Oct 11, 2023

Agenda

- RTO estimation
- Packets to Bytes
- TCP Flow control
- TCP fairness
- TCP cross layer
- TCP over wireless

⊕ TCP RTO Estimation

RTO = weighted avg. of historical RTTs + safety factor

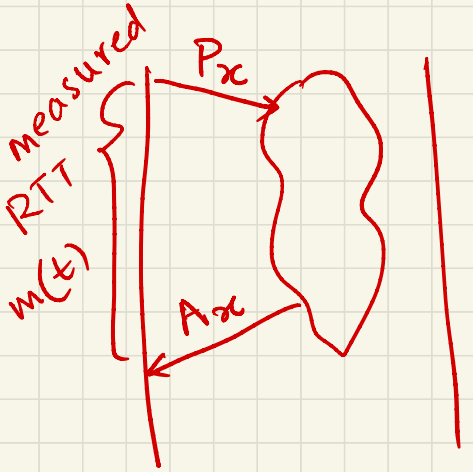
Historical RTTs → Estimated RTT ($\hat{R}$)

Safety factor → Deviation of RTT ($\hat{\Delta}$)

$$\hat{R}(t) = \alpha\,\hat{R}(t-1) + (1-\alpha)\,m(t)$$

weights

$\alpha = 0.8$



$$\hat{R}(t-1) = \alpha\,\hat{R}(t-2) + (1-\alpha)\,m(t-1)$$

$$\hat{R}(t) = \alpha\{\alpha\,\hat{R}(t-2) + (1-\alpha)\,m(t-1)\} + (1-\alpha)\,m(t)$$

$$\hat{R}(t) = \alpha^2\,\hat{R}(t-2) + \alpha(1-\alpha)\,m(t-1) + (1-\alpha)\,m(t)$$

$$= 0.64\,\hat{R}(t-2) + 0.16\,m(t-1) + 0.2\,m(t)$$

Called

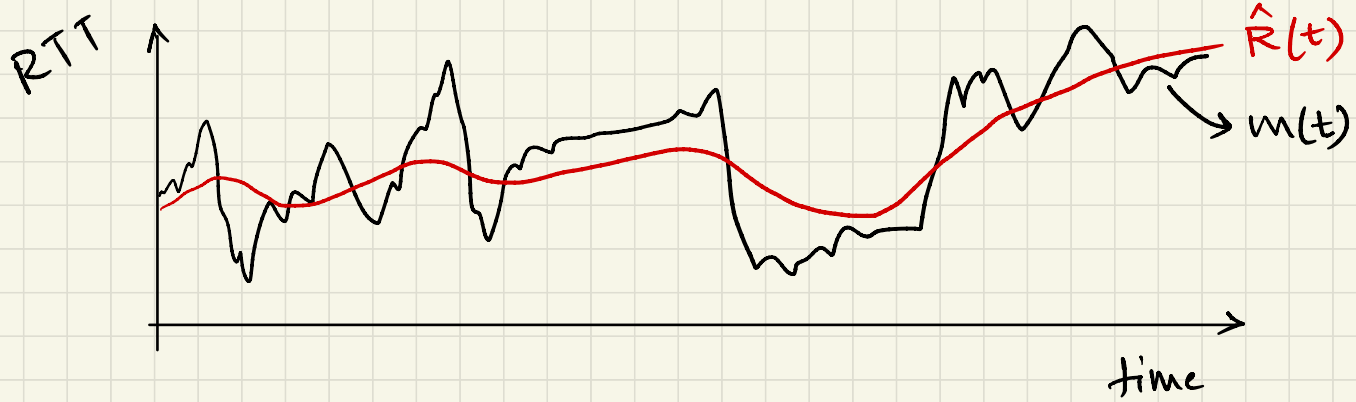
All RTT history up until time $(t-2)$ has a weight of 64%.

EWMA ⇒ Exponential weighted moving average.

① Deviation of RTT $\hat{\Delta}(t) = \hat{R}(t) - m(t)$

$$\hat{\Delta}(t) = \underbrace{\alpha\,\hat{\Delta}(t-1) + (1-\alpha)\{\hat{R}(t) - m(t)\}}$$

smoothen the $\hat{\Delta}(t)$ as well.



$$RTO(t) = \hat{R}(t) + 4\hat{\Delta}(t)$$

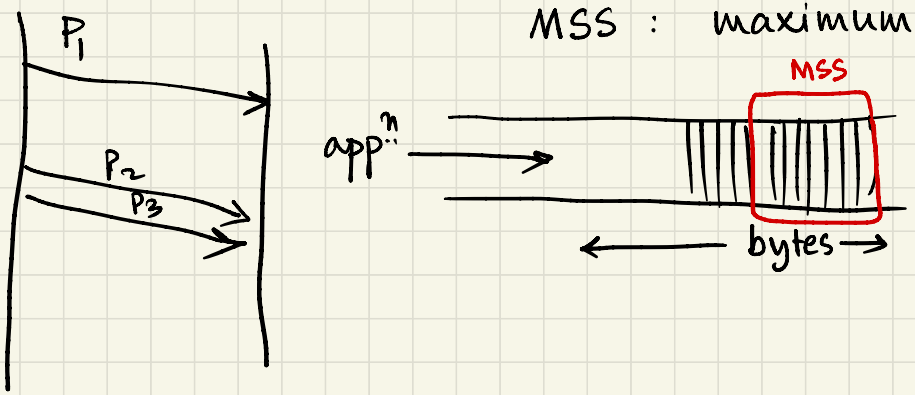
→ Retransmit timeout estimation over time.

When estimating $\hat{R}(t)$ ignore dupACKs, and any stale ACKs that is left of Tx's window base.

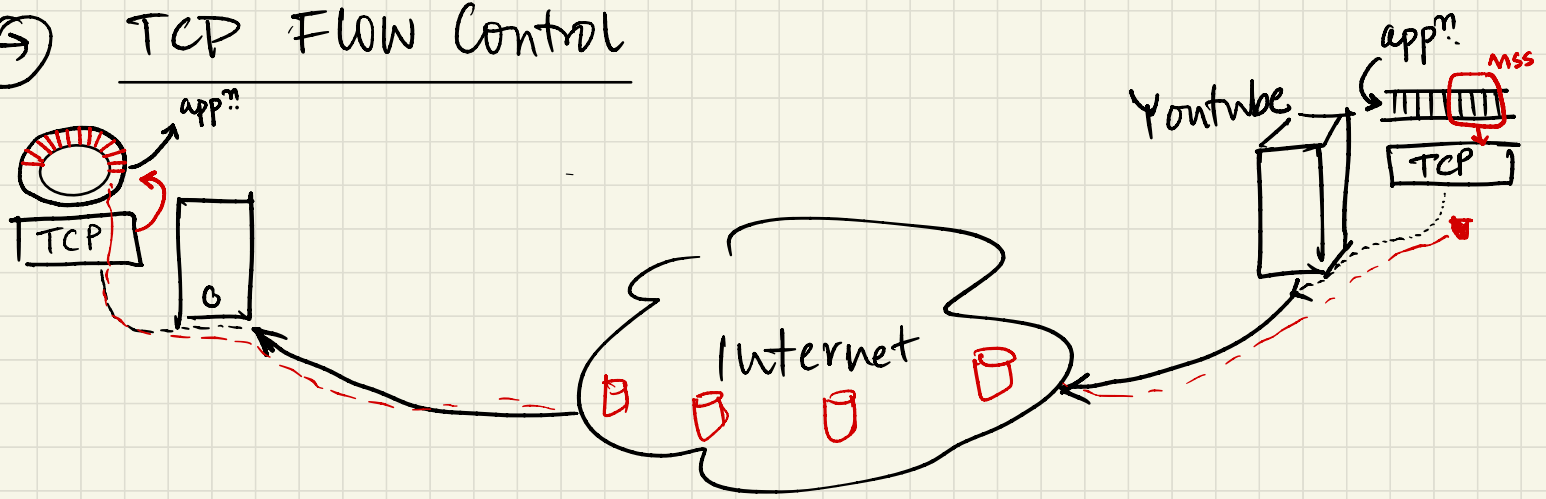
⊕ Packets to Bytes

In reality, TCP operates as a byte stream.

MSS : maximum segment size

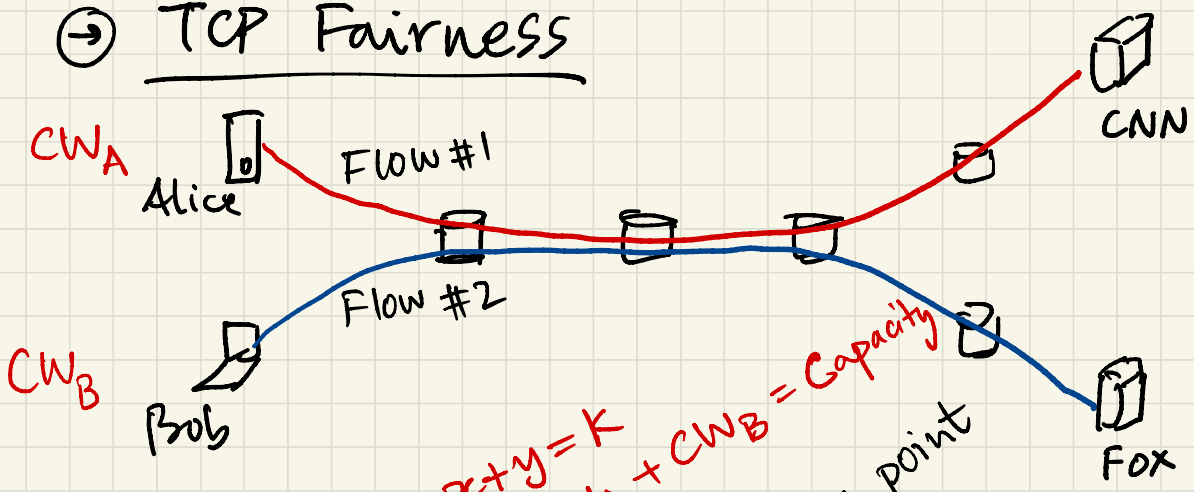


⑤ TCP FLOW Control

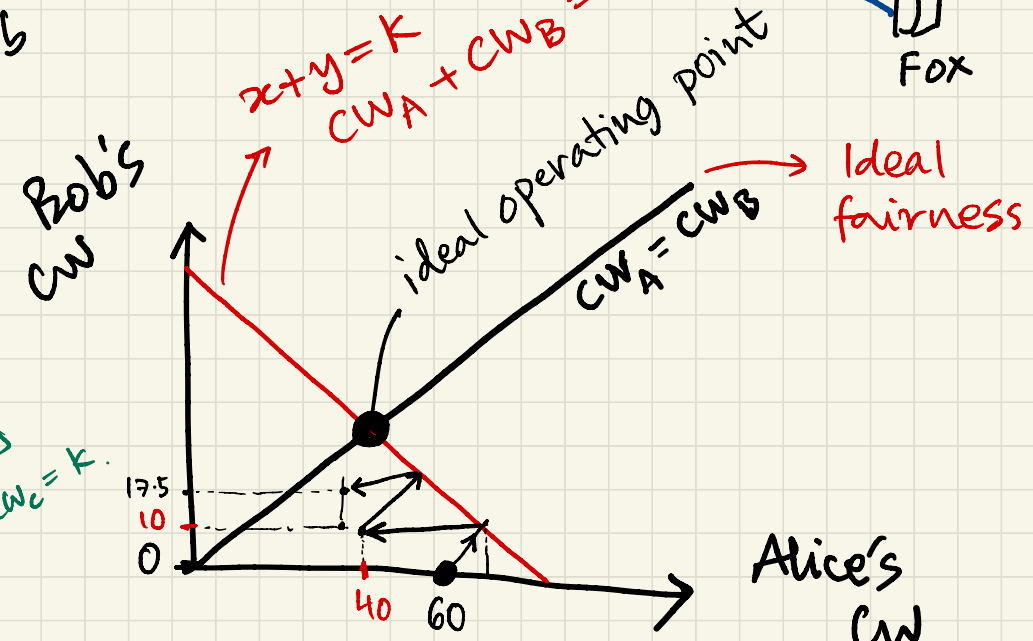


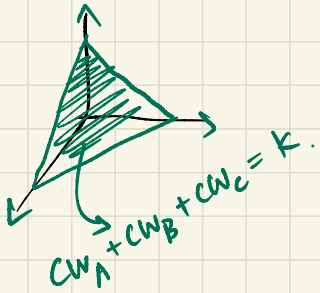
- If R_x's buffer fills up, it's sad because congestion happened right at the entrance of the app^n layer.
- R_x TCP includes how much space is available in its socket buffer. → B bytes. → ACK includes value B.
- When T_x TCP gets ACK, it decides to send: min { B, CW }
  - The CW does not change..... CW keeps following the protocol.

TCP Fairness



How is capacity divided between Alice & Bob.?





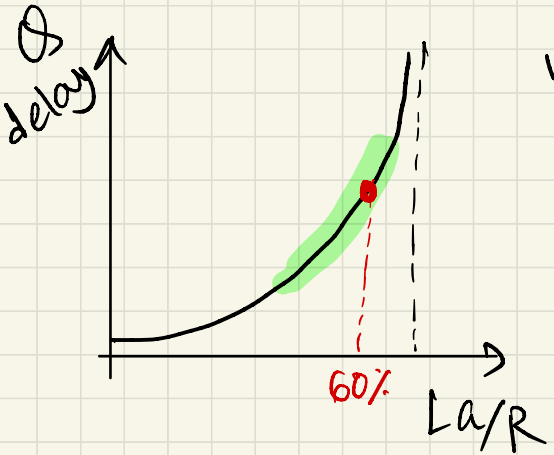
If capacity = 100

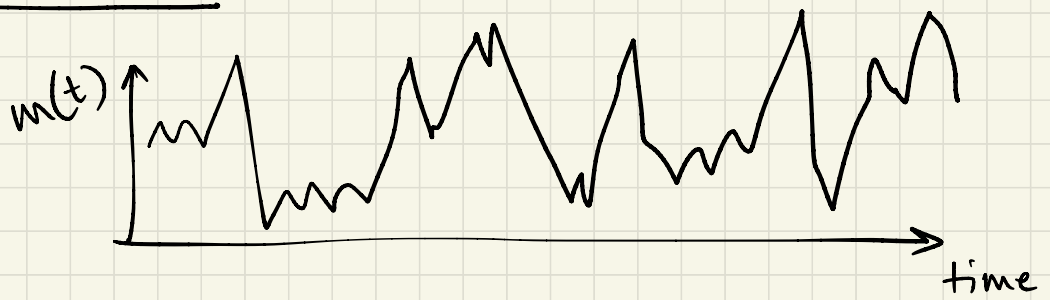
CW_A	CW_B
60	0
~~80~~	~~20~~
40	10
~~65~~	~~35~~
32.5	17.5
⋮	⋮
50	50

40 → 20 + 20

50 → 25 + 25

TCP RED (Random Early Drop) → Cross Layer Hints

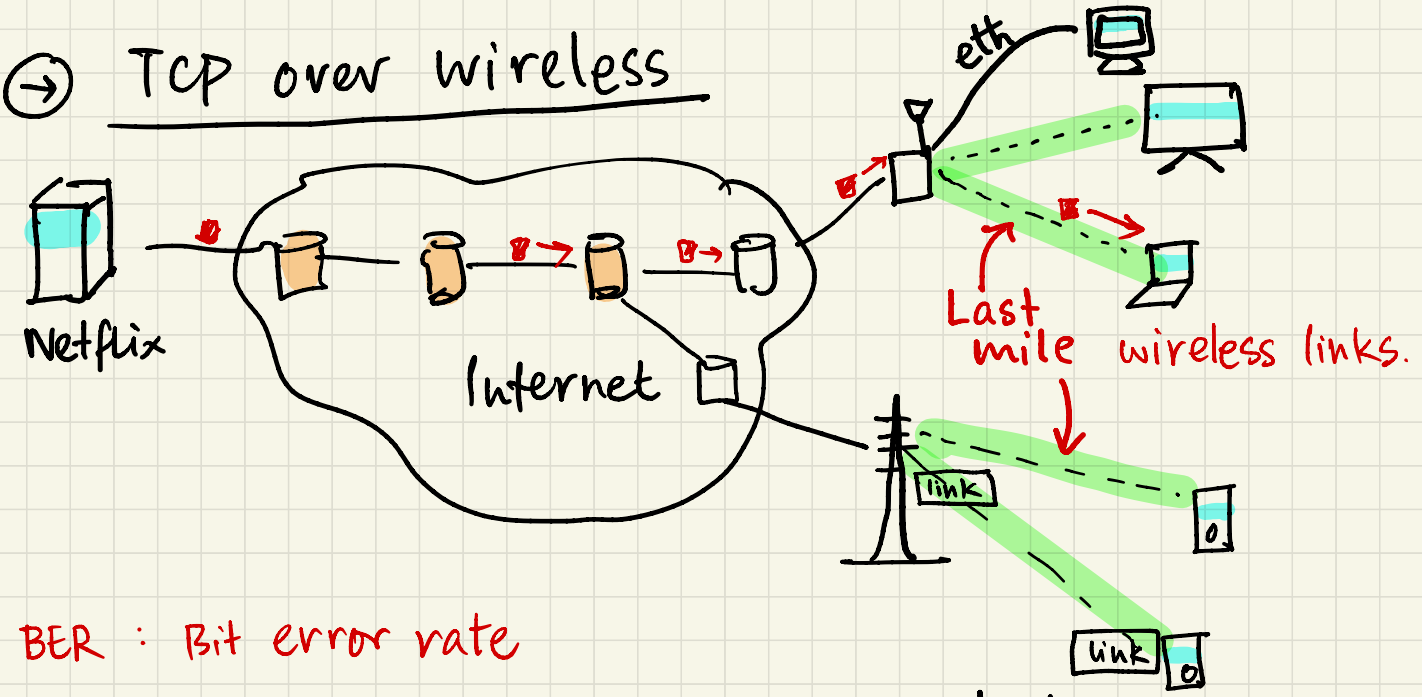




⊖ Hack/workaround for the network layer to give feedback to TCP about it's queue size/congestion.

RED → Network layer drops a random data packet from each flow, so that the router queue does not fill up above some threshold (say 60%.)

TCP over wireless



BER : Bit error rate

Wireless ~ 1 bit in error every 10^{4-6} bits $\Rightarrow \frac{1}{100}$ pkts corrupt

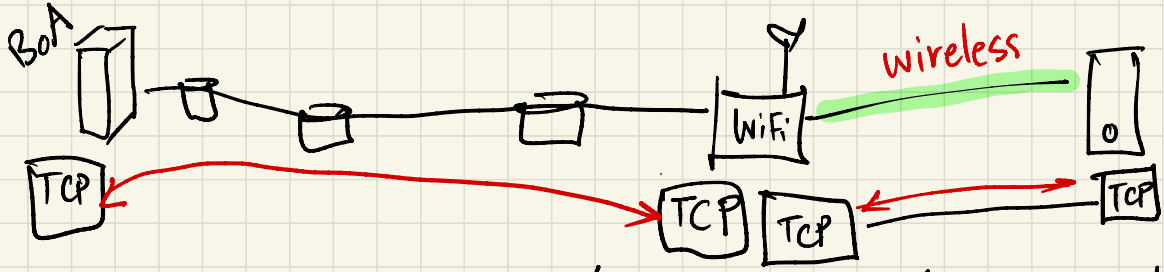
Wired $\overset{BER}{\sim}$ 10^{-12} $\rightarrow$ $\frac{1}{\text{million}}$ pkt corrupt

10^{-5}

- TCP must differentiate between wireless losses & congestion losses (at router) because :

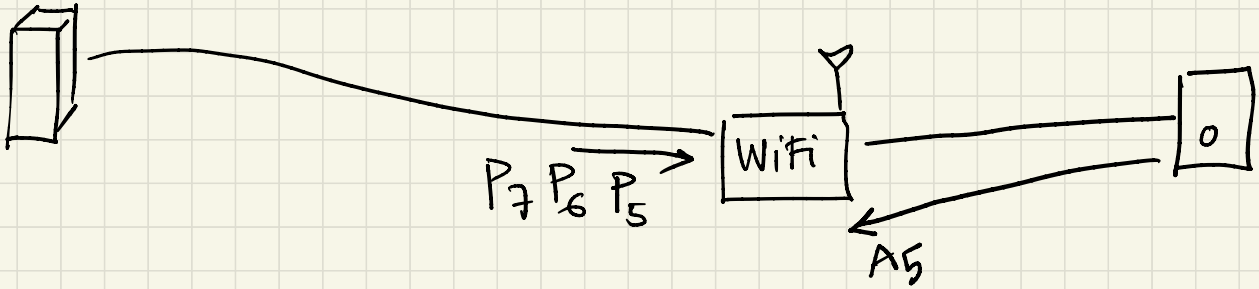
  If congestion loss
  then reduce CW
  else if wireless loss
  then simply retransmit & don't change CW.

① Split TCP.



violates TCP's end to end principle

②



TCP snoop because link layer is snooping into TCP header so that it can retransmit lost packets.

End of Transport / TCP.