**INTRO**

* buffer space -
  + too little: high loss rates and low link utilization
  + too much: long queuing delays
* Traditional method -
  + routers - intentionally discarding packets when their queues are too long
  + Internet traffic sources - when loss - decreases send data rate
  + limited buffering and loss feedback - for congestion control
  + The protocol that controls sources’ send rates degrades rapidly - if the network cannot store at least a few packets per active connection - highly variable delay - perceptible by users
  + heavy load - loss feedback - segregates
    - unlucky ones (which see timeouts)
    - lucky ones (which do not)
* Paper proposal
  + router queue lengths - vary with the no of active connections
  + routers -
    - physical memory - proportional to maximum number of active connections
    - dropping policy - keeping the actual queue size proportional to the actual no of active connections
  + FPQ (FlowProportional Queuing) - automatically chooses good tradeoff - queuing delay and loss rate
    - congestion feedback using queuing delay - proportional to the number of connections
  + TCP’s window flow control - send rate inversely proportional to delay
  + TCP + FPQ - send rate inversely proportional to no of TCPs sharing a link
  + FPQ - heavy load - every transfer - same queuing delay - fairer - overall same delay as loss feedback

**TCP in high loss rates**

* A TCP sender - sets send data rate using window flow control - one window packets - one round trip time
* TCP adjusts window size - reflects network conditions
  + send one window of packets - increase window size by 1
  + TCP decides - network discarded a packet - decrease window in half
  + detects loss fast using “fast retransmit” - if window > 3 packets - if fails - conservative retransmission timeout of 1+ seconds
  + Loss affects delay in two ways: by decreasing TCP’s window size and send rate, and by forcing TCP into timeouts.