CS348 Notes
TCP Vegas
Video Numbers: 27

OjMaha

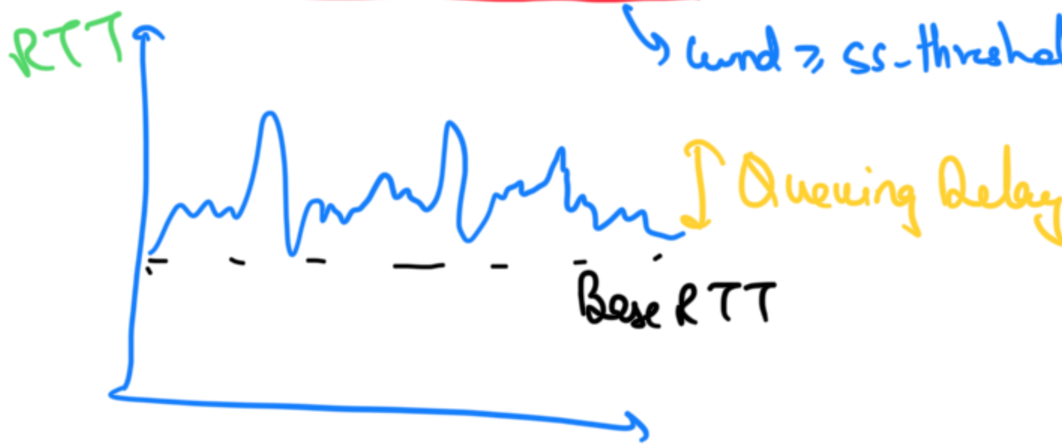
I have prepared these notes by watching the videos from Networks Playlist. The following notes may be asynchronous and irrelevant to what Prof. Vinay teaches in class (cuz I do not pay attention during lectures lol). Further, these notes might not cover *everything* as explained in the video lectures. Consider these to be a supplemental read :). If you find any errors, do notify me so they can be edited.

TCP Vegas

RTT may not seem to be a good indicator of congestion since higher RTT can be caused due to various other reasons (→ Q delay ↑ or smth) apart from congestion. Further, packet loss doesn't necessarily imply congestion. Packets may be lost due to lower layers as well.

TCP Vegas nevertheless tries to factor in RTT as a congestion indicator.

- detect loss by T.O. & 3 Dup ACKs. Modify cwnd & ss_thresh as in Reno.
- slow start rules are same as Reno.
- congestion avoidance is different.
  - ↳ cwnd ≥ ss_threshold



empty queue ⇒ expected rate = $\frac{cwnd}{Base\ RTT}$

∴ RTT ≥ Base RTT; rate = $\frac{cwnd}{RTT}$ ≤ expected rate

RTT: smoothened RTT over a time interval.

Diff = exp. rate − actual rate.

→ like Reno

* if $diff < \alpha$ → $cwnd \mathrel{+}= (MSS)^2 / cwnd$ @ every ACK for new data.

parameters

* if $diff > \beta$ → $cwnd \mathrel{-}= (MSS)^2 / cwnd$

* $\alpha < diff < \beta$ → $cwnd = cwnd$ (not modified)

$$diff = \frac{cwnd}{baseRTT} - \frac{cwnd}{RTT} = cwnd \times \left(\frac{RTT - base\ RTT}{base\ RTT \times RTT} \right) = \frac{cwnd \times queue\ delay}{base\ RTT \times RTT}$$

Say the queue is as follows:

Queuing delay $= B/c$. further, $RTT \approx cwnd/c$

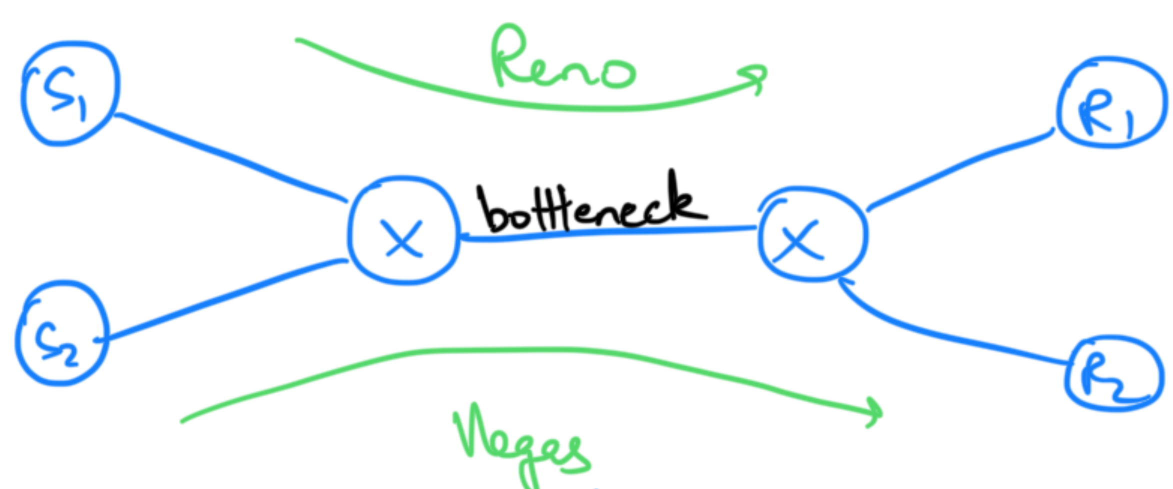
$$\therefore \quad diff \approx \frac{B}{Base\ RTT} \quad \text{(if there is only one flow into the router)}$$

For stability, we'd want $\alpha < \frac{B}{base\ RTT} < \beta$. (no changes to cwnd) eg:- $\alpha = 30$ kbps ; $\beta = 60$ kbps; Base RTT = 100ms.

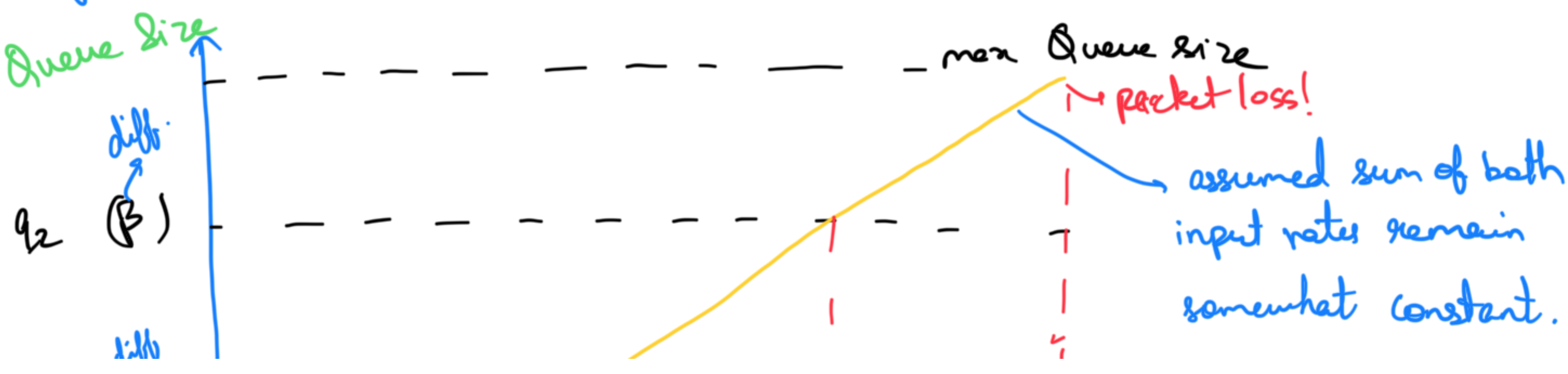
Then; we want $3kb < B < 6kb$.

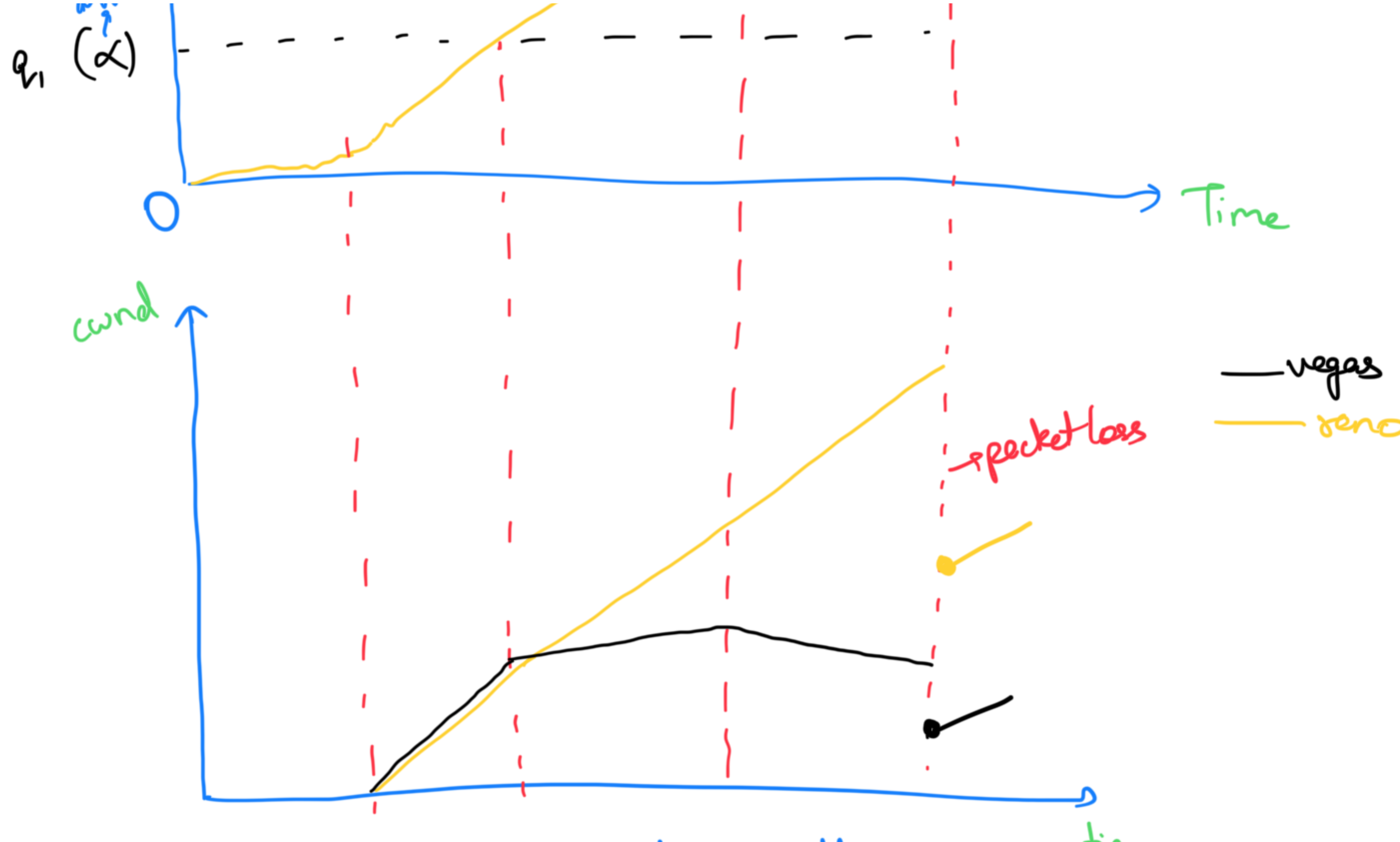
Drawback is choosing α, β difficult since queue sizes @ diff routers may be different.

TCP flow v/s Reno flow:



Ignore slow start in analysis.



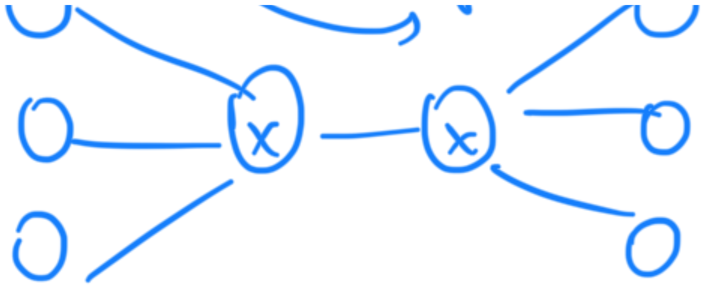


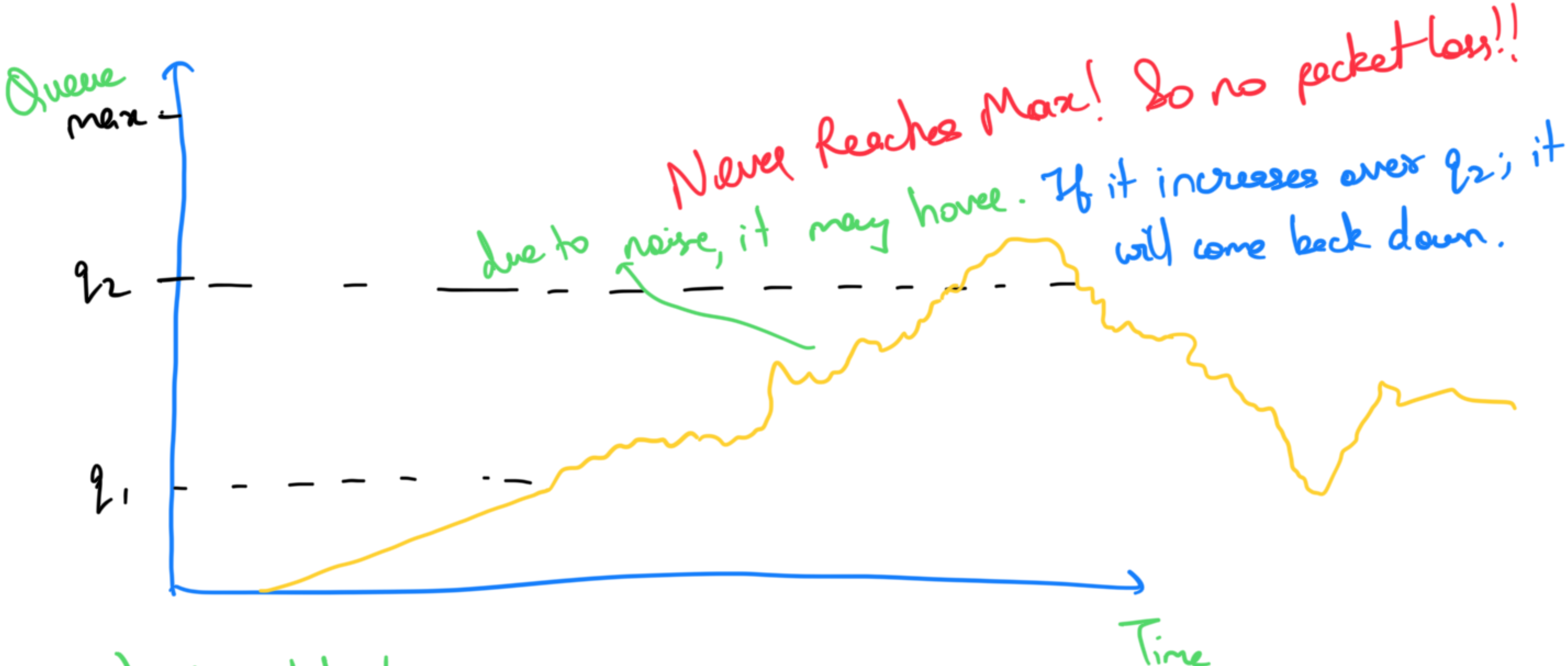
Thus, the Reno flow will eat up all the bandwidth. Reno is more aggressive than Vegas.

What if only Vegas flows?

all vegas

Assume everyone has approx same RTT.





i) No pkt loss

ii) Queue delays are stabilized b/w q_1/c & q_2/c.

iii) Throughput can be higher than for all Reno case.

Why? Because in all reno case; the input rate might become less than c due to severe packet losses. Thus, the queue remains empty and unutilized. However, TCP vegas manages to maintain & stabilize the queue size as here the queues never empty.

Claim: all vegas case gave 50% higher throughput than all Reno case.