TCP over Variable Capacity Links:
A Simulation Study for Bandwidth Allocation

Ian Marsh (SICS, KTH) & **Henrik Abrahamsson** (SICS)

Olof Hagsand (Dynarc AB)

Big Picture

- Reverse approach to statistical multiplexing, allocate bandwidth to individual flows
- If we can allocate exactly what is needed, significant bandwidth savings can be made
- Probably don’t want to allocate bandwidth for an individual web page access, better suited to large company backup for example
- A network technology which varies its capacity & TCP which varies its rate can lead to potential problems!

Goals

- Allocate bandwidth close to TCPs needs
- Investigate bandwidth allocation on a Dynamic Transfer Mode (DTM) network

Method

- Use ns-2
- Compare number of different scenarios of dynamic allocation with a static link case
- Number of plots coming up: static link cases have 2 plots (cwnd, rate) variable 3 (cwnd, rate, allocation)

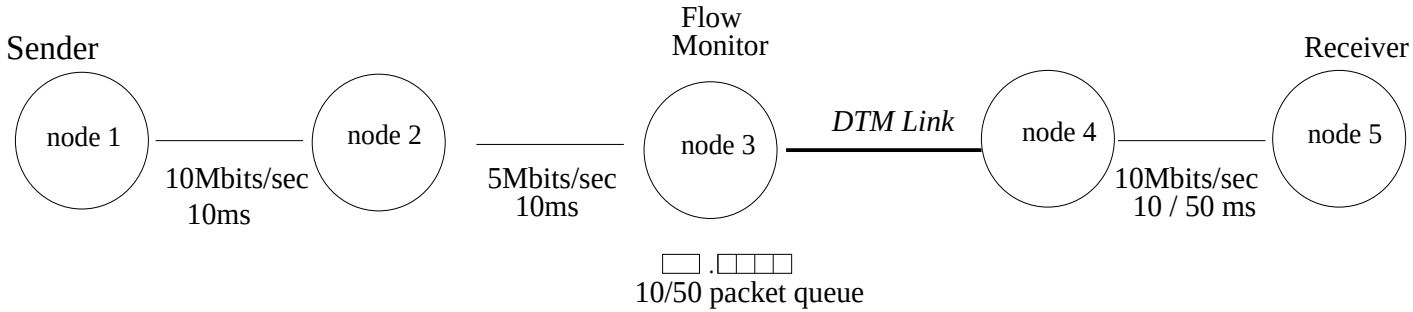
DTM in a slide

- Time Division Multiplexed (TDM) scheme at gigabits/sec
- Channel abstraction, containing a number of slots
- Number of slots determines the bandwidth
- 1 slot per channel is 512kbits/sec
- Slots statically or dynamically allocated
- Channels can have 1 or more TCP flows
- Sample flow bandwidth every 100ms (default)
- Bandwidth changes on 1 second boundaries (default)

Bandwidth Estimation

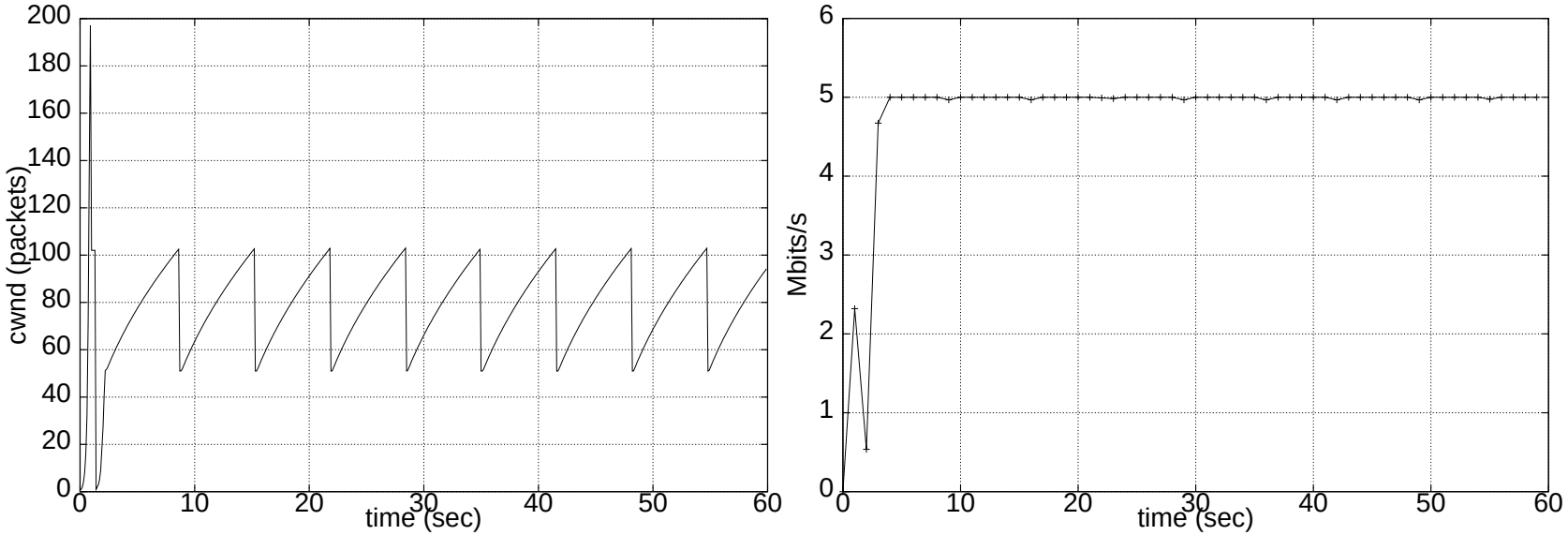
- Measure throughput of each flow
- Compare recently recorded value with old value
- When lower increase allocation by 1/8
- When higher decrease allocation by 1/8
- 1/8 default, higher/lower values more/less aggressive
- DTM works in slots (512kbits) so quantise to slots
- Pseudo code in paper

Experimental Set-up 1



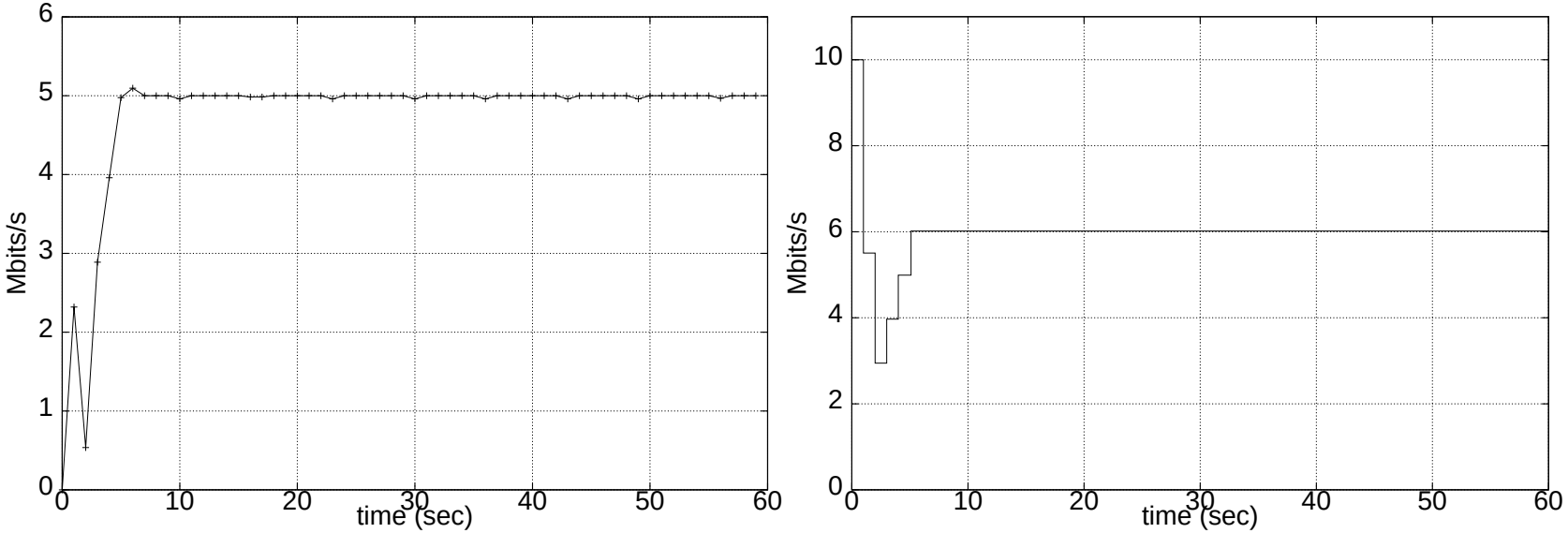
- Single Reno flow
- 40ms (80 RTT)
- 50 packet buffer in DTM access node
- 5Mbits bottleneck between nodes 2 and 3 (others 10Mbits)
- Look at static link case then variable link
- Goal: How allocation adjusts & if TCP is affected

Static Link, Reno, 50 pkt, 40ms delay



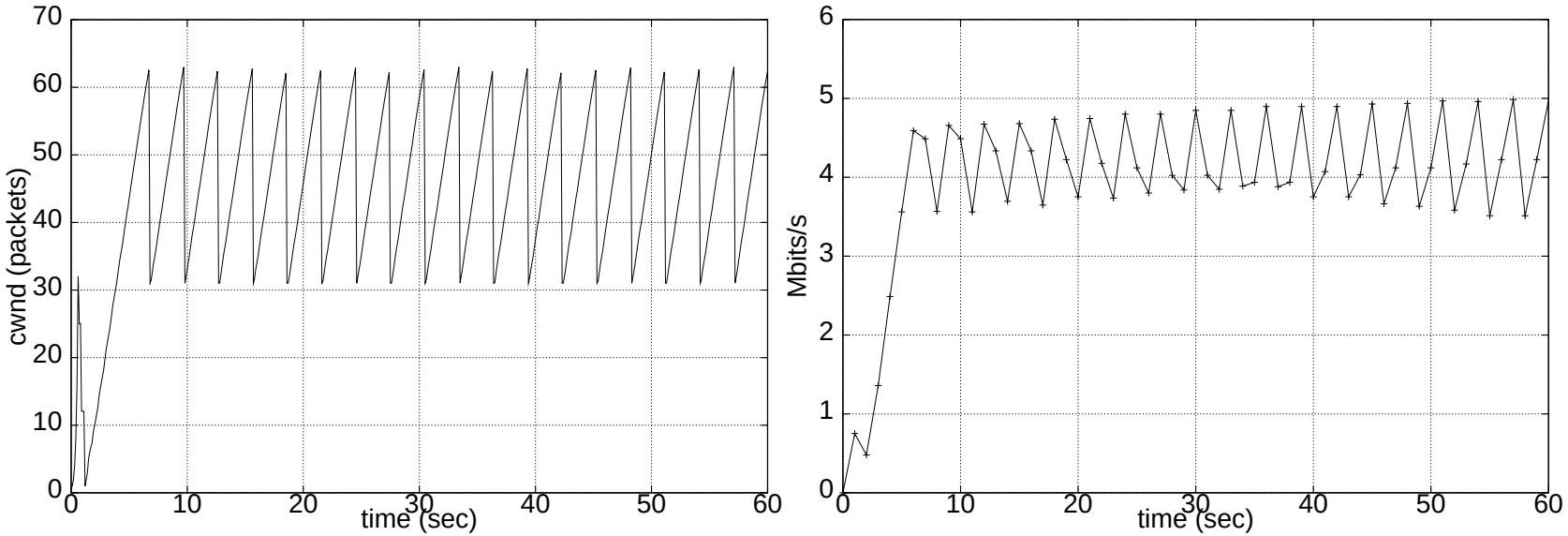
- Typical Reno saw-tooth behaviour
- TCP aggressively fills buffer at the bottleneck
- Throughput constant due to 50 packet buffer (see later)

Variable Link, Reno, 50 pkt, 40ms delay



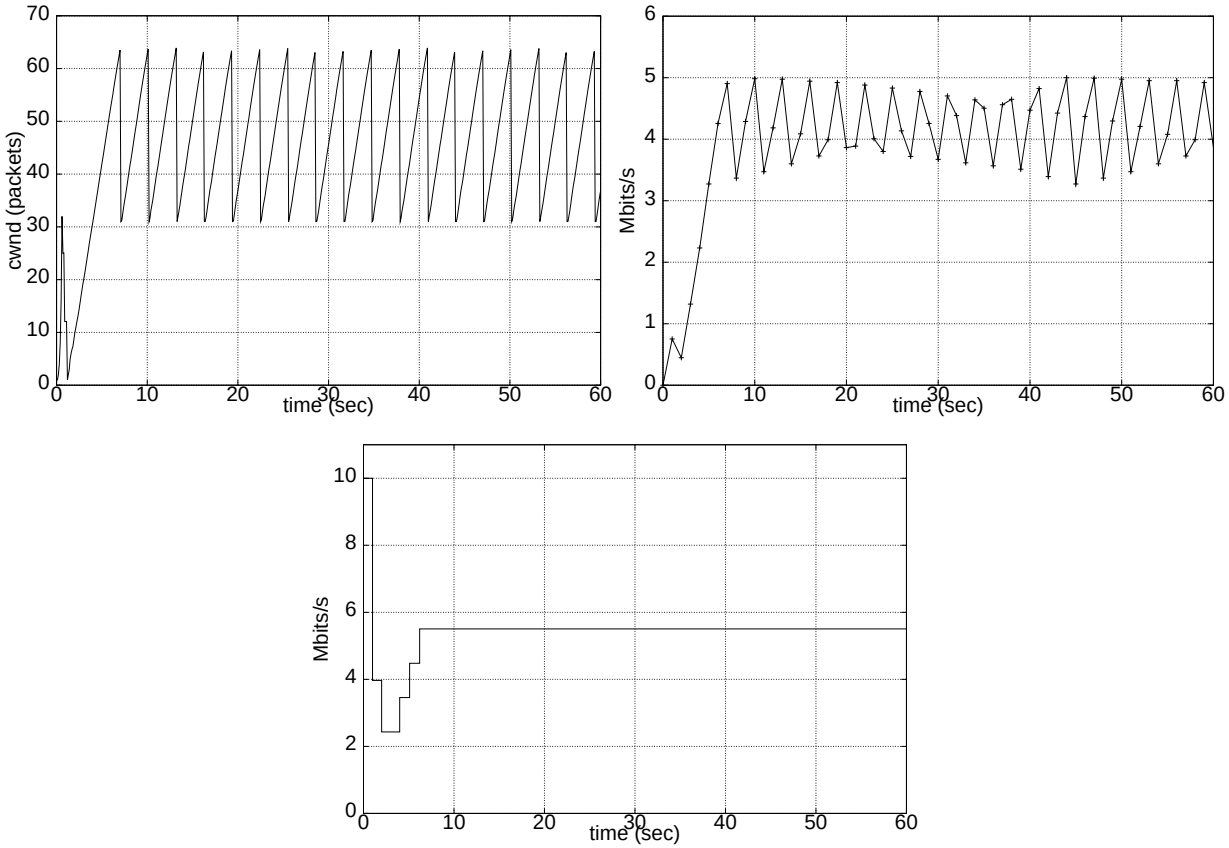
- Throughput & DTM allocation shown (cong. win same)
- Starts at 10Mbits then steady at 6 Mbits (due to margin)

Static Link, Reno, 10 pkt, 40ms delay



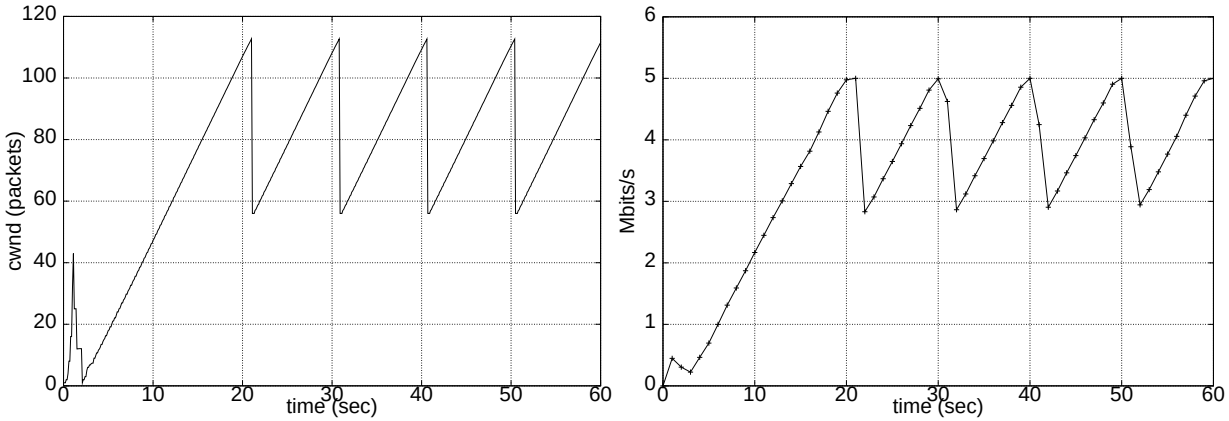
- Reduce buffer length in DTM node to 10 packets
- Lose packets more frequently

Variable Link, Reno, 10 pkt, 40ms delay



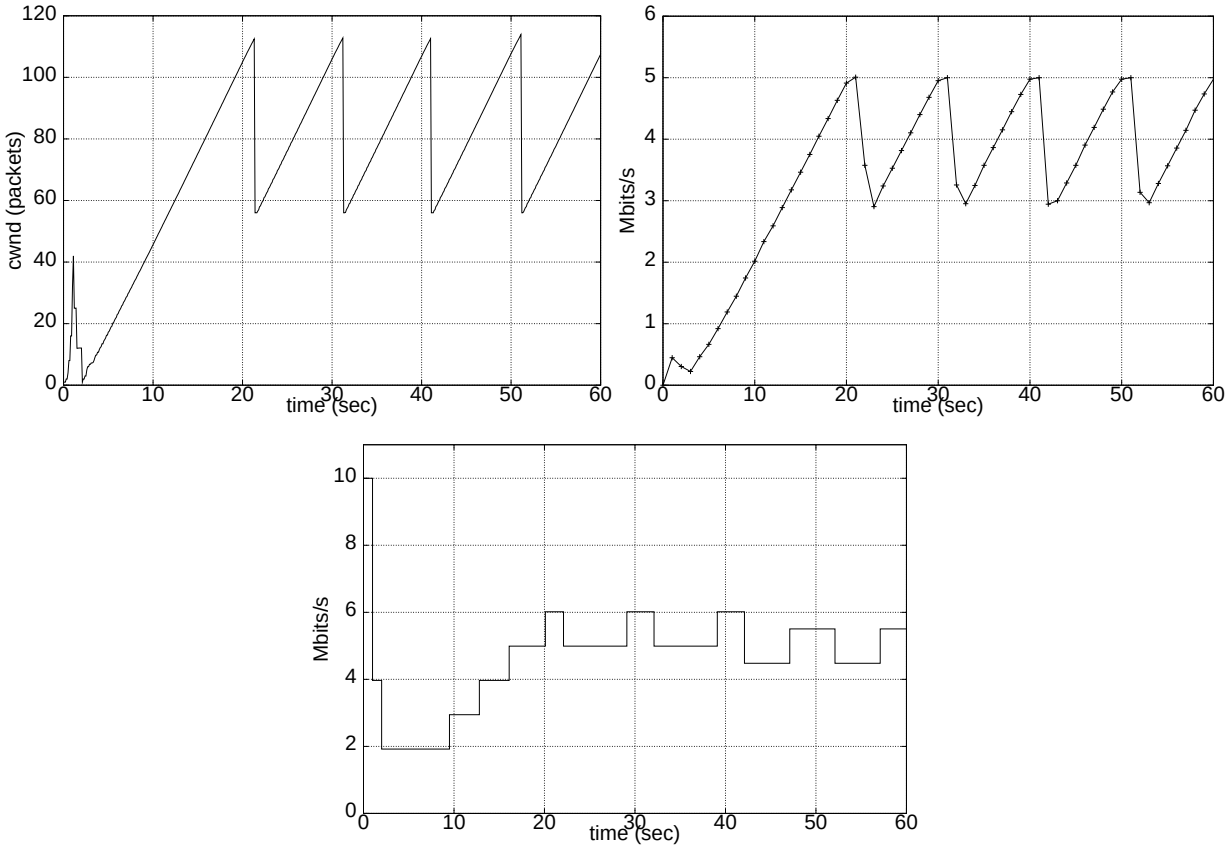
- Rate remains unchanged

Static Link, Reno, 10 pkt, 80ms delay



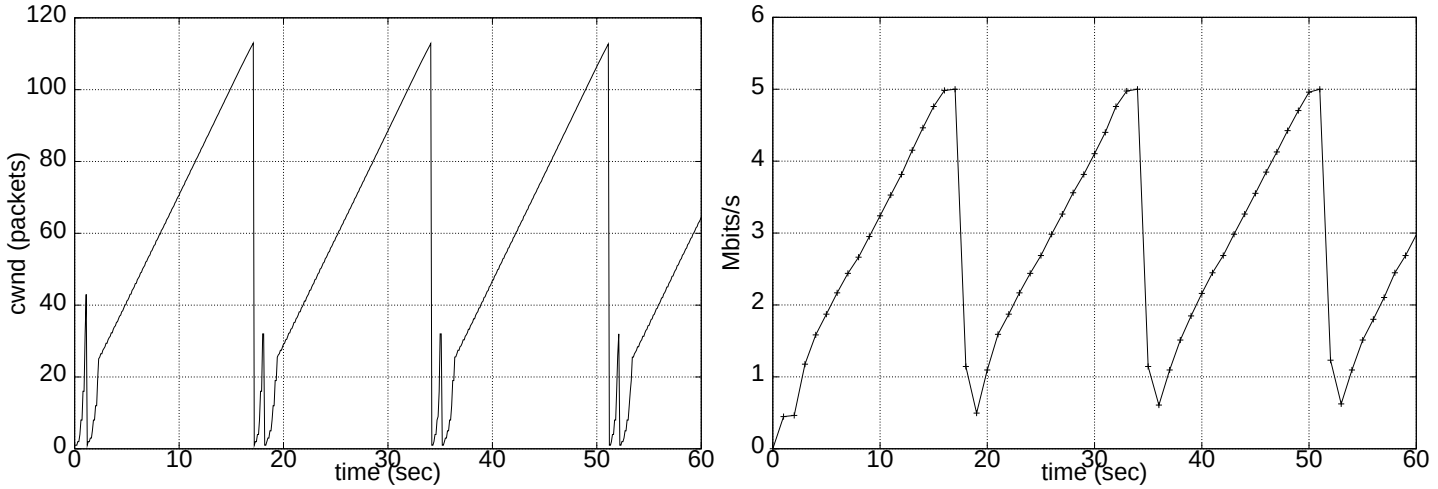
- Link between last 2 nodes extended to 50ms (added 40ms)

Variable Link, Reno, 10 pkt, 80ms delay



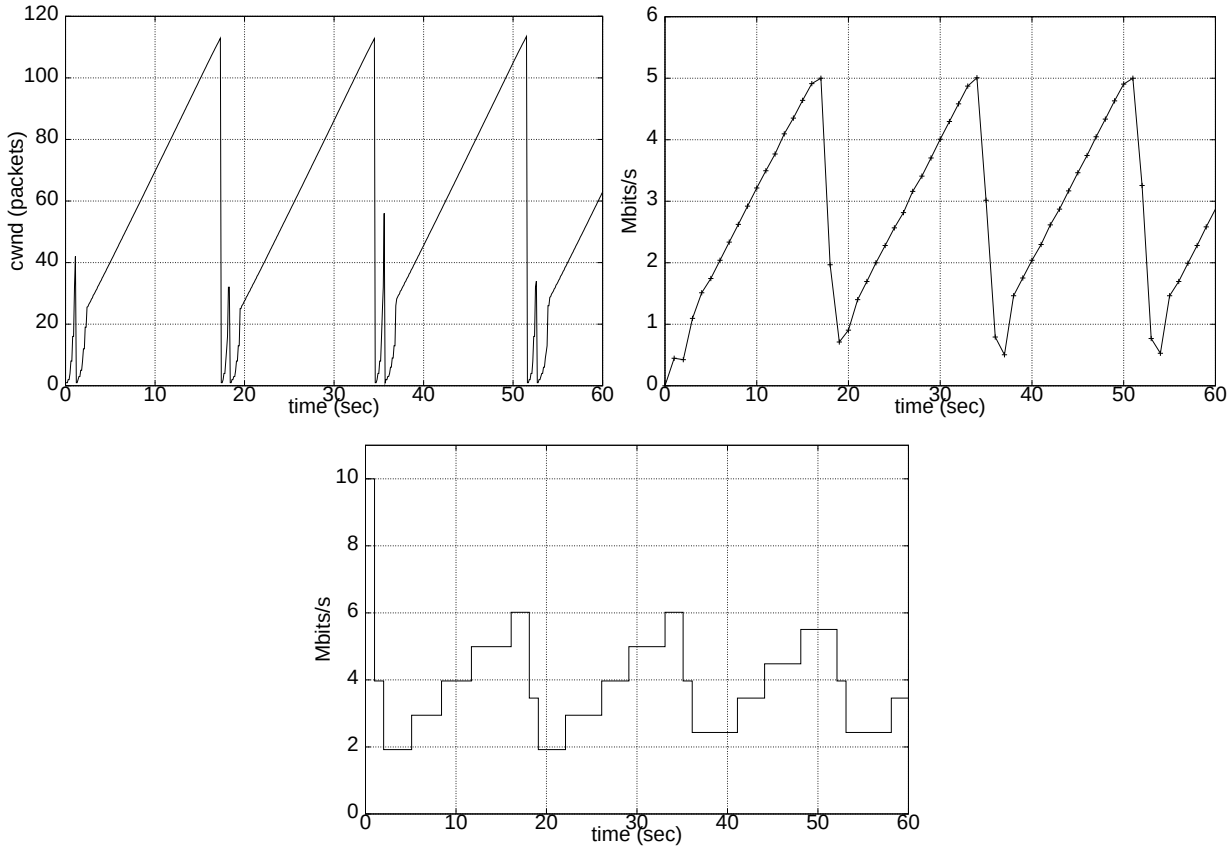
- Throughput follows window, DTM alloc. “mirrors” this

Static Link, Tahoe, 10 pkt, 80ms delay



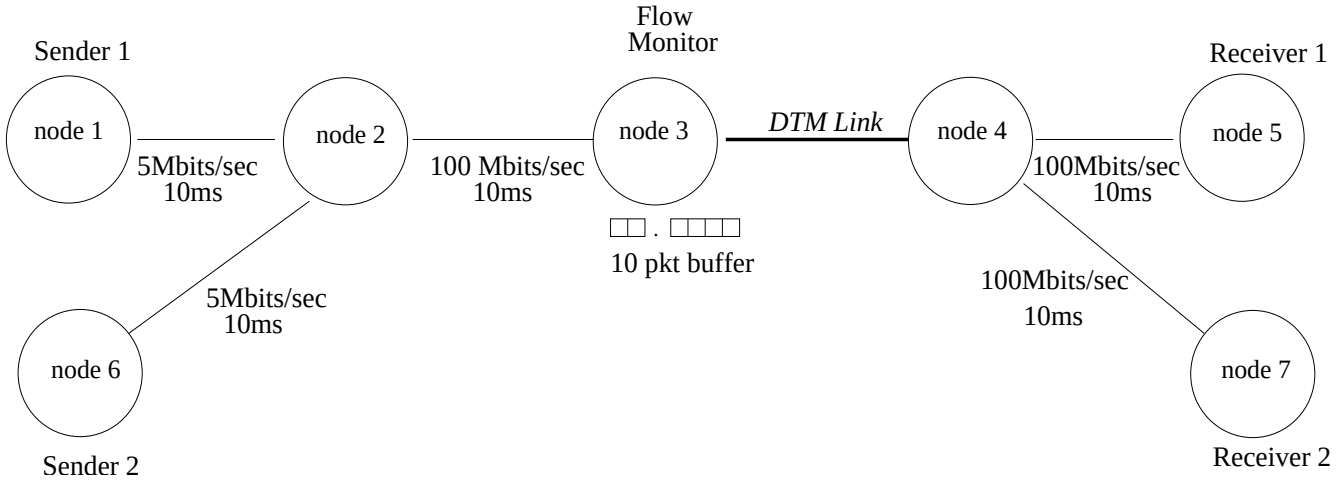
- No fast retransmit (RTO timer initiated repair)
- Window decreased to 1

Variable Link, Tahoe, 10 pkt, 80ms delay



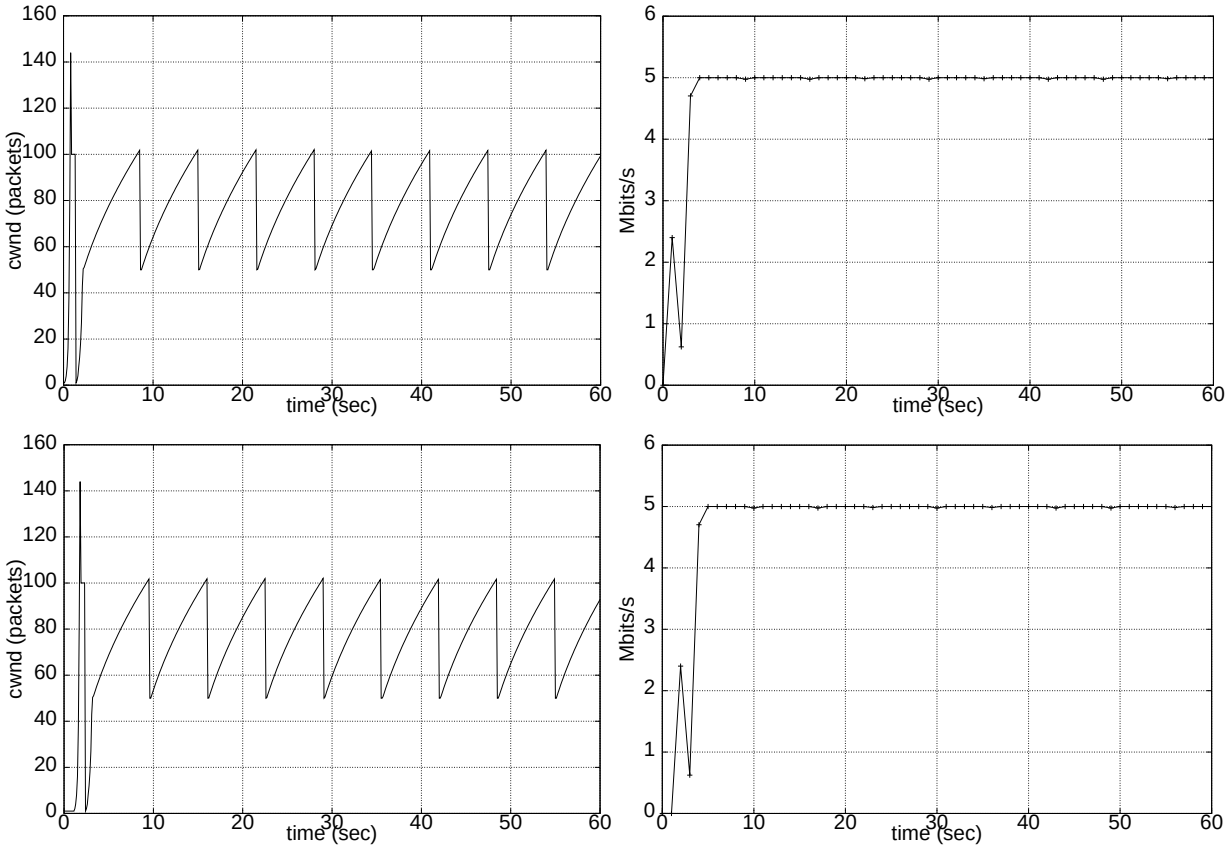
- Allocation in accordance with measured rate

Experimental Set-up 2



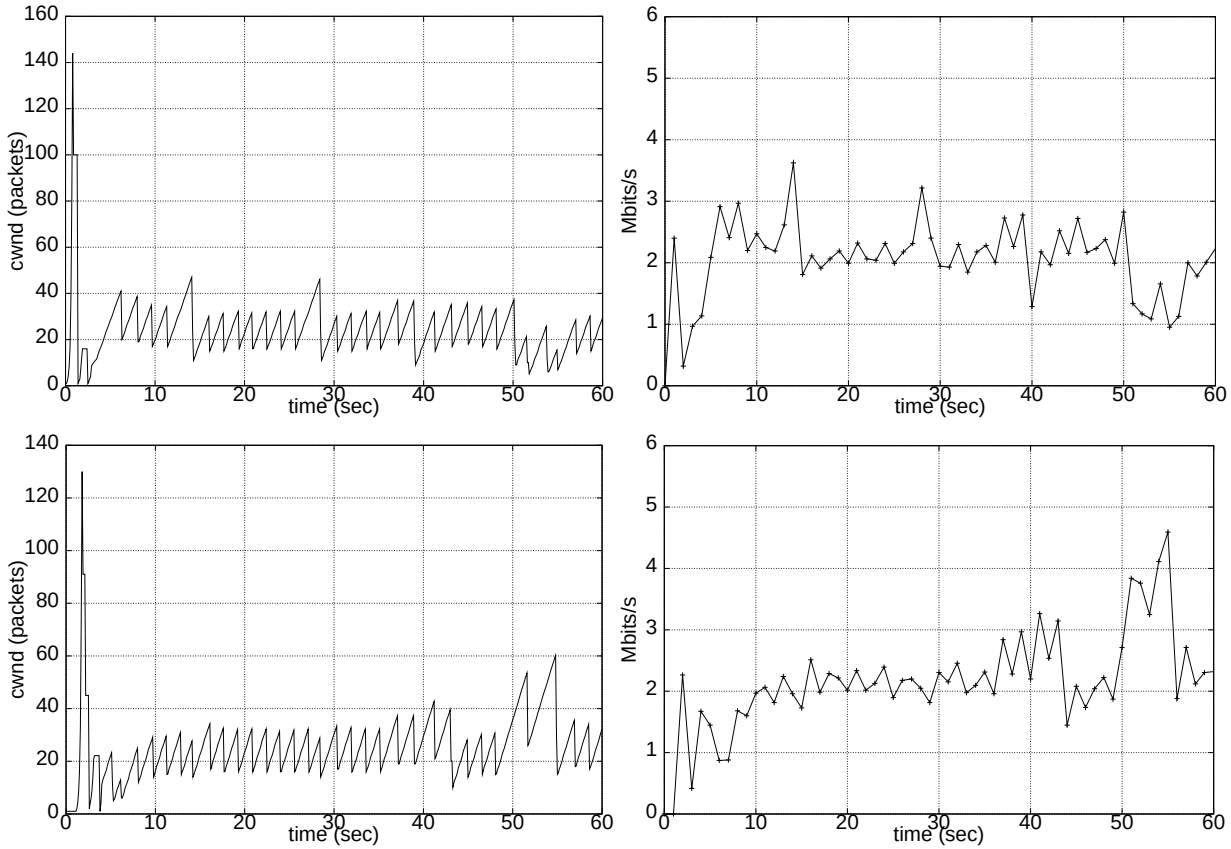
- Run 2 simultaneous flows
- Incoming links limited to 5Mbits/sec
- Access and out-going link(s) 100Mbits/sec

Static Link, Reno, 10 pkt, 40ms, 2 flows



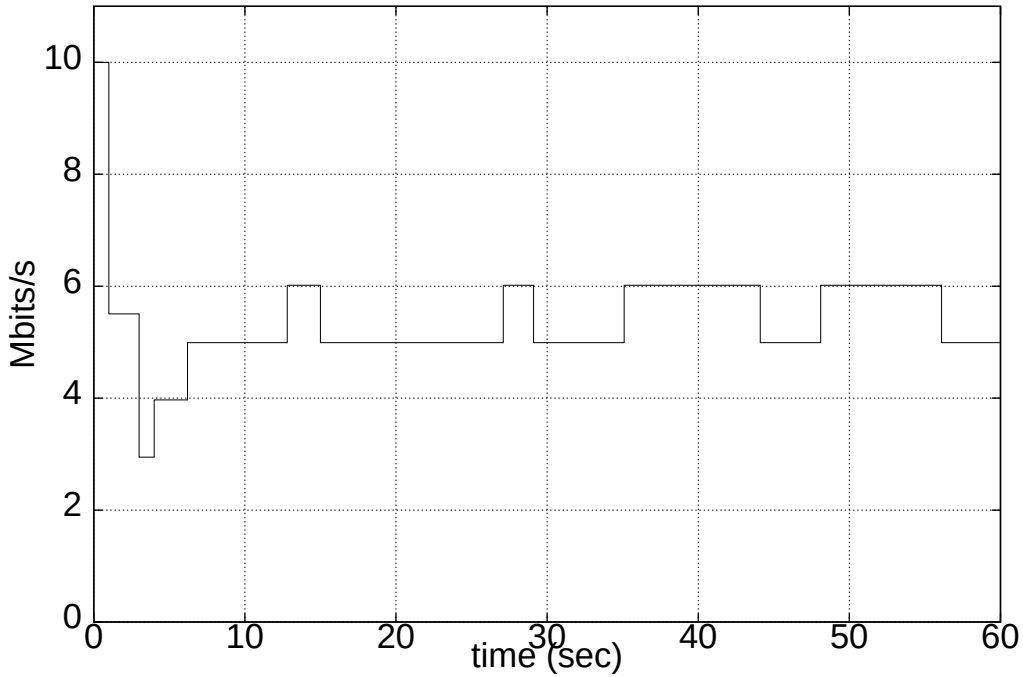
- 2 flows reach their input limited 5Mbits/sec

Variable Link, Reno, 10 pkt, 40ms, 2 flows



- Flows never reach their 5Mbits/sec

Variable Link, Reno, 10 pkt, 40ms delay, 2 flows



- DTM faithfully allocates calculated rates
- Packets lost in small buffer at DTM link

Related Work

- Surprisingly difficult to find other work in this area
- Other ideas look at buffer lengths (and losses) in router in order to allocate bandwidth for TCP flows
- Conclusions - parameter choice *important*
  - James Sterbenz - Load Reactive Links
  - Henrik Lundquist - Allocation scheme for DTM

Conclusions

- In nearly all cases we tried the allocation works well
- *However* might need to take account of lost packets as well as TCP throughput
- Not so easy when loss is not in the monitoring node
- Time scales surely important
- Large parameter space: TCP type, RTT, buffer, sampling rate, allocation frequency etc.
- Paper, technical report and Tcl scripts available from: http://www.sics.se/~ianm/VarTCP/VarTCP.tcl