# Improving End-to-End Throughput of Mobile IP using SCTP

**Mohammed Atiquzzaman**

(Co-author: Shaojian Fu)

School of Computer Science

University of Oklahoma,

Norman, OK 73019-6151.

Email: atiq@ou.edu

# Introduction

- High probability of packet losses during Mobile IP handover causes TCP Reno long time to recover.
- Most previous research on end-to-end throughput over Mobile IP carried out on TCP-Reno.
- TCP SACK can recover from multiple packet losses in a single window of data, but performance is degraded by a limited number of available SACK blocks.
- We investigate impact of SCTP SACK on the transport layer throughput during Mobile IP handovers.

# Problem statement

- Non-consecutive multiple packet *losses* in a single window of data during Mobile IP handovers
  - *signal fading* at the edge of the cell
  - *signaling delay* arising due to handovers
- TCP Reno cannot recover efficiently from non-consecutive multiple losses in a window.
  - Reno has to wait for a timeout after Fast Retransmit
- TCP Reno with SACK option recovers non-consecutive losses, but the recovery time is limited by its maximum 3 SACK blocks.
- Problem: Loss recovery time is very long

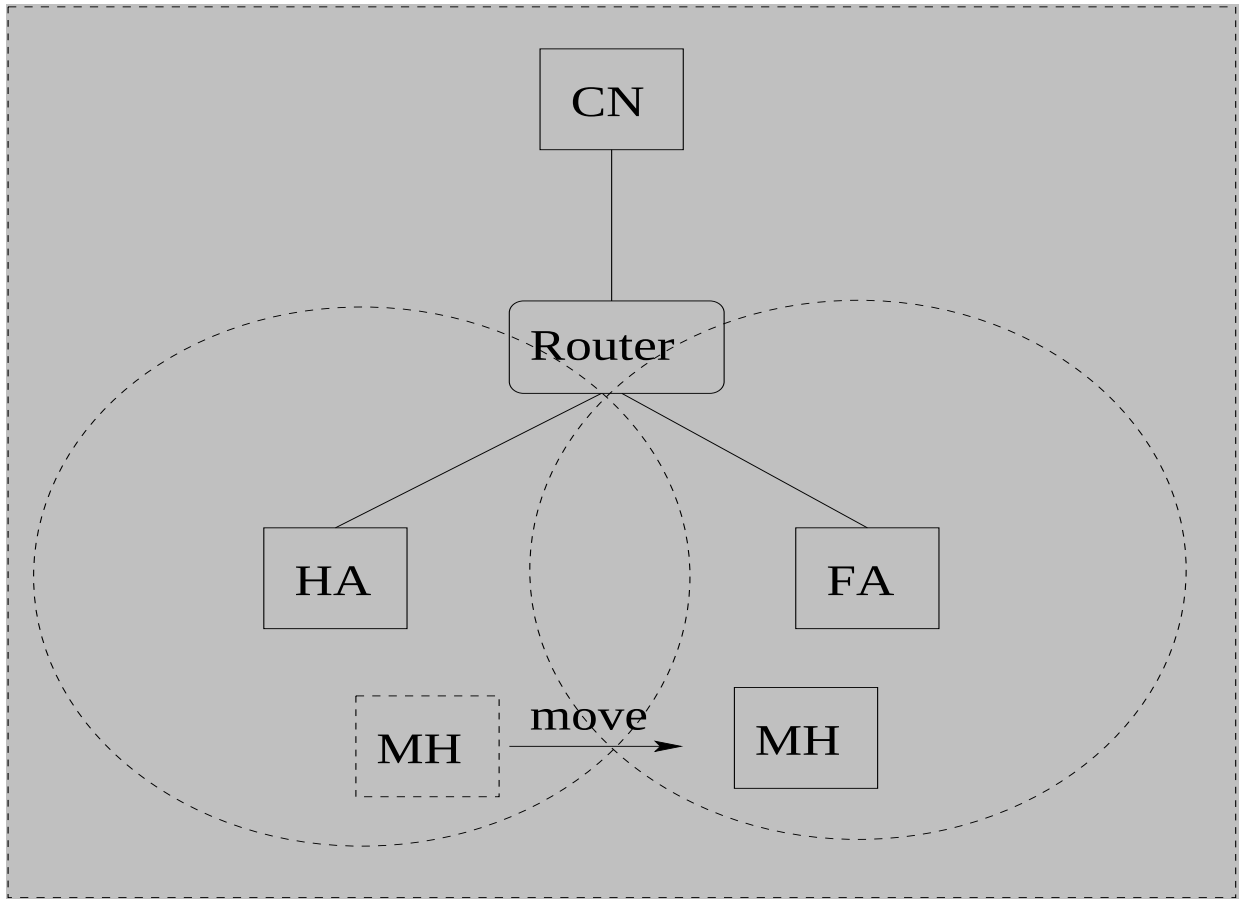
# A typical configuration of Mobile IP

CN: Correspondent Node

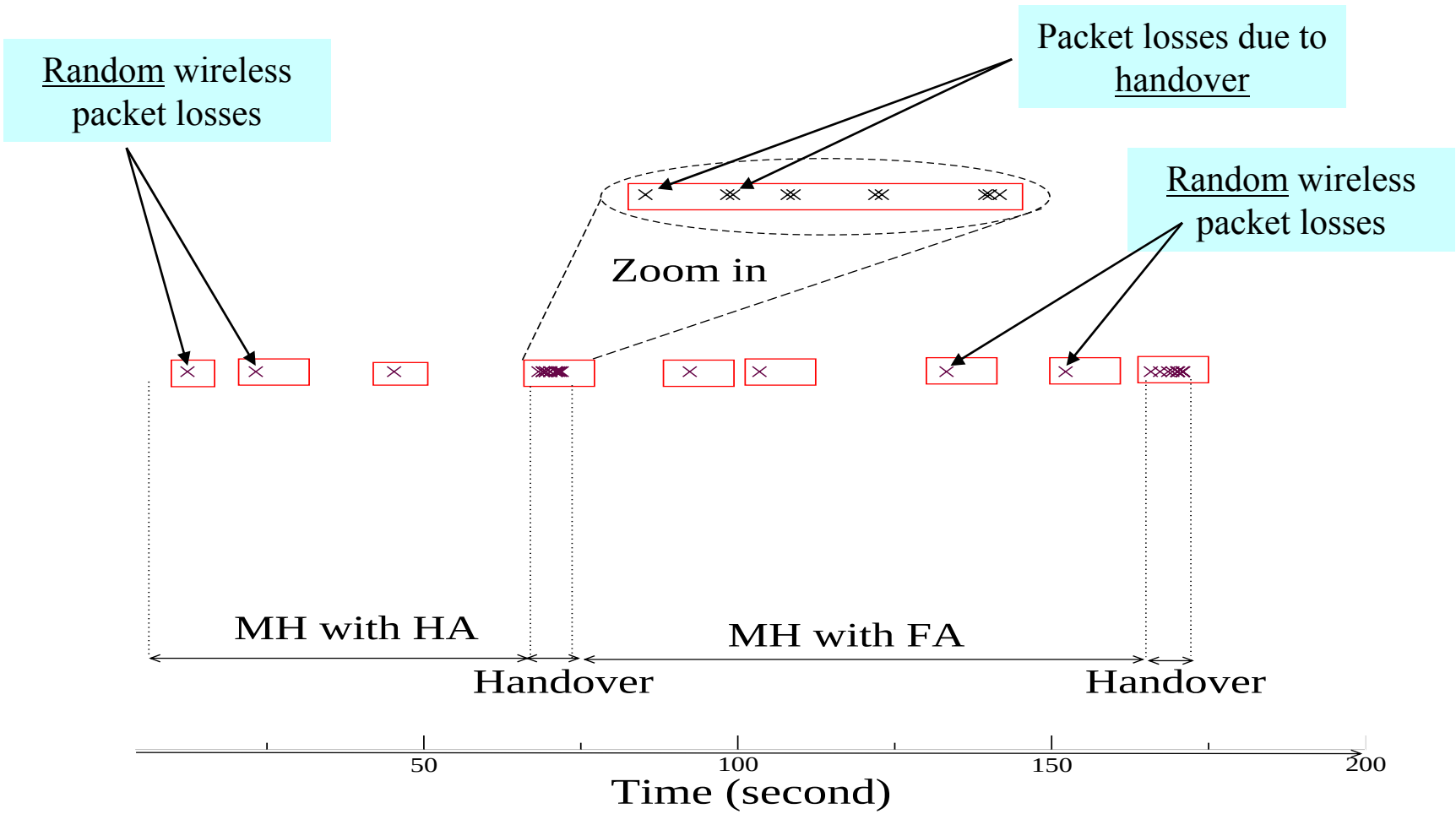
HA: Home Agent

FA: Foreign Agent

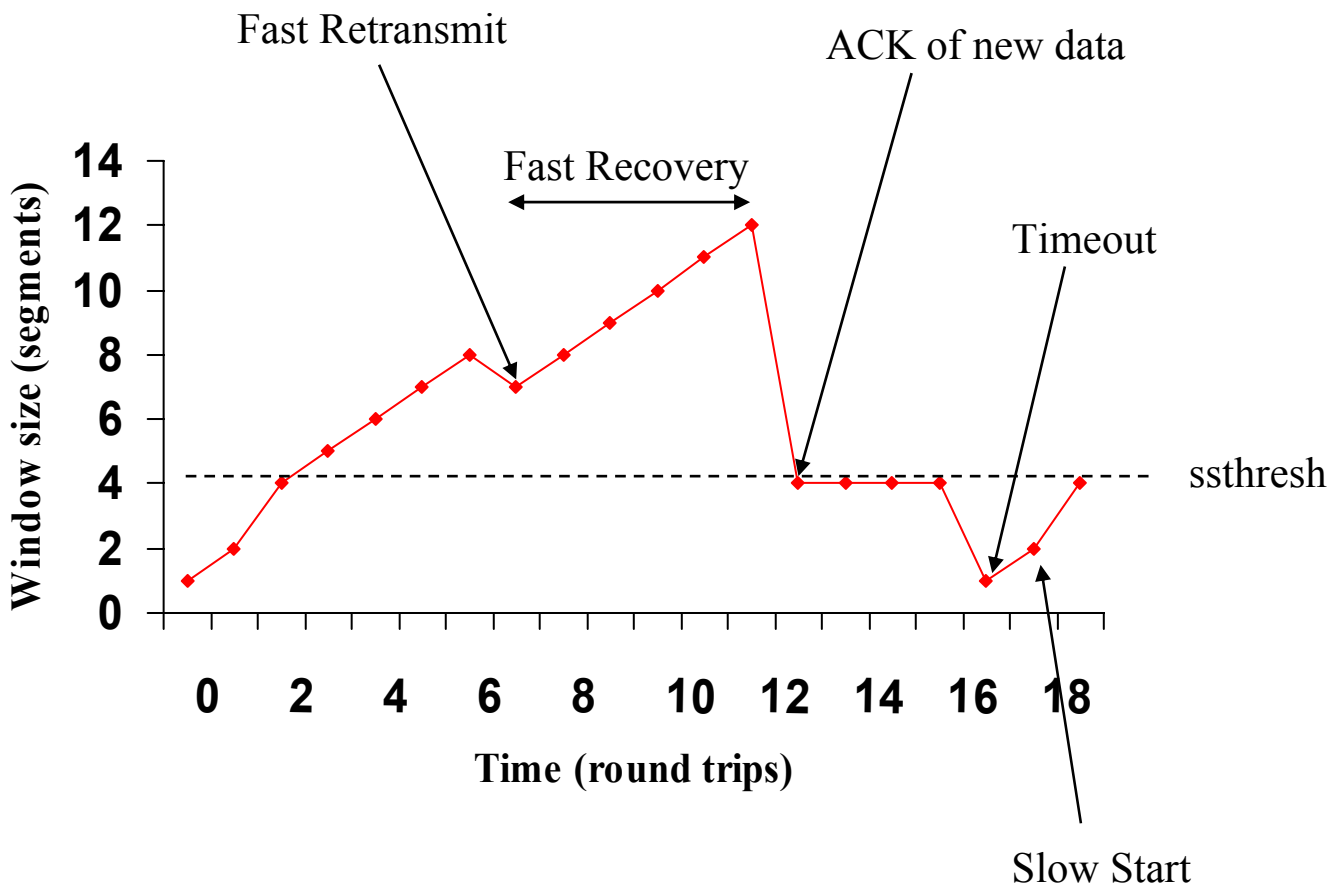
MH: Mobile Host



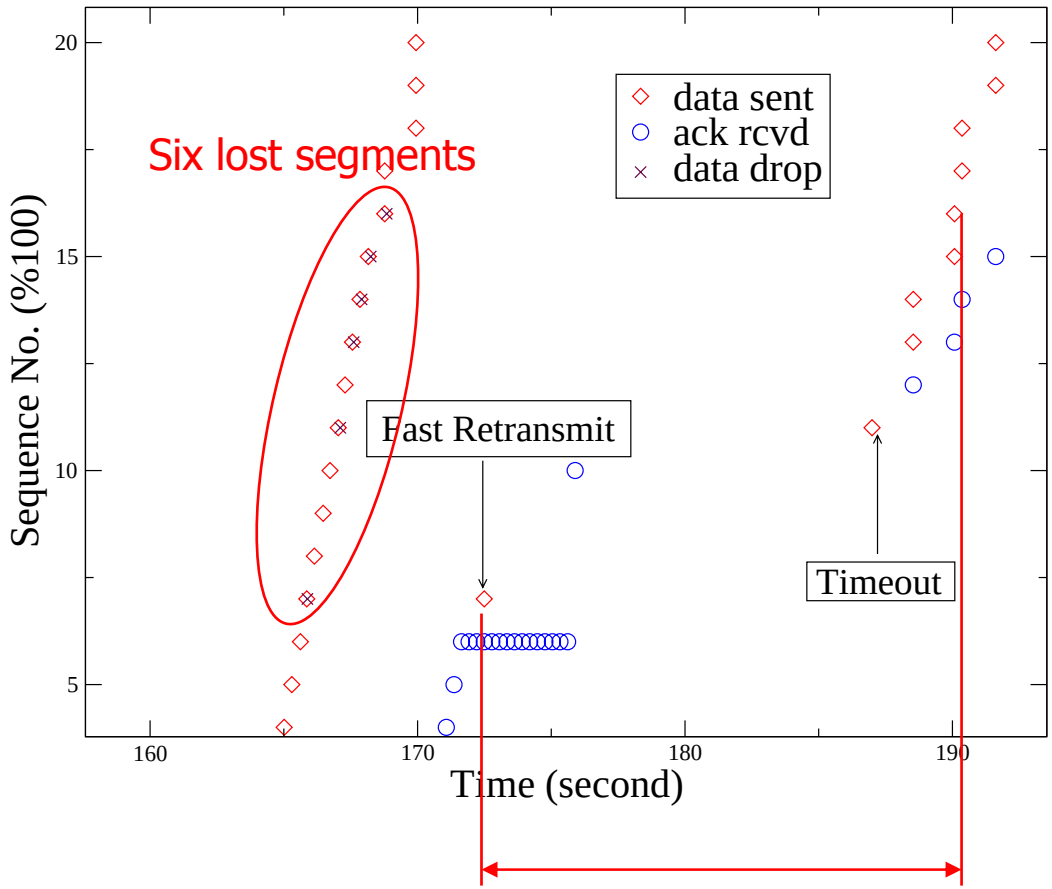
# Multiple non-consecutive packet losses during the handover



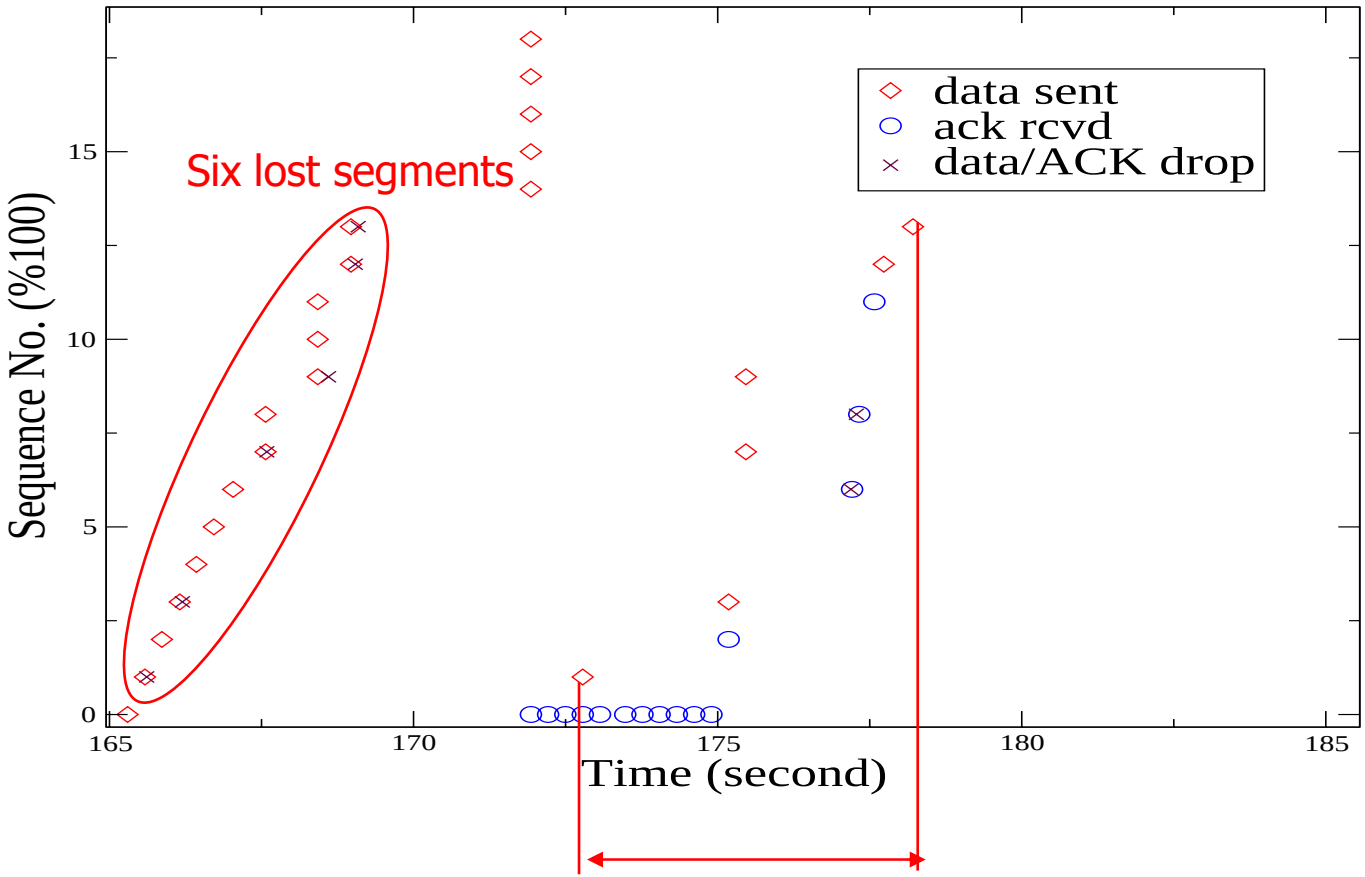
# Fast Retransmit and Fast Recovery



# Long recovery time of TCP Reno



# Loss Recovery of TCP SACK




reduced time to recover 6 segments compared to Reno (still about 6.5 secs)

# Objective & Contribution

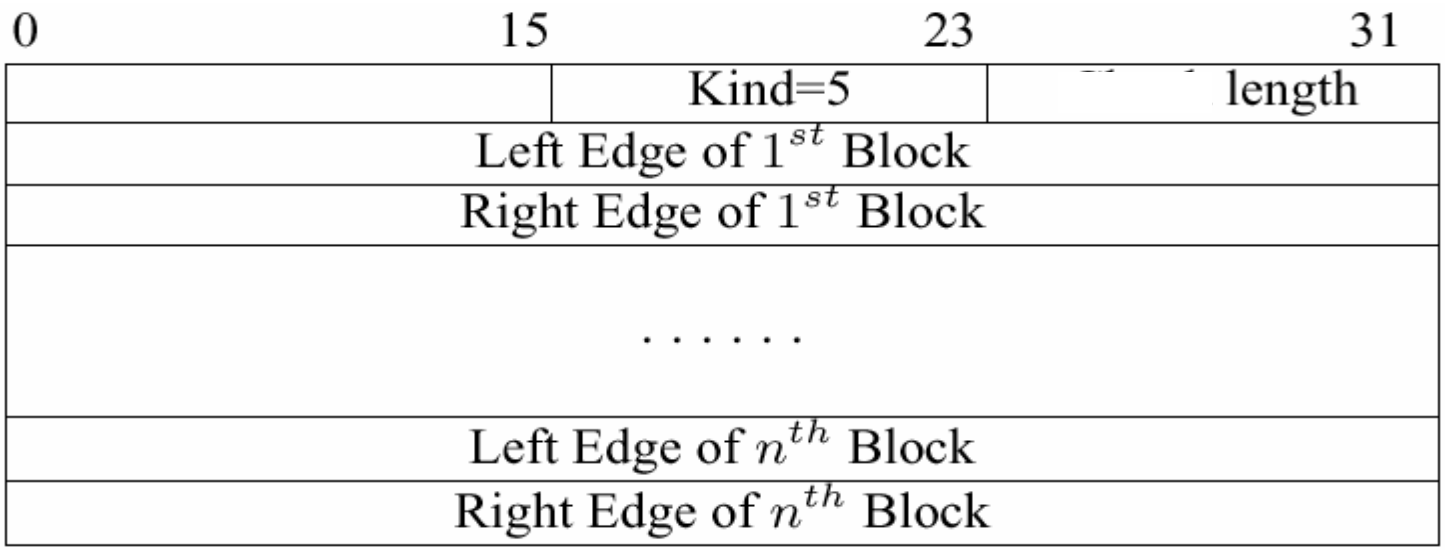
- To investigate whether SCTP's unlimited number of SACK blocks results in a higher throughput than that of TCP during handovers.

- Compare the throughput of TCP Reno, TCP-SACK, and SCTP during handovers for different network bandwidth.

# SCTP ---- A new transport protocol

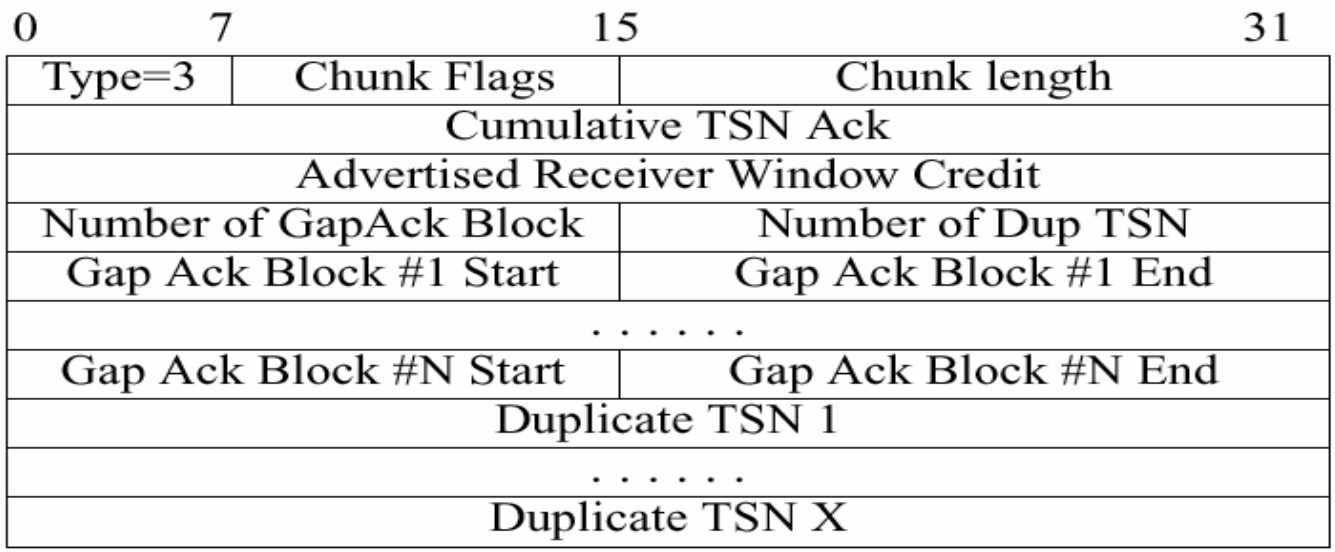
- Stream Control Transmission Protocol;
- Originally designed to support PSTN signaling messages over IP Networks;
- Standardized by IETF RFC 2960;
- Reliable transport protocol operating on top of IP (same level as TCP);
- Congestion control is similar to that of TCP.

# TCP SACK format



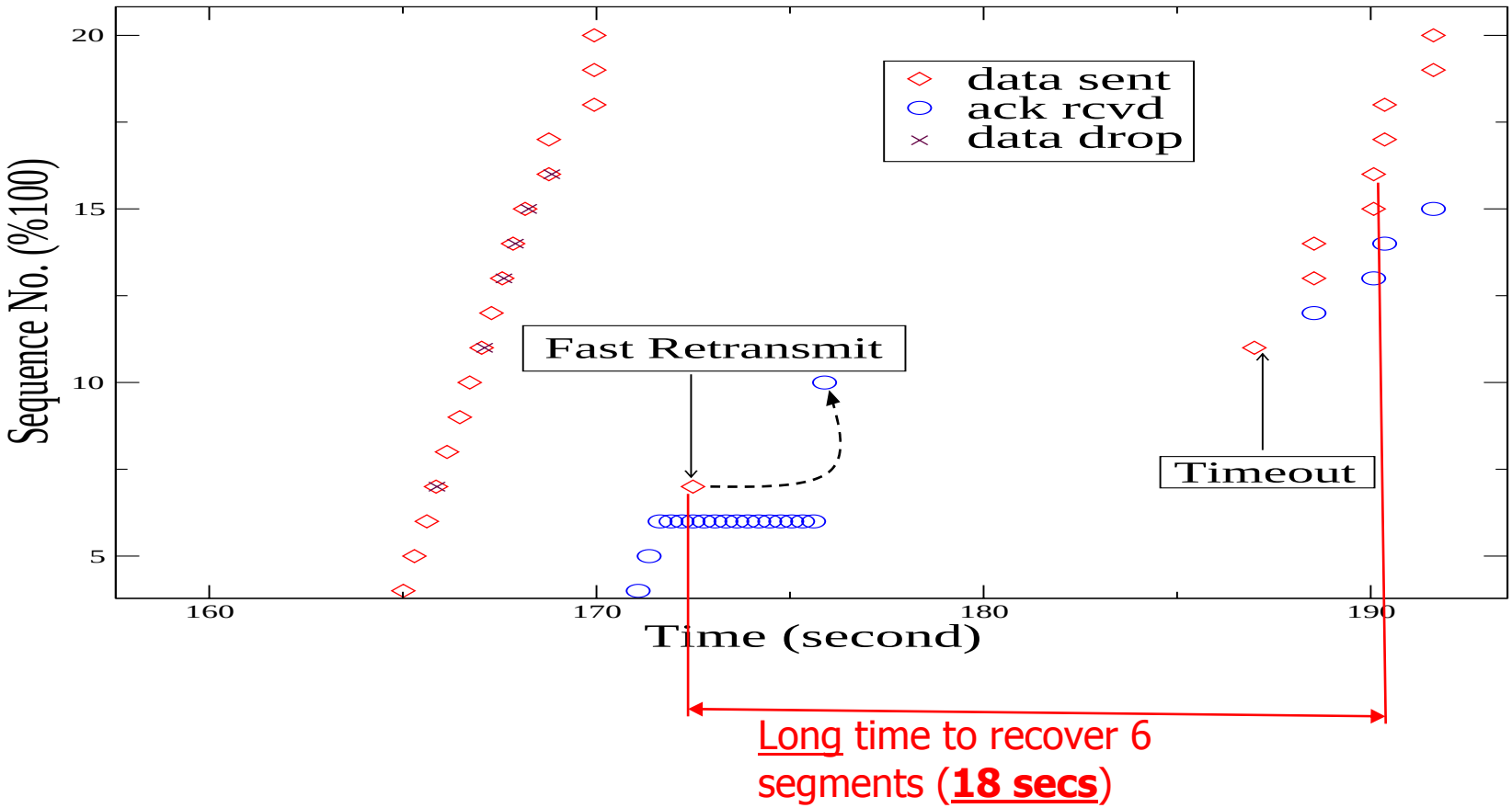

- TCP header length 4bit, which shows the number of 32bit word in the header;
- Maximum TCP header size: $(2^4 - 1)$ x (32/8) byte =60 byte;
- TCP header size without TCP options: 20 bytes;
- Length of the TCP *Options* field is limited to 40 bytes;
- *n* blocks requires a length of 8*n* +2 bytes;
- If SACK is used together with time-stamp option (requiring 12 bytes), the maximum SACK blocks allowed would be three.

# SCTP SACK format



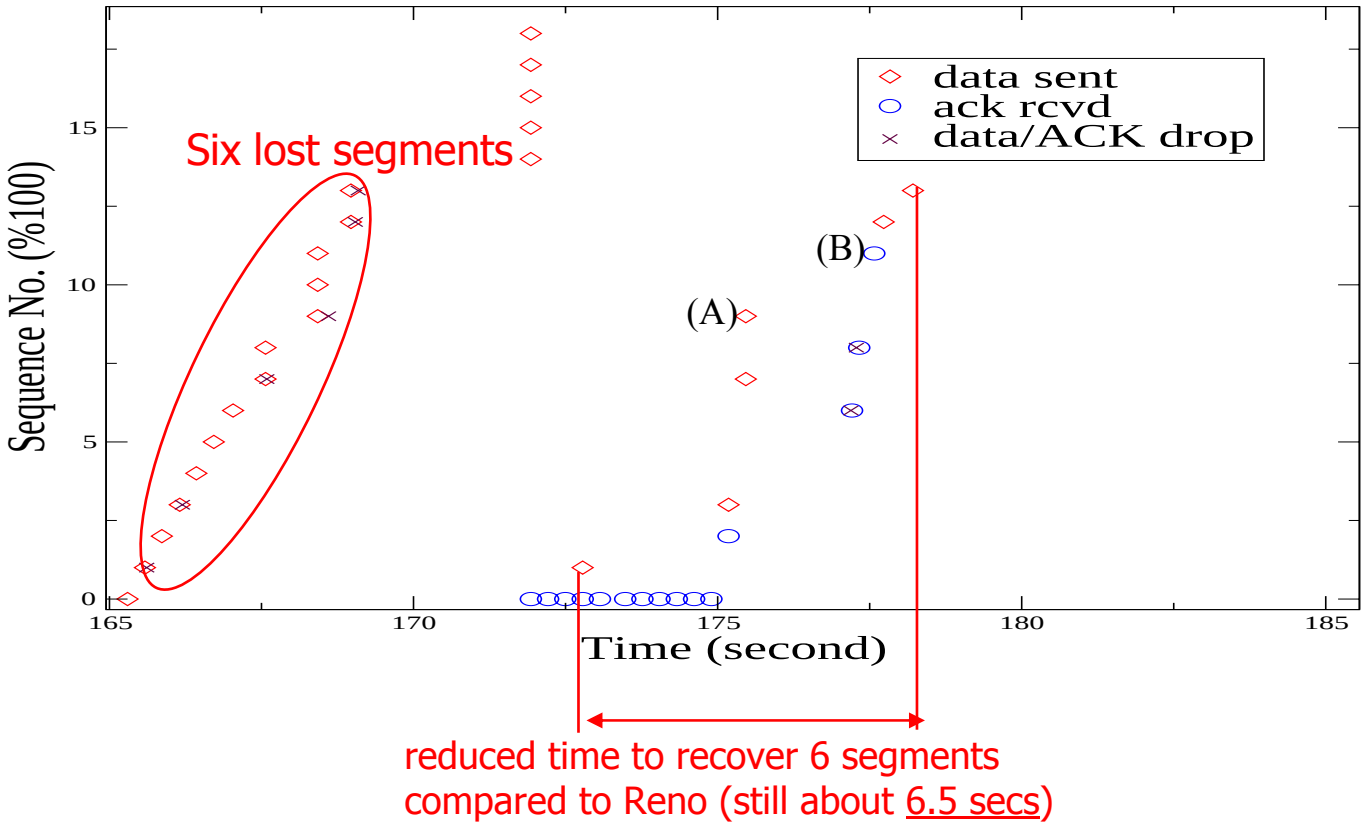

- Total available chunk space is determined by the “Chunk Length" field which is ($2^{16}$ -1) bytes;
- 16 bytes required for description of a SACK chunk ( first four rows );
- Every Gap Ack block needs 4 bytes;
- total number of blocks allowed is 16380.

# Segment loss recovery by TCP Reno



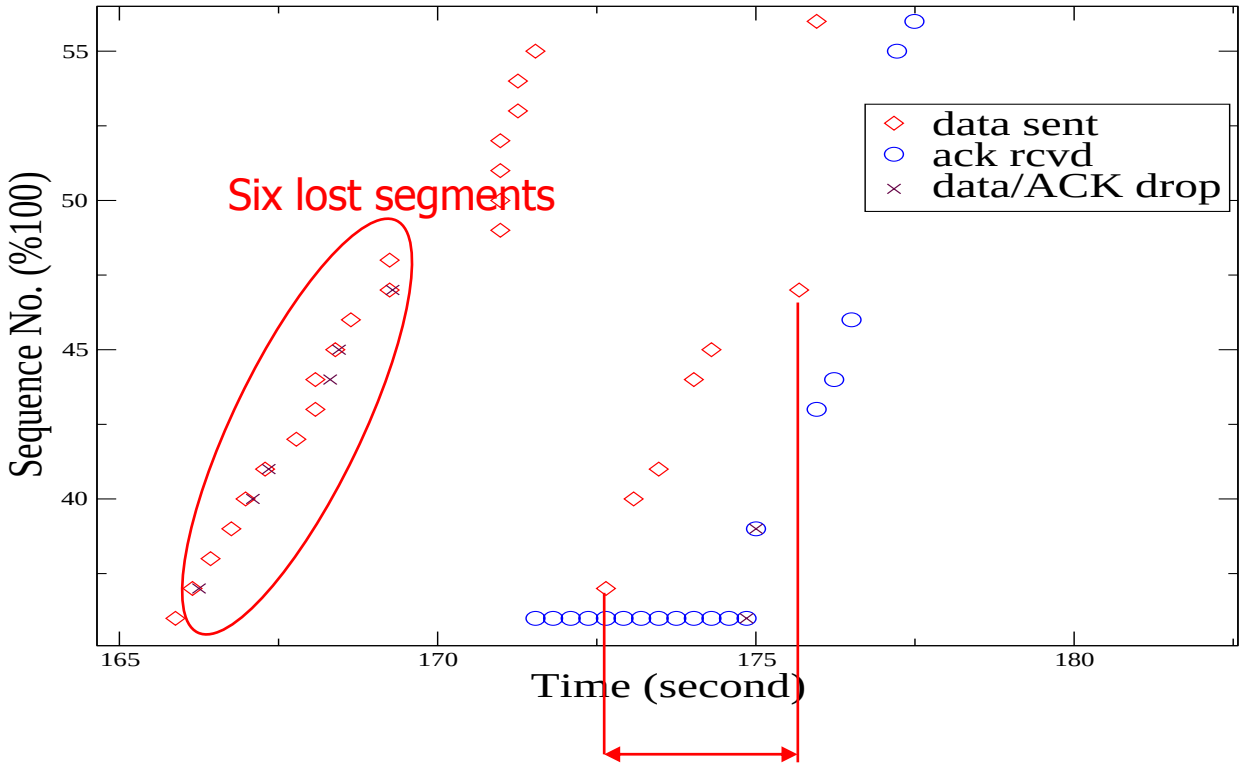

- After fast retransmit, TCP Reno must wait until another timeout to retransmit the remaining lost segments. This is because the flight size > cwnd.

# Segment loss recovery by TCP SACK




- SACK allows flight size to be reduced faster -- TCP sender does not encounter timeout.
- Due to the limitation of three SACK blocks, the sender can't reduce the flight size any more after it retransmits segment No. 9 (point A) and have to wait for new acknowledgment (point B).

# Segment loss recovery by SCTP



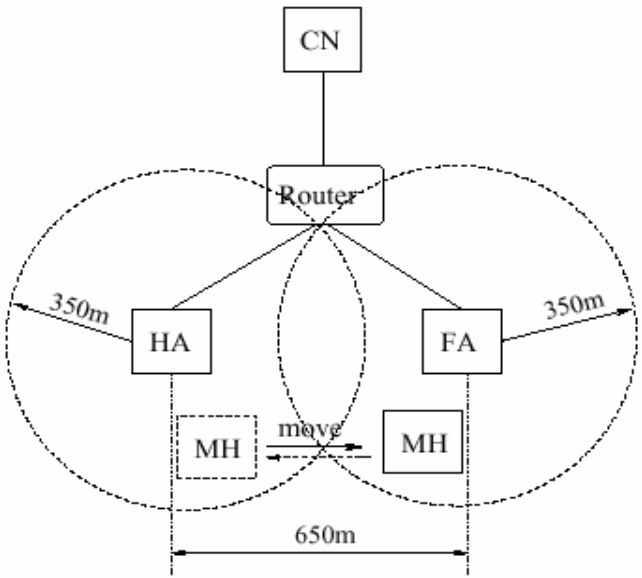

reduced time to recover 6 segments compared to TCP Reno, TCP SACK (about 3.5 secs)

- SCTP is not limited by the number of available SACK blocks, the sender can reduce the flight size more accurately to reflect the actual segments that have left the network, and the sender's window can be opened faster than in TCP-SACK.

# Results

- Performance criteria
  - **Throughput** = Number of segments received at the receiver during a period of time.
  - **Throughput gain:**
    - ( T(SCTP) - T(Reno) ) / T(Reno)
    - ( T(SCTP) - T(SACK) ) / T (SACK)
- Simulation setup
  - ns-2 simulator ver 2.1b8
  - SCTP patch from University of Delaware
  - Wireless extension to ns (from ACIRI) to extend Mobile IP module to overlapping cells.
  - Ftp was used to send data from the MH to CN for 200 secs.
  - Wireless error depends on distance between MH and HA/FA.

# Simulation topology and parameters




Topology

| Links | Link Bw (Kbps) | Prop. delay (ms) |
|---|---|---|
| CN-Router | 28.8-1500 | 200 |
| HA-Router | 500 | 200 |
| FA-Router | 500 | 200 |
| MH-HA/FA | 150-2000 | vary by distance |

Link parameters

# Throughput

## TCP Reno

| Wireline Bw. (Kbps) | Wireless Bw.(Kbps) | | | | |
|---|---|---|---|---|---|
| | *150* | *200* | *384* | *1000* | *2000* |
| *28.8* | **287** | **316** | **451** | **544** | 566 |
| *54* | **283** | **489** | **774** | **971** | 1010 |
| *200* | **373** | **487** | **1205** | 2148 | 2376 |
| *500* | **300** | **616** | 1399 | 2492 | 2723 |
| *1000* | **419** | 680 | 1626 | 2278 | 2695 |
| *1500* | 519 | 717 | 1224 | 2660 | 2715 |

## TCP Reno with SACK

| Wireline Bw. (Kbps) | Wireless Bw.(Kbps) | | | | |
|---|---|---|---|---|---|
| | *150* | *200* | *384* | *1000* | *2000* |
| *28.8* | **323** | **391** | **455** | **563** | 598 |
| *54* | **431** | **560** | **934** | **973** | 1046 |
| *200* | **478** | **717** | **1262** | 2528 | 2501 |
| *500* | **430** | **793** | 1472 | 2614 | 2826 |
| *1000* | **468** | 625 | 1643 | 2672 | 2957 |
| *1500* | 567 | 727 | 1816 | 2750 | 2796 |

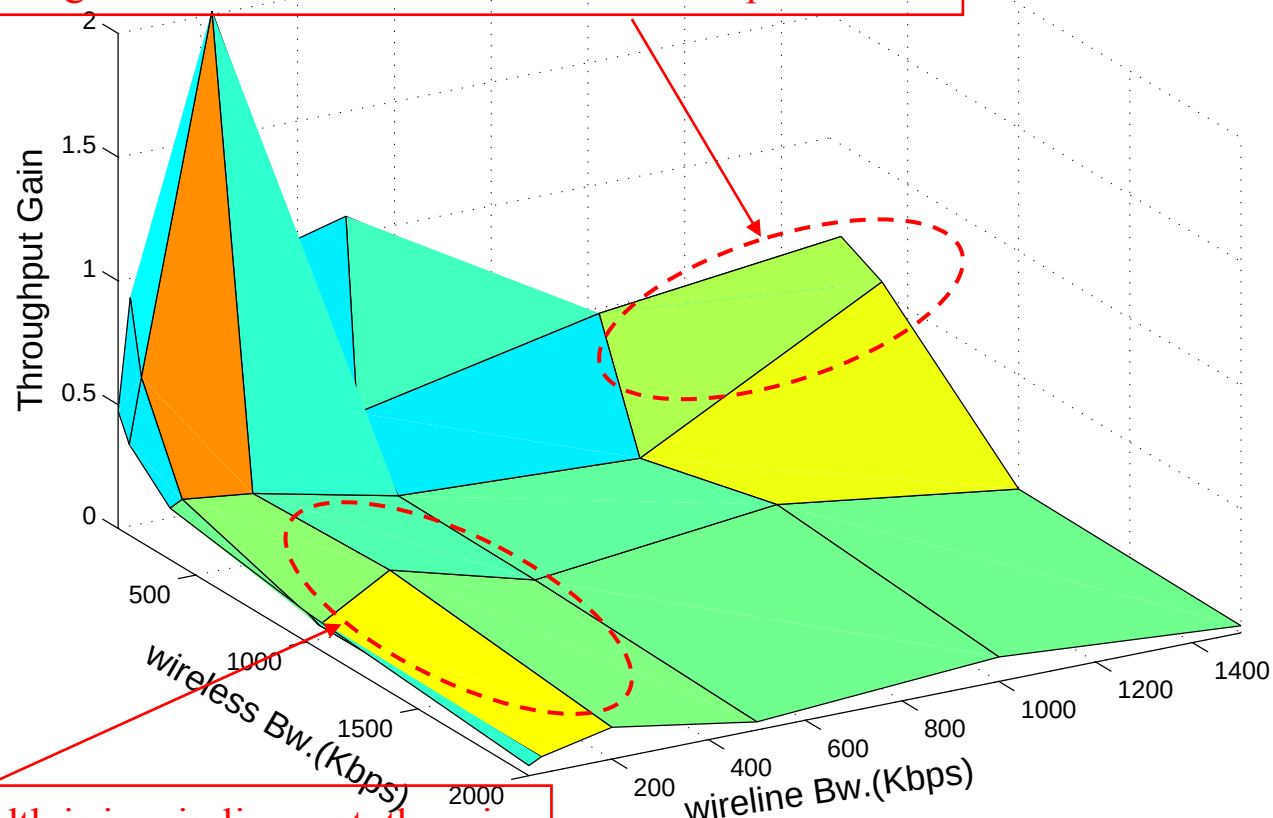
## SCTP

| Wireline Bw. (Kbps) | Wireless Bw.(Kbps) | | | | |
|---|---|---|---|---|---|
| | *150* | *200* | *384* | *1000* | *2000* |
| *28.8* | **423** | **432** | **545** | **609** | 589 |
| *54* | **544** | **796** | **955** | **1028** | 1077 |
| *200* | **716** | **1485** | **1446** | 2629 | 2681 |
| *500* | **623** | **806** | 1502 | 2657 | 2812 |
| *1000* | **605** | 1030 | 1672 | 2676 | 2963 |
| *1500* | 685 | 1168 | 1892 | 2768 | 2793 |

Throughput improvement of SACK is more prominent at low bottleneck bandwidths

# Throughput gain of SCTP over TCP-Reno

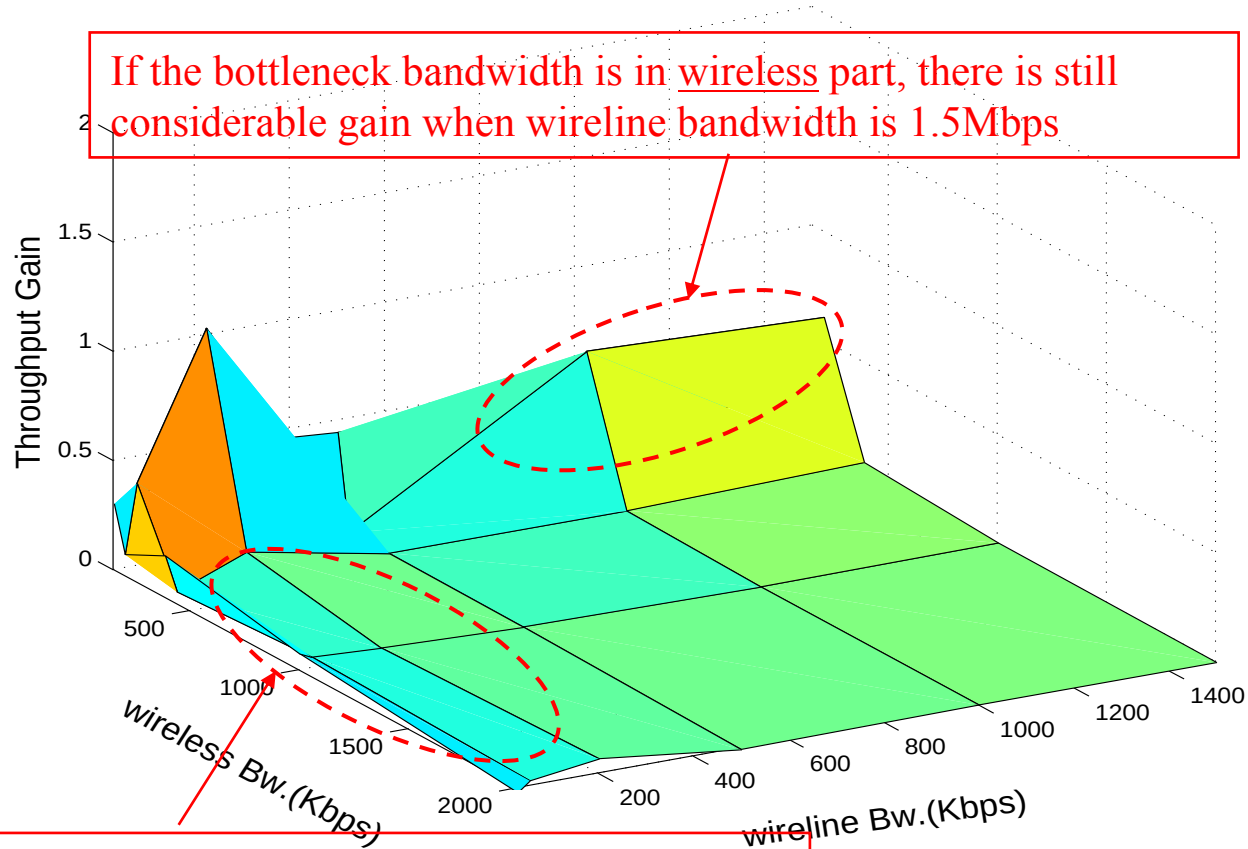
If the bottleneck bandwidth is in wireless part, there is still considerable gain when wireline bandwidth is 1.5Mbps




If the bottleneck bandwidth is in wireline part, there is small gain when wireless bandwidth get over 500Kbps

# Throughput gain of SCTP over TCP-SACK

If the bottleneck bandwidth is in wireless part, there is still considerable gain when wireline bandwidth is 1.5Mbps




If the bottleneck bandwidth is in wireline part, there is small gain when wireless bandwidth get over 500Kbps

# Gain Sensitivity to wireless bandwidth

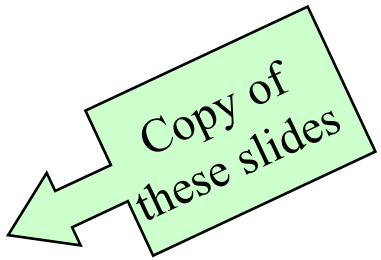
- **Throughput gain** is more sensitive to wireless bandwidth changes. For example:
  - if the bottleneck bandwidth is in wireline part, there is small gain when wireless bandwidth get over 500Kbps;
  - If the bottleneck bandwidth is in wireless part, there is still considerable gain when wireline bandwidth is 1.5Mbps.

# Conclusions

- SCTP benefits from large number of SACK blocks
  - higher throughput than TCP-Reno and TCP-SACK.
- The performance (throughput) improvement of SCTP is more when the bottleneck link bandwidth (wireline or wireless) is low.
- Throughput improvement is more sensitive to wireless bandwidth changes than to wireline bandwidth change.

# Further Information

Mohammed Atiquzzaman,
Web: **www.cs.ou.edu/~atiq**
Email: atiq@ou.edu,
Tel: (405) 325 8077, Fax: (405) 325 4044.




Thank you!