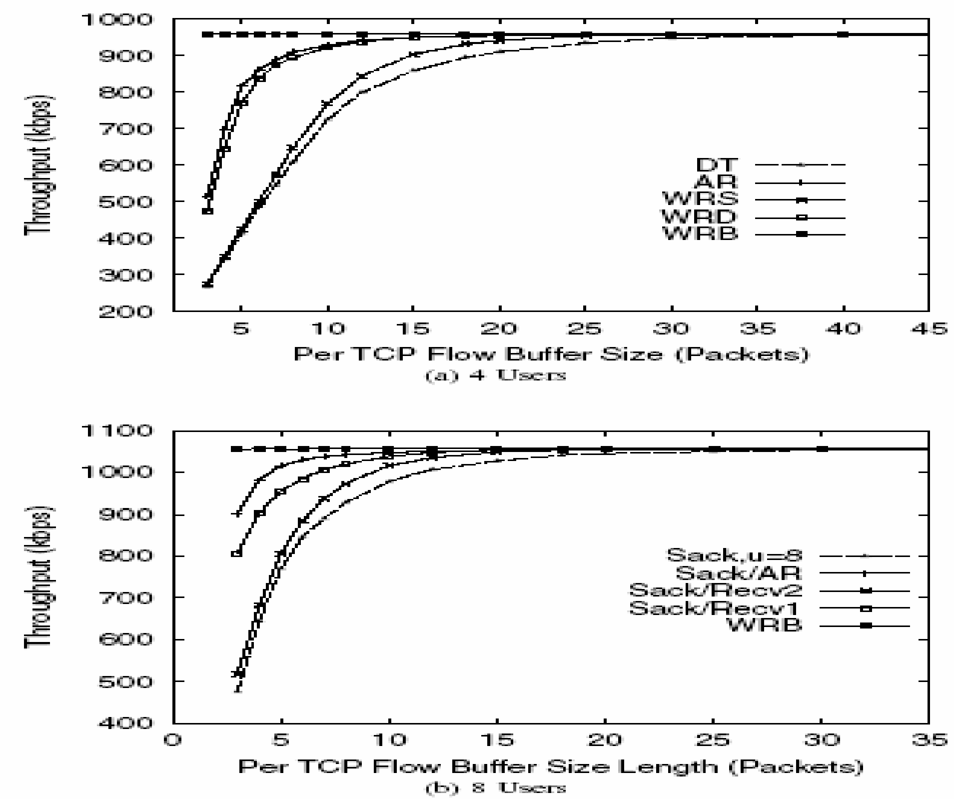



Fig. 8. Aggregate Throughput vs. Queue Length for TCP Sack with multiple users/flows

- Similar to Single user
- Except AR outperforms WRD
- Absolute performance is better with multiple users.
- 8 users are similar to 4 users except gap between AR and WRD widens.
- WRB still performs best.
- WRD is best considering RTT tradeoff.

# Throughput vs. Wired Latency

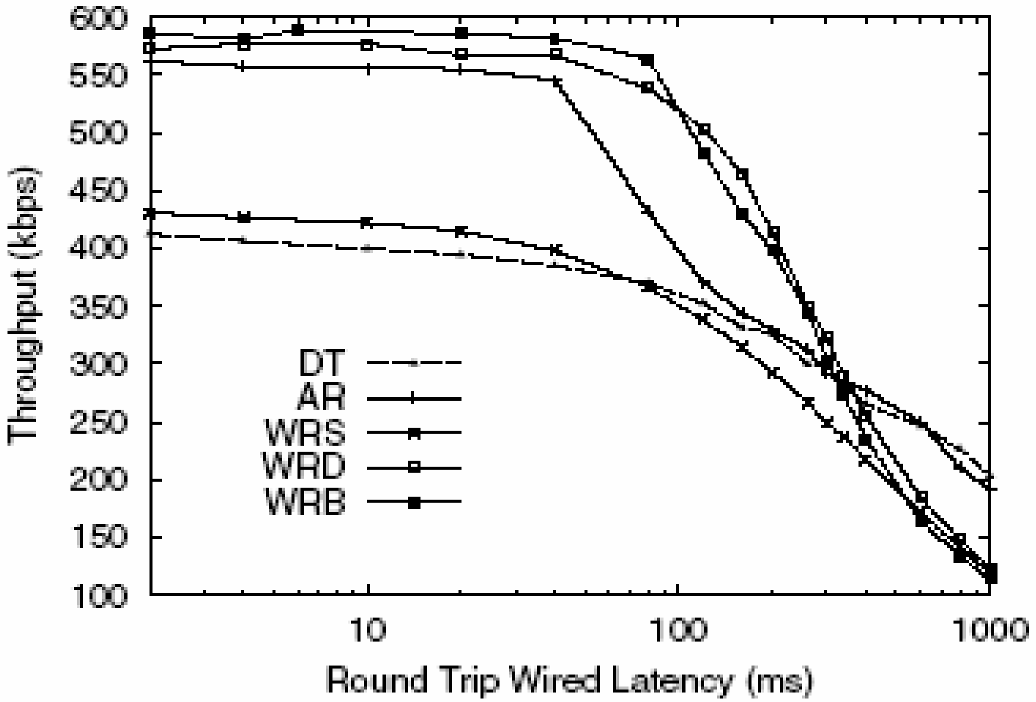


Fig. 9. Throughput vs. Round Trip Wired Latency

- For D < 70ms, WRS is better than DT. It is degrading>70ms
- For D<40ms AR performs well
- WRD is fairly robust till 200 ms
- WRB is 25% better than DT at 200ms
- AR degrades after d=40ms but it is always better than dt
- >400ms, WRD/WRB are worse than DT
- Increasing RLP buffer size will help AR,WRD,WRB

# Throughput Vs Loss

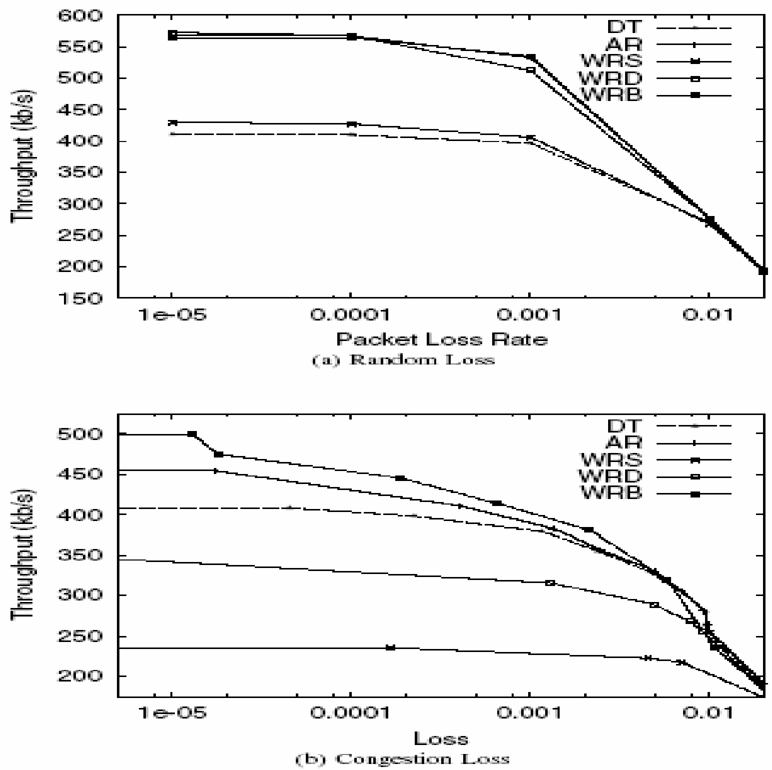


Fig. 10. Throughput vs. Loss

**Random Loss**

- **AR,WRB,WRD perform well for small amount but start degrading after $10^{-3}$**
- **After $10^{-2}$ all algorithms have same low perf as random error is dominant factor**

**Congestion loss**

- **AR,WRB perform better than DT below congestion loss rate $10^{-3}$**
- **WRD/WRS performs poorly than DT**
- **AR/WRB don't degrade as they have ack buffer to provide fast feedback**

# Comparison Summary

- DT and WRS cannot adapt to the large rate and delay variation in wireless channel
- AR adapts well to the large variations but does not perform well with latency as estimation errors cause throughput degrade
- WRD performs well against latency but poorly against congestion loss
- WRB is best and robust against latency and packet loss

# Short flow differentiation

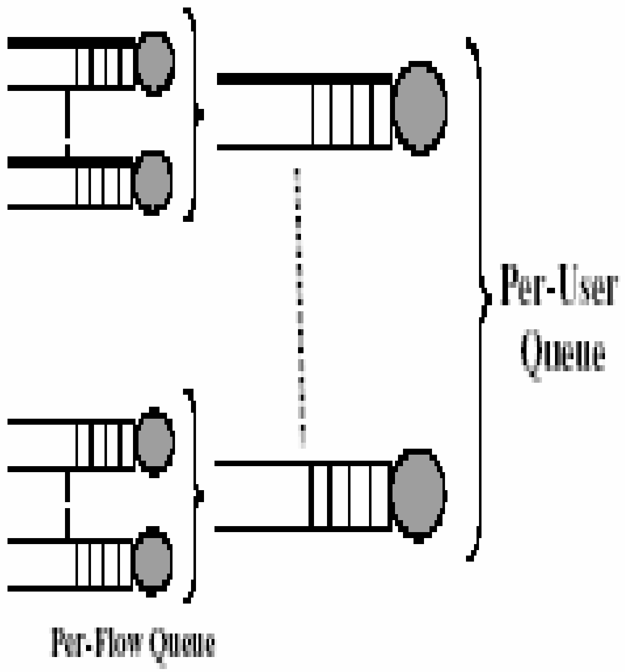



Fig. 11. Queuing Structure of Scheduler

- Per-flow queue.
- Per-user queue
- Simple flow differentiation without

Exploiting user diversity does not

Improve short flow latency and throughput

# Scheduling

- Intra-user scheduling

SFP : short flow priority algorithm

Short flow/Long flow based on length

Reclassification short-long-short

- Inter-user scheduling

PF: Priority Fair queuing

PF-SP: priority fair with Strict Priority

PF-RP: priority fair with Rate priority

# PF scheduler

- Each user reports measured channel conditions to the PF scheduler (RNC)
- User with best channel is selected to transmit in different time slot, better than round robin but unfair
- PF weight the current rate achievable by average rate received by user and select user  (i) with max Ri/Ai

# PF-SP Scheduler

Let set of short flow user be $S$

Let set of long flow user be $L$

if $S$ is non empty

  select the user $i$ with the largest $R_i$

else if $L$ is non empty

  select the user $i$ with the largest $\frac{R_i}{A_i}$

Update $A_i$ for all users

Fig. 12. The PF-SP Algorithm

- **Short flows have higher priority**
- **It is unfair to long flow.**
- **Average rate for each user is maintain over all short and long flows**

# PF-RP scheduler

Run the PF and let user $i$ be selected
Run the PF-SP and let user $j$ be selected
if $R_i > R_j$
    select user $i$
else
    select user $j$
update $A_i$ for all users

Fig. 13. The PF-RP Algorithm

- **Better balance between minimizing Short term latency and fairness**
- **It relatively sacrifices fairness to users with long flows**
- **Latency reduction improves with more no of users**

# Performance Comparison

- **Ns-2 simulation setup**

**Parameters in Web traffic model in table I are used. FTP sessions are used in background.**

**ftp packet size 1000 packets(1 MB)**

**Short flow is below 15 Kb**

**Reset duration 1 sec.**

**User 1 has web and 2-4 have medium load ftp traffic components**

# Performance comparison cont.

| Web Model Elements | Attributes | Distribution and values |
|---|---|---|
| Web Page | Time interval between Pages (s)<br>Number of Web objects per pages | Exponential, mean=15<br>Exponential, mean=5 |
| Web Object | Time interval between Web objects (s)<br>object size | Exponential, mean=0.01<br>Bounded Pareto, mean = 12, shape = 1.2, max=200 |

| FTP Model Elements | Attributes | Distribution and values |
|---|---|---|
| File Size | Time interval between FTP Session (s)<br>FTP File Size | Constant, mean=200/80 (medium/heavy traffic)<br>Constant, mean=1000 |

TABLE I

TRAFFIC MODEL PARAMETERS

# Performance comparison

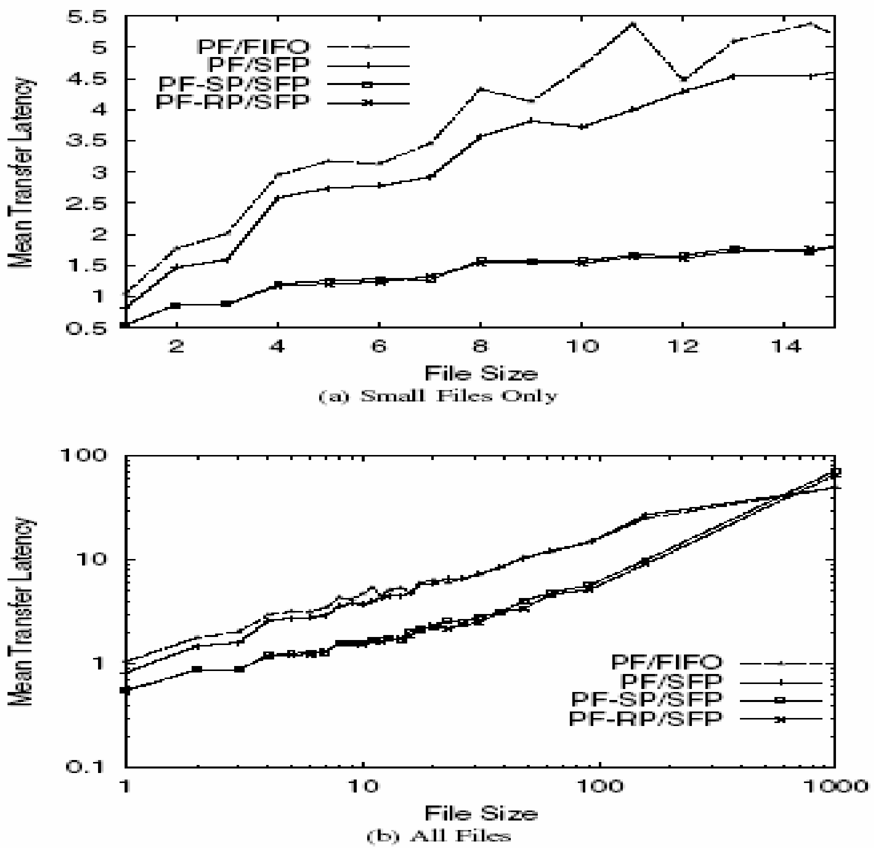



Fig. 14. Download Latency for 4 users, Different Base SNR

2 users with SNR +4dB and 2 users with -4dB

- PF/SFP has better latency under 15 KB then similar to PF/FIFO
- PF-SP and PF-RP can exploit differences in channel conditions

# Performance comparison cont.

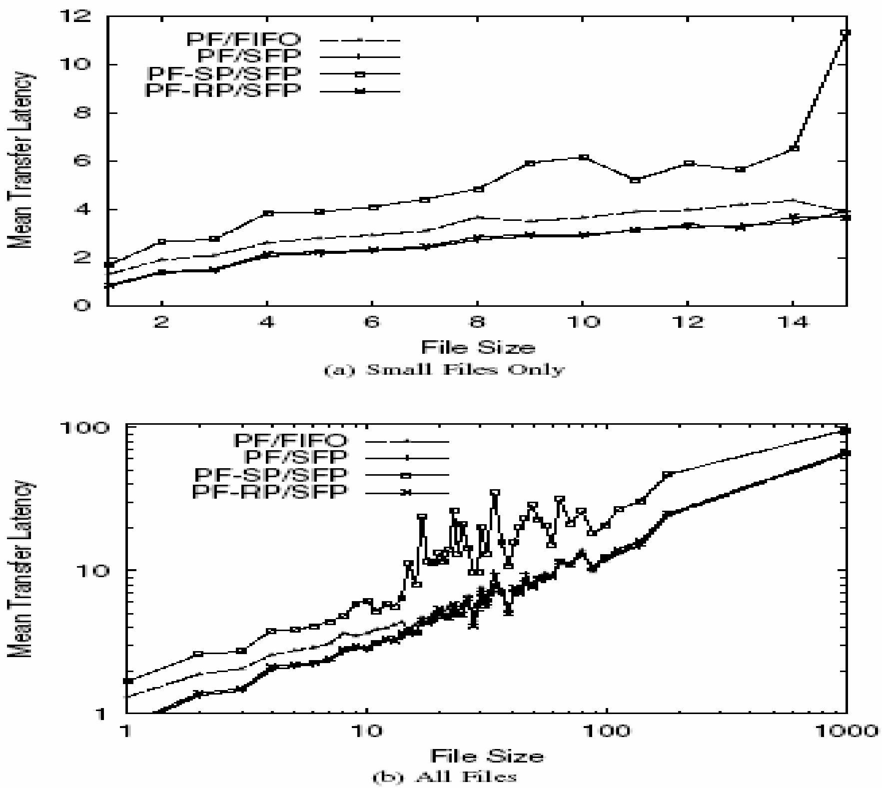



Fig. 15. Download Latency for 4 users, Same Base SNR

**All 4 users with same -4dB SNR**

- **PF-SP performs worst for all file sizes**
- **PF and PF-RP is better than PF-FIFO**

- **PF/SFP is always better than PF/FIFO**
- **PF-SP is better or worse than PF depending on channel condition**
- **PF-RP is the most robust**

# Conclusion

- WRS a common algorithm used in wired routers perform poorly
- WRB which adapts wireless channel conditions and performs ack regulation improves 100% over DT
- PF-RP/SFP provides robust performance over different user channel conditions. 54% over PF/FIFO

# References

1 M C Chan and Ramchandran Ramjee Improving TCP/IP performance over Third generation Wireless networks

Infocom 2004

2 M C Chan and Ramchandran Ramjee:TCP/IP performance over 3G wireless link with rate and delay variation ACM mobicom 2002

# Q & A

??