97-0423
Selective Acknowledgements and UBR+ Drop Policies to Improve TCP/UBR Performance over Terrestrial and Satellite Networks

Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy, Bobby Vandalore, Xiangrong Cai
The Ohio State University

Seong-Cheol Kim, Samsung Electronics Co. Ltd.

Sastri Kota, Lockheed Martin Telecom/Astrolink

http://www.cis.ohio-state.edu/~jain/

- Selective Acknowledgements
- A recent modification to FRR (New Reno)
- When are these useful? How much?
- LAN, WAN, Satellite Simulation Results
- A Problem in TCP Slow Start Implementations

Past Work

- TCP over
  - UBR
  - UBR + EPD
  - UBR + EPD + Selective Drop
  - UBR + EPD + Fair Buffer Allocation
  - All of the above + Fast Retransmit and Recovery (FRR)

Policies

End-System Policies

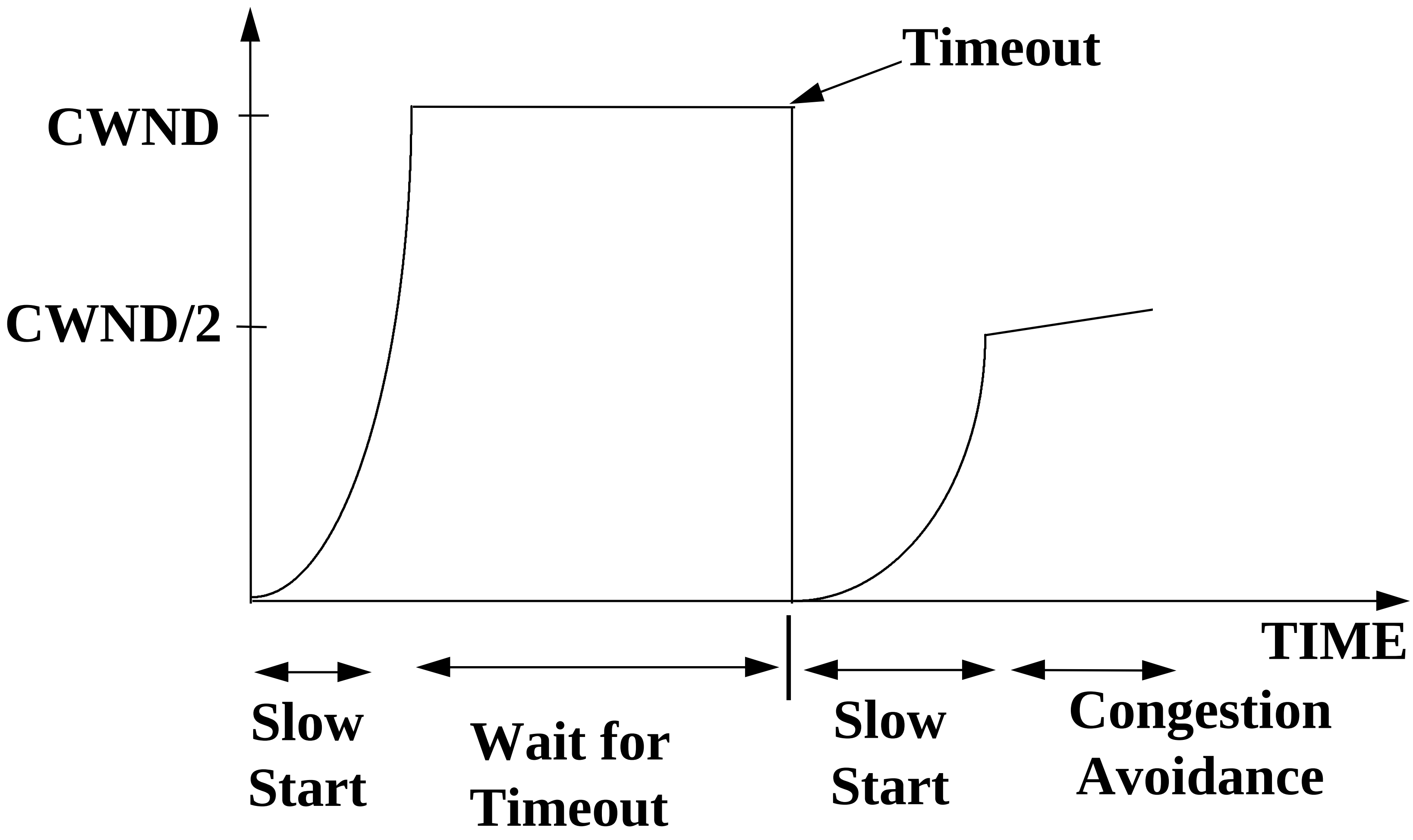
Switch Policies

		No FRR	FRR	New Reno	SACK + New Reno
No EPD					
EPD	Plain EPD				
	Selective Drop				
	Fair Buffer Allocation				

Slow Start

- Congestion Window (CWND) and Receiver Window
- Slow Start Threshold
  SSThresh = 0.5 × Congestion Window
- Exponential increase (Slow Start)
- Linear increase (Congestion Avoidance)
- Horizontal line = Timer granularity of 100 to 500 ms

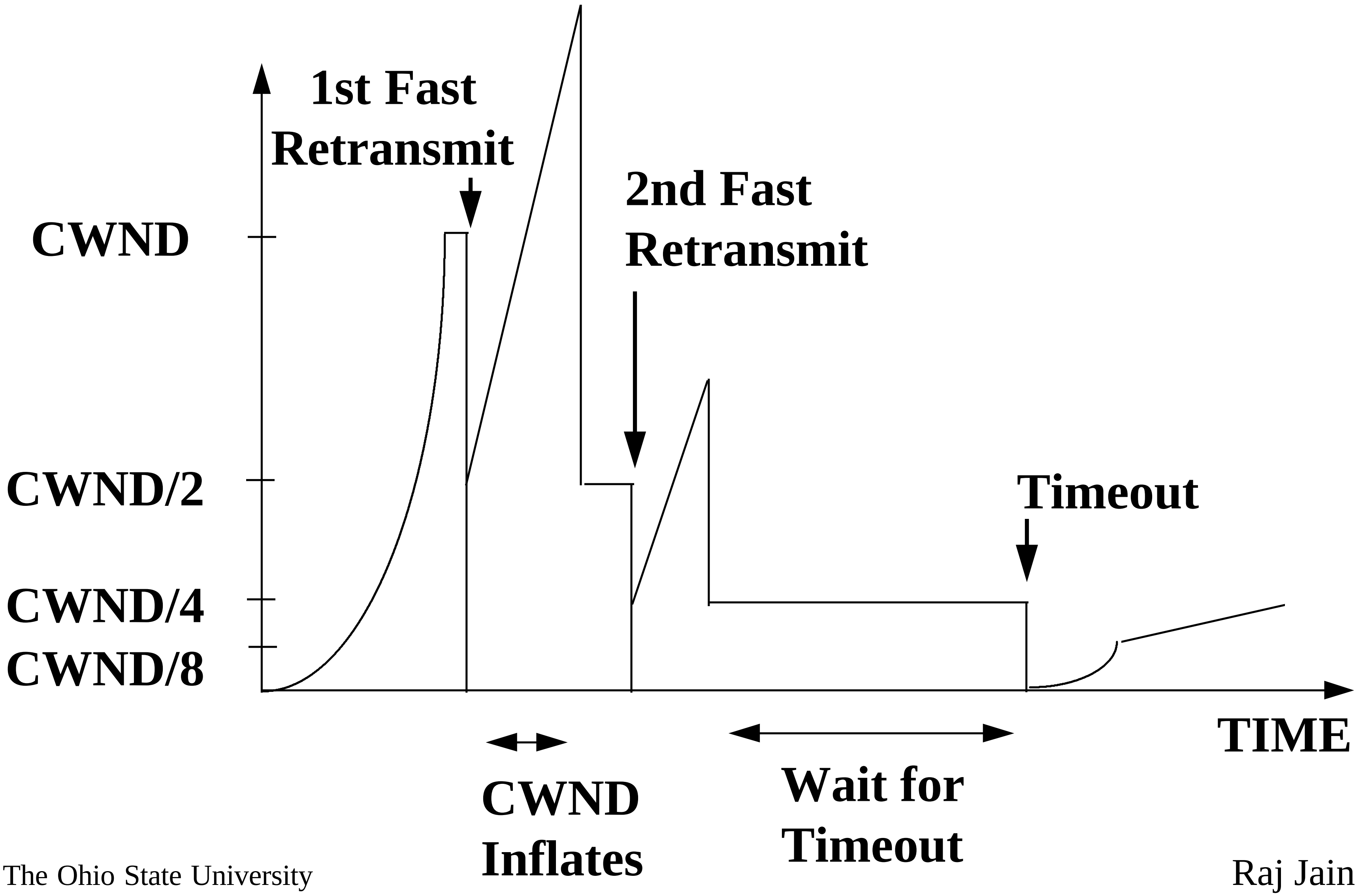
Slow Start (cont)



Fast Retransmit and Recovery

- Ideas:
  - Don't have to wait for timeout on a loss
  - Don't reduce on single loss due to error
  - Duplicate acks ⇒ Loss
- On three duplicate acks
  - Set SSThresh to 0.5 × CWND
    ⇒ Linear increase from now on
  - Reduce CWND to 0.5 × CWND + 3 (instead of 1)
- For each subsequent duplicate ack, inflate CWND by 1 and send a packet if permitted
- Problem with FRR:
  Cannot recover from bursty (3+) losses

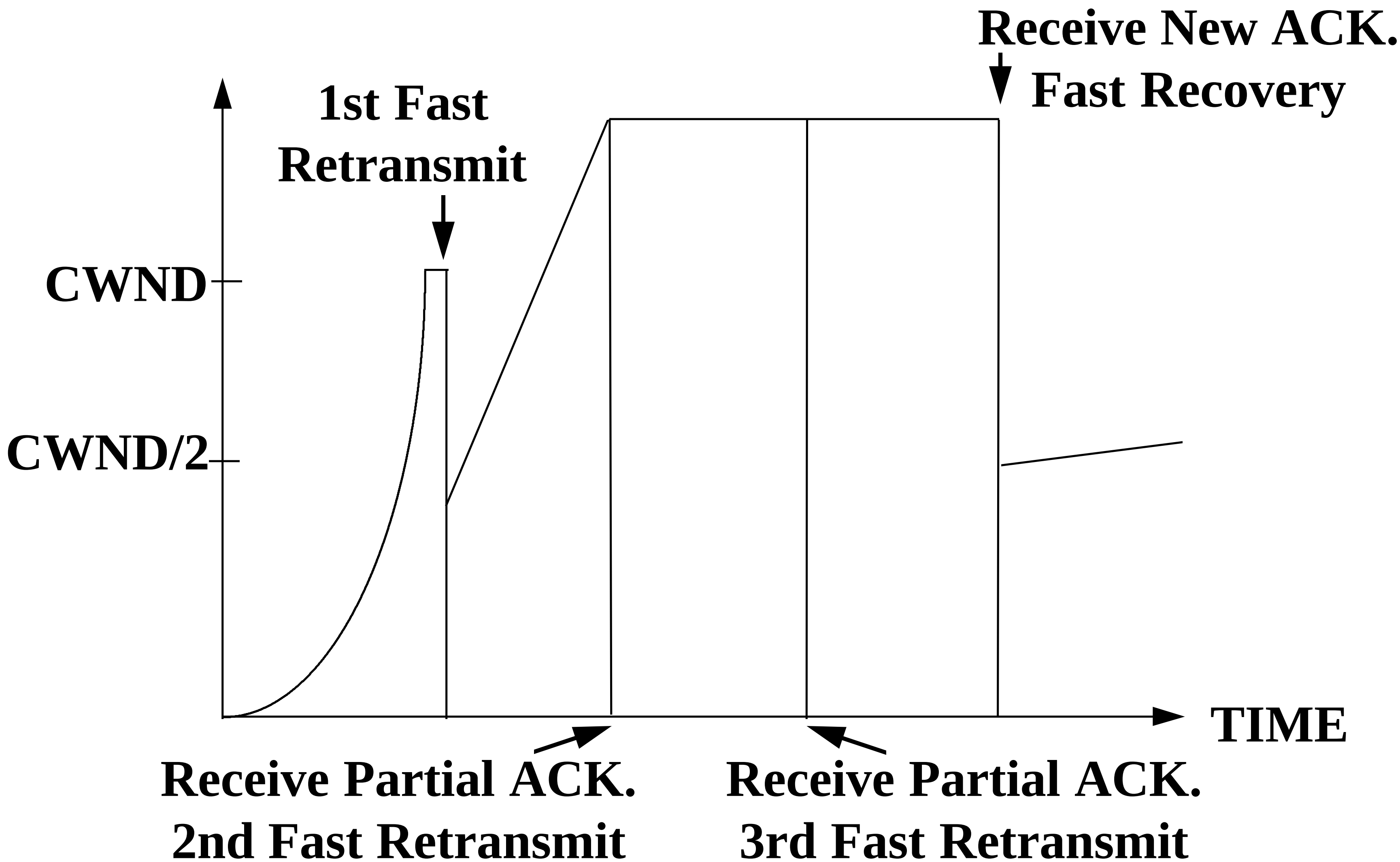
FRR (Cont)


1st Fast
Retransmit
CWND
2nd Fast
Retransmit
CWND/2
Timeout
CWND/4
CWND/8
TIME
CWND
Inflates
Wait for
Timeout

New Reno

- Janey Hoe's MS Thesis from MIT
  Published in SIGCOMM'96
- Solution: Determine the end-of a burst loss
  Remember the highest segment sent (RECOVER)
  Ack < RECOVER $\Rightarrow$ Partial Ack
  Ack $\geq$ RECOVER $\Rightarrow$ New Ack
- New Ack $\Rightarrow$ Linear increase from $0.5 \times$ CWND
- Partial Ack $\Rightarrow$ Retransmit next packet,
  let window inflate
- Recovers from N losses in N round trips

New Reno (Cont)



Selective Ack

- RFC 2018, October 1996
- Receivers can indicate missing segments
- Example:
  Using Bytes: Ack 500, SACK 1000-1500, 2000-2500
  ⇒ Rcvd segment 1, lost 2, rcvd 3, lost 4, rcvd 5
- On a timeout, ignore all SACK info
- SACK negotiated at connection setup
- Used on all duplicate acks

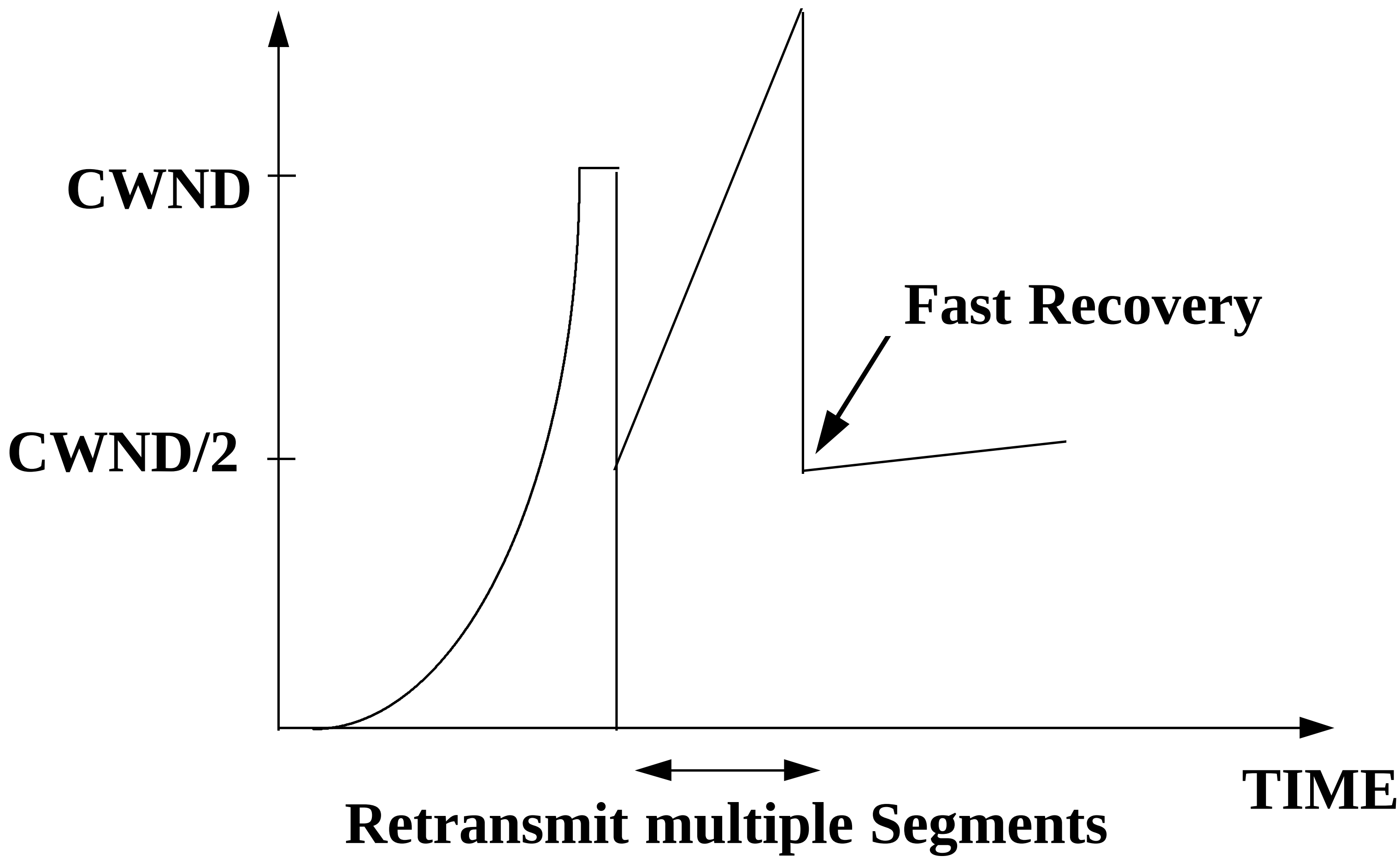
0-499	500-999	1000-1499	1500-1999	2000-2499
	Lost		Lost	

SACK with New Reno

- On 3 duplicate acks,
  retransmit the missing segment
- Then if permitted,
  retransmit the holes before new segments
- PIPE represents number of bytes on the path
- When FRR triggers, PIPE is set to CWND,
  then CWND is reduced to half
- On every duplicate ack, PIPE is reduced by 1
- Send new or retransmitted packet only if
  PIPE < CWND

- PIPE is incremented by 1 when a segment is sent
- PIPE is decremented by 2 when a "partial" ack is received

SACK (Cont)


CWND
CWND/2
Fast Recovery
TIME
Retransmit multiple Segments

Analytical Result

- SACK TCP can recover in one RTT from 1/4th window loss (Worst case)
- SACK TCP can recover from 1/*n*th window loss in $\lceil \log_2[n/(n-2)] \rceil$ RTTs, $n > 2$
  $\Rightarrow$ In k RTTs, recover from $2^{k+1}/(2^k - 1)$th of window
  $\Rightarrow$ 3/8th in 2 RTTs, 7/16th in 3 RTTs, 15/32th in 4 RTTs, ...
  $\Rightarrow$ Cannot recover from half or more window loss
- See contribution for derivation
- Assumption: Retransmitted segements are not lost.

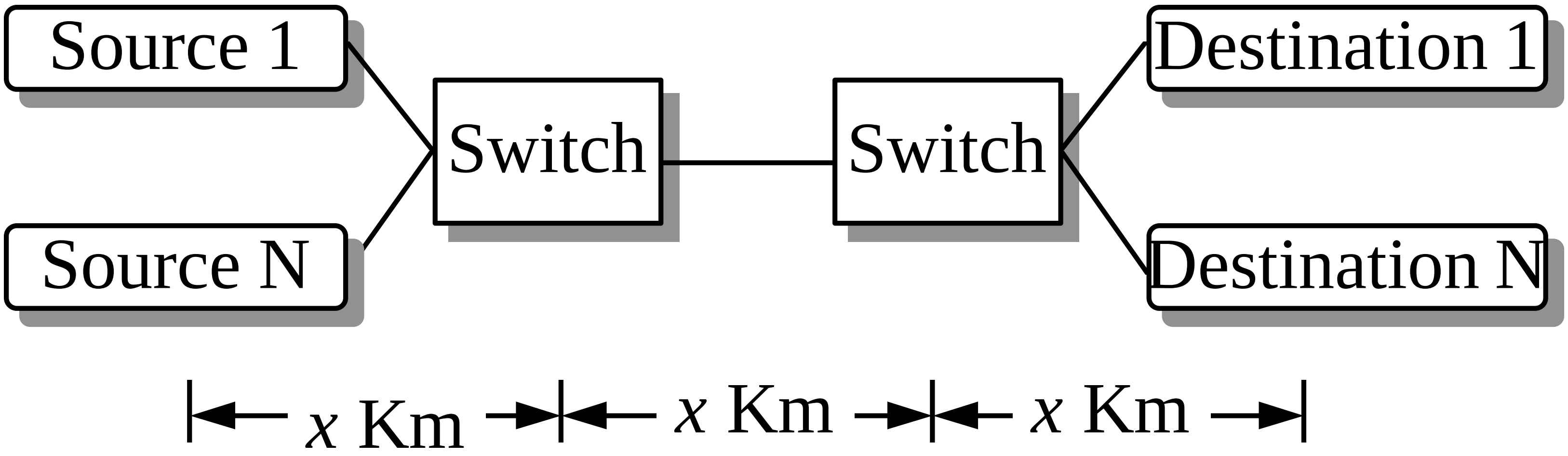
Past Results on TCP over UBR+

- Need buffers = Σ Windows
- Poor performance with limited buffers
- EPD improves efficiency but not fairness
- In high delay-bandwidth paths,
  too many packets are lost
  $\Rightarrow$ Effect of EPD reduces
  $\Rightarrow$ EPD has little effect in satellite networks.

Past Results (Cont)

- Selective drop (only above-average users punished) improves fairness and even efficiency.
- Fair buffer allocation (more sophisticated selective drop) improves fairness and efficiency more.
- FRR improves performance over LANs but <u>degrades</u> performance over WANs and Satellites

Simulation model



- N identical persistent TCP sources
- Link Delay: LAN: 5 μs, WAN: 5 ms.
- Link Capacity = PCR = 155.52 Mbps
- Unidirectional traffic

TCP Parameters

- MSS = 512 Bytes (LANs and WANs), 9180 (Satellites)
- Window = 64 K (LANs) 600,000 (WANs) 34000 × 8 (Satellites)
- Buffer sizes = 1k and 3k cells (LANs) 1 to 3 times RTT (WANs and Satellites)
- No TCP delay ack timer
- All processing delay, delay variation = 0
- TCP timer granularity = 100 ms

Performance Metrics

- Efficiency = Sum of throughputs/Max poss. throughput
  - Maximum Segment Size = 512 data
    = 512 data + 20 TCP + 20 IP + 8 LLC + 8 AAL5
    = 12 cells = 12×53 bytes = 636 bytes in ATM Layer
  - Maximum possible throughput = 512/636 = 80.5%
    = 125.2 Mbps on a 155.52 Mbps link
- Fairness = $$\frac{(\Sigma x_i)^2}{n \Sigma x_i^2}$$

  Where x_i = throughput of the *i*th TCP source

SACK TCP: Efficiency

Config-uration	# of Sources	Buffer (cells)	UBR	EPD	Selective Drop
LAN	5	1000	0.76	0.85	0.94
LAN	5	3000	0.98	0.97	0.98
LAN	15	1000	0.57	0.78	0.91
LAN	15	3000	0.86	0.94	0.97
SACK			0.79	0.89	0.95
Vanilla TCP			0.34	0.67	0.84
Reno TCP			0.69	0.97	0.97
WAN	5	12000	0.90	0.88	0.95
WAN	5	36000	0.97	0.99	1.00
WAN	15	12000	0.93	0.80	0.88
WAN	15	36000	0.95	0.95	0.98
SACK			0.94	0.91	0.95
Vanilla TCP			0.91	0.9	0.91
Reno TCP			0.78	0.86	0.81

SACK TCP: Fairness

Config-uration	# of Sources	Buffer (cells)	UBR	EPD	Selective Drop
LAN	5	1000	0.22	0.88	0.98
LAN	5	3000	0.92	0.97	0.96
LAN	15	1000	0.29	0.63	0.95
LAN	15	3000	0.74	0.88	0.98
SACK			0.54	0.84	0.97
Vanilla TCP			0.69	0.69	0.92
Reno TCP			0.71	0.98	0.99
WAN	5	12000	0.96	0.98	0.95
WAN	5	36000	1.00	0.94	0.99
WAN	15	12000	0.99	0.99	0.99
WAN	15	36000	0.98	0.98	0.96
SACK			0.98	0.97	0.97
Vanilla TCP			0.76	0.95	0.94
Reno TCP			0.90	0.97	0.99

Simulation Results

- In LANs, switch improvements (PPD, EPD, SD, FBA) have more impact than end-system improvements (Slow start, FRR, New Reno, SACK). Different variations of increase/decrease have little impact due to small window sizes.
- Previously retransmitted holes may have to be retransmitted on a timeout
  $\Rightarrow$ SACK can hurt under extreme congestion.

Simulation Results (Cont)

- **SACK is more helpful in WANs** (Whereas FRR hurts in WANs) due to multiple losses
- Switch-based improvements are helpful even with SACK
- Fairness depends largely on the drop policy and not so much on end-system policies

A Problem in Slow Start Implementations

- Linear Increase in Segments: CWND/MSS = CWND/MSS + MSS/CWND
- In Bytes: CWND = CWND + MSS*MSS/CWND
- All computations are done in integer
- If CWND is large, MSS*MSS/CWND is zero and CWND does not change. CWND stays at 512*512 or 256 kB.

Solutions

- **Solution 1**: Increment CWND after N acks (N > 1)
  CWND = CWND + N*MSS*MSS/CWND
- **Solution 2**: Use larger MSS on Satellite links such that MSS*MSS > CWND. MSS $\geq$ Path MTU.
- **Solution 3**: Use floating point
- **Recommendation**: Use solution 1. It works for all MSSs.

Satellite Networks: Efficiency

Config-uration	# of Sources	Buffer (cells)	UBR	EPD	Selective Drop
SACK	5	200000	0.86	0.6	0.72
SACK	5	600000	0.99	1.00	1.00
Reno	5	200000	0.84	0.12	0.12
Reno	5	600000	0.30	0.19	0.22
Vanilla	5	200000	0.70	0.73	0.73
Vanilla	5	600000	0.88	0.81	0.82

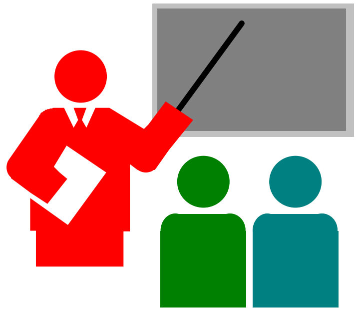
Satellite Networks: Fairness

Configuration	# of Sources	Buffer (cells)	UBR	EPD	Selective Drop
SACK	5	200000	1.00	0.83	0.94
SACK	5	600000	1.00	1.00	1.00
Reno	5	200000	0.96	0.97	0.97
Reno	5	600000	1.00	1.00	1.00
Vanilla	5	200000	1.00	0.87	0.89
Vanilla	5	600000	1.00	1.00	1.00

Simulation Results on Satellites

- SACK helps significantly
- FRR hurts badly
- Switch-based improvements have relatively less impact than end-system improvements
- Fairness is not affected by SACK

Summary



- In LANs, switch improvements (PPD, EPD, SD, FBA) have more impact than end-system improvements (Slow start, FRR, New Reno, SACK).
- In WANs and satellite networks, end-system improvements have more impact than switch-based improvements
- FRR hurts in WANs and satellite networks.
- Fairness depends upon the switch drop policies and not on end-system policies
- Unless implemented properly, congestion window may get stuck at 256 kB