Goal: q = 0.

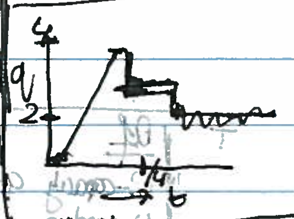
Lecture Notes (3/17/03)

(lets 5 pks 4 for w)

$w = R * rtt$

(8) One conn

M

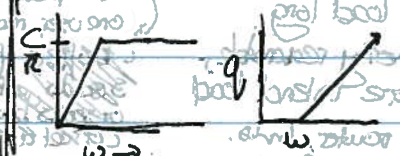


$R = w/rtt$

buffer smooths window:
packet leave at bottleneck rate
receivers sends ack as pkt arrives.
sender limited by recv acks.
⇒ nice feedback:
· self-pacing.
· cool.
not very sensitive to w size!

But:

(9)



large windows ⇒ unnecessary delays

B (10) Choosing w — (ignore receiver window).

$w = C * d$ (last lec)

$w = R * Rtt$? (rtt includes q.)

w too big → rtt bigger → w bigger → q bigger

nice puzzle

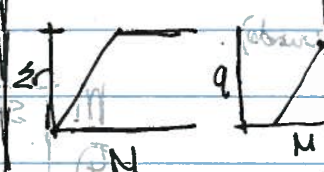
AIMD.

lets assume reasonable window

timeout came as a surprise as possible solution direction

in equilibrium, add one to window

M (11) Sharing a link.



(each conn 1 window to q.)

F (12) Sharing-r²

$R = \frac{w}{rtt}$ (steady state).

$rtt = q/C$

$q = N * w$

⇒ $R = C/N$. ⇒ as N increase, adapt.

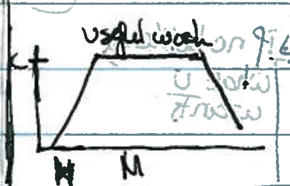
cool, delay goes up, sender responds correctly.
feedback is variation in rtt.

(works well for moderate overload.
cheat: · multiple conn.
· cheat w. large w!

long term, fix large N with incr C.

(12) If you don't, q gets longer, with # users. (bandwidth is cheap).

(13) B



↓ good: pkt slow down sender.
bad:

If q is unbounded ⇒ delay goes ⇒ timeouts ⇒ retrans.
⇒ all pkts are retran!
(bad interactive perf)
congestion collapse).

bigger point about exponential backoff ⇒ allows to repair things

Ethernet

M (14) max q

pkt wait too long q ⇒ drop pkts ⇒ ....

B (15) Soluts

(1) adaptive TO.
(2) selective
(3) flow w = L/N on turn on reset w.

⇒ flow control good too!