Lec22

Saturday, March 4, 2023 10:59 AM

Bring handout

Mid-term Exam:

1. Midterm: 11/7 Thursday 8am: available on Blackboard
    a. Due next day at 12:00pm.
    b. Take 2-4 hours
2. Take home. No discussion with others. Can consult book or paper.
3. Coverage up to lectures 1-24.
    a. Convex sets
    b. Convex functions
    c. Convex programs and optimal solutions
    d. Duality, KKT conditions, duality gap.
    e. Optimization algorithms, convergence.
    f. Some application: background info will be provided
4. Sample exam on the web.
5. Review notes and homework problems!

- IP is best effort
- TCP provides reliability on top of IP
- Window-based control increases throughput but also congestion
- TCP window adjustment:
    - Congestion avoidance
    - Three goals
    - Desirable features

Congestion control

Sunday, March 08, 2009 4:00 PM

So far we have discussed

- formulation of optimization problems into convex programs
- Optimality conditions
- Duality
- Optimization algorithms with practical interpretations

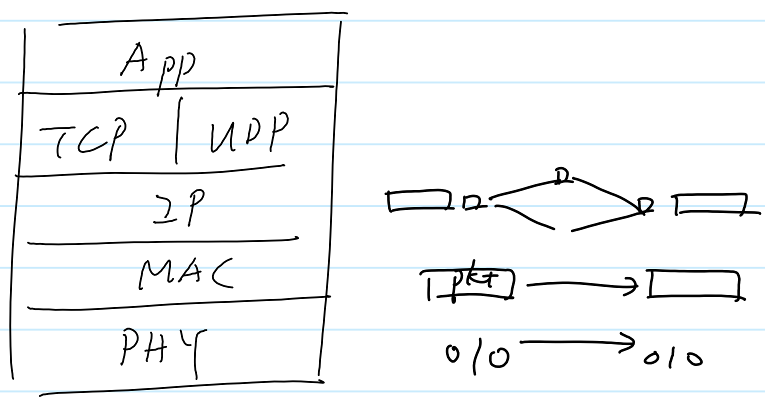
Next, we are going to use congestion control & TCP as practical examples of how to use convex optimization to design protocols

- which will introduce additional issues
    - delayed feedback
    - noise / randomness

TCP

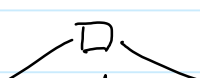
Sunday, March 08, 2009 4:22 PM

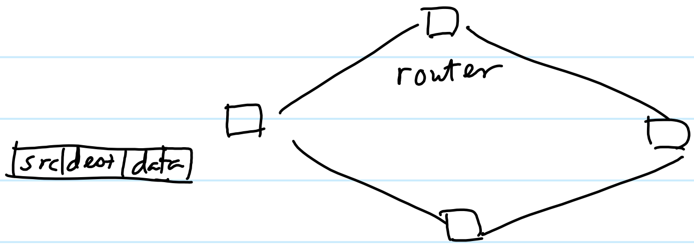
- The TCP/IP protocol stack



- Key summary:
  - IP is best effort
  - TCP provides reliability on top of IP
  - Window-based control increases throughput but also congestion
  - TCP window adjustment:
    - Congestion avoidance
    - Three goals
    - Desirable features

- IP provides best-effort packet delivery service over heterogeneous networks
  - each computer has an IP address
  - each data packet contains both the source IP addr & the destination IP addr.
  - routers will route IP packets from the source to destination





- Best effort:
    - packet may be lost, duplicated, form loops or out-of-sequenced.
    - no guarantee.

Skip UDP

- UDP: just add a port number in the packet header
    - (port + addr) identifies the specific application on a specific computer
    - no improvement in service quality.
- TCP: provides not only a port number for each application, but also a connection-oriented, reliable, in-order packet delivery service over IP.
    - connection-oriented and in-order: both end-points maintain the "state" of the connection
    - open / close: an OS call binds (src IP,

' src port) to (dest IP, dest port)

- sequence number:

  - source maintains seq # for the next byte to be sent

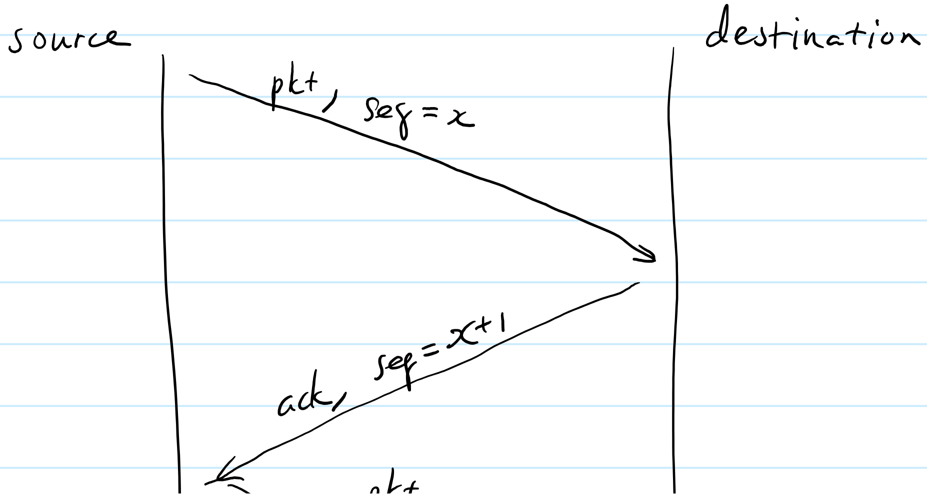
  - receiver maintains seq # for the next byte expected

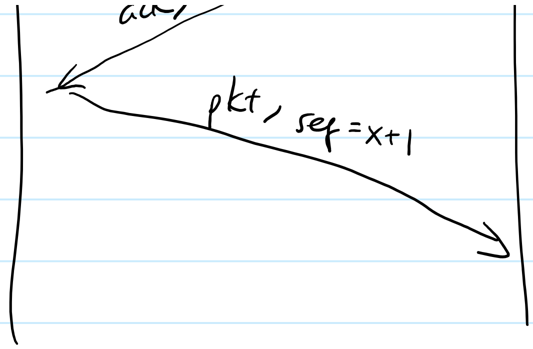
  - The two sync the seq # at the beginning when the connection is opened.

- reliable:

  - The seq # advances only after the packet is successfully acknowledged.

  - If the packet is not acked within a time-out period (on the order of a round-trip-time RTT), the source will retransmit the packet.





- While this is reliable, it is too slow
    - 1 packet every round-trip-time (RTT)

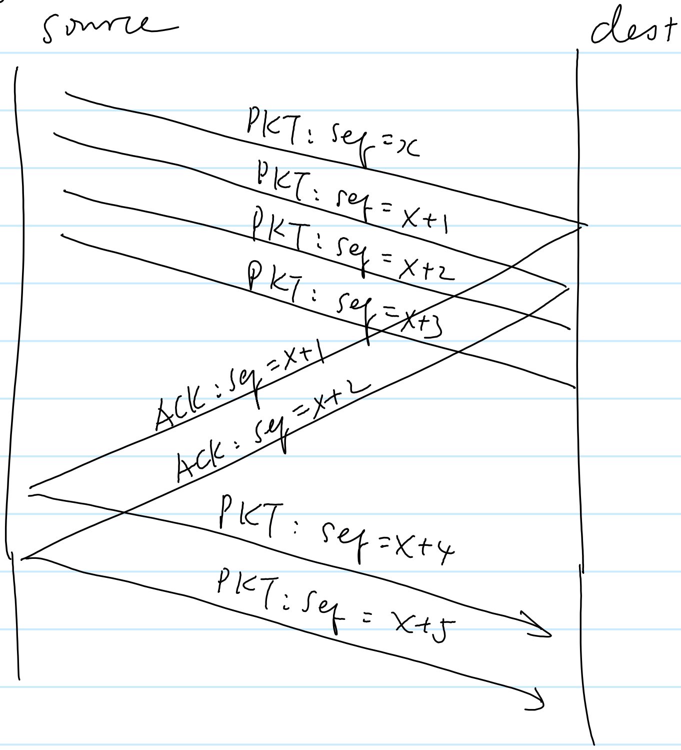
Windows

Wednesday, October 25, 2023 4:36 PM

- To increase throughput, the source can send multiple packets into the network before waiting for an ack.

- Window: the # of outstanding packets that the source can send before waiting for an acknowledgement

Window = 4



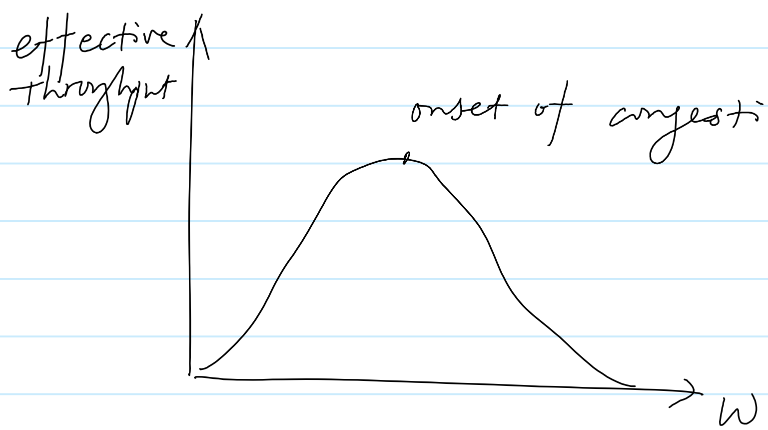
- Clearly, the rate that packets are sent into the network increases with the window-size the network increases with the window-size

- If the window-size is too small, the network is under-utilized.

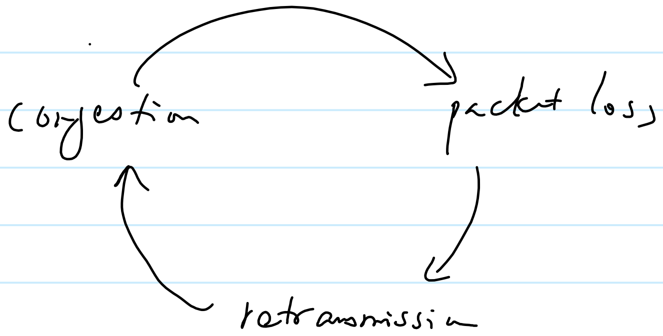
- If the window-size is too large, the network will become congested.

- In reality, when intermediate routers can not accommodate these packets, the packets are dropped, which leads to time-out & retransmission of the packet at the source

- If not controlled properly, the retransmission may further aggravated the congestion, which then leads to the so-called "congestion collapse."



Congestion collapse: no one is able to send any useful packets. packets keep being dropped & retransmitted.


congestion
packet loss
retransmission

TCP window control

Wednesday, March 11, 2009 2:38 PM

Ⓠ How does TCP control the window sizes.

Skip - flow control: the receiver will advertise a window-size that corresponds to the # of outstanding packets that it can accommodate.

(This is due to a finite receiver buffer and that the application may not pick up packets fast enough.)

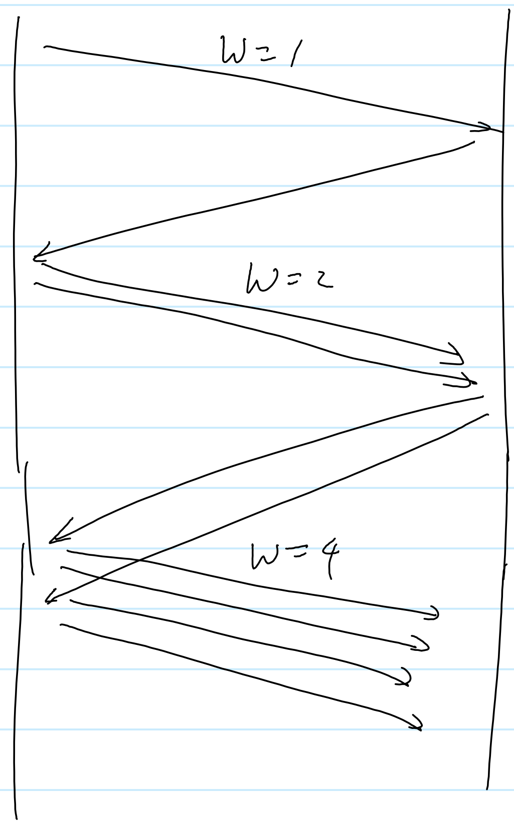
- flow control puts an upper bound on the window size.

- Congestion control: our main focus

Two phases: skip

① "Slow start", which is not slow

- Used when starting a connection.
- Increase window by 1 at each ack
- Doubles the window size each RTT.



- "Slow start" stops when the window size exceeds the slow start threshold, or when packets are time-out/lost.

② "Congestion avoidance": Our focus

- On each new ack, cwnd increased by $\frac{1}{cwnd}$
  - After one RTT, cwnd increased by 1
- On each time-out, cwnd cut in half
- Known as AIMD (Additive Increase/Multiplicative Decrease)

- Believed to achieve three goals:
  - to fully utilize the available bandwidth,
  - to prevent the onset of congestion
  - to be fair to different flows.
- Some highly desirable features of TCP
  - Only need to be deployed at the end-points
  - Very little support from routers (just drop packets when congested).

- The above basic principles of TCP was published in Van Jacobson's 1988 paper after the first congestion collapse
- They forms the basis of TCP today.
- Many researchers have tried to understand the behavior of TCP
  - Easier if there is only one link
    - Utilization
    - Congestion avoidance
    - Fairness

- However, the case of multiple links is hard to track.

- Until Kelly's 1998 paper, which view congestion control as the solution of an optimization problem.

Formulation

Wednesday, March 11, 2009 2:51 PM

- Kelly proposed to formulate congestion control as the following optimization problem, and view TCP as an iterative solution to this problem:

$$\max \sum_s U_s(x_s)$$

$$\text{sub to} \quad \sum_s H_s^l x_s \le R^l \quad \forall l$$

- x_s : rate of user s
- $U_s(\cdot)$: utility function
- $H_s^l = 1$: if user s uses link l
  $= 0$: o/w
- R^l : capacity of link l.

- Note that all three objectives are incorporated
  - High utilization : using increasing $U_s(\cdot)$
  - Congestion avoidance
  - Fairness : by choosing the utility function

e.g.: - $U_s(x_s) = \log x_s$

$$\Rightarrow \sum_s \frac{x_s - x_s^*}{x_s^*} \le 0$$

proportional fair

- $U_s(x_s) = -\frac{x_s^{1-\alpha}}{1-\alpha}, \alpha > 0$

(a) Is TCP solving such a problem?

(Q) If yes, what utility function is TCP implicit using?

(b) Are there other ways to develop protocols that also solve such an optimization problem?

(Q) Can we work with more complex settings, e.g. cross-layer control?

References:

(1) "End-to-end Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks," S. Kunniyur and R. Srikant, IEEE/ACM Transactions on Networking, Vol. 11, No. 5, October 2003

(2) F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for Communication Networks: Shadow prices, proportional fairness and Stability", Journal of Operation Research Society, Vol. 49, No. 3, pp 237-252, March 1998.

Primal solution: penalty approach

Wednesday, March 11, 2009 2:57 PM

- Consider the following problem:

$$\max_{[x_s]} \sum_s U_s(x_s) - \beta \cdot \sum_l \int_0^{\sum_l H_s^l x_s} p_l(R_l, x)\, dx$$

where the function $p_l(R, x)$ is chosen to penalize the objective when $\sum_l H_s^l x_s \geq R_l$.

$$p_l(R, x) = \begin{cases} 0 & x \leq R \\ > 0 & x > R \end{cases}$$

e.g. $p_l(R, x) = \dfrac{(x-R)^+}{x}$

- We will later interpret p_l as the packet dropping prob. at link l, when the packet incoming rate is x.

- This new problem can be viewed as a penalty function approach to solve the original constrained problem.

- Let $f(\vec{x})$ denote the objective function minus penalty.

- The gradient is given by

$$\frac{\partial f}{\partial x_s} = U_s'(x_s) - \beta \cdot \sum_l H_s^l p_l(R_l, \sum_l H_s^l x_s)$$

- Each user can employ the following gradient-ascent iteration:

$$\dot{x}_s = K_s \left[U_s'(x_s) - \beta \cdot \sum_l H_s^l p_l(R_l, \sum_l H_s^l x_s) \right]$$

Example:

① If $U_s(x_s) = \gamma_s \cdot \log x_s$, then skip

$$\dot{x}_s = K_s \left[\frac{\gamma_s}{x_s} - \beta \cdot \sum_l H_s^l p_l(R_l, \sum_l H_s^l x_s) \right]$$

or

$$\dot{x}_s = \hat{K}_s \left[\gamma_s - \beta x_s \cdot \sum_l H_s^l p_l(R_l, \sum_l H_s^l x_s) \right]$$

- Recall that p_l is the packet dropping prob. at link l

- Then, $\sum_l H_s^l p_l$ is approximately the packet dropping prob. along the path of user s.

- Let $\gamma_s = x_s \sum_l H_s^l p_l(R_l, \sum_l H_s^l x_s))$ which represents the rate of packets that are lost. Then are lost. Then

$$\dot{x}_s = K_s \left[\gamma_s - \beta z_s \right]$$

— AIAD

② If $U_s(x_s) = -\frac{\gamma_s}{x_s^{\alpha}}$

$$\dot{x}_s = K_s \left(\frac{\alpha \gamma_s}{x_s^{\alpha+1}} - \beta \cdot \sum_l H_s^l P_L(R_L, \sum_l H_s^l x_s) \right)$$

- Recall that P_L is the packet dropping prob. at link l

- Then, $\sum_l H_s^l P_L$ is approximately the packet dropping prob along the path of user s.

- Let $z_s = x_s \sum_l H_s^l P_L(R_L, \sum_l H_s^l x_s))$ which represents the rate of packets that are lost. Then

$$\dot{x}_s = \tilde{K}_s \left(\alpha \gamma_s - \beta \cdot x_s^{\alpha} \cdot z_s \right)$$

If $\alpha = 1$, the latter represents AIMD.

(26)