CS 4251/6250 · Quiz 2 — iPad problem set

Work these by hand, in order, right after reading the matching section of **exam2_notes.html**. Every problem number matches a dashed orange box in the notes; the solution is in the notes under that box (*Show solution*). Write your answer before you open it.

- **Units (same as Quiz 1):** 1 Mb/s = 10^6 b/s · 1 MB = 10^6 bytes · 1 kB = 1000 bytes · 1 byte = 8 bits.
- **MCQs are “select all that apply”, +1 per correct mark, −1 per wrong mark, floor 0 per question.** Practice leaving a choice blank when unsure.
- Mark each problem ✓ (clean), ~ (needed a peek), ✗ (wrong). Redo every ~ and ✗ on Day 3.
- The last section is a full **mock quiz**: do it timed (75 min) on Day 3 without notes.

1 · Congestion control after Quiz 1 — model-based vs PCC (Allegro / Vivace)

Notes §1 · Lecture 9 · HW2 Q1

1.1 What should the sender do after loss?

A sender at 100 Mb/s observes 1% loss. Which statements are correct?

☐ **A.** If the loss is random, non-congestive loss with spare capacity, the useful response is to increase the rate.

☐ **B.** If another, more aggressive flow is occupying the shared buffer, the useful response is to back off substantially.

☐ **C.** Reno's rule (halve on loss) embeds one interpretation of loss — that this flow overloaded the link — and applies it in every scenario.

☐ **D.** PCC resolves the ambiguity by first identifying which of the four hidden scenarios is happening and then applying the matching response.

☐ **E.** If a shallow buffer overflowed from a small burst while the link is far from capacity, a slight decrease or better pacing is the useful response.

1.2 Allegro randomized controlled trials

A PCC Allegro sender is at r = 50 Mb/s with ε = 0.04. Each monitor interval runs two randomized pairs of trials.

a. What two rates are tested?
b. Results: pair 1 (48 then 52): U = 40.1, 41.0. Pair 2 (52 then 48): U = 40.6, 39.8. What does Allegro do?
c. Results: pair 1 (52 then 48): U = 41.2, 40.3. Pair 2 (48 then 52): U = 40.9, 40.2. What does Allegro do now, and what changes for the next MI?
d. Why is each trial's utility only computed about one RTT after that trial's rate begins?
e. Why randomize the order inside each pair?

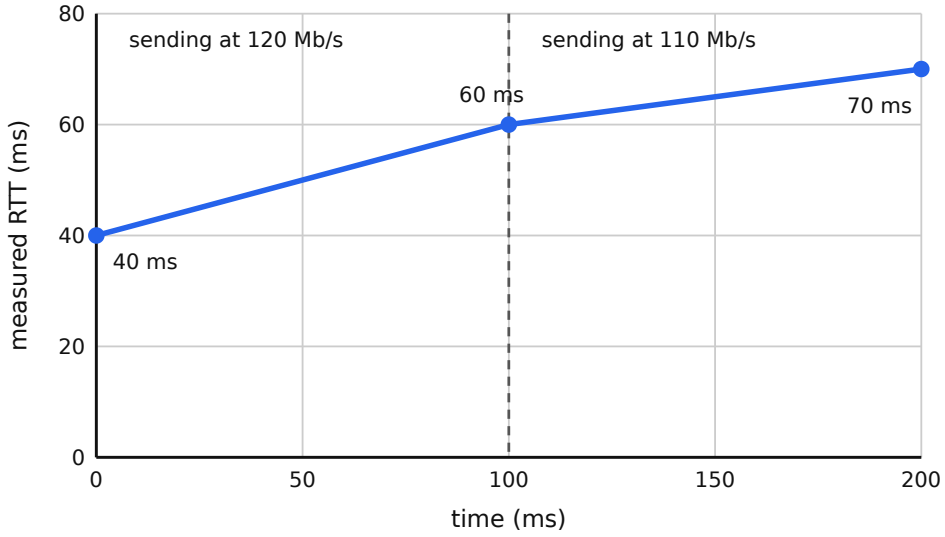
1.3 Allegro's utility and the 5% loss cliff

Allegro uses $u(x) = T \cdot S(L) - x \cdot L$ with $T = x(1-L)$ and $S(L) = 1 / (1 + e^{\alpha(L - 0.05)})$, $\alpha = 100$.

a. Compute u for x = 100 Mb/s at L = 1%, 4%, 6%.
b. Explain what this implies when Allegro shares a bottleneck with CUBIC.
c. Name the metric missing from this utility and the consequence.

1.4 Vivace: RTT gradient vs RTT magnitude

Capacity C = 100 Mb/s; the queue is non-empty and doesn't overflow. For 100 ms the sender sends at 120 Mb/s and RTT rises 40 → 60 ms. For the next 100 ms it sends at 110 Mb/s and RTT rises 60 → 70 ms. Use the simplified utility $u(x) = x - b \cdot x \cdot (dRTT/dt)$ with b = 2 (no loss).



a. Average RTT in each interval? Which rate would a utility penalizing average RTT prefer?
b. RTT slope in each interval, and u(120), u(110)? Which rate wins?
c. Show that both slopes match dRTT/dt = (x − C)/C. Using that, what is u(90)? Why is that term a reward?

1.5 Vivace step vs Allegro step

A Vivace sender at r = 80 Mb/s tests 76 and 84 Mb/s and measures U(76) = 61.0, U(84) = 58.6. Its confidence amplifier is θ = 10 (Mb/s per unit of gradient).

a. Estimate the gradient g and the proposed change Δr. New rate?
b. The next decision again points down with g = −0.25, and θ has doubled. New rate?
c. Allegro with a fixed step of ε·r (ε = 0.05) — where does it go from 80? When is a fixed step worse?
d. Why does Vivace bound Δr relative to r?

1.6 PCC with a delay penalty (HW2 Q1.2 style)

$U = g - \beta \cdot q$ (g goodput in Mb/s, q average queueing delay in ms). Measurements: low rate $g = 60$, $q = 2$; high rate $g = 72$, $q = 14$.

a. Utilities and choices for $\beta = 0.5$ and $\beta = 2$. Which gives lower delay?
b. For which β are the two rates tied?
c. Does a larger β always select a strictly lower-delay rate than a smaller β? Give a numerical counterexample (both configurations choose from the same two measured options).
d. Interpret β's unit.

1.7 BBRv1 after a route change (HW2 Q1.1 with numbers)

BBRv1, bottleneck 100 Mb/s (unchanged), continuous demand. The route changes: RTprop 20 ms → 80 ms. Assume cwnd = 2·BtlBw_est·RTprop_est and ignore pacing-gain cycling. The max-BW filter spans 10 rounds; the min-RTT filter spans 10 s.

a. Right after the change: cwnd (in Mb and in kB), the BDP the new path needs, and the throughput the flow can achieve.
b. What happens to BtlBw_est when the old samples leave the BW filter? Next cwnd and throughput? Continue one more step.
c. When does it recover, and why?
d. Propose a modification and how it distinguishes longer propagation delay from queueing.

1.8 **Learning rates, learning policies, running AI**

Which statements are correct?

☐ **A.** Remy updates its table of control rules during live operation on the Internet.

☐ **B.** A fixed learned policy can still produce a constantly changing sending rate.

☐ **C.** Moving a neural congestion controller to user space lets it react to every ACK at microsecond timescales.

☐ **D.** Linux kernel code generally avoids floating point, one reason heavy learned controllers run in user space behind an API.

☐ **E.** PCC Vivace learns its sending rate through live experiments.

☐ **F.** Because PCC is rate-based, it cannot limit the number of packets in flight.

☐ **G.** A good way to find the best rate is a binary search that alternates very low and very high rates.

1.9 **Exploration vs exploitation (HW2 Q1.3)**

Which statements about experiment (trial) length in BBR/PCC-style probing are correct?

☐ **A.** A trial shorter than one RTT risks attributing performance to the wrong rate.

☐ **B.** Longer trials always produce better decisions.

☐ **C.** Longer trials average out measurement noise, but if conditions change during the trial the result mixes regimes and may be stale.

☐ **D.** Short flows may finish before PCC has run enough trials to converge.

☐ **E.** Exploration is free when the network is idle, so trials should be as long as possible.

2 · QUIC

Notes §2 · Lecture 10 · HW2 Q2

2.1 Time to first response

RTT = 30 ms; ignore serialization and processing. A client requests a small object (fits in one packet). When does the response arrive (time from the client's first packet) for:

a. TCP + TLS 1.3, fresh connection?
b. QUIC, fresh connection?
c. QUIC 0-RTT resumption, accepted by the server?
d. What two conditions does 0-RTT require?
e. Why can't the industry just merge TCP's SYN with the TLS ClientHello?
f. For (a)–(c), when can the client *send* its HTTP request?

2.2 Head-of-line blocking (HW2 Q2.1 style)

Objects: A = image (3 packets), B = text (1 packet), C = caption (2 packets). Each displays as soon as all its bytes are delivered to the application; they're independent. Transmission order and arrival times are identical for every transport:

packet	A1	B1	C1	A2	C2	A3
arrives (ms)	lost → retransmission at 70	10	14	18	22	26

a. Display times of A, B, C with one TCP connection (HTTP/2-style multiplexing).
b. With QUIC, one stream per object.
c. With three separate TCP connections (HTTP/1.1-style). Why not just do that?

2.3 Packet numbers vs stream offsets

Packet 40 = [A: off 0, len 1000]; packet 41 = [A: off 1000, len 1000] + [B: off 0, len 500]; packet 42 = [A: off 2000, len 1000]. Packet 41 is lost. The ACK frame reports 40 and 42.

a. What can the receiver hand to the application for stream A? For stream B?
b. The sender retransmits the lost data. What packet number does it use, and which identifiers stay the same? Must it reuse the same packetization?
c. What would TCP do differently with the bytes after the gap?
d. Why is it useful that the retransmission has a new packet number?

2.4 Stream scheduling under a congestion window (HW2 Q2.2 style)

All data is buffered at t = 0: A (image, 4 chunks) and B (text, 2 chunks), one chunk per packet. RTT = 40 ms (20 ms each way), ACKs immediate, no loss. cwnd starts at **2 packets** and grows by 1 after each RTT. Send whenever the window permits. Fill a table (time | ACKs received | cwnd | packets sent) and give A's and B's display times for:

a. B-first
b. round robin, starting with A
c. A-first

d. Why does prioritization matter even though all the data is already buffered? What is the cost?

2.5 Connection migration

A phone moves from Wi-Fi to cellular mid-download. Which statements are correct?

☐ **A.** A TCP connection is identified by its 4-tuple, so the client IP change breaks it.

☐ **B.** QUIC packets carry a connection ID, letting the server associate packets from the new address with the existing connection state.

☐ **C.** Because the QUIC connection survives, it should keep its Wi-Fi congestion window on the cellular path.

☐ **D.** Path validation (PATH_CHALLENGE / PATH_RESPONSE with random data) checks that the new path works.

☐ **E.** The client must keep using exactly the same connection ID on the new path so the server can recognize it.

2.6 QUIC loss detection

Packets 20–26 are sent. ACK frames acknowledge 21, 22, 23, and later 25. Use a packet threshold of 3 (RFC 9002 default).

a. Which packets are declared lost by the packet threshold? Which are still undecided?
b. A short flow's last packet's ACK is lost and nothing else is outstanding. Which mechanism helps, and what does its expiry do?
c. Why does QUIC's ACK format cost some goodput compared with TCP's?

2.7 **Why QUIC looks the way it does**

Which statements are correct?

☐ **A.** A QUIC stream can still be blocked by its own missing bytes.

☐ **B.** Because streams are independent, a congested bottleneck slows down only the stream whose packet was dropped.

☐ **C.** QUIC runs over UDP partly because UDP is broadly allowed by Internet paths and supported by endpoint operating systems.

☐ **D.** Encrypting most transport header fields makes it harder for middleboxes to depend on or rewrite them, reducing future ossification.

☐ **E.** Deploying new QUIC behavior typically requires a kernel upgrade on both endpoints.

☐ **F.** Different QUIC implementations can have interoperability problems even though they follow the same RFCs.

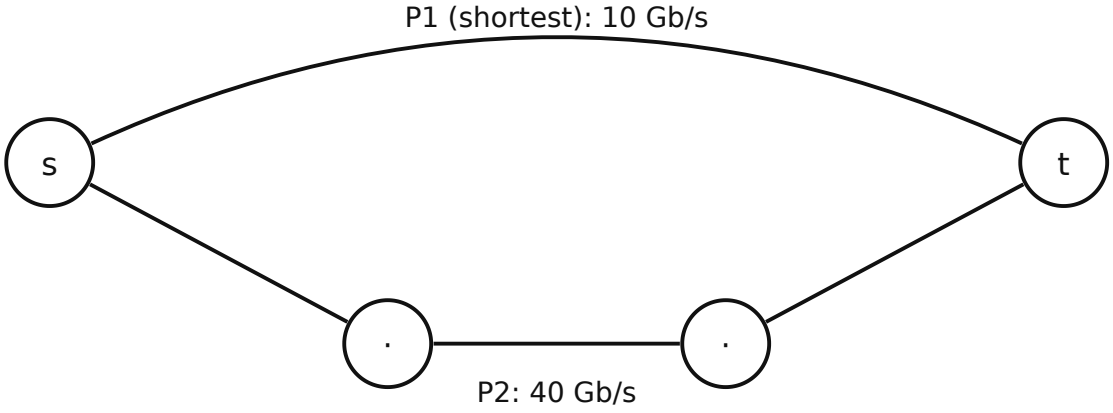
3 · Traffic engineering foundations

Notes §3 · Lecture 11 · HW2 Q3

3.1 Shortest path vs ECMP vs min-MLU

s→t has two disjoint paths: P1 (the shortest) with 10 Gb/s and P2 with 40 Gb/s. Demand 30 Gb/s.

demand s → t: 30 Gb/s (later 55 Gb/s)



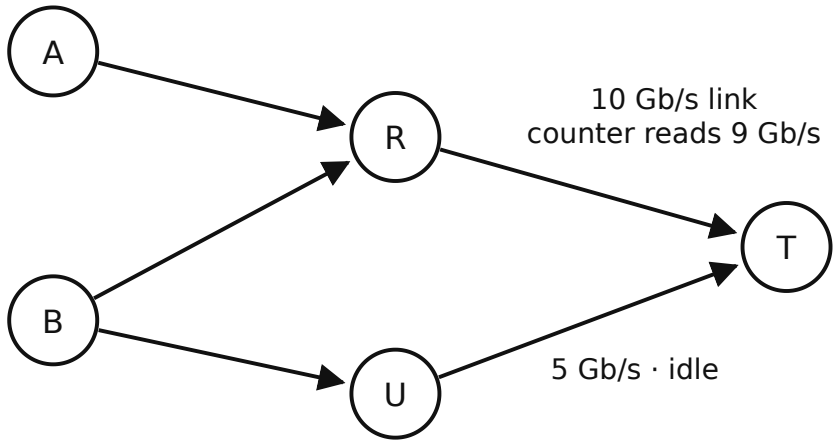
a. Utilization of each path and MLU under shortest-path routing; under ECMP.
b. The min-MLU split and MLU*.
c. Demand becomes 55 Gb/s. What can TE do?
d. Give the professor's main reason why minimizing peak utilization is a good TE objective.

3.2 Min-MLU on three parallel paths

Three disjoint paths with capacities 10, 20, 30 Gb/s carry a 42 Gb/s demand. Find the min-MLU allocation and MLU*. State the general rule.

3.3 Measurement: can link counters reveal demand?

A and B both reach T through router R over a 10 Gb/s link whose counter reads 9 Gb/s. B also has an idle 5 Gb/s alternative via U.



a. What are the possible (A, B) demand pairs?
b. Why does this matter for TE?
c. Why is “link load” also not the same as “delivered rate”?

3.4 Staged swap with headroom

Two 10 Gb/s paths. A = 7 Gb/s on P1, B = 7 Gb/s on P2. Goal: swap them. Each move shifts part of one aggregate and must keep every path ≤ 10 Gb/s, even after only that move completes.

a. What happens if A moves entirely first?
b. Give a safe sequence of moves. What is the minimum number of moves?
c. What does the plan require from the network and the controller?

3.5 Deferring batch traffic

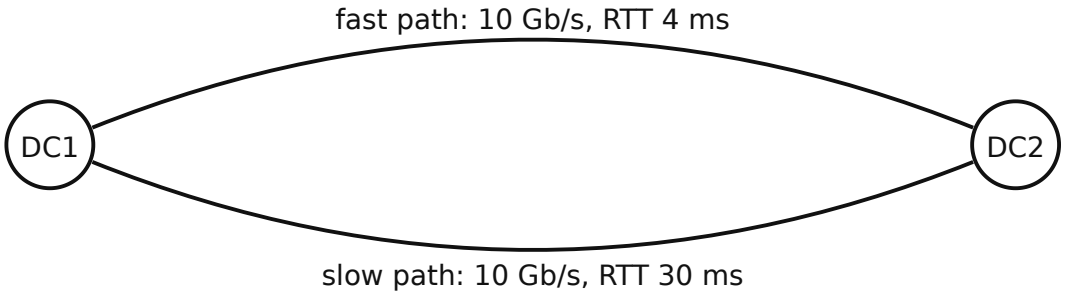
Path capacity 20 Gb/s. Urgent traffic: 16 Gb/s in hour 1, 12 Gb/s in hour 2. A batch transfer needs 6 Gb/s for one hour (6 Gb/s·h of volume) and must finish by the end of hour 2.

a. Run the batch in hour 1: utilization? Defer it to hour 2?
b. If the batch may be spread over both hours at any rates, what schedule minimizes the peak utilization?

3.6 RPC vs replication (HW2 Q3 twin)

Two paths between DCs: fast (10 Gb/s, RTT 4 ms) and slow (10 Gb/s, RTT 30 ms). Policy: RPC traffic needs RTT < 10 ms; replication is latency-tolerant. Demands: RPC 5 Gb/s, replication 9 Gb/s.

demands DC1 → DC2: RPC 5 Gb/s (needs RTT < 10 ms) · replication 9 Gb/s



a. The controller aggregates both classes into one demand and routers use ECMP. Link loads, MLU, and RPC on the slow path?
b. Find a configuration that meets the policy. What is the best MLU it can reach? Can you get it by just changing how the aggregate is split? What information and forwarding state does it need?
c. RPC grows to 12 Gb/s. Redo (a) and (b).

3.7 **TE vs congestion control**

Which statements are correct?

☐ **A.** TE typically acts on aggregates every few seconds to minutes; congestion control acts per flow every RTT.

☐ **B.** Minimizing MLU is attractive mainly because it minimizes the average path latency.

☐ **C.** If total demand exceeds the capacity of the minimum cut, a better TE split can still carry all of it.

☐ **D.** ECMP balances flows, which isn't the same as balancing bytes.

☐ **E.** If both the current and the target configurations are feasible, any order of router updates is safe.

☐ **F.** Besides routing, TE's control surfaces include traffic management such as rate limiting or admission.

4 · MPLS and SDN

Notes §4 · Lecture 12

4.1 The MPLS shim header

a. List the MPLS header fields with sizes. What's the total?
b. How many label values can one router distinguish?
c. A packet carries a 3-label stack. Overhead in bytes, and as a % of a 1500-byte packet? How does a router know which label is the last?
d. Give two reasons label switching helps TE beyond raw lookup speed.

4.2 Walking MPLS forwarding tables

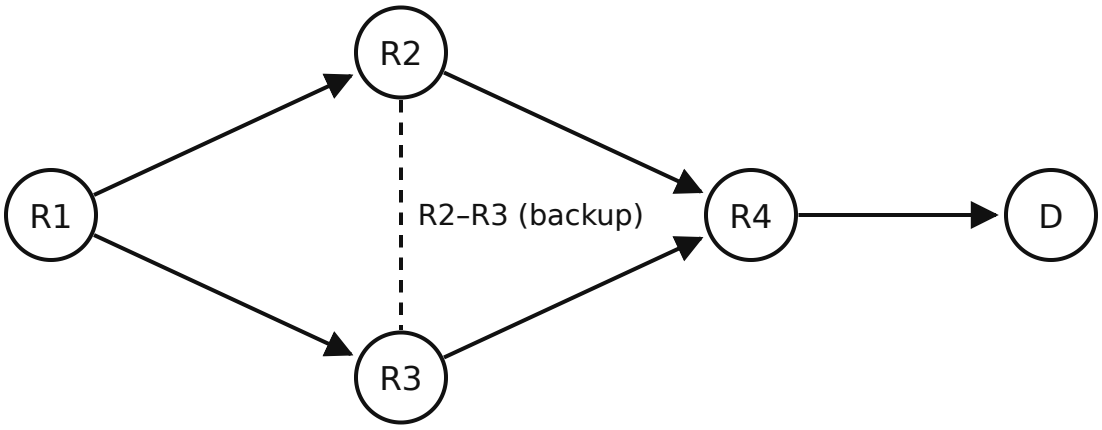
Ingress R1 classifies two FECs to destination network D:

R1: `video→D: push 17, out to R2` · `backup→D: push 23, out to R3`

R2: `in 17 → swap 31, out to R4` · R3: `in 23 → swap 44, out to R4`

R4: `in 31 → pop, out to D` · `in 44 → pop, out to D`

video FEC: R1→R2→R4→D · backup FEC: R1→R3→R4→D



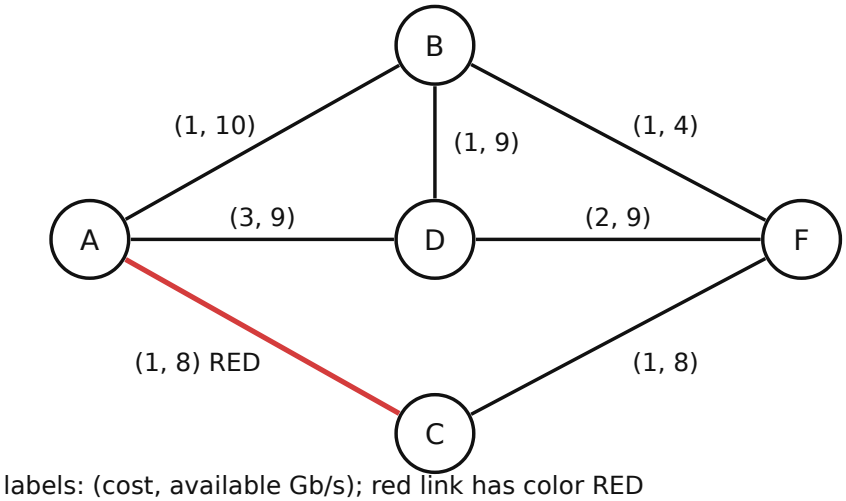
a. List the label each FEC carries on every link.

b. Both FECs reach R4 and leave to D. Why do they carry different labels at R4, and is it a problem that R3 could also have used label 31 internally?

c. The R2–R4 link fails. R2 has a precomputed backup “in 17 → swap 23, out to R3” (R2–R3 link exists). Trace a video packet.

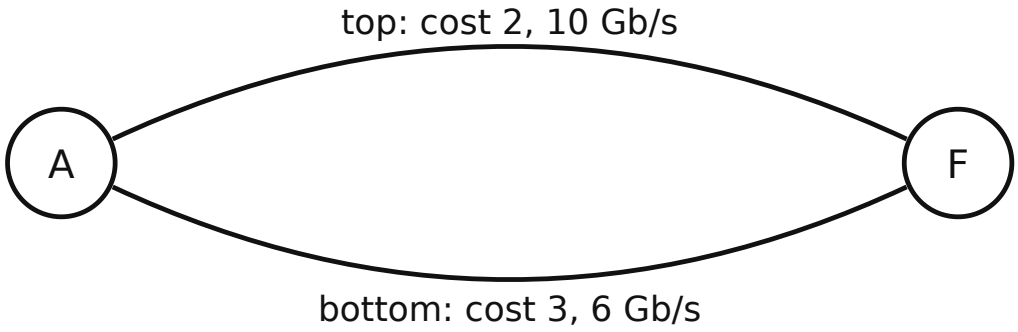
4.3 Constraint-based path selection (CSPF)

Undirected links (cost, available Gb/s, color): A–B (1, 10), B–F (1, 4), A–C (1, 8, RED), C–F (1, 8), A–D (3, 9), D–F (2, 9), B–D (1, 9).



labels: (cost, available Gb/s); red link has color RED

trunks X = 6 Gb/s and Y = 8 Gb/s, single LSPs



a. Trunk T1: A→F, 6 Gb/s, exclude RED. Prune, then find the path. What would plain shortest path have picked?
b. Then trunk T2: A→F, 5 Gb/s, no color constraint, placed after T1 is reserved. Path?
c. Separate network: A→F via top path (cost 2, 10 Gb/s) or bottom path (cost 3, 6 Gb/s). Trunks X = 6 and Y = 8 Gb/s, each a single LSP. Place them X then Y, then Y then X. Lesson?

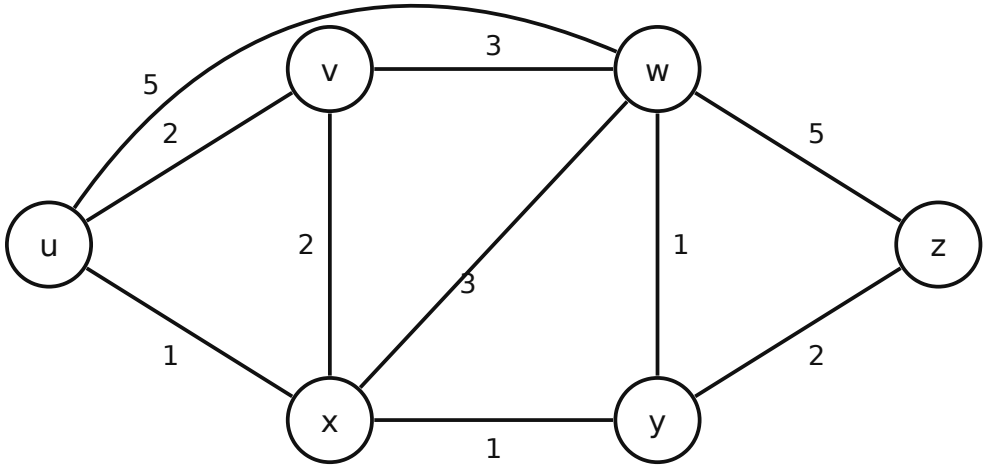
4.4 Resilient trunk reservations

A 120 Gb/s trunk can use four parallel 400 Gb/s links.

a. Reserve on one link only: what happens on a failure?
b. Must survive any single link failure without recomputation: per-link reservation and total reserved?
c. Any two failures?
d. How do operators keep this from wasting capacity?

4.5 TE with only link weights

Undirected weights: u–v 2, u–x 1, u–w 5, v–x 2, v–w 3, x–w 3, x–y 1, w–y 1, w–z 5, y–z 2.



a. Least-cost u→z path and cost?

b. The operator wants u→z on u–v–w–z. Propose weight changes that make it the unique least-cost path; check the closest competitors.

c. What are the side effects, and what two TE wishes can't destination-based link-state routing express at all?

4.6 SDN flow table with priorities

prio	match	action
30	src = 10.1.2.3	send to controller
20	src = 1.2.*.*	drop
10	dst = 3.4.*.*	forward(2)
0	*	forward(1)

What happens to: (a) 1.2.9.9 → 3.4.5.6 (b) 10.1.2.3 → 3.4.0.1 (c) 7.7.7.7 → 3.4.1.1 (d) 7.7.7.7 → 8.8.8.8? (e) What changes for (a) if the priorities of the drop and forward(2) rules are swapped? (f) Which table feature supports TE measurement?

4.7 MPLS and SDN in practice

Which statements are correct?

☐ **A.** MPLS and SDN are alternatives: a network uses one or the other.

☐ **B.** A centralized controller (PCE) can compute paths and program an MPLS data plane.

☐ **C.** Under purely centralized control, link-failure recovery takes at least a report to the controller, recomputation, and reprogramming, often hundreds of ms or more.

☐ **D.** The southbound API (e.g., OpenFlow) defines what in a switch is controllable.

☐ **E.** A logically centralized SDN controller must physically run on a single machine.

☐ **F.** MPLS labels have network-wide meaning, so every router must use the same label for an LSP.

☐ **G.** In distributed MPLS TE, the ingress router uses RSVP-TE to set up and reserve the LSP at downstream routers, using bandwidth information flooded by modified OSPF/IS-IS.

☐ **H.** When an SDN switch sees a link fail, it recomputes its own routes locally and then informs the controller.

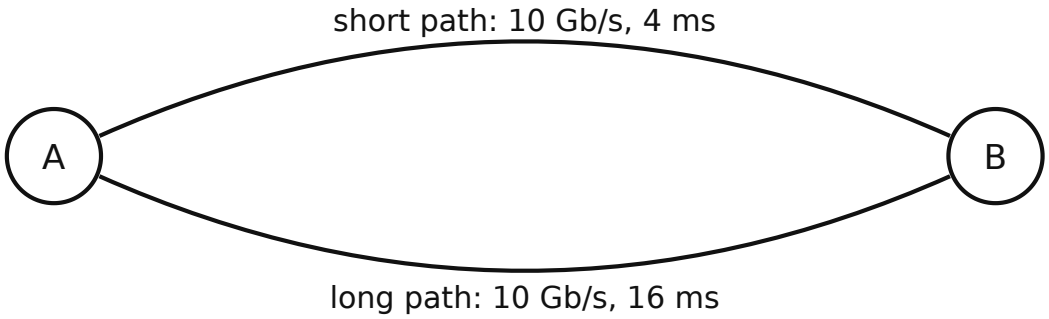
5 · Meta EBB

Notes §5 · Lectures 12–14 · HW2 Q5

5.1 Min latency vs min MLU

DC A sends 15 Gb/s to DC B over two disjoint paths: short (10 Gb/s, 4 ms) and long (10 Gb/s, 16 ms).

DC A → DC B: 15 Gb/s



a. The minimum-latency allocation: split, MLU, mean fixed latency.
b. The min-MLU allocation: split, MLU, mean latency.
c. Which objective does Meta use for gold, and which for silver/bronze? Why restrict silver/bronze to K-shortest paths?

5.2 Gold first, then silver, then bronze

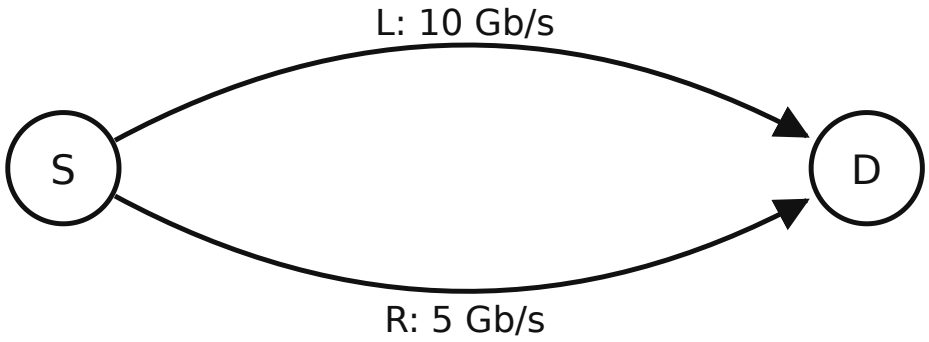
Paths: short (10 Gb/s, 5 ms) and long (10 Gb/s, 20 ms). Gold demand 6 Gb/s with a 2 Gb/s burst margin; silver 8 Gb/s; bronze 2 Gb/s. Allocate in priority order. Gold uses shortest path. Silver and bronze minimize the maximum link utilization given what is already placed (count gold's margin as used).

a. Gold placement and what's left.
b. Silver placement and MLU.
c. Bronze placement and the final MLU.
d. Later, bronze actually sends 9 Gb/s on its configured paths. What happens, and what mechanisms protect gold and silver? What does Meta add so bronze owners don't just see mysterious loss?

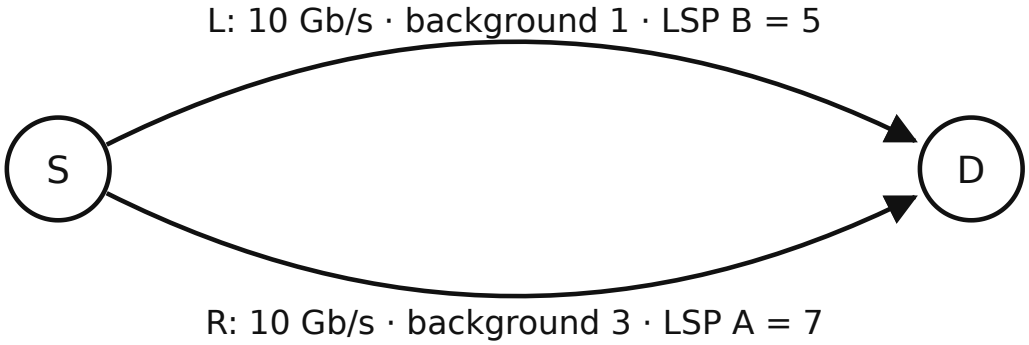
5.3 HPRR: granularity and getting stuck (HW2 Q5.1–Q5.2 twin)

HPRR reroutes one LSP at a time and accepts a move only if the **max utilization along the new route after the move** is lower than the **max utilization along the LSP's current route before the move**. Links are directed and parallel S→D.

(a) demand S → D: 9 Gb/s, no background



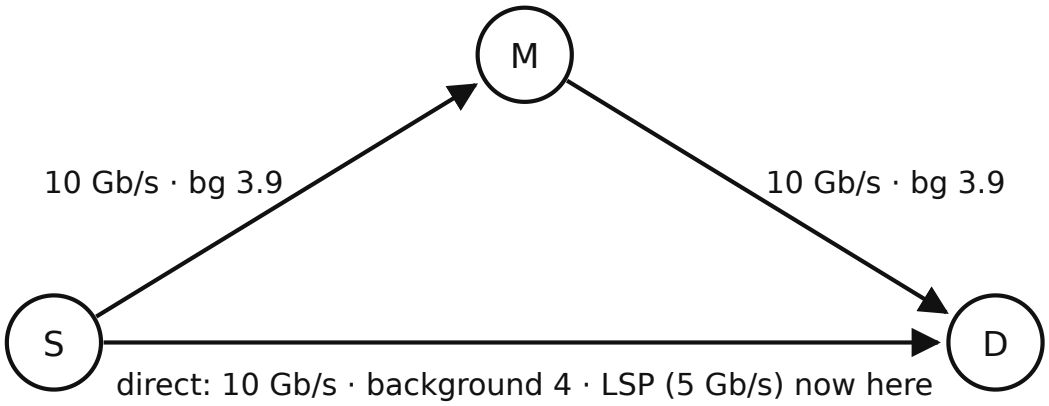
(b) unsplittable LSPs; initial placement shown



a. 9 Gb/s S→D over links L (10 Gb/s) and R (5 Gb/s), no background. Best MLU with arbitrary splitting? With the demand carried by 2, 3, or 4 equal LSPs?
b. Links L and R, both 10 Gb/s, with fixed background L = 1, R = 3. Unsplittable LSPs A = 7 (currently on R) and B = 5 (currently on L). Evaluate every single-LSP move. Where does HPRR end? Is it optimal?
c. Propose a fix for (b) and its cost.

5.4 HPRR's exponential link cost (HW2 Q5.3 twin)

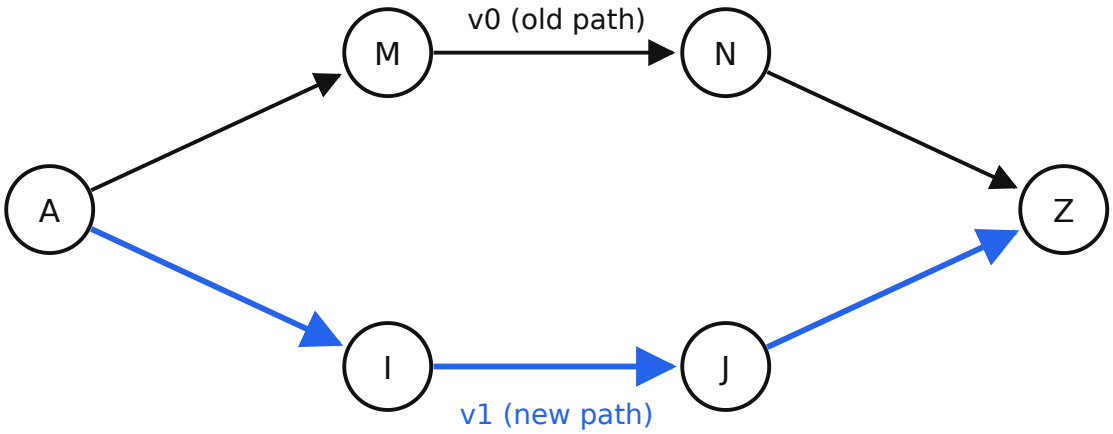
One unsplittable LSP of 5 Gb/s S→D, currently on the direct link. All links are 10 Gb/s. Fixed background: direct link 4 Gb/s; each of the two links via M (S→M, M→D) 3.9 Gb/s. HPRR runs Dijkstra with link cost $w_e = \exp[\alpha(u'_e/u^* - 1)]$, where u'_e is the link's utilization if the LSP uses it; route cost = sum of link costs; $\alpha = 50$, $u^* = 0.8$. The current route is a candidate.



a. Cost of the direct route and of the route via M. Which does HPRR pick?
b. MLU of each choice. Which is better?
c. Explain the discrepancy and propose a fix. Does your fix solve the problem in 5.3(b)?

5.5 **Versioned path updates**

Traffic A→Z moves from v0 (A–M–N–Z) to v1 (A–I–J–Z) using Meta's version bit. Steps: (1) install v1 rules at I; (2) install v1 rules at J; (3) A starts tagging packets v1; (4) remove v0 rules at M and N. Which statements are correct?



☐ **A.** Order 1, 3, 2, 4 can black-hole packets at J.

☐ **B.** Order 2, 1, 3, 4 is safe, provided step 3 waits for acknowledgments from I and J.

☐ **C.** Step 4 may run immediately after step 3 is issued, because new packets are tagged v1.

☐ **D.** A single bit suffices because only the old and the new version ever coexist; the bit alternates 0/1 across updates.

☐ **E.** With this scheme, packets of the same flow can briefly take two different paths, which the sender may see as an RTT change unrelated to congestion.

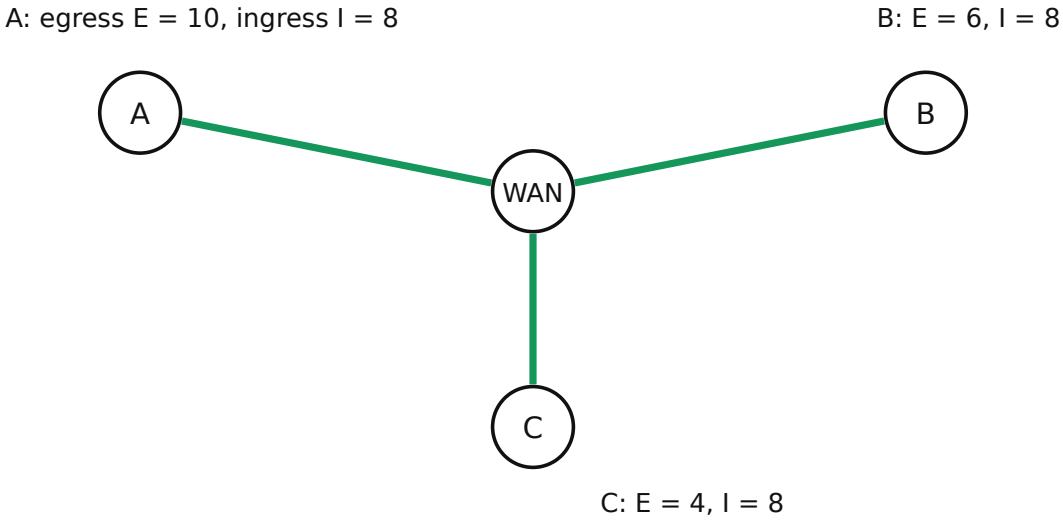
5.6 Building the traffic matrix with sFlow

An EBB edge router samples 1 in 1000 packets. In one second the collector receives 5,000 samples whose headers map to (DC1 → DC2, silver), with an average packet size of 1,500 bytes.

a. Estimate that cell of the traffic matrix in Gb/s.
b. Why doesn't sampling lose the information TE needs?
c. Why not keep a per-flow byte counter in the switch ASIC? Where must sampling happen to build the full DC×DC matrix?

5.7 Pipe vs hose

Three sites. Hose limits: egress $E_A = 10$, $E_B = 6$, $E_C = 4$; ingress $I_A = I_B = I_C = 8$ (Gb/s).



a. How many numbers does a pipe model need for n = 3 sites? For n = 50? A hose model?
b. Largest possible A→B demand allowed by the hose?
c. For the cut separating {A} from {B, C}: the most traffic that can leave A across it, and the most that can enter A.
d. Why is the hose model better for capacity planning months ahead? What does Meta's “segmented hose” add?

5.8 Label-stack depth and Binding SIDs

Assume each hop needs one adjacency label, EBB hardware handles at most 3 labels, and a Binding SID takes one slot: the router it names pushes a new stack (up to 3 labels). A path has 7 hops: SRC→R1→R2→R3→R4→R5→R6→DST.



a. Can SRC encode it alone? How many Binding-SID routers are needed, and which (one valid choice)?
b. When only the R5→DST portion changes, which routers need new state?
c. Why is static-label source routing better than installing per-path rules at every router?

5.9 Meta EBB architecture

Which statements are correct?

☐ **A.** EBB's planes are parallel physical networks with their own controllers, so new controller software can be rolled out to a fraction of capacity and A/B tested.

☐ **B.** An EBB “plane” is the same thing as a traffic class such as gold or bronze.

☐ **C.** When a primary path fails, an LSP agent activates a preinstalled backup within hundreds of milliseconds; the central controller re-optimizes in a later cycle.

☐ **D.** Restoring reachability after a failure guarantees that the rerouted demands have enough capacity.

☐ **E.** Open/R's KvStore replicates link state with strong consistency, so all routers see an update at the same instant.

☐ **F.** Gold traffic is allocated before silver and bronze, on shortest paths measured by RTT.

6 · Google: intent, Orion, bandwidth functions, B4, BwE, dSDN

Notes §6 · Lectures 14–16 · HW2 Q4

6.1 Intent, policy, or configuration?

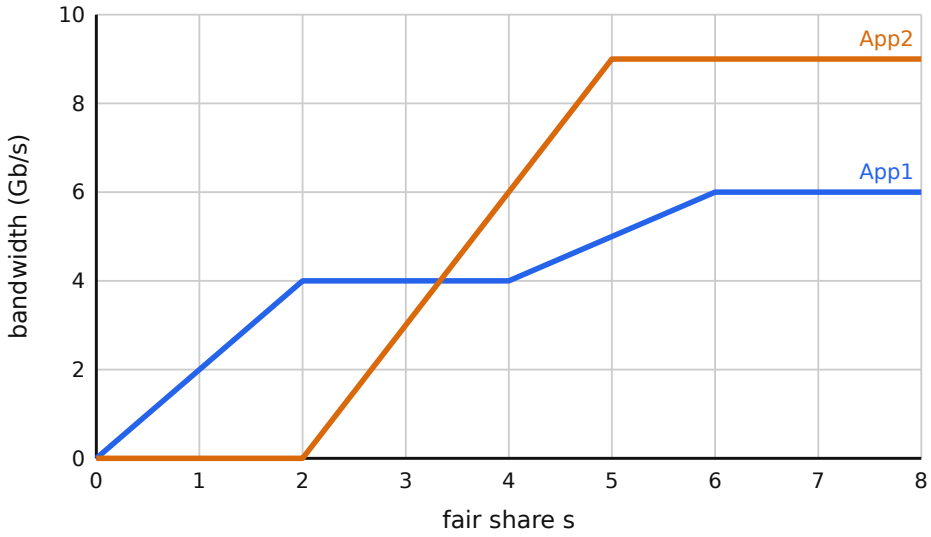
Classify each as intent, policy, configuration, or a non-example (rule / access policy): (i) “Keep at least 8 Gb/s available to service S while link e is drained.” (ii) “Prefer path P1 for S when it has capacity.” (iii) “Install next-hop group {P1: 0.7, P2: 0.3} on R7.” (iv) “When interface utilization exceeds 90%, emit an alert.” (v) “Avoid routing customer X's traffic through vendor V's equipment, even at reduced service levels.” (vi) “Deny traffic to prefix P1 unless it comes from prefix P2.”
(b) For (i), only 6 Gb/s can cross the cut while e is drained. Which lifecycle stage finds this and what should happen? (c) S offers only 2 Gb/s and receives 2 Gb/s. Does this show the intent is met?

6.2 Sharing policies on one link

Three applications share a 12 Gb/s link with demands A = 3, B = 4, C = 9. Give each one's rate and the unused capacity under: (a) equal shares with hard rate limits (non-work-conserving), (b) max-min fairness, (c) weighted max-min with weights 1 : 1 : 2, (d) strict priority C > B > A.

6.3 Bandwidth functions with a priority switch

App1: $B_1(s) = 2s$ for $s \le 2$ (reaches 4), flat at 4 for $2 \le s \le 4$, then rises with slope 1 to its cap of 6 at $s = 6$. App2: $B_2(s) = 0$ for $s \le 2$, then slope 3 to its cap of 9 at $s = 5$. They share one bottleneck.

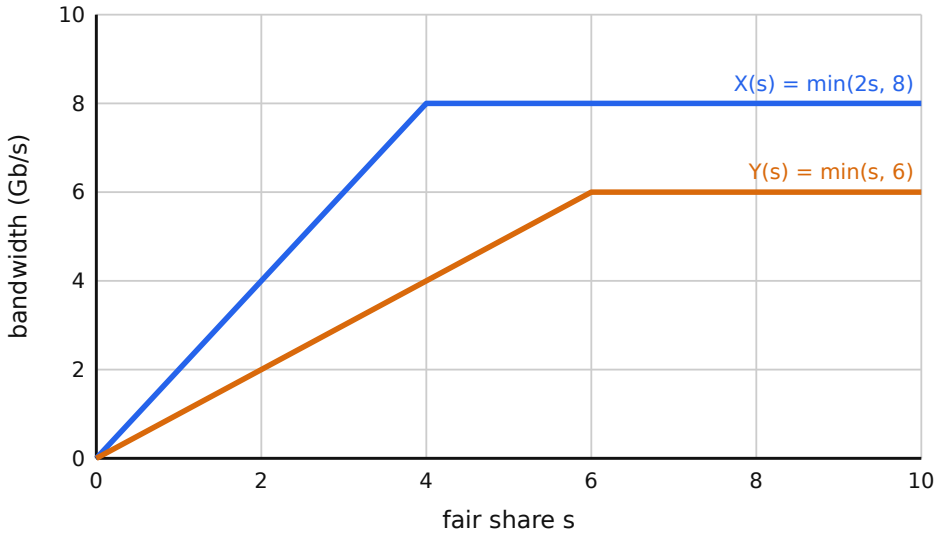


a. Allocations for C = 3, 10, 13, 16 Gb/s (and any unused capacity).

b. Describe the policy these curves express.

6.4 Bandwidth functions vs fixed weights (HW2 Q4.1–4.2 twin)

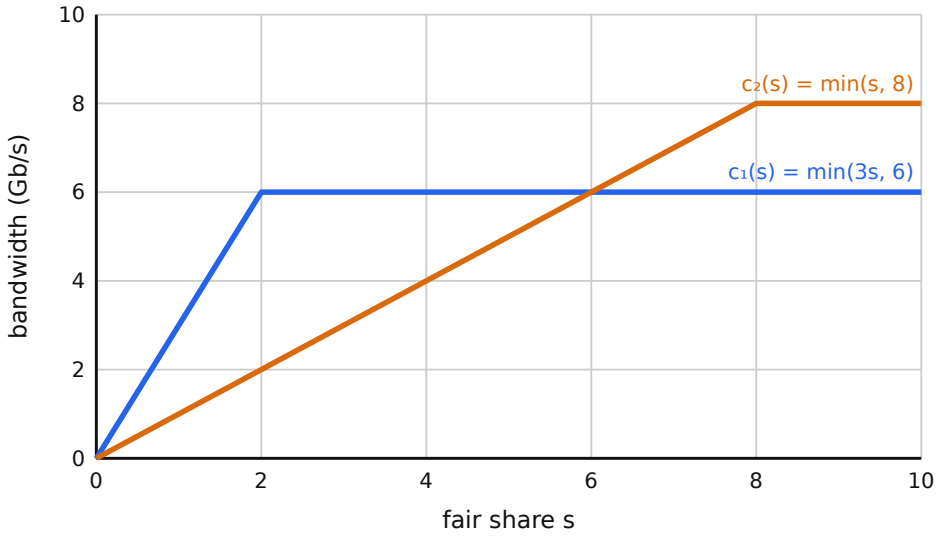
Two flow groups share a bottleneck: X(s) = min(2s, 8), Y(s) = min(s, 6) (demands 8 and 6).



a. Allocations at C = 6, 12, 16.
b. Same capacities with non-work-conserving 2:1 sharing (reserve 2C/3 and C/3, then cap by demand). Unused capacity?
c. With work-conserving 2:1 sharing.
d. Construct a bandwidth-function policy that no single fixed weight ratio can implement, and show why.

6.5 Aggregating and decomposing

Children: $c_1(s) = \min(3s, 6)$, $c_2(s) = \min(s, 8)$.

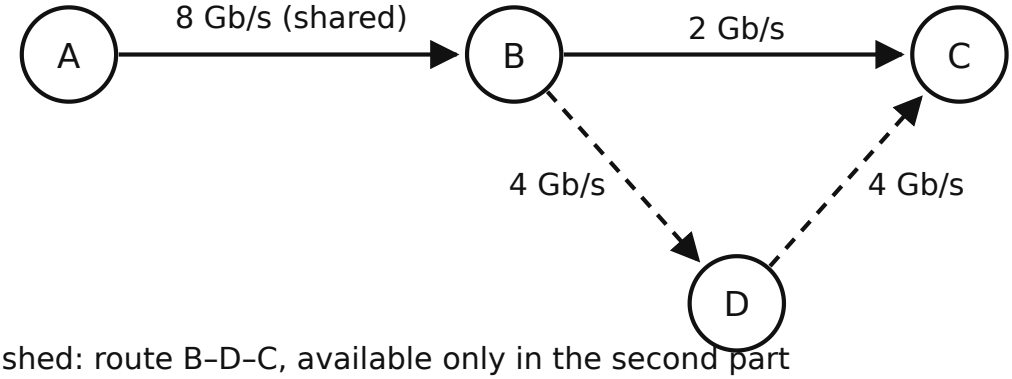


a. Write the aggregate P(s): breakpoints and slopes.
b. The parent is allocated 11 Gb/s. Each child's rate? Allocated 6?
c. Why aggregate with functions instead of passing up one total-demand number (14)?

6.6 Freezing and routing freedom

App1 (A→B→C): min(3s, 6). App2 (A→B): min(s, 10). A–B = 8 Gb/s shared; B–C = 2 Gb/s used only by App1.

App1: A→B→C, min(3s, 6) · App2: A→B, min(s, 10)



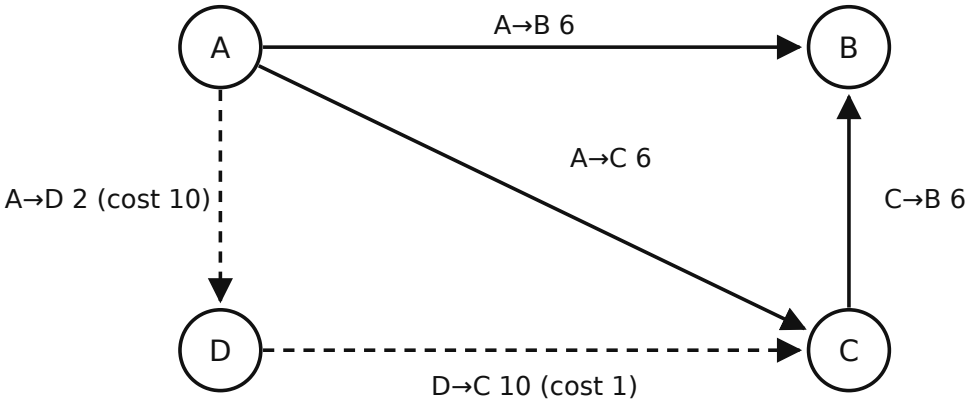
dashed: route B–D–C, available only in the second part

a. With fixed routes, find the allocation. When does App1 freeze?
b. An unused route B–D–C with 4 Gb/s becomes available to App1. New allocation?
c. What does this show about TE vs bandwidth allocation?

6.7 Run the B4 waterfill yourself

Directed links, cost 1 each: A→B 6, A→C 6, C→B 6 (Gb/s). FG1 (A→B): min(2s, 12), paths AB then ACB. FG2 (A→C): min(s, 8), path AC.

FG1: A→B, min(2s, 12) · FG2: A→C, min(s, 8)



dashed links are added only in the last part

a. At what s does the first link fill? Group rates then?
b. Continue until no group can grow. Final rates and per-path split. Unused capacity?
c. Now add A→D (2 Gb/s, cost 10) and D→C (10 Gb/s, cost 1), giving FG1 the path ADCB and FG2 the path ADC. Continue from (b). Final rates?

6.8 **Quantization overload**

Use the lecture's continuous B4 result: FG1 = 20 Gb/s split ½ AB, 5⁄12 ACB, 1⁄12 ADCB; FG2 = 5 Gb/s split ⅓ AC, ⅔ ADC. Capacities: AB, AC, CB 10; AD, DC 5.

a. Load on every link (continuous).
b. Round every fraction independently to the nearest multiple of 50%. Loads now?
c. Fix it with 50% units without reducing either group.

6.9 **BwE update period waste (HW2 Q4.3 twin)**

A parent splits a 20 Gb/s link between teams A and B (equal shares when both need them; unused capacity is redistributed). Current budgets: 10 / 10. New demands for the next 500 ms: A = 4 Gb/s, B = 18 Gb/s. The next parent update is 500 ms away.

a. How much data (MB) does each team send in the interval?
b. What could the network deliver under the stated policy? How much is wasted?
c. Propose a mechanism change with its benefit and cost.

6.10 **Orion, BwE, dSDN**

Which statements are correct?

☐ **A.** In BwE, usage reports flow up the hierarchy and bandwidth budgets flow down to host enforcers.

☐ **B.** Host enforcers cap tasks' rates (e.g., with Linux HTB); TCP congestion control still operates underneath that cap.

☐ **C.** A single Orion controller instance manages all of B4; domains don't need BGP between them.

☐ **D.** Orion's NIB is a pub-sub system: components subscribe to state changes rather than the NIB invoking every component's callback.

☐ **E.** dSDN gives up programmability to gain reliability.

☐ **F.** In dSDN, every router learns the TE inputs by flooding and computes the global TE solution; the ingress encodes its chosen path as an MPLS label stack.

☐ **G.** 52% of the 41 largest B4 outages studied had a control-related root cause.

6.11 **Short essay: did SDN simplify WAN operations?**

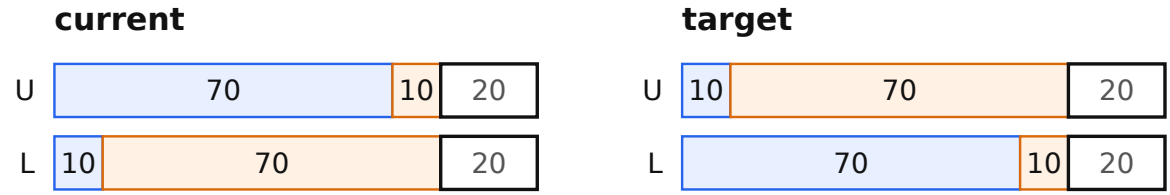
Answer in 5–6 sentences: what SDN simplified, what it complicated, what changed in the last ~20 years, and what dSDN keeps and changes.

7 · Microsoft: SWAN, TeaVaR, OneWAN

Notes §7 · Lectures 16–17

7.1 **Plan a congestion-free update**

Two links U and L, 100 Gb/s each. F1 = 80 Gb/s, F2 = 80 Gb/s. Current: F1 (U 70, L 10), F2 (U 10, L 70). Target: F1 (U 10, L 70), F2 (U 70, L 10). Every link keeps 20 free.



blue = F1, orange = F2, grey = free · links U and L: 100 Gb/s each

a. Write the condition a transition from stage i to i+1 must satisfy on each link.
b. Can it be done in one step? Show the worst case.
c. Construct a 3-stage plan (two intermediate states) and verify every transition.
d. What does SWAN's bound say about the number of steps with 20% scratch? What's the trade-off in choosing the scratch?

7.2 Check the lecture's plan, and catch a bad one

Links 100 Gb/s. Stage 0: U (F1 64, F2 16), L (F1 16, F2 64). Stage 1: U (44, 36), L (36, 44). Stage 2: U (24, 56), L (56, 24). Stage 3: U (16, 64), L (64, 16).

a. Worst-case load on U for each transition.
b. A colleague proposes going straight from stage 0 to U (40, 40), L (40, 40). Valid?
c. Why is the scratch capacity still 20 after each completed stage?

7.3 Changing tunnels with a small rule table

An ingress switch's tunnel-rule table holds 4 entries. Current tunnels {T1, T2, T3}; the new optimal allocation uses {T4, T5, T6}.

a. With 1 spare entry, list the steps. How many intermediate allocation LPs and how many congestion-free shift plans are needed?
b. The table holds 5 entries (2 spare). Steps and counts?
c. Why not overwrite T1 with T4 directly?

7.4 Deterministic failure guarantees

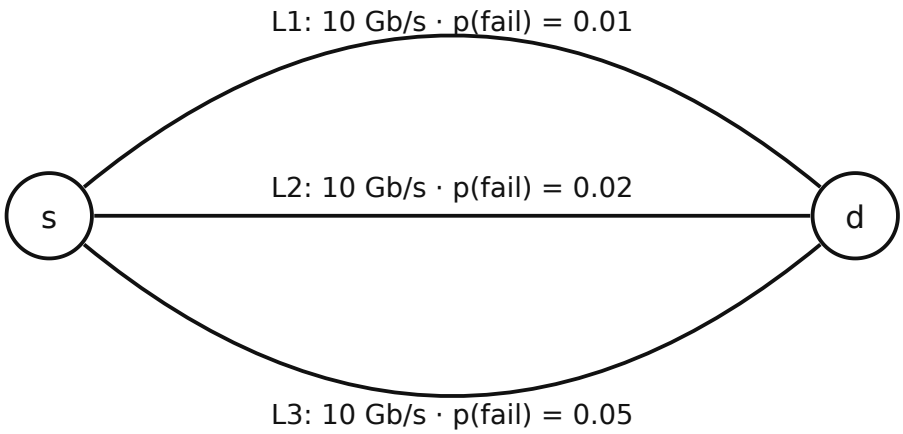
Four disjoint links of 10 Gb/s connect s to d. What bandwidth can be guaranteed with 0, 1, or 2 simultaneous link failures (spreading traffic optimally and rerouting after a failure)? Why is the 2-failure guarantee wasteful if two simultaneous failures happen 0.1% of the time?

(b) The lecture's slide. Three 10 Gb/s links, independent failure probabilities: top 10^{-3}, middle 10^{-1}, bottom 10^{-3}.

i. P(no failure), P(exactly one failure), P(two or more failures)?
ii. Allocation X: 20 Gb/s spread so that any two surviving links carry it. What fraction of the time is 20 Gb/s delivered?
iii. Allocation Y: 10 + 10 on top and bottom only. What fraction of the time is 20 Gb/s delivered?
iv. Which one meets “20 Gb/s, β = 99.98%”?

7.5 TeaVaR: scenarios, loss distribution, VaR

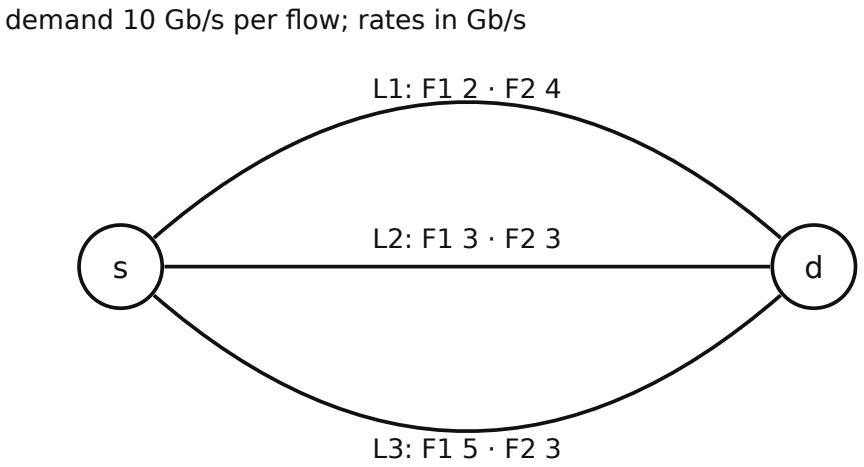
Three links L1, L2, L3 (10 Gb/s each) carry one s→d flow; links fail independently with probabilities 0.01, 0.02, 0.05. Use the lecture's loss model: tunnels = links; in a scenario, traffic assigned to a failed link is lost (no rerouting).



a. Probability of each of the 8 scenarios.
b. Allocation E: split demand equally over the 3 links. Allocation T: split equally over L1 and L2 only. Loss distribution of each.
c. VaR (the β-percentile loss) of each for β = 0.90, 0.95, 0.99.
d. Max demand each allocation can carry (capacity), and the bandwidth it can guarantee β of the time (demand × (1 − VaR)) for each β. Which is better?

7.6 TeaVaR's loss function (lecture example extended)

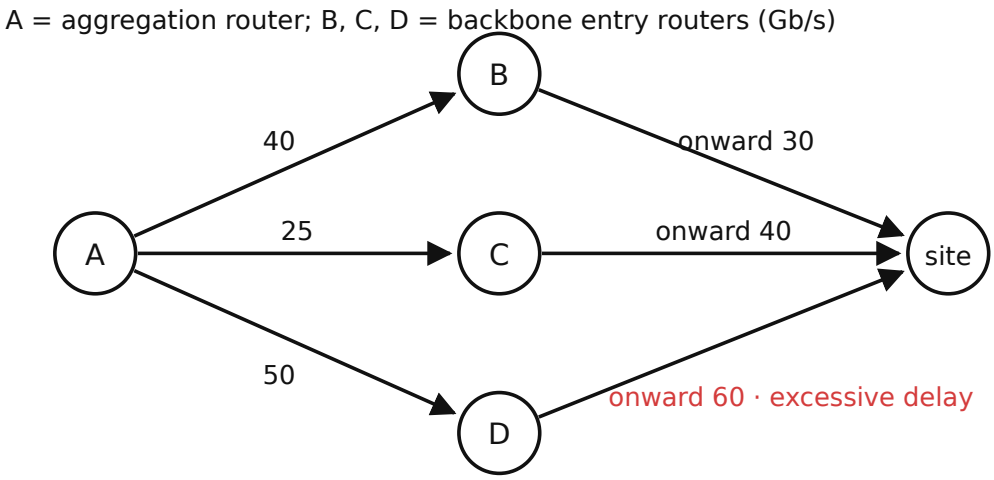
Flows F1 and F2 (demand 10 each) send F1: (2, 3, 5) and F2: (4, 3, 3) Gb/s on links (L1, L2, L3).



a. L(x, q) for: no failure; L1 down; L3 down; L2 and L3 down.
b. Suppose these four are the only scenarios, with probabilities 0.90, 0.05, 0.04, 0.01. VaR at $\beta = 0.95$ and $\beta = 0.99$?
c. What rates can be guaranteed 95% of the time?

7.7 OneWAN traffic steering

An aggregation router A connects to backbone entry routers B, C, D with links A–B 40, A–C 25, A–D 50 Gb/s. Their onward capacity toward the destination site is B 30, C 40, D 60. D's onward delay is excessive.



a. Apply OneWAN's three steering steps. What weights result? How would 1,100 flows be distributed?

b. Why do aggregation routers hold the full IP tables while backbone routers don't? What do the outer label and the inner SID identify?

7.8 SWAN, TeaVaR, OneWAN

Which statements are correct?

☐ **A.** SWAN's controller configures only the ingress (edge) switches; core routers forward on statically configured label paths.

☐ **B.** SWAN allocates the background class first and the interactive class with the remaining capacity.

☐ **C.** Congestion-free updates are needed because switches apply updates at different times, so links can see a mix of old and new rates.

☐ **D.** TeaVaR assumes independent link failures, so a scenario's probability is a product of per-link probabilities.

☐ **E.** Guaranteeing service under any 2 simultaneous failures always uses capacity more efficiently than a probabilistic β-guarantee.

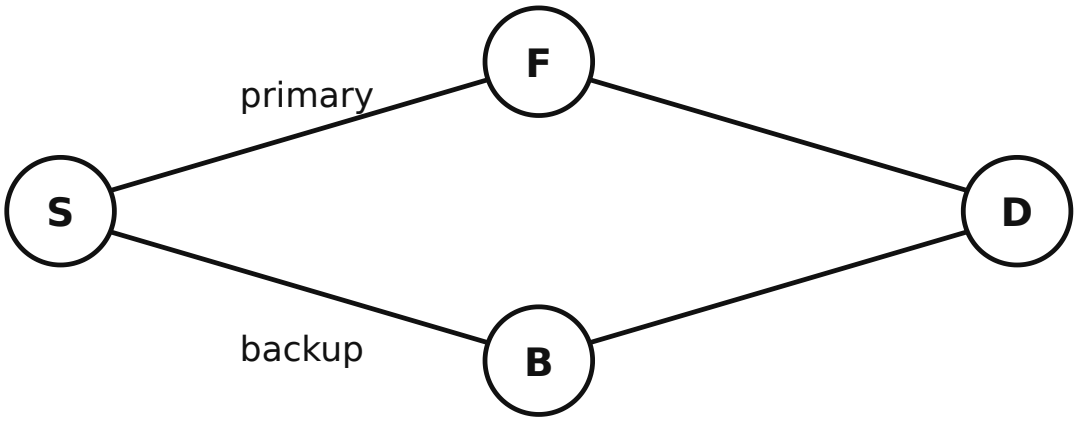
☐ **F.** A split WAN can strand capacity: one WAN overloaded while the other has spare.

☐ **G.** OneWAN removes trunks below 500 Mb/s from the optimization, cutting variables about 7× while affecting about 2.5% of traffic.

8 · Network verification

Notes §8 · Lecture 18

8.1 **Scenario table for the running example**



Routing: S uses S–F–D when S–F and F–D are both up; otherwise it settles on S–B–D. F's ACL: deny TCP/23, permit everything else. B's ACL (first match): TCP/443 → deny; TCP/22 → permit; default → deny. Requirements: web (443) delivered under any ≤ 1 link failure; SSH (22) goes through F in normal operation; telnet (23) never delivered.

a. For each scenario (none, S–F, F–D, S–B, B–D down), is web / SSH / telnet delivered, and via which node?
b. Which requirements are violated, and in which scenarios?
c. Repair B's ACL minimally and state the verification result after the repair.

8.2 Reachability layers

Same network and buggy ACL as 8.1, for a web packet. Using N(u,v) = selected ∧ up ∧ permit and backward layers $R_0 = \{D\}$, $R_k = R_{k-1} \cup \{v : N(v,u) \text{ for some } u \in R_{k-1}\}$:

a. Compute R_0, R_1, R_2 with F–D down.
b. With S–B down.
c. With F–D down after the repair.
d. Why do two layers suffice here?

8.3 Encoding an ordered ACL

B's ACL in order: r1: src ∈ 66.6.0.0/16 → deny; r2: TCP/443 → permit; r3: TCP/22 → permit; r4: default → deny.

a. Decide: web from 66.6.1.1; web from 10.0.0.1; telnet from 10.0.0.1; SSH from 66.6.7.7.
b. Write $\text{hit}_2(h)$ and permit(h) in first-order logic.
c. An engineer swaps r1 and r2. Which decision changes, and what kind of requirement does that violate?

8.4 **Writing requirements and the query**

For the network in 8.1 (links SF, FD, SB, BD), write:

a. Isolation for telnet.
b. Waypointing for SSH. Does it imply SSH is delivered?
c. Loop freedom.
d. “At most one link down” as pure Boolean clauses. How many clauses?
e. The solver query that checks “web survives every single-link failure”, and how to read SAT vs UNSAT.

8.5 **Why symbolic search?**

(a) How many failure scenarios are in scope for ≤ 1 failure on 4 links? ≤ 2 failures? (b) For 50 links with ≤ 2 failures? All subsets? (c) Why model packets symbolically (H_{443}) instead of enumerating headers?

8.6 SAT, SMT, Minesweeper

Which statements are correct?

☐ **A.** If the solver returns UNSAT for (allowed failures ∧ model ∧ ¬property), the property is guaranteed in the real network under all conditions.

☐ **B.** A SAT result yields a concrete counterexample: a failure condition, a packet, and forwarding state that violate the property.

☐ **C.** Pure Boolean SAT can't directly express “the sum of competing traffic on a link ≤ its capacity” over integer rates; SMT can.

☐ **D.** Minesweeper derives routing and forwarding constraints from router configurations, including import policies and route preferences, and checks stable routing states.

☐ **E.** Waypointing alone guarantees that packets are delivered.

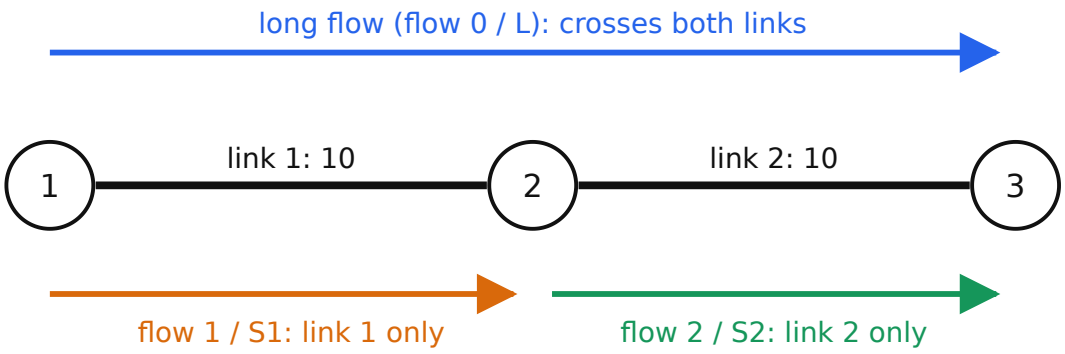
☐ **F.** Loop freedom is checked on forwarding prefixes, because a looping packet may never be delivered.

9 · Cumulative refresher (Quiz 1 material)

Notes §9 · Lectures 1–8

9.1 **NUM: fairness on two links**

Two links, 10 Mb/s each. Flow 0 crosses both; flow 1 uses only link 1; flow 2 only link 2. Find the allocation under (a) max throughput (linear utility), (b) proportional fairness (log utility), (c) max-min fairness.



9.2 **Copa decision**

Copa, δ = 0.5, cwnd = 20 packets, RTT_min = 40 ms, RTT_standing = 50 ms. Compute d_q, the target and current rates (packets/s), and the next adjustment and its size per RTT (v = 1).

9.3 Vegas decision

Vegas, W = 24 packets, BaseRTT = 40 ms, current RTT = 50 ms, α = 1, β = 3. Compute Q = (W/BaseRTT − W/RTT)·BaseRTT and the action.

9.4 Bufferbloat

A 50 Mb/s bottleneck with a FIFO holding 2 MB of waiting data; base RTT 30 ms. (a) Queueing delay for a packet behind 2 MB? (b) Resulting RTT? (c) Max waiting data (kB) to keep queueing ≤ 10 ms?

9.5 BBR estimates

Delivery-rate samples 80, 95, 90 Mb/s; RTT samples 30, 25, 40 ms. (a) BtlBw, RTprop, BDP (kB)? (b) Why does BBR periodically cut inflight in ProbeRTT? (c) Why must BBR pace?

9.6 CoDel

CoDel with target 5 ms and interval 100 ms. Experiment 1: sojourn time stays at 6 ms for 300 ms. Experiment 2: sojourn alternates 3 ms (20 ms) and 30 ms (5 ms). Which are correct?

☐ **A.** Experiment 1 meets the persistent-delay condition after about 100 ms, even though its maximum is lower than experiment 2's.

☐ **B.** Experiment 2 enters dropping as soon as a 30 ms sojourn is observed.

☐ **C.** If experiment 1 persists, successive drop gaps shrink as 100/√count ms (100, 70.7, 57.7, 50 …).

☐ **D.** CoDel's target must be re-tuned whenever the link rate changes because it is measured in bytes.

9.7 Reading fluid models

A sender updates once per RTT R; it sees a loss indication with probability p. What algorithm does each describe? (i) $dW/dt = [3(1-p) - p \cdot W/2]/R$. (ii) $dW/dt = [(1-p) \cdot W/4 - p \cdot W/2]/R$. (iii) For (i), the equilibrium window.

9.8 **ECN, INT, end-to-end**

Which are correct?

☐ **A.** ECN marks let the sender respond to congestion before buffer overflow causes loss, without retransmitting the marked packets.

☐ **B.** With Reno's standard ECN response, if 20% of a window's packets are marked the window shrinks by exactly 20%.

☐ **C.** INT records can show which hop had the largest queue; a single ECN mark can't.

☐ **D.** By the end-to-end argument, reliable link-level retransmission makes an endpoint file checksum unnecessary.

☐ **E.** BBRv1 reduces its rate in response to ECN marks.

9.9 **Bridges between Part 1 and Part 2**

Which statements are correct?

☐ **A.** dSDN's fate-sharing applies Clark's principle to control: control should depend only on the routers and links it manages, so it fails only when they do.

☐ **B.** Host rate limits (BwE) work in Google's WAN for the same reason ECN never deployed on the open Internet.

☐ **C.** Min-MLU traffic engineering and NUM congestion control turn the same knob: both adjust source sending rates.

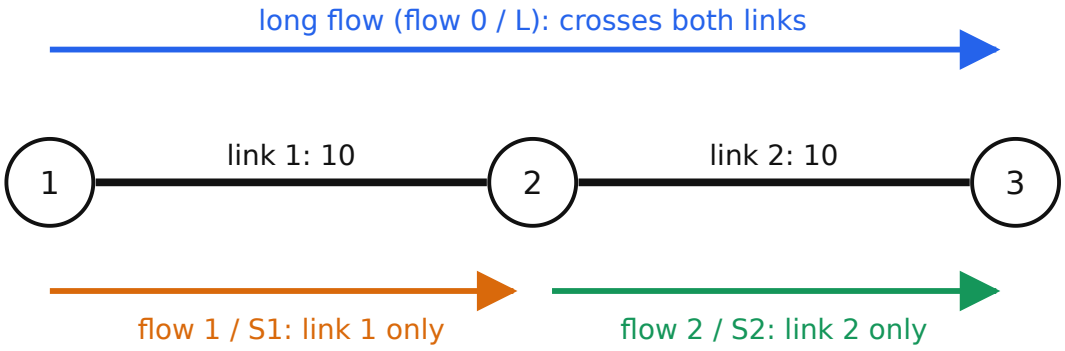
☐ **D.** QUIC's fast rollout fits the Part 1 “de-ossification” story: one company controlling servers, clients and much of the traffic can change transport behavior quickly.

☐ **E.** In B4's waterfill, if every flow group has the same bandwidth function $B(s) = s$ and no cap, the result is max-min fair.

☐ **F.** SDN's single-point-of-failure risk is mainly the controller machine crashing, which replication solves.

9.10 **Parking lot: max-min, proportional, and bandwidth functions**

Two 10 Gb/s links in series. A long flow L crosses both links; S1 uses only link 1; S2 uses only link 2. Everyone has unlimited demand.



a. Max-min fair rates (water-filling)?
b. Proportionally fair rates (maximize log L + log S1 + log S2)?
c. Run B4-style waterfill with bandwidth functions B(s) = s for all three groups.
d. Repeat (c) with S1's function changed to B(s) = 2s. What do you notice?

Mock Quiz 2 — timed, closed notes, 75 minutes

Notes §10 · grade with +1/−1 per MCQ choice

M.1 **Performance-oriented congestion control (12 pts)**

(a) Which statements are correct?

☐ **A.** Allegro's default utility counts throughput only while the loss rate stays below about 5%.

☐ **B.** Vivace penalizes the magnitude of the RTT, as Copa does with queueing delay.

☐ **C.** In Vivace, a larger magnitude of the estimated utility gradient produces a larger rate change, subject to a bound.

☐ **D.** When Allegro's two trial pairs disagree, it moves in the direction of the pair with the larger utility difference.

☐ **E.** Under random non-congestive loss, PCC can keep increasing its rate because higher rates still yield higher utility, without identifying the cause of the loss.

(b) A Vivace sender at 200 Mb/s tests 190 and 210 Mb/s and measures U(190) = 150, U(210) = 146. With θ = 25, compute g, Δr, and the new rate.

M.2 QUIC scheduling and semantics (12 pts)

(a) All buffered at t = 0: A (2 chunks), B (1 chunk), C (2 chunks); one chunk per packet; RTT 60 ms (30 each way); ACKs immediate; no loss; cwnd starts at 1 and grows by 1 per RTT. Give each object's display time under C-first, and under round robin in the order A, B, C.

(b) Which statements are correct?

☐ **A.** With HTTP/2 over one TCP connection, a lost packet of one object can delay delivery of other objects whose bytes already arrived.

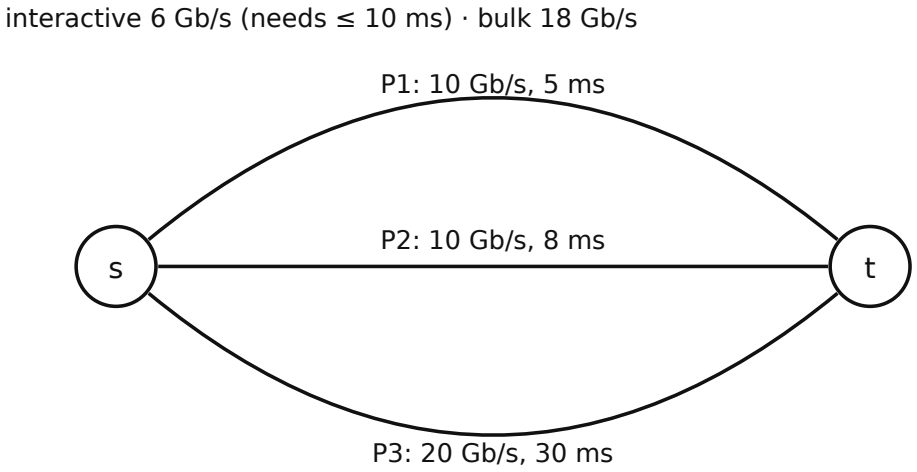
☐ **B.** QUIC 0-RTT can be used on the first-ever connection to a server.

☐ **C.** A QUIC retransmission reuses the original packet number so the receiver can match it to the lost packet.

☐ **D.** All QUIC streams in a connection share one congestion controller.

M.3 Traffic engineering with classes (12 pts)

Three disjoint paths: P1 (10 Gb/s, 5 ms), P2 (10 Gb/s, 8 ms), P3 (20 Gb/s, 30 ms). Demands: interactive 6 Gb/s (requires path latency ≤ 10 ms) and bulk 18 Gb/s.



a. The controller aggregates both classes and minimizes MLU, and each path carries the classes in proportion. Loads, MLU, and interactive traffic on P3?
b. Find a policy-compliant allocation with the same MLU. What does the network need for it?
c. Interactive grows to 22 Gb/s, bulk drops to 10. Is the aggregate feasible? Is the policy feasible? What should the operator do?

M.4 **Mechanisms (10 pts)**

Which statements are correct?

☐ **A.** An MPLS label is 20 bits, and the S bit marks the bottom of the label stack.

☐ **B.** CSPF prunes links that can't meet a trunk's bandwidth or color constraints before running shortest path.

☐ **C.** With Meta's version bit, the source may start tagging packets v1 as soon as it has sent the update messages to the new path's routers.

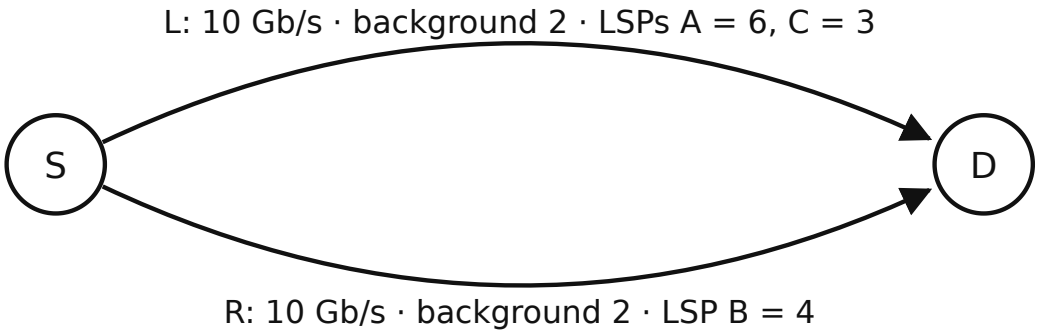
☐ **D.** sFlow lets switches keep exact per-flow byte counts at line rate.

☐ **E.** Under the hose model, a site's total egress is bounded, but how it splits across destinations may change.

☐ **F.** A Binding SID lets an intermediate router supply the rest of a label stack, working around a stack-depth limit.

M.5 HPRR (12 pts)

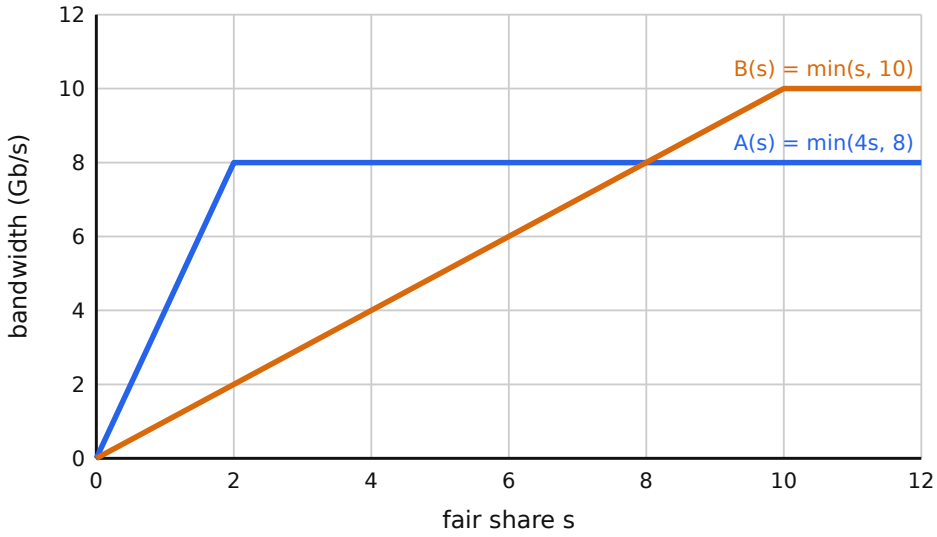
Parallel directed links L and R, 10 Gb/s each, with fixed background 2 Gb/s on each. Unsplittable LSPs: A = 6 and C = 3 on L, B = 4 on R. HPRR considers A, then B, then C, repeating passes until nothing changes. A move is accepted only if the max utilization on the new route after the move is lower than the max utilization on the LSP's current route before it.



a. Run HPRR. Final placement and MLU.
b. Is it optimal among unsplittable placements? Justify.
c. What MLU would arbitrary splitting achieve?

M.6 Bandwidth functions and intent (14 pts)

(a) Two groups share one bottleneck: A(s) = min(4s, 8), B(s) = min(s, 10). Allocations at C = 6, 14, 20 Gb/s. (b) A and B are the children of an aggregate that is allocated 9 Gb/s. Children's rates? (c) Which statements are correct?



☐ **A.** If service S offers 2 Gb/s and receives 2 Gb/s, an intent of "≥ 8 Gb/s for S" is confirmed.

☐ **B.** Orion's reconciliation repairs drift such as rules missing after a component restart.

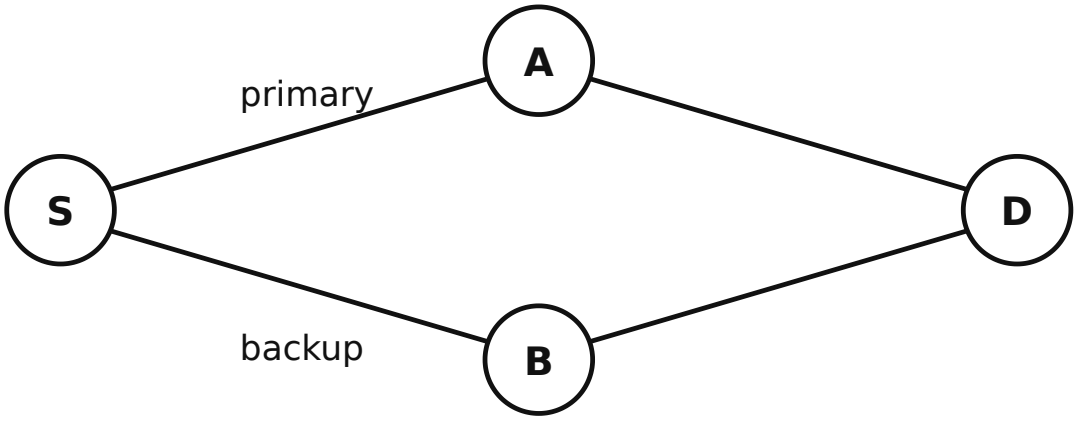
☐ **C.** TE gives BwE paths and split weights; BwE enforces budgets at hosts and reports prioritized demand to TE.

☐ **D.** A non-work-conserving 2:1 policy can leave capacity unused while one group still has unmet demand.

M.7 SWAN and TeaVaR (14 pts)

(a) Links U and L, 50 Gb/s each. F1: (U 30, L 10) → (U 10, L 30). F2: (U 10, L 30) → (U 30, L 10). Can the update happen in one step? Check a plan through the intermediate state F1 (20, 20), F2 (20, 20). What's SWAN's step bound with this 20% scratch? (b) F1 sends (4, 4, 2) and F2 sends (1, 3, 6) on links (L1, L2, L3); demands 10 each. Compute L(x, q) for “L1 down” and “L3 down”. (c) The only scenarios are none (0.90), L1 down (0.06), L3 down (0.04). VaR at β = 0.95 and at 0.99, and the rate per flow guaranteed in each case.

M.8 **Verification (14 pts)**



S routes via A when S–A and A–D are up, otherwise via B. Requirement: DNS (UDP/53) from S reaches D under any single link failure. A's ACL: permit all. B's ACL (first match): r1 UDP → deny; r2 UDP/53 → permit; r3 default → permit.

a. Write the verification query. Is it SAT or UNSAT? Give a witness if SAT.
b. Fix the configuration and re-evaluate.
c. How many failure scenarios are in scope, and how many pairwise clauses encode "≤ 1 failure"?

(d) To also require "DNS + other traffic on B–D ≤ 10 Gb/s", which solver class do you need?