Routing Protocols

CSE 461 Week 7
Winter 2023
Mohan Kukreja

Agenda

1. Review of Intradomain Routing (Lecture Feb 10, 2023)
   a. Distance Vector Routing and Link-State Routing
2. Review of Interdomain Routing (Lecture Feb 13, 2023)
   a. Border Gateway Protocol (BGP)
3. Max-Min Fair Bandwidth Allocation
4. Leftover time for office hours

Review of Intradomain Routing

Overview of Intradomain Routing

- There are 2 main classes of routing approaches:
    - Distance Vector Routing
    - Link-State Routing
- Both approaches are distributed algorithms
    - No centralized authorities, no one can see the entire network from where they are
    - Nodes share network information with each other in a consistent manner
    - Nodes all update their information in the same way
    - Eventually nodes independently converge and have some view of the network
- Distance Vector Routing
    - Node X shares with **neighbors** <u>the distance from Node X to everyone else</u>
    - Node X <u>eventually knows what hop to take</u> to get to everyone else, with the lowest cost
- Link-State Routing
    - Node X shares with **everyone** <u>Node X's distance to Node X's neighbors</u>
    - Node X <u>eventually has a global view</u> of the network and can calculate the best paths

Distance Vector Routing

- Distance Vector Routing
  - Node X shares with **neighbors** the distance from Node X to everyone else
  - Node X eventually knows what hop to take to get to everyone else, with the lowest cost
- A broad class of routing methods that involving sharing “distance vectors”
  - (i.e. an array of numbers representing its distance to other nodes)
  - Everyone shares distance vectors and uses this to update their own distance vectors (by using the lowest cost path)
- Distributed version of Bellman-Ford algorithm
  - “Simply put, the algorithm initializes the distance to the source to 0 and all other nodes to infinity. Then for all edges, if the distance to the destination can be shortened by taking the edge, the distance is updated to the new lower value.”
- Example of a distance vector protocol:
  - Routing Information Protocol (RIP)

Node A's distance vector

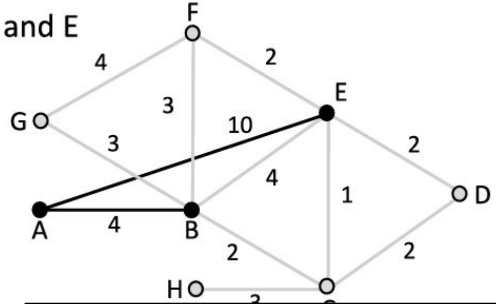
Node E's distance vector

Node B's distance vector

- Consider from the point of view of node A
  - Can only talk to nodes B and E

Initial vector

To	Cost
A	0
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞



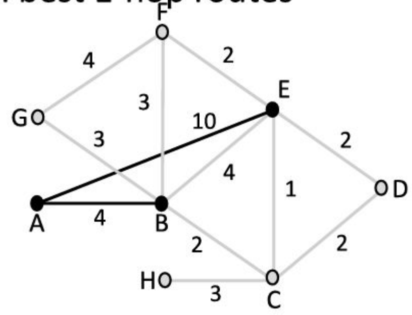
CS 461 Unive

Node A receives distance vectors from B and E and improves its distance vector

- First exchange with B, E: learn best 1-hop routes

To	B's	E's	→	B +4	E +10	→	Cost	Next
A	∞	∞		∞	∞		0	--
B	0	∞		4	∞		4	B
C	∞	∞		∞	∞		∞	--
D	∞	∞		∞	∞		∞	--
E	∞	0		∞	10		10	E
F	∞	∞		∞	∞		∞	--
G	∞	∞		∞	∞		∞	--
H	∞							

Learned better route



This is B/E's distance vector plus A's distance to B/E

Link-State Routing

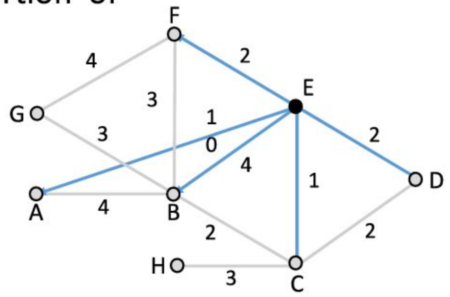
- Link-State Routing
  - Node X shares with **everyone** Node X's distance to Node X's neighbors
  - Node X eventually has a global view of the network and can calculate the best paths
- A broad class of routing methods that involving sharing "link-state"
  - (i.e. the current state of a node's links with its direct neighbors)
  - Everyone floods everyone else's link-state, until everyone knows everyone's link-state
- Once a node has a global view, they use Dijkstra's algorithm
  - Can easily calculate the shortest path from one node to all other nodes
- Examples of link-state routing protocols:
  - Intermediate System to Intermediate System (IS-IS)
  - Open Shortest Path First (OSPF)

Phase 1: Topology Dissemination

- Each node floods <u>link state packet</u> (LSP) that describes their portion of the topology

Node E's LSP flooded to A, B, C, D, and F

Seq. #	
A	10
B	4
C	1
D	2
F	2



Phase 2: Route Computation

- Each node has full topology
  - By combining all LSPs
- Each node simply runs Dijkstra
  - Replicated computation, but finds required routes directly
  - Compile forwarding table from sink/source tree
  - That's it folks!

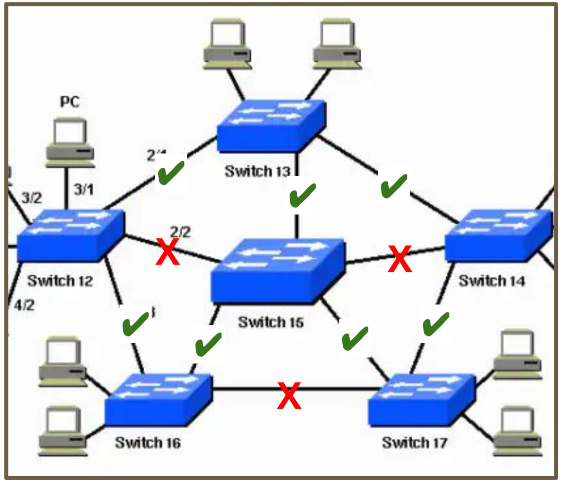
Overview of Intradomain Routing

Goal	Distance Vector	Link-State
Correctness	Distributed Bellman-Ford	Replicated Dijkstra
Efficient paths	Approx. with shortest paths	Approx. with shortest paths
Fair paths	Approx. with shortest paths	Approx. with shortest paths
Fast convergence	Slow – many exchanges	Fast – flood and compute
Scalability	Excellent – storage/compute	Moderate – storage/compute

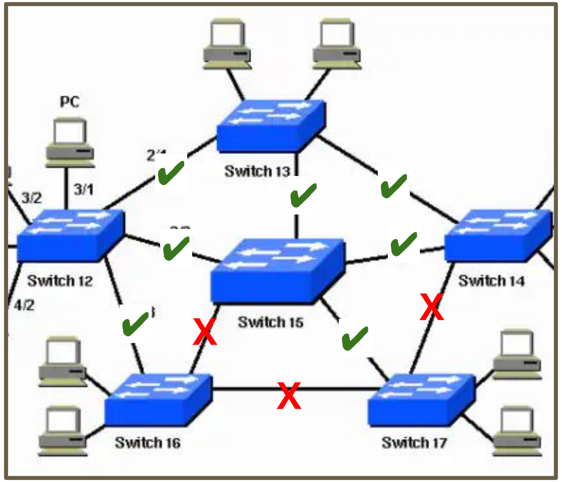
Delay Tolerant Network (DTN) Routing

These distributed routing protocols work well in most scenarios…

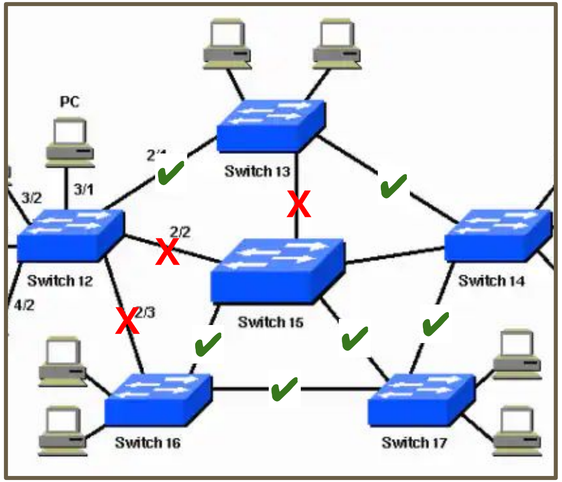
but what if your routers are not always connected?



t=0



t=1



t=2

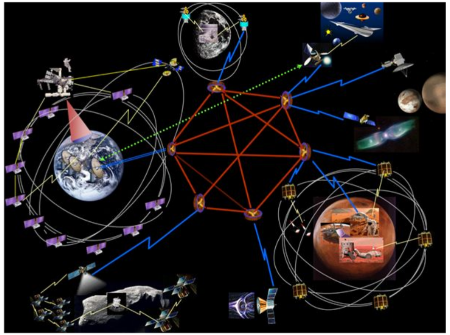
Delay Tolerant Network (DTN) Routing

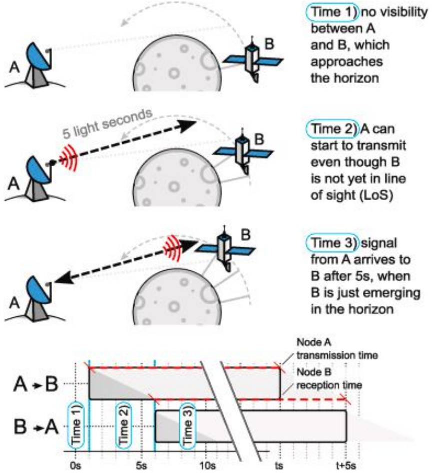
Distributed routing algorithms are hard when links are unreliable/intermittent

- How do you get all nodes an up-to-date view of the network?
- How do you spread messages reliably?
- Will messages ever converge?

Scenarios where these conditions are common:

- Military/War scenarios (see Ukraine/Russia)
- Natural disasters (see Turkey)
- Outer Space (see Solar System Internet)
- Remote and rural areas





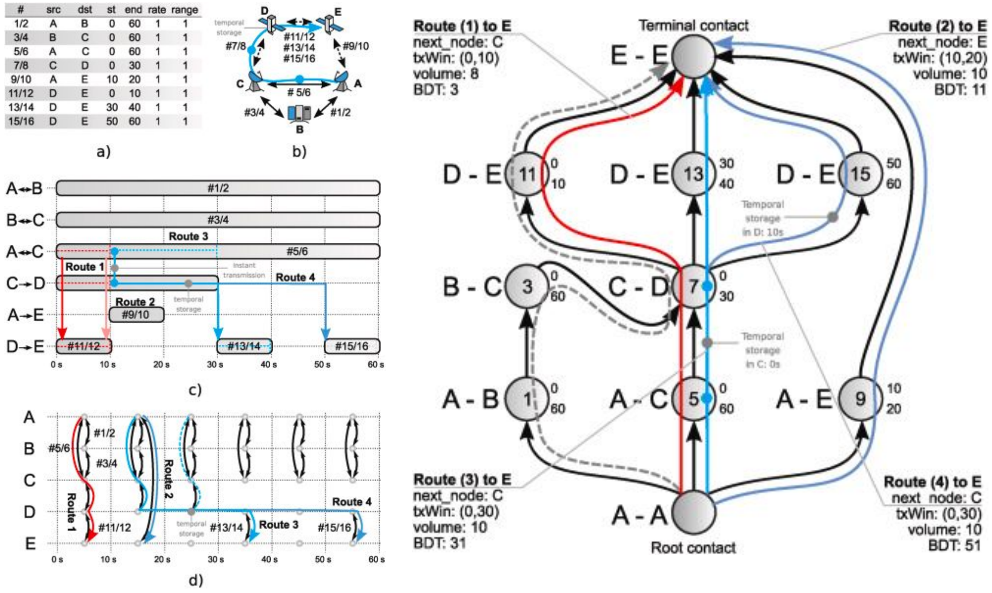
Delay Tolerant Network (DTN) Routing

Space delay tolerant networks currently use “[Contact Graph Routing](#)”

- Each node has a “contact plan”, a schedule of the network links
- Each node uses a variant of Dijkstra’s algorithm on the contact plan
- Assuming contact plan is always accurate, at any point in time you can get the best route

Active research questions:

- What if you can’t predict your network’s contact plan ahead of time?
- How to distribute changes in contact plans? ([software-defined networking somehow?](#))

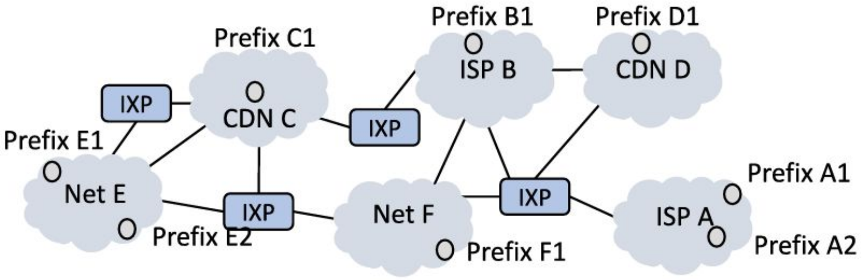


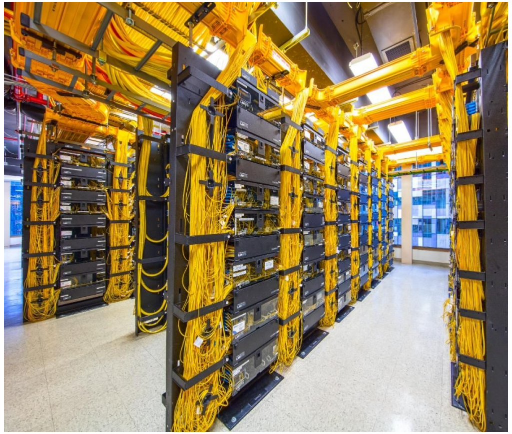
#	src	dst	st	end	rate	range
1/2	A	B	0	60	1	1
3/4	B	C	0	60	1	1
5/6	A	C	0	60	1	1
7/8	C	D	0	30	1	1
9/10	A	E	10	20	1	1
11/12	D	E	0	10	1	1
13/14	D	E	30	40	1	1
15/16	D	E	50	60	1	1

Review of Interdomain Routing

Border Gateway Protocol (BGP)

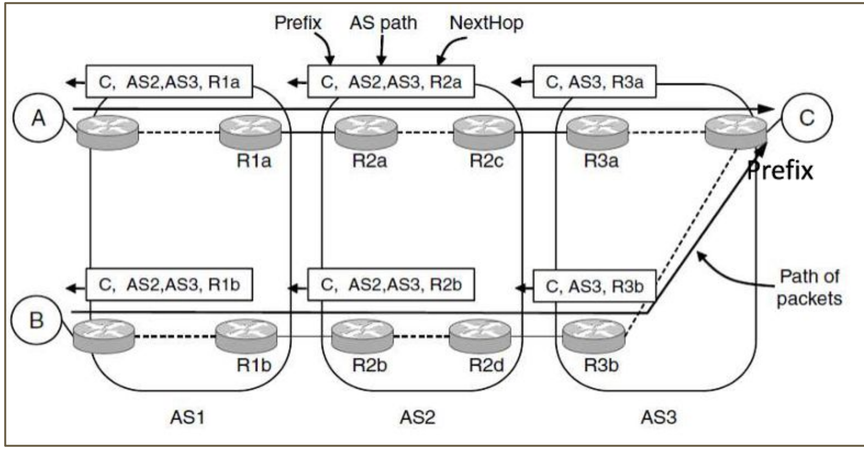
- A protocol used to route between different Autonomous Systems (AS)
  - between entities like Comcast/CenturyLink/T-mobile/Seattle Community Network
  - Involves political/social/monetary considerations
- These BGP conversations happen at Internet Exchange Points (IXP)
  - Such as the Seattle Internet Exchange



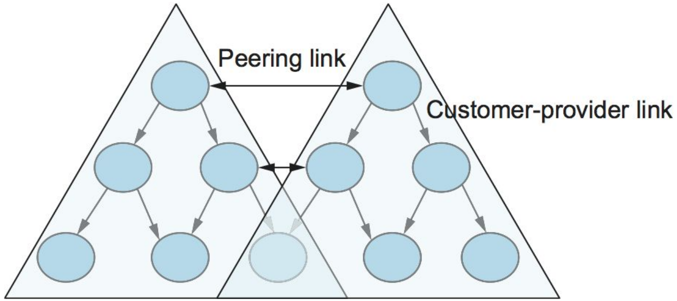


Border Gateway Protocol (BGP)

- Two types of BGP: Interior BGP (iBGP) and exterior BGP (eBGP)
  - We'll focus on eBGP
- AS must have at least one BGP speaker
  - a router that "speaks" BGP to other BGP speakers in other autonomous systems
- BGP speakers advertise routes to IP prefixes (blocks of IP addresses)
- Route announcements include:
  - **IP Prefix, Path Vector, Next Hop**
  - Path vector is a list of ASs on the path to the IP Prefix



Common AS Relationships/Policies

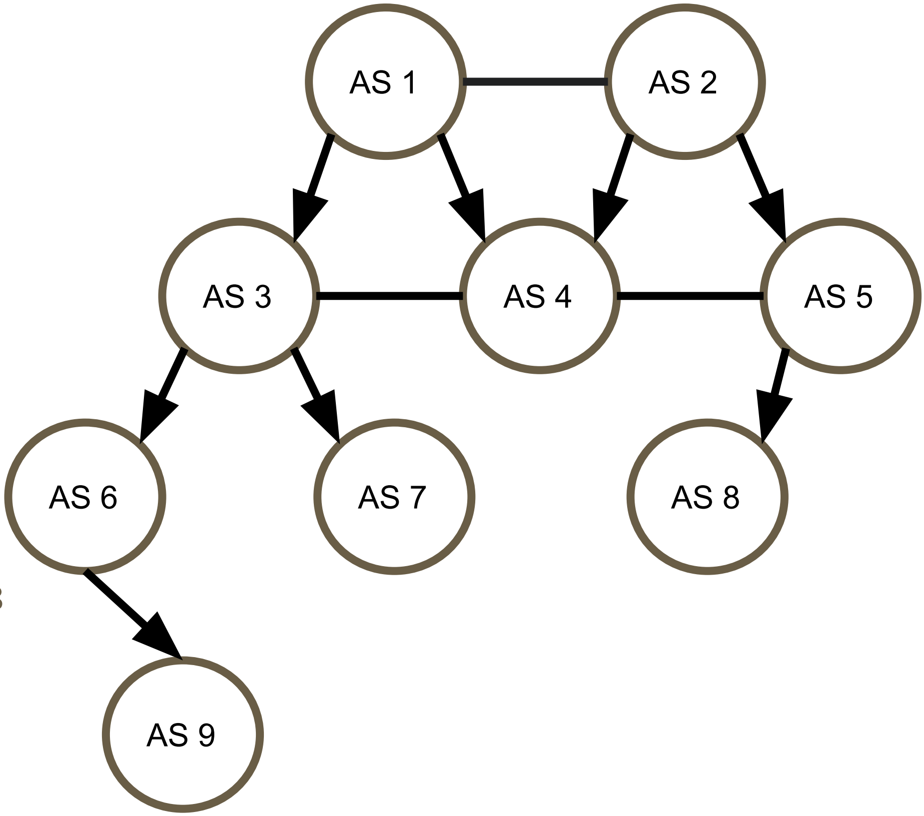


- Transit
  - AS 1 provides transit to AS 2 at some cost
  - AS 1 is the provider, AS 2 is the customer
  - Analogy: Xfinity provides transit to your home
    - If you want to send data through Xfinity's wires, you need to pay them regularly
- Peering
  - AS 3 and AS 4 mutually agree to peer with each other
  - If AS 3 wants to send data to hosts in AS 4's network, AS 3 can do that for free
  - Analogy: Bob and Charlie are neighbors and their kids want to Zoom each other
    - One option: Bob -> Xfinity -> Charlie
      - but Bob would have to pay Xfinity for that :(
    - Another option: Bob -> Charlie (imagine they hooked up a wire to each others house)
      - its free! but only for between hosts in each others networks

Example BGP exercise

Assume when deciding what route a packet should take, an AS's preference is to send it to a customer, otherwise a peer, otherwise to a provider.

1. What path would a packet from a host in AS 9 to AS 5 take?
   - 9-6-3-1-2-5 is correct
   - 9-6-3-4-5 is incorrect
     - AS 4 would not advertise a route to AS 5, it goes against how peering works.
2. Suppose AS 7 and AS 8 were peers, would AS 7 tell AS 8 that AS 7 has a path to AS 3?
   - No, because if AS 7 advertised that, it would lose money because it would be taking on AS 8's traffic for free, while having to pay AS 3

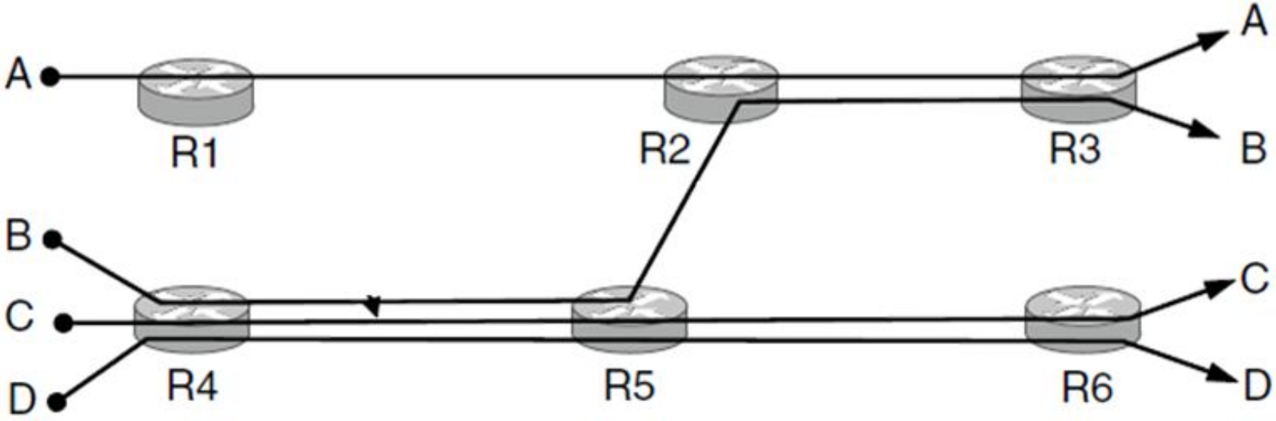


Max-Min Fair Allocation

Fair Bandwidth Allocation

Imagine each router in the network below only has 1 Mbps of bandwidth.

How much bandwidth should each of these flows get? Assume each flow is allocated a constant amount of bandwidth throughout the network.



Max-Min Fair Bandwidth Allocation

One way to “fairly” allocate bandwidth is Max-Min allocation:

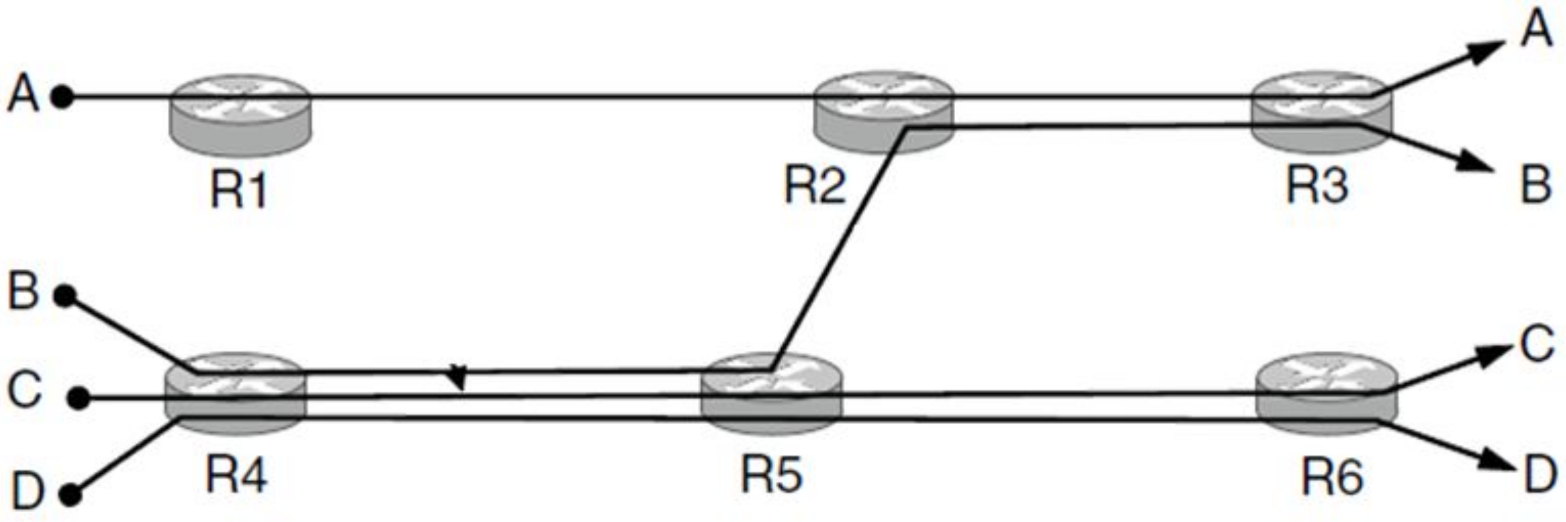
- Maximizes the minimum bandwidth of a flow within a network
- Increasing the rate of one flow would decrease the rate of another flow

Algorithm:

1. Start all flows at rate 0
2. Increase flows until there is a new bottleneck in the network
3. Hold fixed the rate of flows that are bottlenecked
4. Go to step 2 for any remaining flows

Intuition: imagine “pouring water” into a network

Max-Min Fair Bandwidth Allocation



Max-Min Fair Bandwidth Allocation

- End with A=2/3, B, C, D=1/3, and R2/R3, R4/R5 full
  - Other links have extra capacity that can't be used