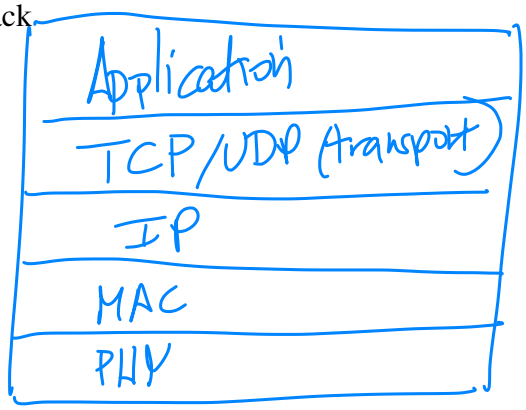
Week 10

Nicolo Michelusi

I. Congestion control in the internet

- Congestion control and TCP are practical examples of how to use convex optimization to design protocols.
- TCP/IP protocol stack



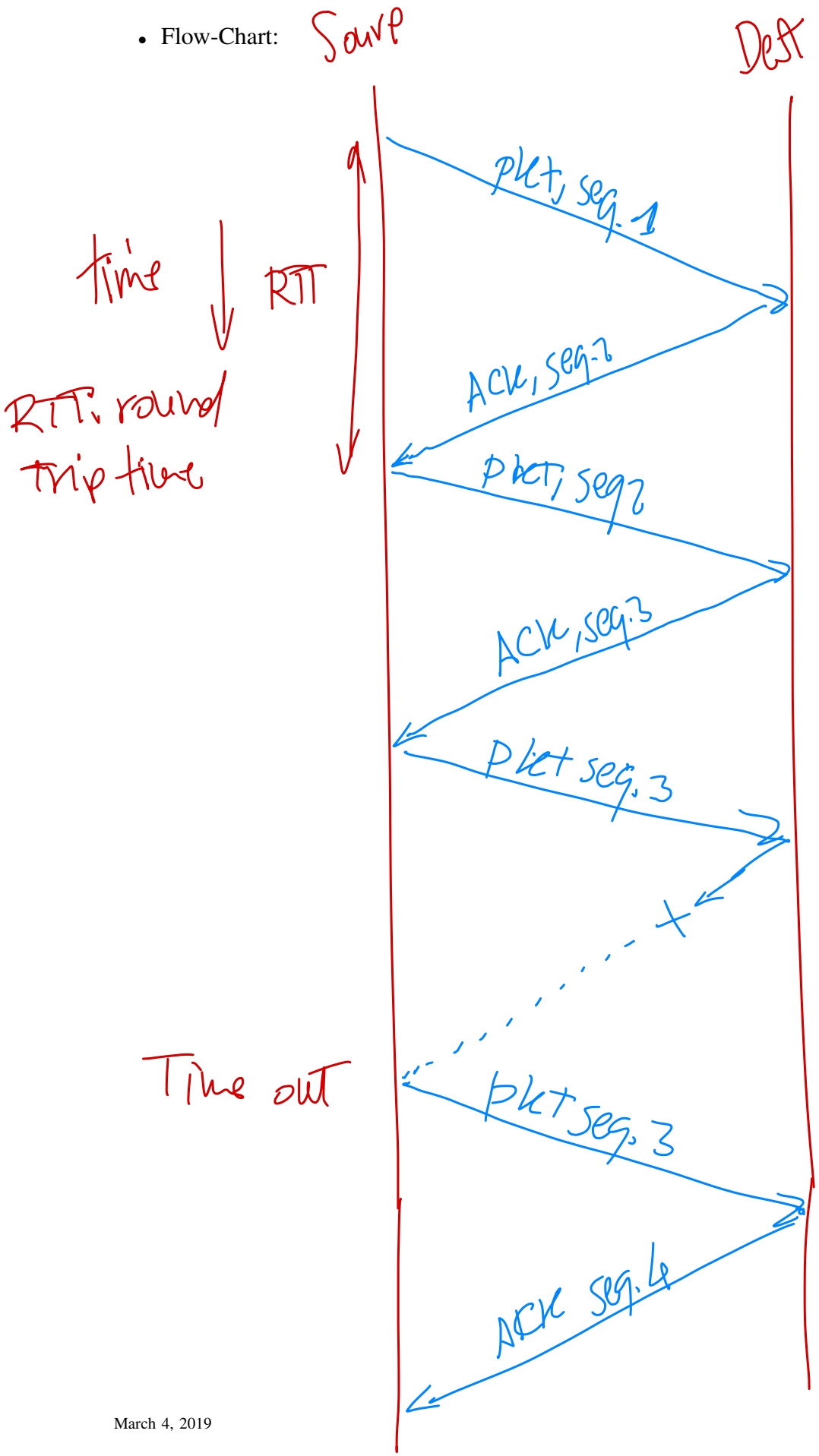
– IP provides best-effort packet delivery service over heterogeneous networks:

- Each node has an IP address
- Each data packet contains both source and destination IP addresses
- Routers will route the IP packet from source to destination
- IP is best effort: packets might be lost, duplicated, travel over loops, or arrive out of sequence; no guarantee

– TCP: provides a port number for each application and a connection-oriented, reliable, in-order packet delivery service over IP:

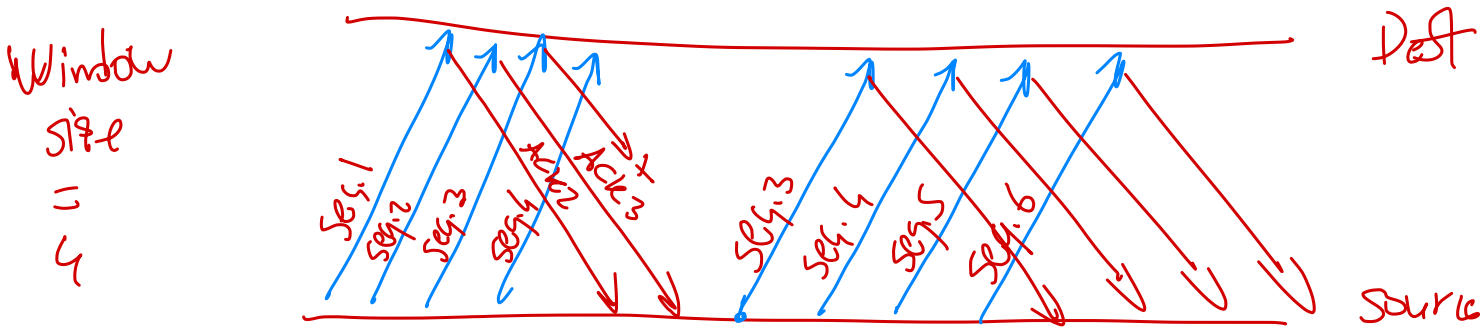
- Connection-oriented and in order: both end-points maintain a state of the connection
- Sequence numbers are used to maintain order and reliability: the source maintains the seq.# for the next byte to be sent; the receiver maintains the seq.# for the next byte expected; the two sync the seq.#
- The seq.# advances only after the packet is successfully acknowledged
- If the packet is not ACKed within a time-out period, the source will retransmit the packet.

- Flow-Chart:


Sourp
Dest
time
RTT
RTT: round trip time
Pkt, seq. 1
ACK, seq. 2
Pkt, seq 2
ACK, seq. 3
Pkt seq. 3
Time out
Pkt seq. 3
ACK seq. 4

- 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 ACK.



  - If the window is too small, throughput is small. If the window is too large, too many packets are injected in the network, resulting in congestion. Therefore, it is important to do TCP window control to adapt the window size to the current congestion level of the network.

- Congestion control:

  - to fully utilize the available bandwidth

  - to prevent the onset of congestion

  - to be fair to different flows

- TCP congestion control has two phases:

  1) Slow start:

  - used when starting a connection

  - increase window size by 1 at each ack

  - doubles the window size after each RTT (round trip time)

  - Stops when the window size exceeds a certain threshold, or when packets are lost/time-out

  2) Congestion avoidance:

  - on each new ack: increase window W by a fraction $1/W$ (increases by one unit after one RTT): $W \leftarrow W + 1/W$

  - on each time-out, cut window in half, $W \leftarrow W/2$

  - This scheme is known as AIMD (additive increase/multiplicative decrease)

  - more conservative than slow-start to avoid congestion

- These basic principles of TCP were published by Van Jacobson's 1988 paper after the first congestion collapse, and forms the basis of TCP today.

- Many researchers have tried to understand the behavior of TCP, until this paper, which viewed congestion control as the solution of an optimization problem

  Kelly, F. P., et al. "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability." The Journal of the Operational Research Society, vol. 49, no. 3, 1998, pp. 237-252. JSTOR, www.jstor.org/stable/3010473.

  Kelly proposed to formulate congestion control as the following problem, and view TCP as an iterative algorithm to solve it
- Formulation:

$$\max \sum_s U_s(x_s)$$
$$\text{s.t. } \sum_s H_{s,l} x_s \leq R_l, \ \forall l$$
$$x_s \in [m_s, M_s], \ \forall s.$$

  - x_s rate of user s
  - U_s: utility function
  - $H_{s,l}$: $= 1$ if user s uses link l, $= 0$ otherwise
  - R_l capacity of link l
  - The utility function is expressed as $U_s(x_s) = \frac{x_s^{1-\alpha}}{1-\alpha}, \ \alpha > 0$
  - However, this problem need to be solved in a distributed fashion: each source has only access to the packet loss rate of its own flow $\rightarrow$ duality
- Lagrangian:

$$L(x, \lambda) = -\sum_s U_s(x_s) + \sum_l \lambda_l (\sum_s H_{s,l} x_s - R_l)$$

- Dual objective function

$$g(\lambda) = \min_{x: x_s \in [m_s, M_s]} \sum_s \left[-U_s(x_s) + x_s \sum_l H_{s,l} \lambda_l \right] - \sum_l \lambda_l R_l$$

  so that each user solves:

$$x_s^* = \arg \max_{x_s \in [m_s, M_s]} \left[U_s(x_s) - x_s \sum_l H_{s,l} \lambda_l \right]$$

For instance, with $\alpha = 1$, $U_s(x_s) = \ln(x_s)$, hence

$$x_s^* = \left[\frac{1}{\sum_l H_{s,l}\lambda_l}\right]^+$$

projection into $[m_s, M_s]$

Note that λ_l can be interpreted as the unit price for using link l. Hence,

$$\sum_l H_{s,l}\lambda_l$$

is the total unit price incurred by user s, for each unit of traffic.

- To maximize dual function:

$$\lambda_l^{k+1} = \left[\lambda_l^k + \gamma(\sum_s H_{s,l}x_{s,k} - R_l)\right]^+$$

- This can be done locally at link l, by measuring the incoming traffic $\sum_s H_{s,l}x_{s,k}$ through the link

- In order to implement x_s^*, each user needs to know the total price $\sum_l H_{s,l}\lambda_l$

- prices are closely related to the queue-length at each link:

$$Q_l^{k+1} = \left[Q_l^k + \sum_s H_{s,l}x_{s,k} - R_l\right]^+$$

This is simply the queue-evolution equation; price is simply a scaled version of the queue length.

- Queue length can be communicated via:

– Explicit control messages

– Packet drops or marks

- $\Rightarrow$ Random exponential marking (REM)

- Each packet is marked at link l with probability $1 - e^{-q_l}$

- The probability that the packet of source s is marked after it passes through the path is

$$1 - e^{-\sum_l H_{s,l}q_l}$$

hence, by counting the fraction of packets that are being marked, the source can estimate $\sum_l H_{s,l}q_l$, hence the price of the route.

- Suggested readings:

- S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," in IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 861-874, Dec. 1999.

- S. H. Low. 2003. A duality model of TCP and queue management algorithms. IEEE/ACM Trans. Netw. 11, 4 (August 2003), 525-536.

II. Cross-layer formulation

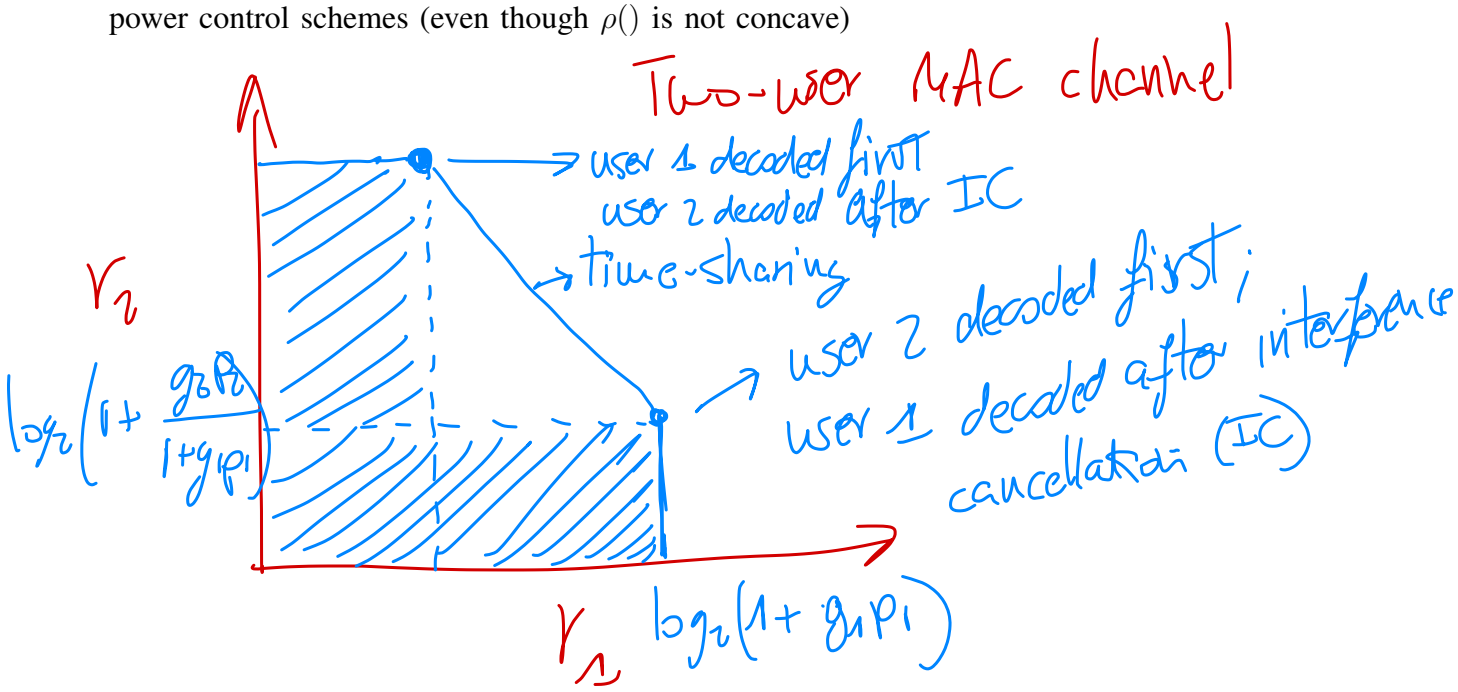
- In an optimization approach, it is not difficult to incorporate controls at multiple layers into a unified optimization problem:
  - Physical layer: power control, rate, bandwidth
  - MAC: scheduling
  - Network layer: multipath routing, node-balance equations
  - Transport layer: utility maximization
- Joint congestion control and scheduling

$$\max_{x,r} \sum_s U_s(x_s)$$

$$\text{s.t.} \sum_s H_{s,l} x_s \le r_l, \ \forall l$$

$$r \in \text{conv}(r | r = \rho(p), p \in \Pi)$$

- Π is a set of feasible power control policies; conv() is achieved by time-sharing over power control schemes (even though $\rho()$ is not concave)



- Lagrangian formulation (do not include last constraint)

$$L(x,r,\lambda) = -\sum_s U_s(x_s) + \sum_l \lambda_l (\sum_s H_{s,l} x_s - r_l)$$

so that x solves

$$x_s^* = \arg\max U_s(x_s) - x_s \sum_l H_{s,l}\lambda_l$$

and r solves

$$r^* = \arg \max_{r \in \text{conv}(r|r=\rho(p), p\in\Pi)} \sum_l \lambda_l r_l$$

- Each user maximizes the net utility

- "Max-weight scheduling": the schedule is chosen to maximize the overall value of the available resources

- Optimization of dual variables

$$\lambda_l^{k+1} = \left[\lambda_l^k + \gamma(\sum_s H_{s,l} x_{s,k} - r_{l,k})\right]^+$$

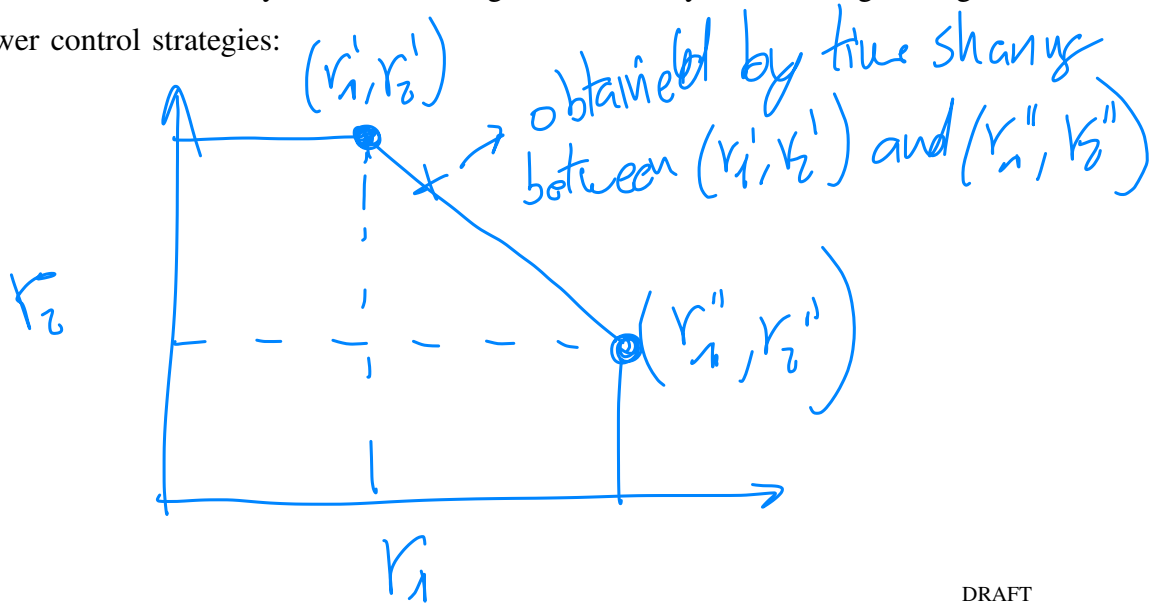
- Suggested reading:

Xiaojun Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), Nassau, 2004, pp. 1484-1489 Vol.2.

- Max-weight scheduling:

$$\max_{r \in \text{conv}(r|r=\rho(p), p\in\Pi)} \sum_l \lambda_l r_l = \max_{p\in\Pi} \rho(p)$$

- note that, under the optimal prices λ_l^*, the optimal r might not be achievable by any $p \in \Pi$: in this case, r is achievable only in a time-average sense, i.e. by time-sharing among two or more power control strategies:



- Channel variations: the solution can be extended to the case with channel variations; in this case, the capacity constraint of each link is expressed as

$$\sum_s H_{s,l} x_s \leq \sum_i \theta_i r_l^{(i)}, \ \forall l,$$

where

$$r^{(i)} \in \text{conv}(\{r | r = \rho(p; g_i), \ p \in \Pi\})$$

and $\theta_i = \mathbb{P}(g = g_i)$ is the channel probability distribution of the channel g.

To address this scenario, use the idea of stochastic approx to replace the mean value by an unbiased estimate.

- If the channel distribution is known, $r^{(i)}$ is the maximizer of

$$\max_{p \in \Pi} \rho(p; g_i)$$

when the channel state is g_i, and λ is updated as

$$\lambda_l^{k+1} = \left[\lambda_l^k + \gamma (\sum_s H_{s,l} x_{s,k} - \sum_i \theta_i r_{l,k}^{(i)}) \right]^+$$

- With stochastic approx, after observing the channel state $g = g_{i_k}$ in slot k, the network chooses $r^{(i_k)}$, and $\sum_i \theta_i r_{l,k}^{(i)}$ is replaced with the unbiased estimate $r_{l,k}^{(i_k)}$ measured under the current channel state g_{i_k}, yielding

$$\lambda_l^{k+1} = \left[\lambda_l^k + \gamma (\sum_s H_{s,l} x_{s,k} - r_{l,k}^{(i_k)}) \right]^+$$

- Random arrivals: consider the case where packets arrive randomly. Then, the link controller does not know the rate x_s, but only the stochastic number of packet arrivals $\sum_s H_{s,l} A_{s,k}$ arriving at link l in slot k, where $A_{s,k}$ is the number of packets generated at user s in slot k. Note that

$$\mathbb{E}[A_{s,k}] = x_s$$

is an unbiased estimator of the packet arrival rate, hence we can replace it in the update of λ, yielding the following stochastic approximation algorithm:

$$\lambda_l^{k+1} = \left[\lambda_l^k + \gamma (\sum_s H_{s,l} A_{s,k} - r_{l,k}^{(i_k)}) \right]^+$$

- Routing: we can also optimize the routing algorithm. Let

  - x_s: rate of user s

  - f_s, d_s source-destination pair of user s

  - $r_{i,j}^d$: amount of capacity on link (i,j) allocated for data towards destination d. This is set to zero if $i = d$, i.e. $r_{i,j}^i = 0$ (since the destination is the sink of a given flow).

$$\begin{aligned} &\max \sum_s U_s(x_s) \\ &\text{s.t.} \sum_{s:f_s=i,d_s=d} x_s + \sum_j r_{j,i}^d \leq \sum_j r_{i,j}^d, \ \forall d, \forall i \neq d \\ &[\sum_d r_{i,j}^d]_{\forall i,j} \in \text{conv}(\{\rho(p) : p \in \Pi\}) \end{aligned}$$

- Lagrangian formulation:

$$L(x,r,\lambda) = -\sum_s U_s(x_s) + \sum_{d,i} \lambda_{d,i} \left[\sum_{s:f_s=i,d_s=d} x_s + \sum_j r_{j,i}^d - \sum_j r_{i,j}^d \right]$$

  where $\lambda_{i,i} = 0$.

  Hence, each user solves

$$x_s^* = \arg\max U_s(x_s) - x_s \sum_{d,i} \chi(f_s = i, d_s = d) \lambda_{d,i}$$

  the schedule r should be chosen to maximize

$$r^* = \arg \max_{r:[\sum_d r_{i,j}^d]_{\forall i,j} \in \text{conv}(\{\rho(p):p\in\Pi\})} \sum_{i,j,d} r_{i,j}^d \left[\lambda_{d,i} - \lambda_{d,j}\right]$$

  The dual variables can be updated as

$$\lambda_{d,i}^{(k+1)} = \left[\lambda_{d,i}^{(k)} + \gamma \left(\sum_{s:f_s=i,d_s=d} x_s + \sum_j r_{j,i}^d - \sum_j r_{i,j}^d \right) \right]^+, \ d \neq i.$$

  Note that $\lambda_{d,i}$ can again be interpreted as the backlog of packets at node i that are destined to node d, hence it can be estimated using techniques such as REM or control messages.

- Backpressure routing: the scheduling component corresponds to the so called backpressure routing:

$$\max_{r:[\sum_d r_{i,j}^d]_{\forall i,j} \in \text{conv}(\{\rho(p):p\in\Pi\})} \sum_{i,j,d} r_{i,j}^d \left[\lambda_{d,i} - \lambda_{d,j}\right]$$

  the difference $\lambda_{d,i} - \lambda_{d,j}$ is called *differential backlog* (recall that λ can be interpreted as the scaled backlog)

- To solve this problem, note that, for a given link (i,j) and fixed $\sum_d r_{i,j}^d = r_{i,j}$, there is no reason to schedule $r_{i,j}^d > 0$, unless

$$\lambda_{d,i} - \lambda_{d,j} > 0 \text{ and } d \in \arg\max\{\lambda_{\delta,i} - \lambda_{\delta,j}\}.$$

Hence, on each link (i,j), we only schedule the packets that correspond to the destination with the largest differential backlog. This decision determines how packets are routed.

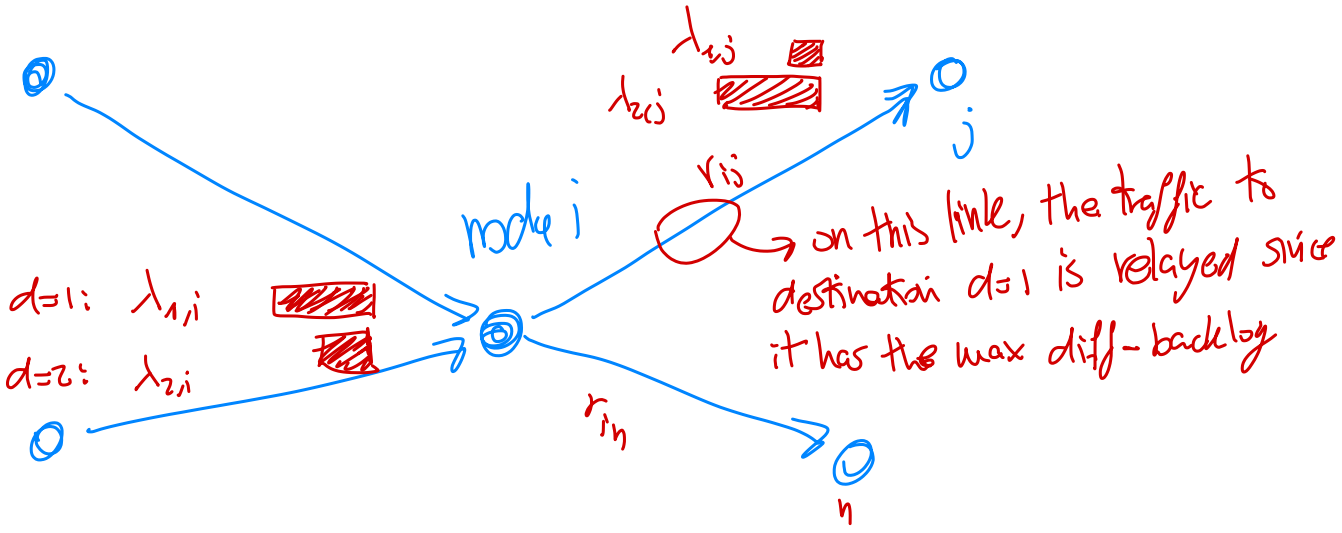
- Then, the scheduling problem becomes

$$\max_{r \in \text{conv}(\{\rho(p): p \in \Pi\})} \sum_{i,j} r_{i,j} \max_d [\lambda_{d,i} - \lambda_{d,j}]$$

- As earlier, can also be extended to the case with channel variations.

- Note that, even if the dual vars converge, the routing and scheduling decisions do not converge. Packets will be served in a time-interleaved fashion, based on the instantaneous backlog. Link will be turned on/off in a time-interleaved fashion.

- In this sense, the primal variables r are optimal in the sense that the node-balance equations are satisfied on average.



- This has effect of reducing the backlog associated to d=1 on node i, and augmenting the backlog of node j (closer to the destination)
- Also, node i waits for node j to reduce its backlog for d=2 before relaying the traffic destined to d=2.