Network Layer:
Distance Vector Protocols with Safety; Link-State Protocol; Global/Internet-Scale Routing

Y. Richard Yang

http://zoo.cs.yale.edu/classes/cs433/

11/29/2018

Outline

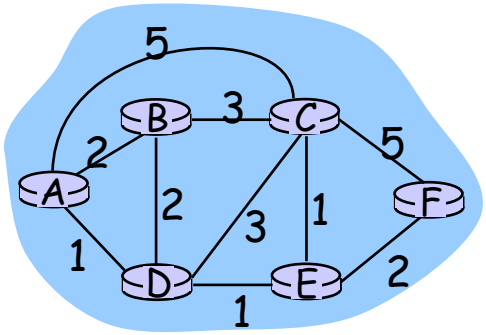
- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Distance vector protocols (distributed computing)
      - Link state protocols (distributed state synchronization)
      - *Internet-scale routing*

Admin

- Assignment four meeting
  - Today 2:30 – 4:00; 5:30-7:00 pm
  - Friday: 11:30-12:30 pm; 4:00-5:00 pm
- Assignment five posted
- Reminder: Exam 2: 7-8:30 pm Tuesday Dec. 11

Recap: Routing Design Space

- Routing has a large design space
  - who decides routing?
    - source routing: end hosts make decision
    - network routing: networks make decision
  - how many paths from source s to destination d?
    - multi-path routing
    - single path routing
  - what does routing compute?
    - network cost minimization (shortest path routing)
    - QoS aware
  - will routing adapt to network traffic demand?
    - adaptive routing
    - static routing

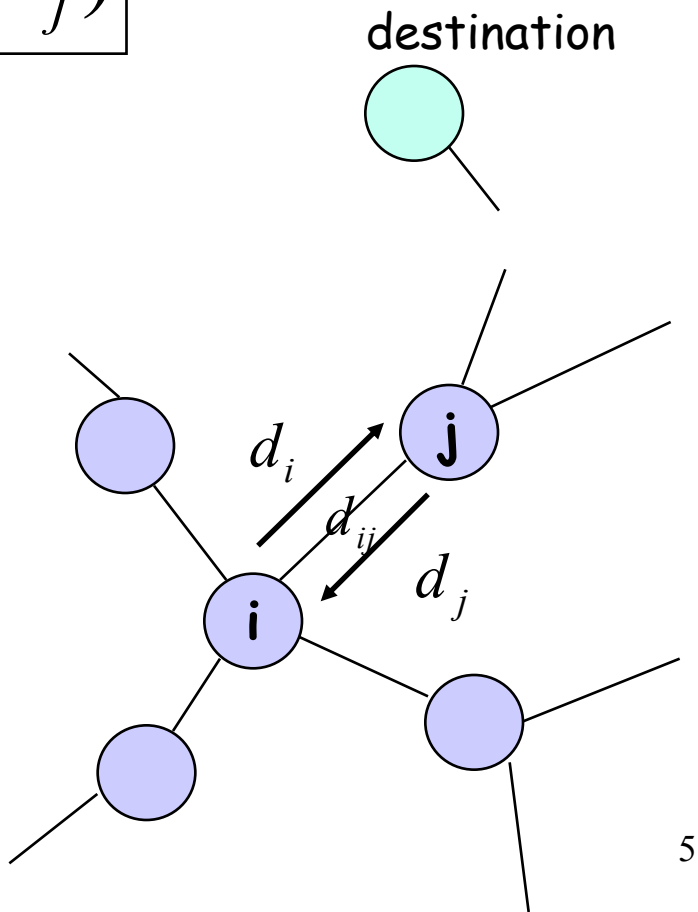


Recap: Distributed Distance Vector Routing Protocols

- Basic Idea: Bellman-Ford (BF) update rule

$$d_i = \min_{j \in N(i)} (d_{ij} + d_j)$$

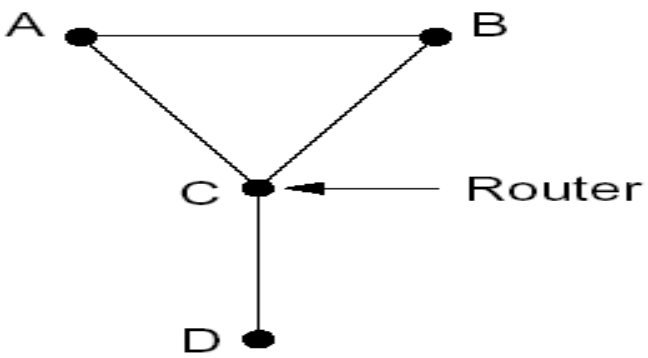
- Synchronous BF
  - Two extreme states
    - Monotonicity
    - Convergence
- Asynchronous BF
- Issues of BF
  - counting-to-infinity in settings such as disconnection



Recap: Counting-To-Infinity

- Counting-to-infinity is caused by a routing loop, which is a global state (consisting of the nodes' local states) at a global moment (observed by an oracle).

- Initial solution:
  - Reverse-poison
  - Solution integrated into RIP, the first major routir protocol



Discussion

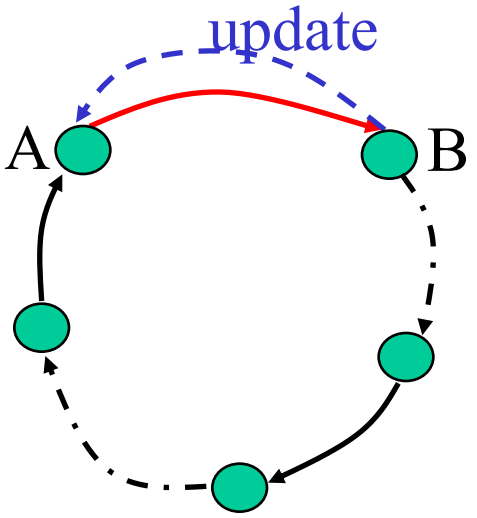
- Any proposal that can enforce global property (i.e., no loop) using local condition (i.e., decision made by each node locally)?

The No Increase Cost (NIC) Condition

- Claim: If a DV protocol enforces that the cost to a destination cannot increase, there will not be any routing loops

Claim: NIC will NEVER Form a Loop

- Initially no loop (no one has next hop so no loop)
- Derive contradiction if a loop forms after a node processes an update,
  - e.g., when A receives the update from B, A decides to use B as next hop and forms a loop



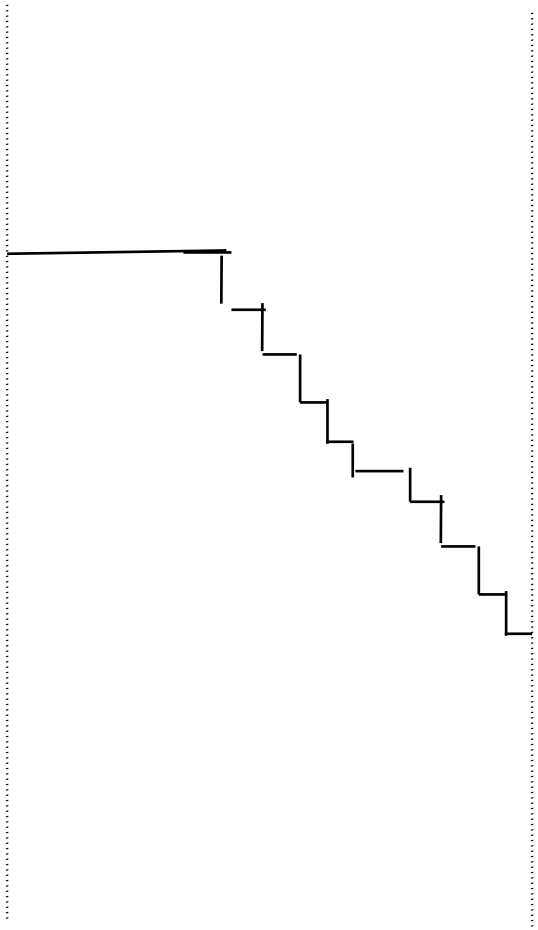
Analysis Technique: Global Invariants

- Global Invariant is a very effective method in understanding distributed asynchronous protocols
- Invariants are defined over the states of the distributed nodes

- Consider any node B in a DV protocol for a destination D.
- What is the state of B?

Invariant at a Single Node B

- [I1] d^B is non-increasing



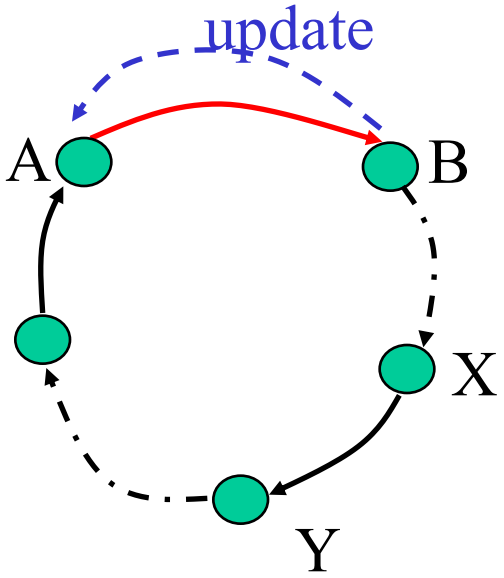
Invariants when A Considers B as Next Hop

- Invariant if A considers B as next hop

  - [I2] $d^B < d^A$
    - because d^A is based on d^B which B sent to A some time ago, $d^B < d^A$ since all link costs are positive;
    - d^B might be decreased after B sent its state (I1)

Loop Freedom of NIC

- Consider a critical moment
  - A starts to consider B as next hop, and we have a loop
- According to invariant I2 for each link in the loop (X considers Y as next hop)
  - $d^Y < d^X$ for each link
- By transition along the loop $d^B > d^B$

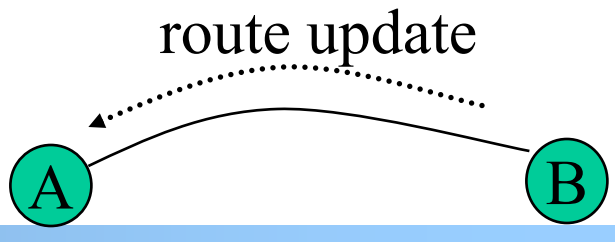


Problem of enforcing NIC: achieves safety but stops the protocol when cost is increased?

Outline

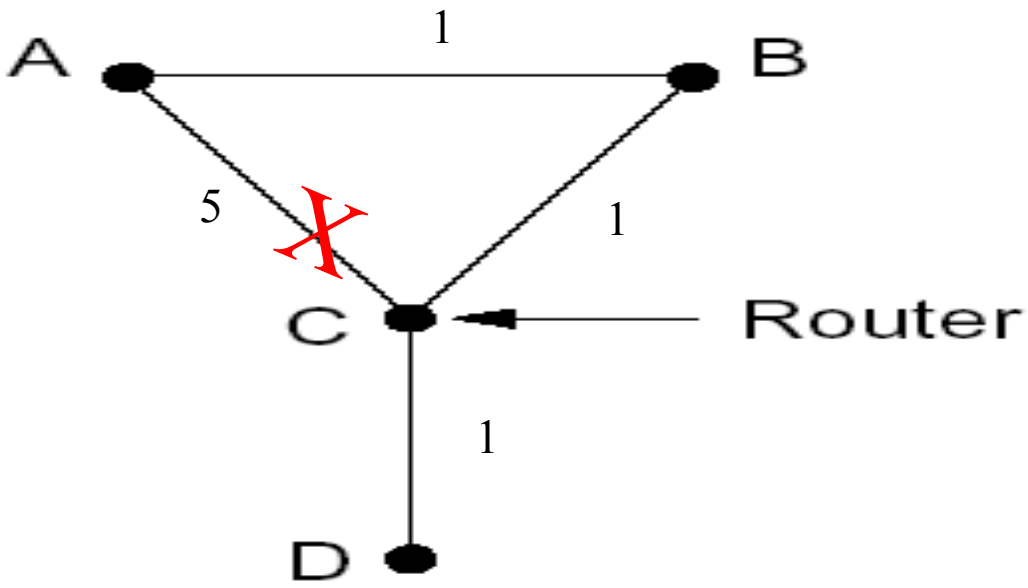
- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Distance vector protocols (distributed computing)
        - Synchronous Bellman-Ford (SBF)
        - Asynchronous Bellman-Ford (ABF)
        - Properties of DV
        - Distributed protocols w/ safety (loop prevention)
          - Reverse poison safety and Routing Information Protocol (RIP)
          - *No increase cost safety and the Destination-Sequenced DV Protocol (DSDV)*

Destination-Sequenced Distance Vector Protocol (DSDV)



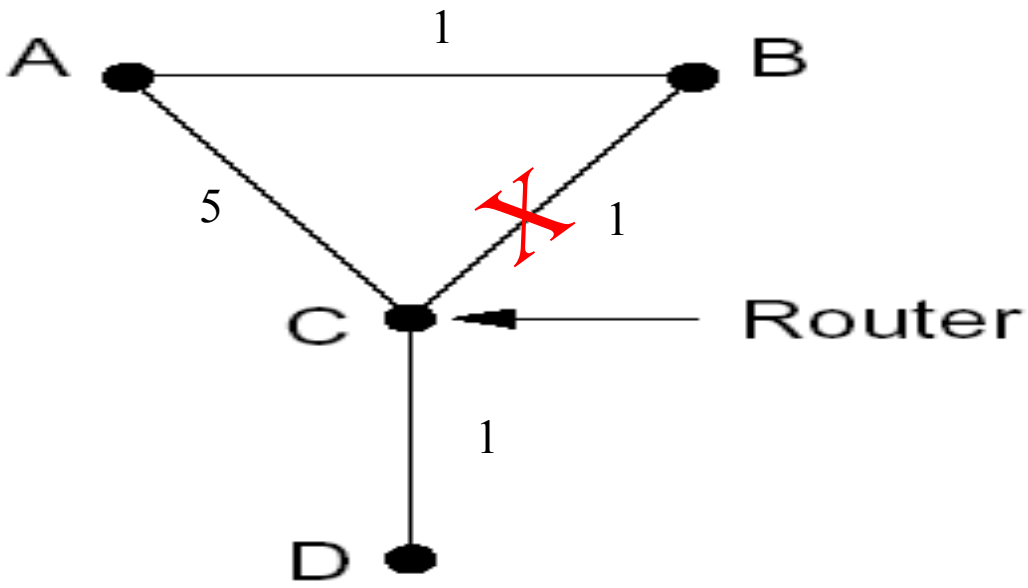
- Basic idea: to enforce NIC but still allow link cost increase (e.g., when link broken), use sequence numbers to partition computation
  - tags each route with a sequence number: each node B maintains (S^B, d^B) for each destination D, where S^B is the sequence number at B for destination D and d^B is the best distance using a neighbor from B to D
- External triggered messages
  - Timer-triggered: periodically each destination D increases its seq. by 2 and broadcasts with $(S^D, 0)$
  - Link-event triggered: if B is using C as next hop to D and B discovers that C is no longer reachable
    - B increases its sequence number S^B by 1, sets d^B to infinite ∞, and sends (S^B, d^B) to all neighbors

Example



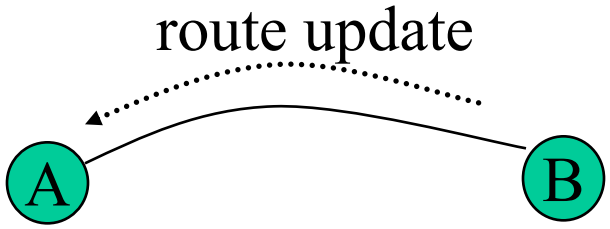
Will this trigger an update?

Example



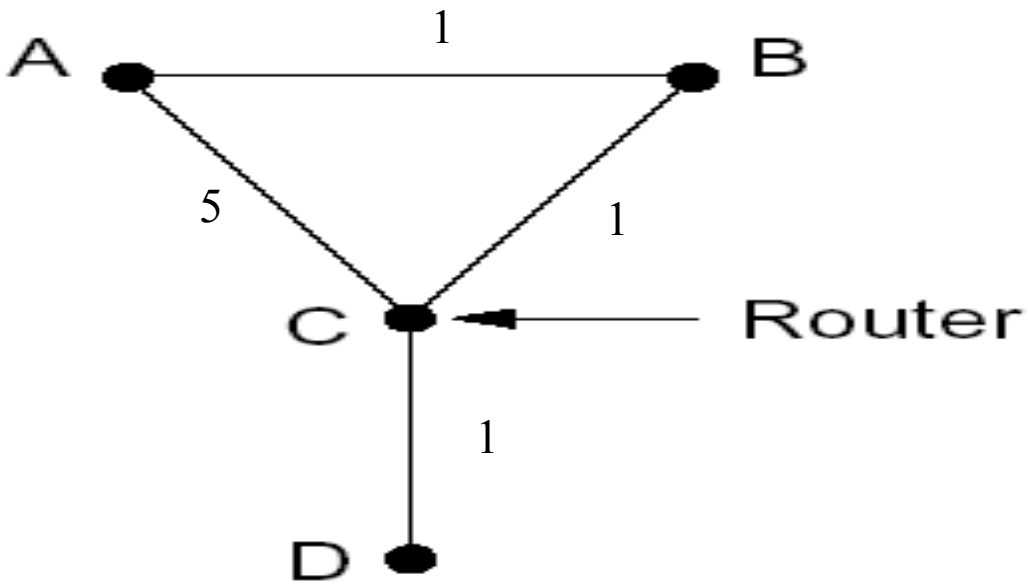
Will this trigger an update?

DSDV: Msg-Triggered Update



- Update after receiving a message
  - assume B sends to A its current state (S^B, d^B)
  - when A receives (S^B, d^B)
    - if $S^B > S^A$, then
      // always update if a higher seq#
      - » $S^A = S^B$
      - » if ($d^B == \infty$) $d^A = \infty$; else $d^A = d^B + d(A,B)$
    - else if $S^A == S^B$, then
      - » if $d^A > d^B + d(A,B)$
        // update for the same seq# only if better route
        $d^A = d^B + d(A,B)$ and uses B as next hop

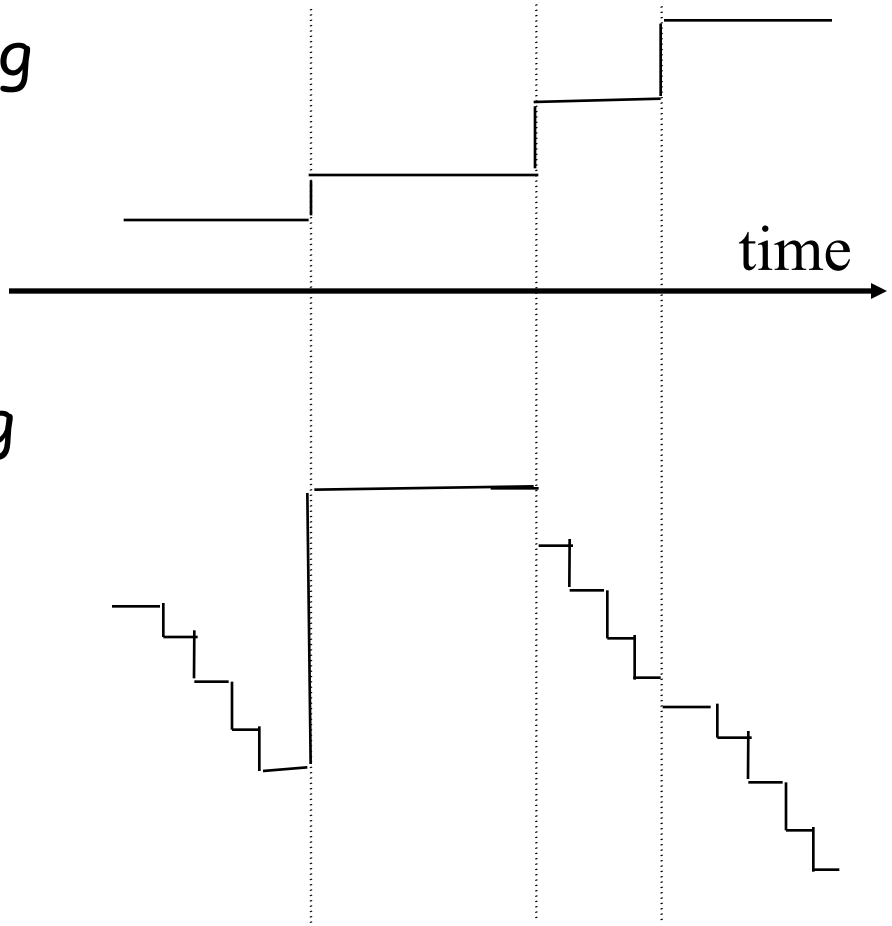
Example



Exercise: update process after D increases its seq# to next even number.

Global Invariants of DSDV: Single Node B

- Some invariants about the state of a node B
  - [I1] S^B is non-decreasing
  - [I2] d^B is non-increasing for the same sequence number



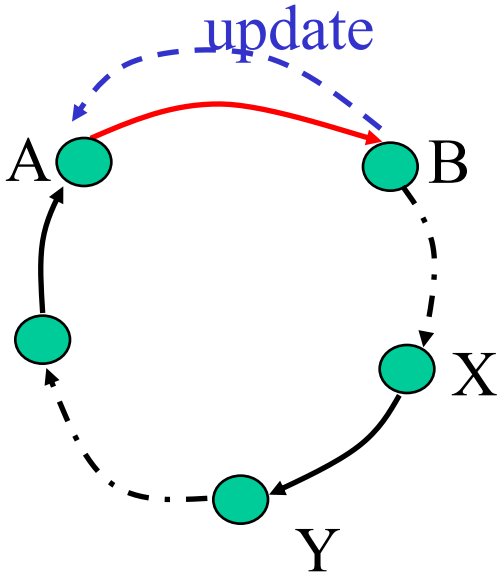
Global Invariants: if A Considers B as Next Hop

- Some invariants if A considers B as next hop

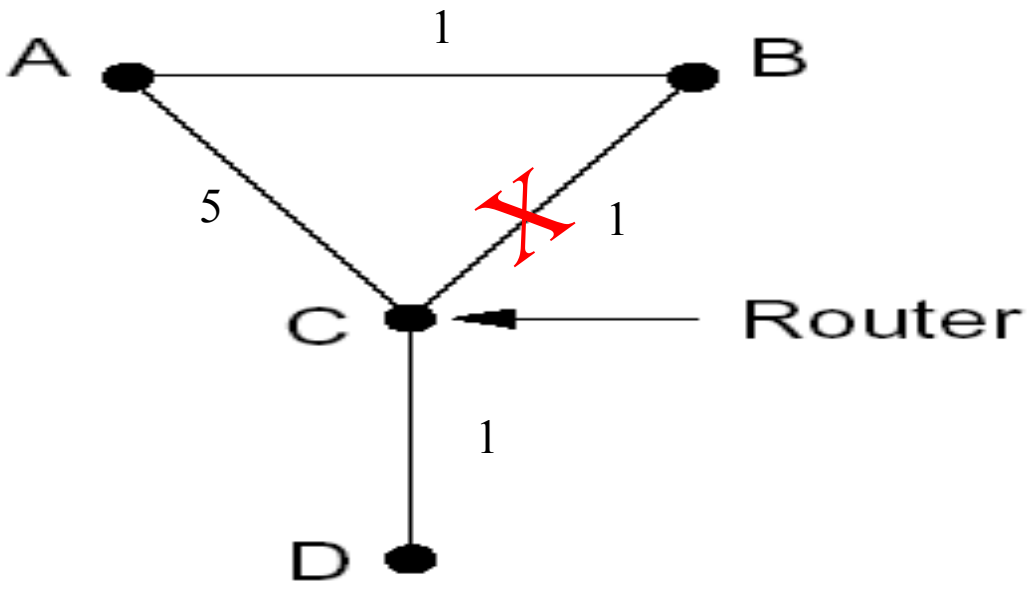
  - [I3] d^A is not ∞
  - [I4] $S^B >= S^A$
    - because A is having the seq# which B last sent to A; B's seq# might be increased after B sent its state
    - [I5] if $S^B == S^A$
    - then $d^B < d^A$ because d^A is based on d^B which B sent to A some time ago, $d^B < d^A$ since all link costs are positive; d^B might be decreased after B sent its state

Loop Freedom of DSDV

- Consider a critical moment
  - A starts to consider B as next hop, and we have a loop
- According to invariant I4 for each link in the loop (X considers Y as next hop) : $S^Y >= S^X$
- Two cases:
  - exists $S^Y > S^X$
    - by transition along the loop $S^B > S^B$
  - all nodes along the loop have the same sequence number
    - apply I5, by transition along the loop $d^B > d^B$



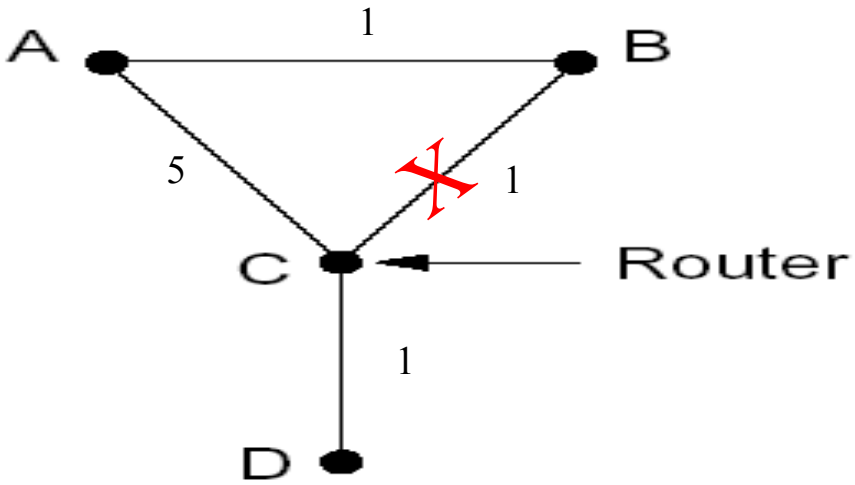
Discussion



- When the BC link fails, B loses its path to D, but A actually has C as an alternative (backup) path.
- Is there a local condition for A to realize that C is usable?

Issues of DSDV

- DSDV guarantees no loop, but need global recomputation after each cost increase (not using any backup path).
- DSDV does not include a mechanism to use multiple paths (backup, load balancing)

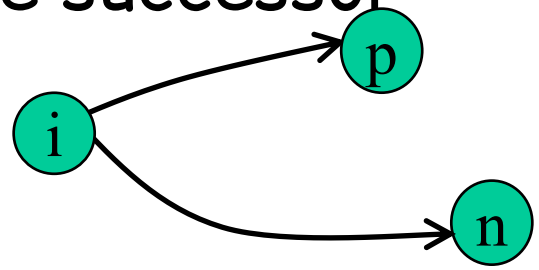


Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Distance vector protocols (distributed computing)
        - Synchronous Bellman-Ford (SBF)
        - Asynchronous Bellman-Ford (ABF)
        - Properties of DV
        - Distributed protocols w/ safety (loop prevention)
          - Reverse poison safety and Routing Information Protocol (RIP)
          - *No increase cost safety and the Destination-Sequenced DV Protocol (DSDV)*
          - *Feasible Distance safety and Diffusive Update algorithm (DUAL) and EIGRP*

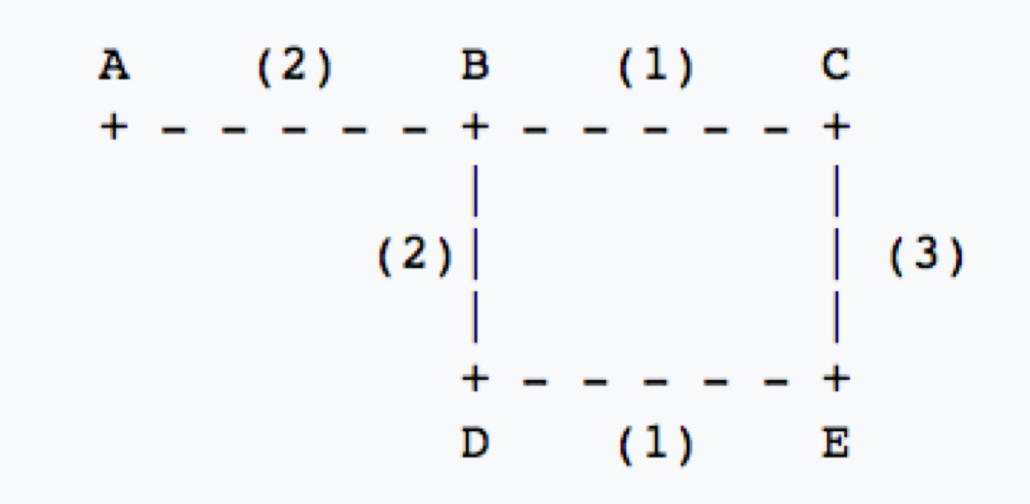
Key Idea: Feasible Successors [EIGRP]

- If the reported distance of a neighbor n is lower than the lowest ever total distance to reach destination D (i.e., neighbor n is closer than this node has ever been), the neighbor n is a feasible successor



$$d_n + d_{i \to \text{n}} \geq d_{\text{primary}} + d_{i \to \text{primary}} > d_{\text{n}}$$

Example



- Assume A is destination, consider E

	Reported Distance	Total Distance
Neighbor C	3	6
Neighbor D	4	5

Q: Which one is primary path?

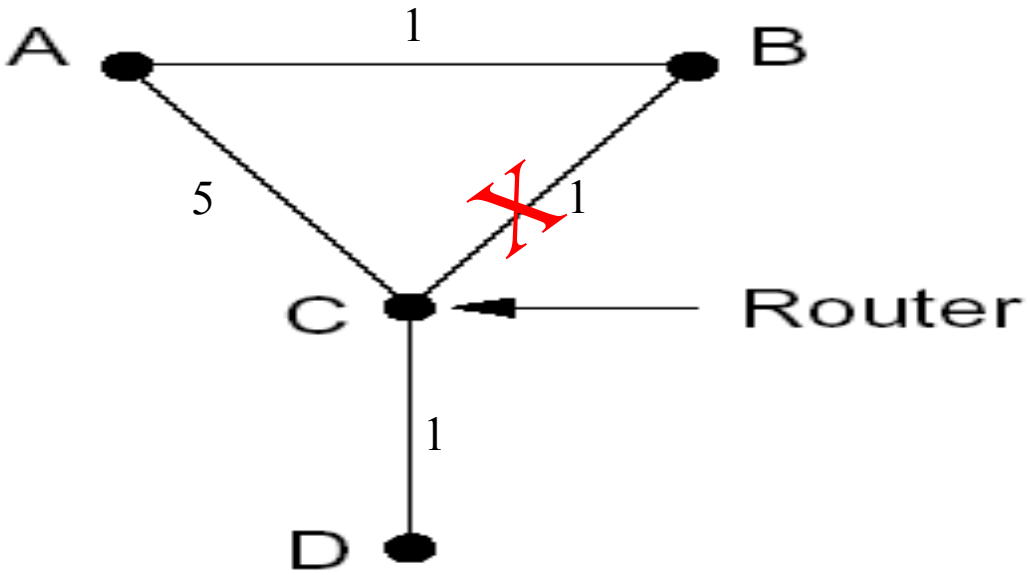
Q: If link to primary fail, can E use the back up?

Offline Exercise

- Identify the invariants and prove the safety property of the feasible distance condition

Pointer

- What if a node has no feasible successor?
  - Hint: a diffusive distributed computing protocol to compute the next route
    - See Dijkstra-Scholten algorithm



Summary: Distance Vector Routing

- Basic distance vector
  - take away: use monotonicity as a technique to understand liveness/convergence
    - highly recommended reading of Bersekas/Gallager chapter; see Schedule page
- Fix counting-to-infinity problem
  - Take away: use local condition to enforce global properties; use global invariants to understand/design safety/no routing loops
  - Diffusive computing model can be a powerful tool to design distributed protocols [not fully covered, but highly recommended reading; see schedule page]

Discussion: Distance Vector Routing

- What do you like about distributed, distance vector routing?

- What do you not like about distributed, distance vector routing?

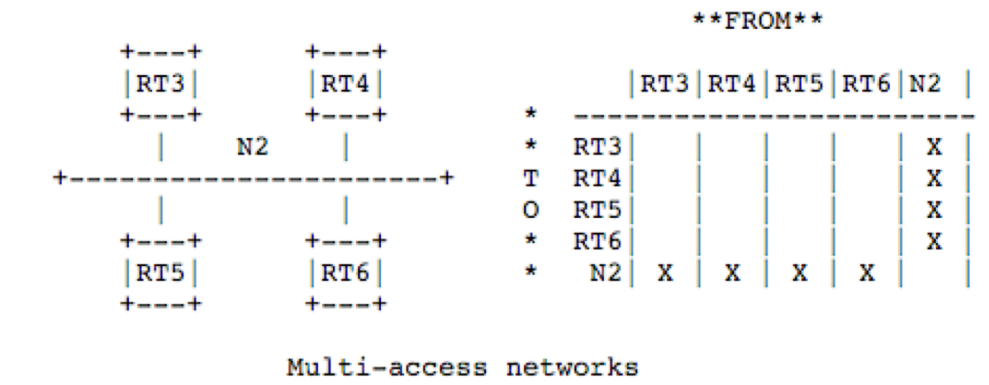
Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Distance vector protocols (distributed computing)
      - *Link state protocols (distributed state synchronization)*

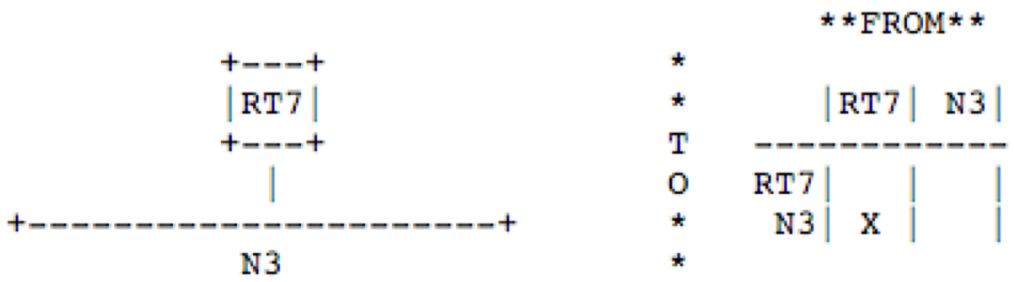
Link-State Routing

- Basic idea: Not distributed computing, only distributed state distribution
- Net topology, link costs are distributed to all nodes
  - All nodes have same info
  - Each node computes shortest paths from itself to all other nodes
    - Standard Dijkstra's algorithm as path compute algorithm
    - Allows multiple same-cost paths
    - Multiple cost metrics per link (for type of service routing)
- Often used by large networks (OSPF by large enterprises; ISIS by service providers)

Example: Link State and Directed Graph (OSPFv2)



Multi-access networks



Stub multi-access networks

https://tools.ietf.org/html/rfc1583#page-12

Example: Link State and Directed Graph (OSPFv2)

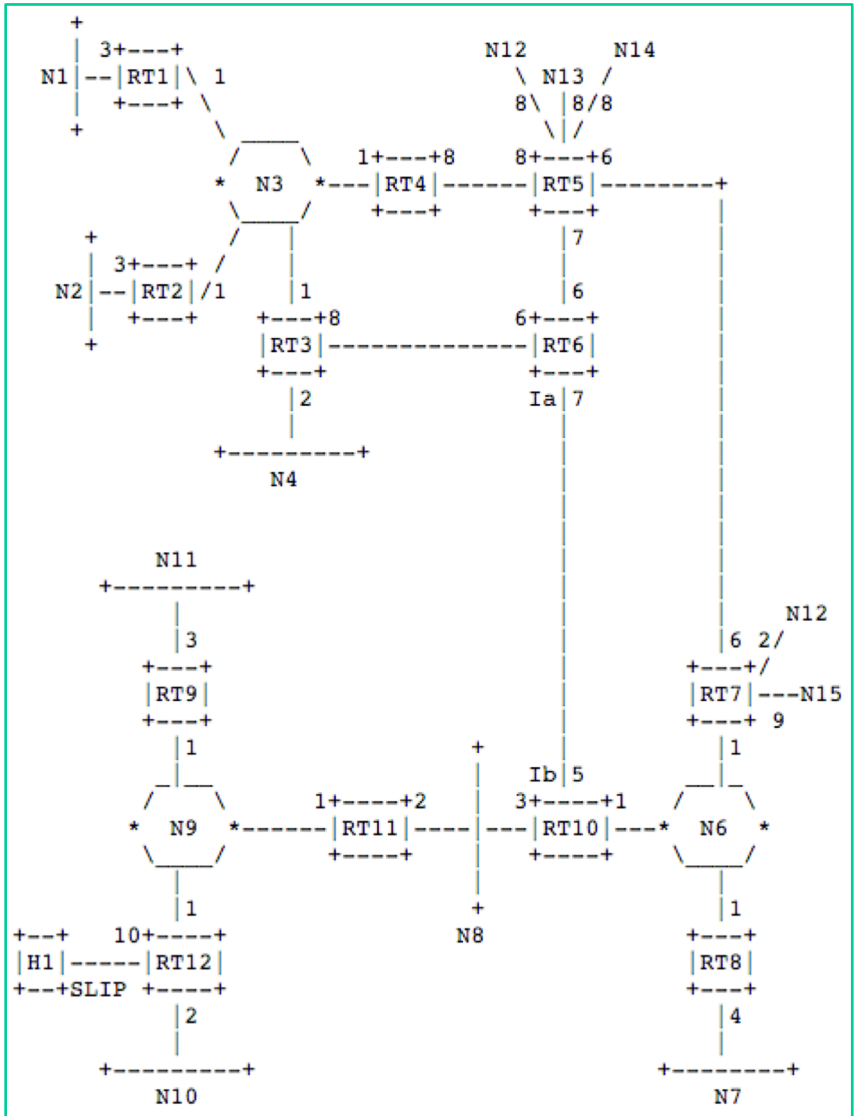


Figure 2: A sample Autonomous System

FROM

TO	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	N3	N6	N8	N9
RT1													0			
RT2													0			
RT3						6							0			
RT4					8								0			
RT5				8		6	6									
RT6			8		7					5						
RT7					6									0		
* RT8														0		
* RT9																0
T RT10						7								0	0	
O RT11															0	0
* RT12																0
* N1	3															
N2		3														
N3	1	1	1	1												
N4			2													
N6							1	1		1						
N7								4								
N8										3	2					
N9									1		1	1				
N10												2				
N11									3							
N12					8		2									
N13					8											
N14					8											
N15							9									
H1												10				

Figure 3: The resulting directed graph

Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Distance vector protocols (distributed computing)
      - *Link state protocols (distributed state synchronization)*
        - data structure to be distributed
        - *state distribution protocol*

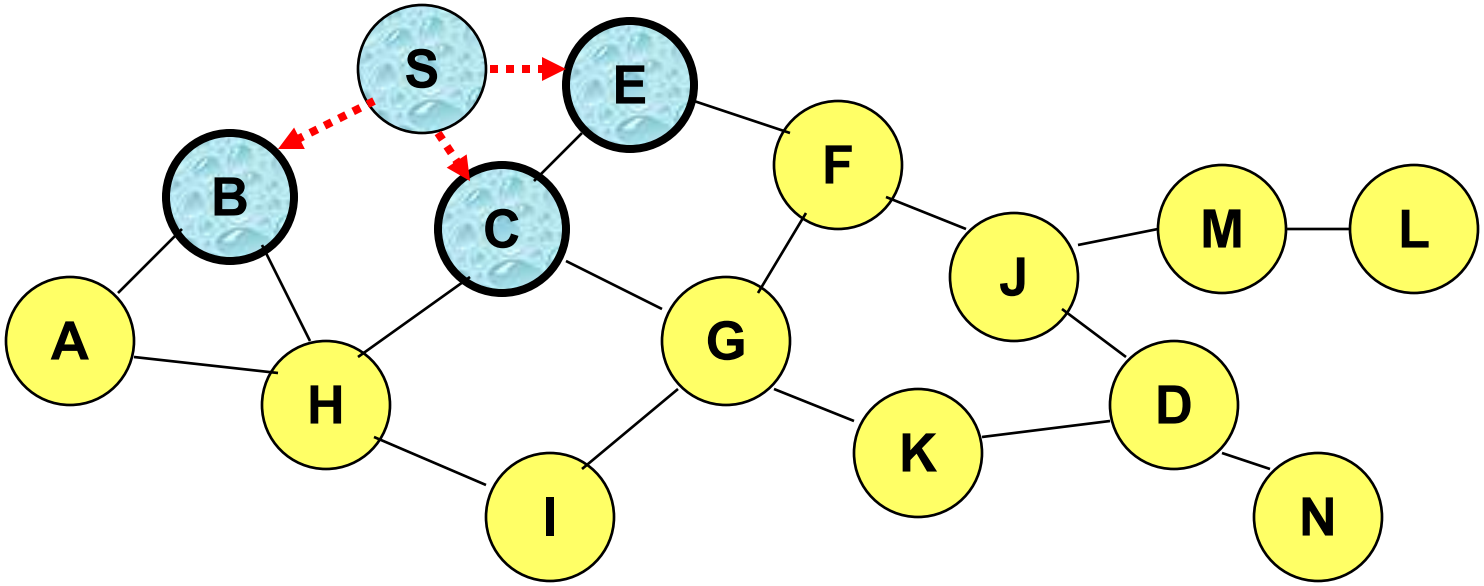
Basic Link State Discovery Broadcast Protocol

Basic event structure at node n

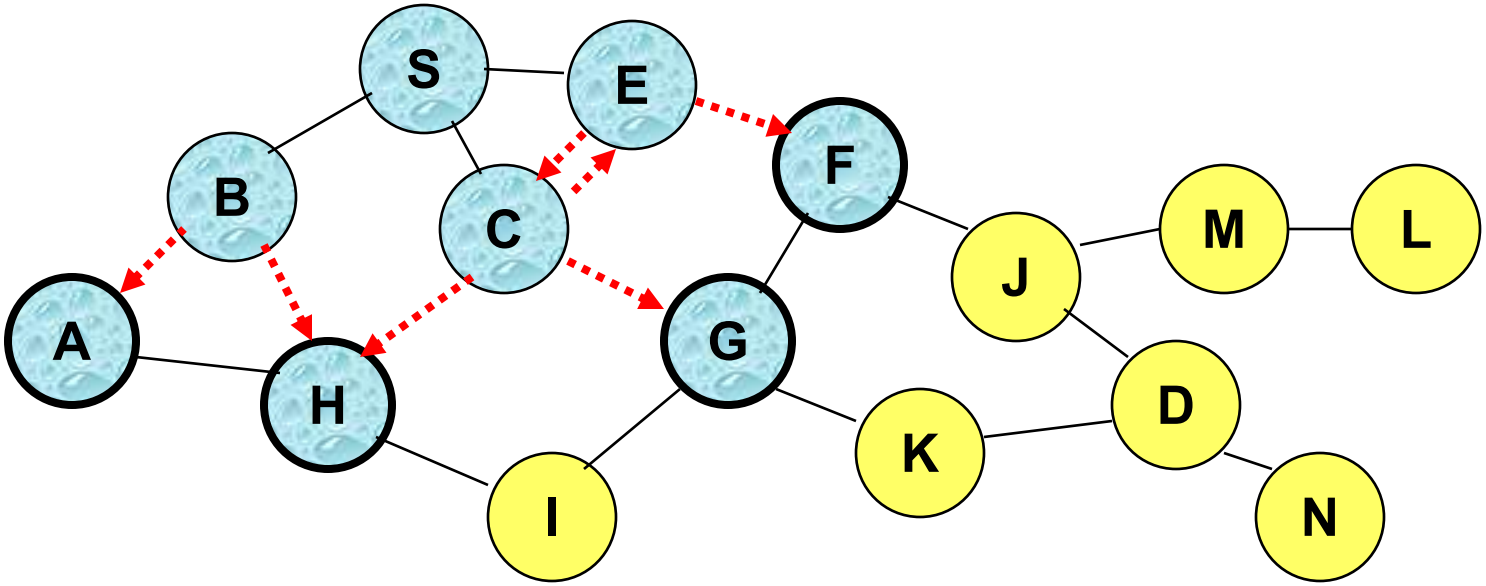
- periodically:
  - broadcast at each interface HELLO msg so that neighbor can discover me
- on discovering new status of a directly connected link e (receive HELLO from a new neighbor or a neighbor misses N HELLO msgs)
  - broadcast new status of LSA[e]
- on receiving an LSA[e]:
  - if (does not have LSA[e])
    forwards LSA[e] to all links except the incoming link

Link State Broadcast

Node S updates link states connected to it.



Link State Broadcast



To avoid forwarding the same link state announcement (LSA) multiple times (forming a loop), each node remembers the received LSAs.
- Second LSA[S] received by E from C is discarded
- Second LSA[S] received by C from E is discarded as well
- Node H receives LSA[S] from two neighbors, and will discard one of them

Discussion

- Issues of the basic link state protocol?
  - Recall: goal is to efficiently distribute to each node to a correct, complete link state map

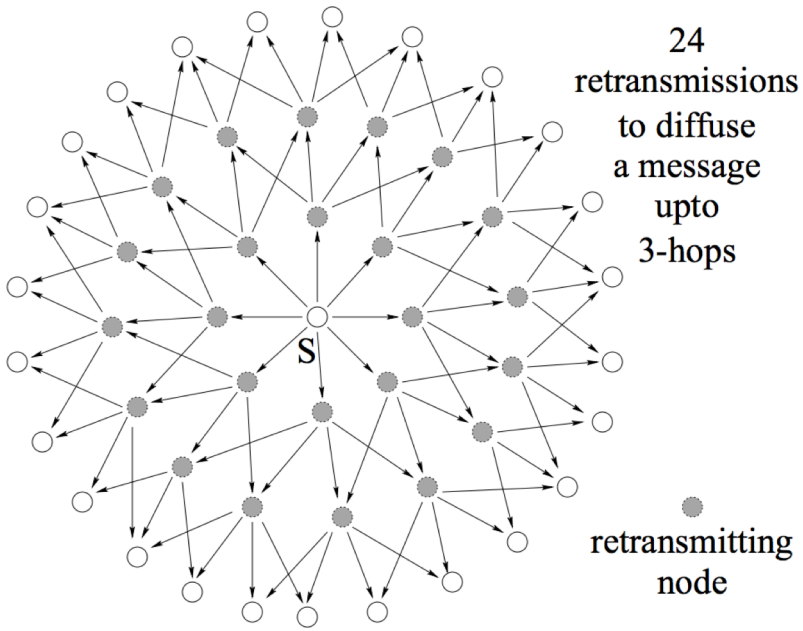
Link State Broadcast: Issues

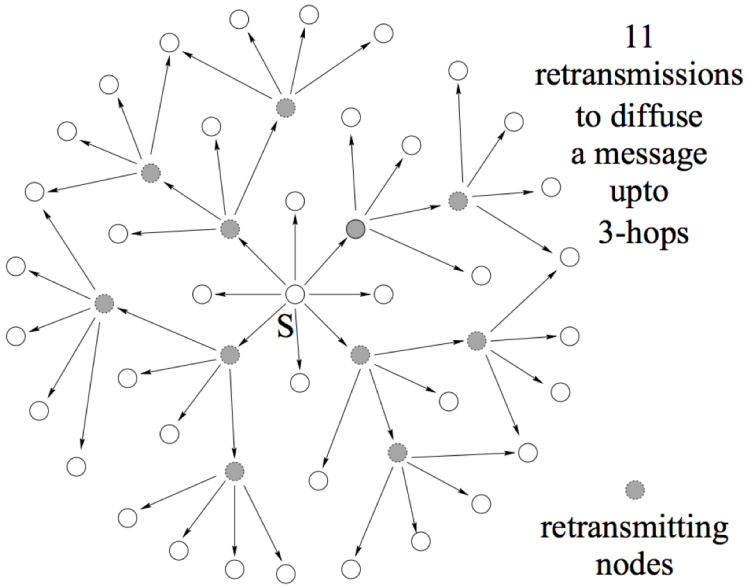
- Correctness problem: network partition and then reconnect, basic protocol propagates only the single reconnection link, how to sync across the reconnected components?

- Solution: updates are sent periodically

Link State Broadcast: Issues

- Problem: Broadcast redundancy

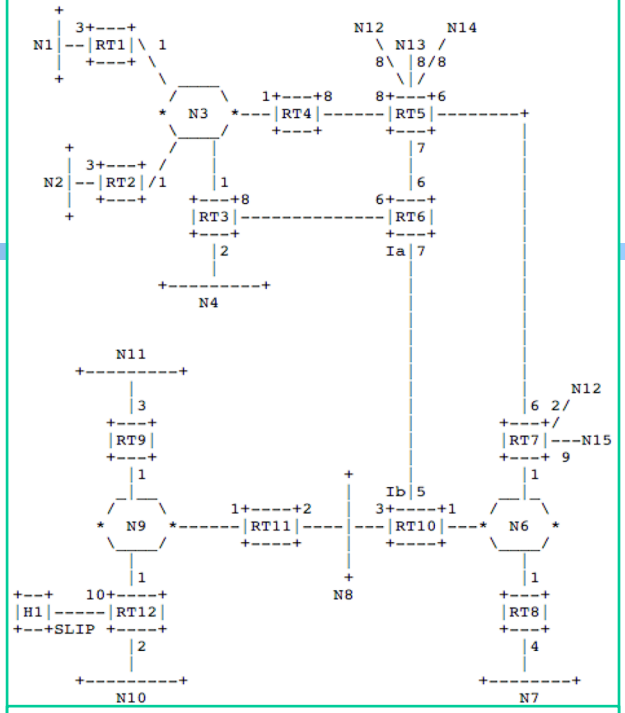




https://hal.inria.fr/inria-00072756/document

Summary: Link State

- Basic LS protocol
  - take away: instead of computing routing results using distributed computing, distributed computing is for only link state distribution (synchronization)
- Link state distribution can still have much complexity, e.g., partition and reconnect, scalability



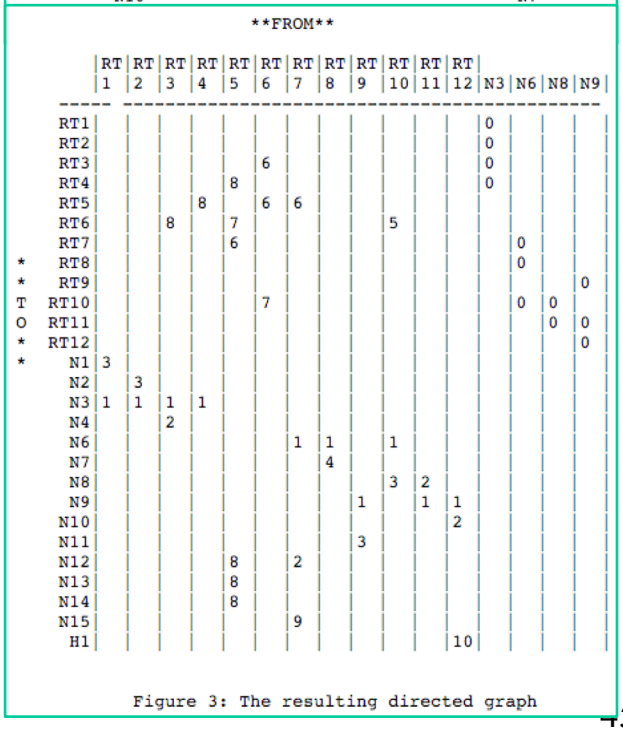


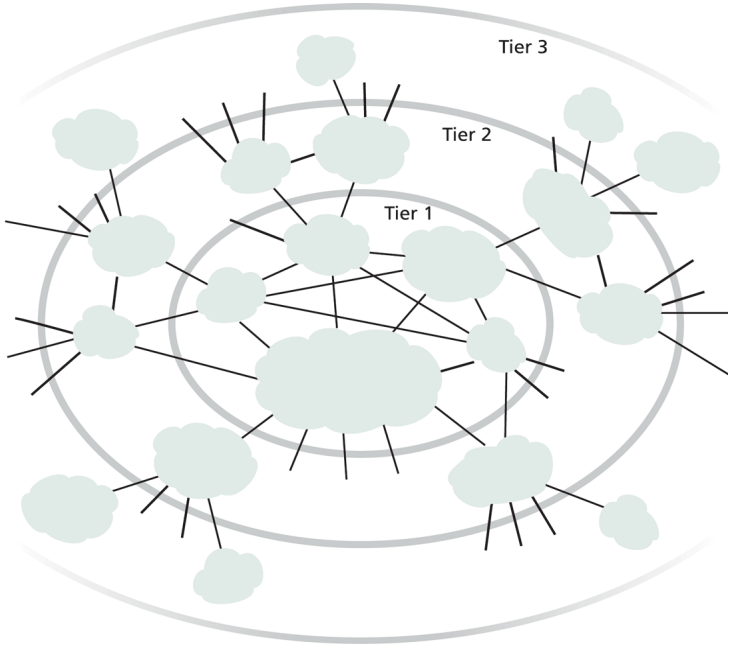
Figure 3: The resulting directed graph

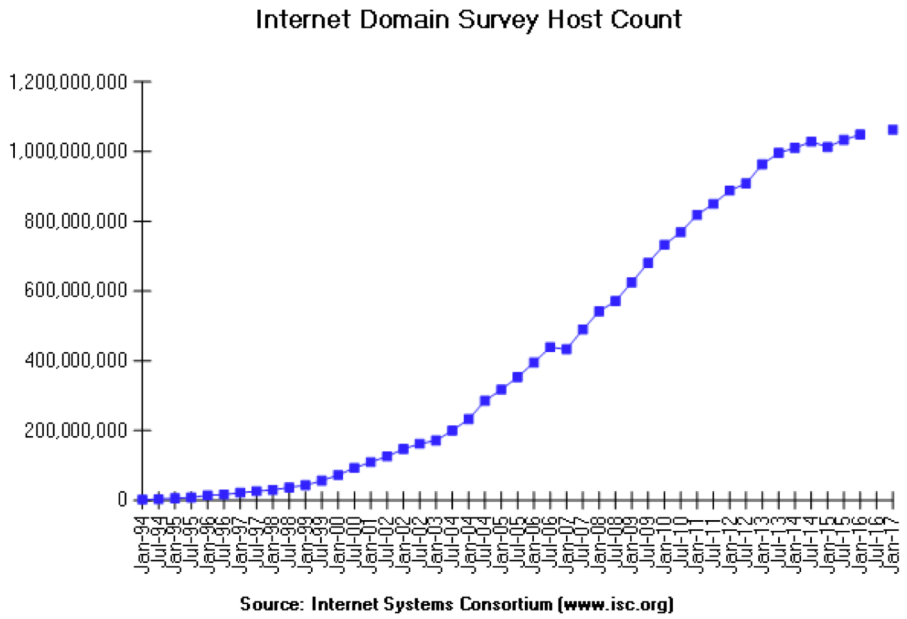
Outline

- Admin and recap
- Network control plane
  - Routing
    - Link weights assignment
    - Routing computation
      - Basic routing computation protocols
        - Distance vector protocols (distributed computing)
        - Link state protocols (distributed state synchronization)
    - Global Internet routing

Discussion

- Does it work to use DV or LS as we discussed for global Internet routing?





Requirements and Solution of Current Global Internet Routing

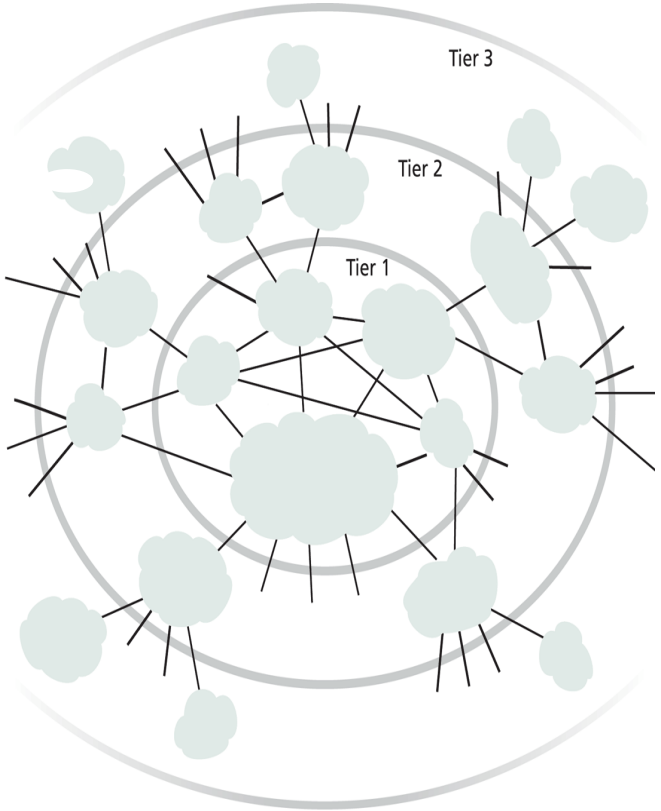
- Scalability: handle network size (#devices) much higher than typical DV or LS can handle
  - Solution: Introduce new abstraction (hierarchy) to reduce network (graph) size

- Autonomy: allow each network to have individual preference of routing (full control of its internal routing; control/preference of routing spanning multiple networks)
  - Solution: autonomous, policy routing

New Abstraction: Autonomous Systems (AS)

- Abstract each network as an autonomous system (AS), identified by an AS number (ASN)

- Conceptually the global routing graph consists of only autonomous systems as nodes



Exercise: https://www.ultratools.com/tools/asnInfo

Global Routing: Bigger Picture

- Global Internet routing is divided into inter-AS routing and intra-AS routing
  - Inter-AS routing (also called interdomain routing)
    - A protocol runs among autonomous systems is also called an Exterior Gateway Protocol (EGP)
    - The de facto EGP protocol is BGP
  - Intra-AS routing (also called intradomain routing)
    - A protocol running insides an AS is called an Interior Gateway Protocol (IGP), each AS can choose its own protocol, such as RIP, E/IGRP, OSPF, IS-IS

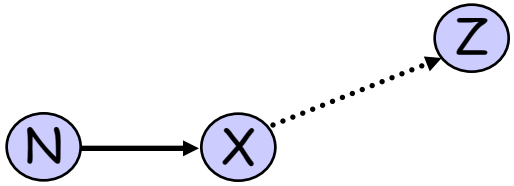
Border Gateway Protocol (BGP) Interdomain Routing

- BGP is **a Path Vector** protocol
  - similar to Distance Vector protocol
  - a border gateway sends to a neighbor *entire path* (i.e., a sequence of ASNs) to a destination, e.g.,
    - gateway X sends to neighbor N its path to dest. Z:

$$\text{path } (X,Z) = X,Y1,Y2,Y3,\ldots,Z$$

  - if N selects path(X, Z) advertised by X, then:

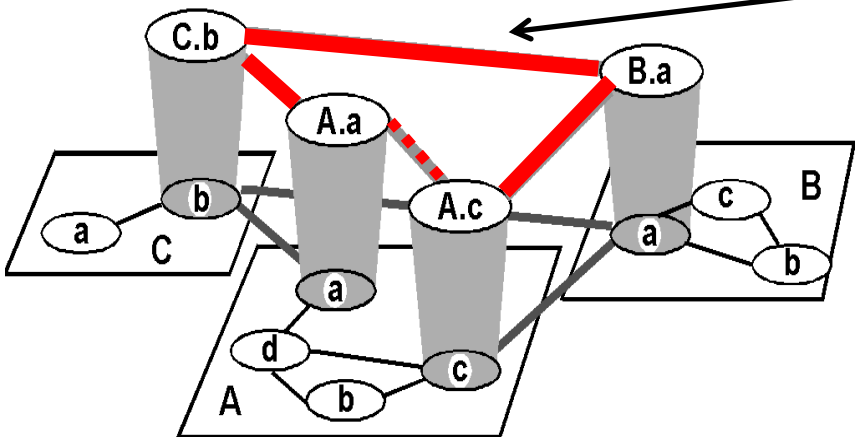
$$\text{path } (N,Z) = N, \text{ path } (X,Z)$$



Exercise: Observing BGP Paths

- Use one of the looking glass servers (http://www.bgp4.as/looking-glasses)
  - List of destinations announced by an autonomous system:
    - http://irrexplorer.nlnog.net/search/29
  - Click on AS number of each address to see all paths announced
    - https://lg.de-cix.net/alice/search

Bigger Picture: Integration of Intra- and Inter-Domain Routing



A potential view: Interdomain routers as an overlay

Gateway routers of same AS share learned external routes using iBGP.

Gateway routers participate in intradomain to learn internal routes.

AS C (RIP intra routing)

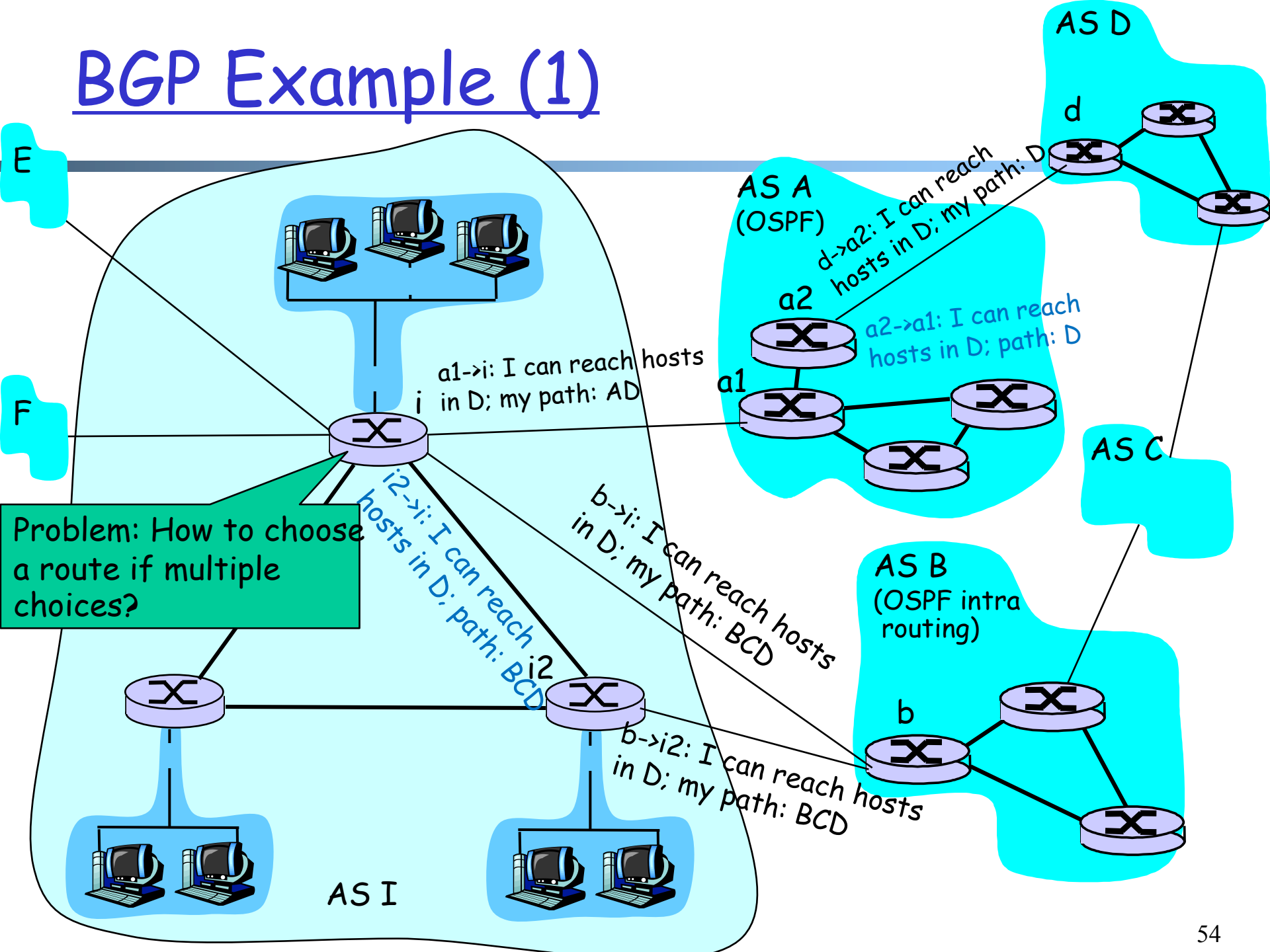
AS A (OSPF intra routing)

AS B (OSPF intra routing)

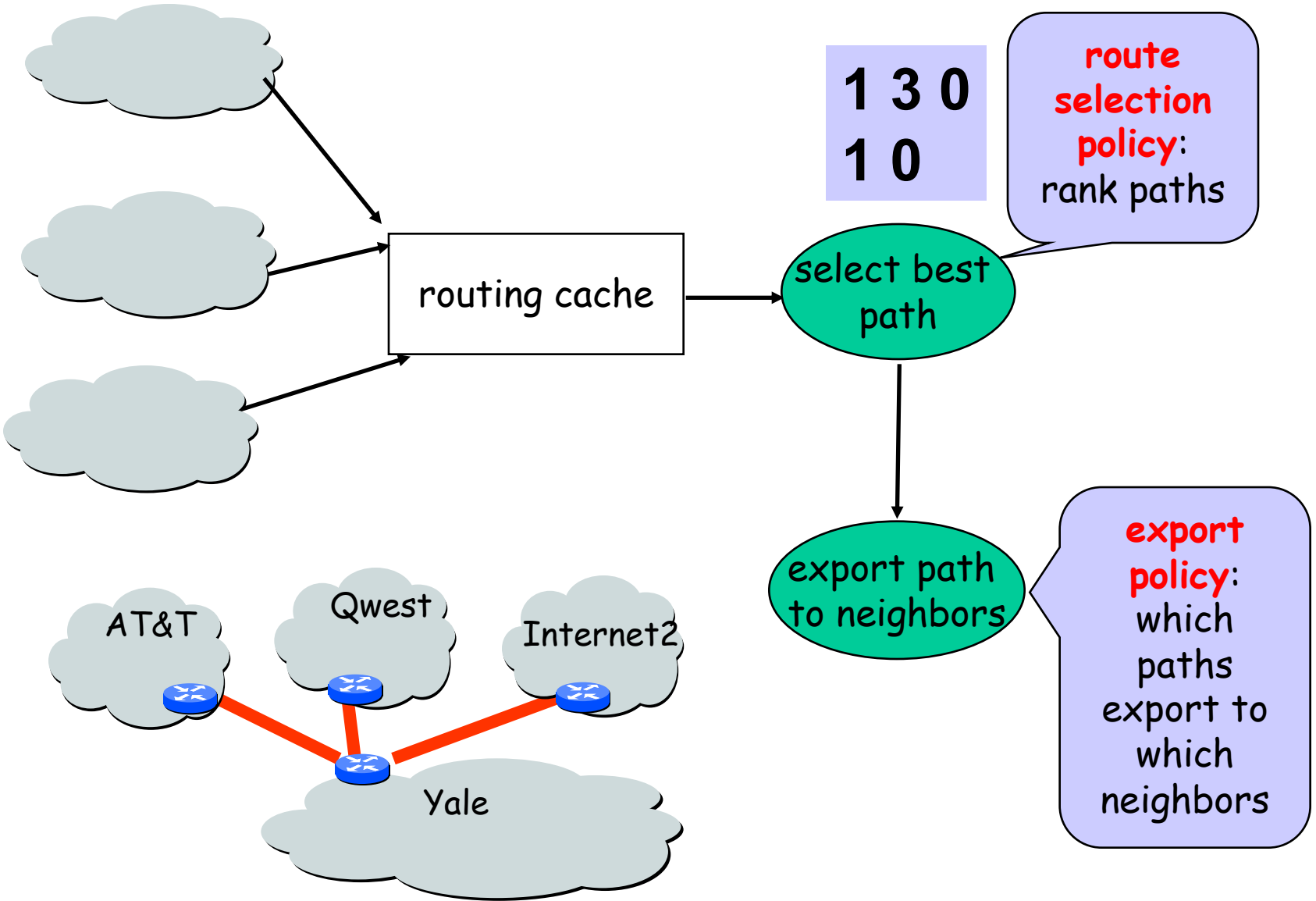
a b e eBGP iBGP d c f g i h

Gateway routers of diff. auto. systems exchange routes using eBGP

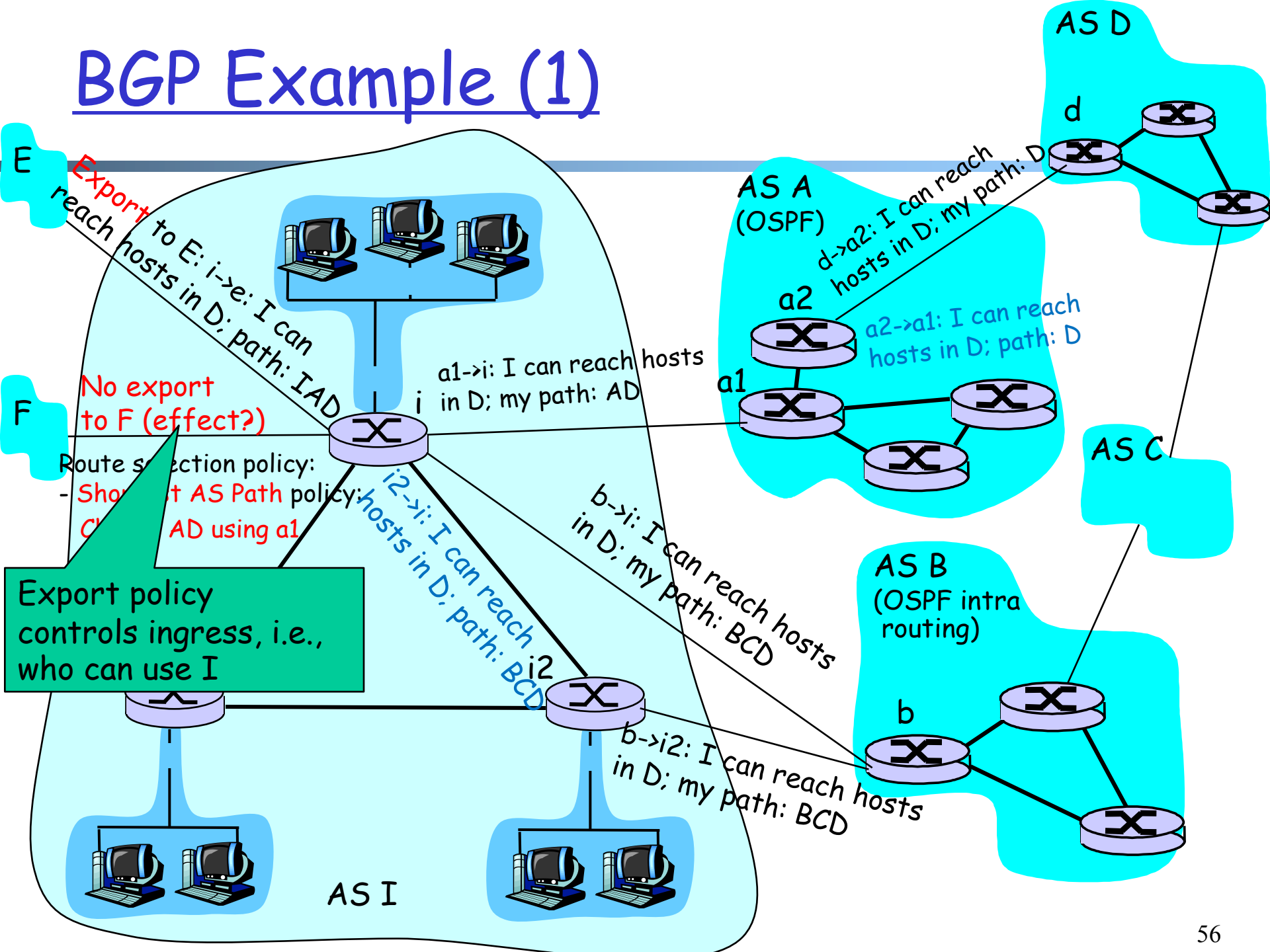
BGP Example (1)



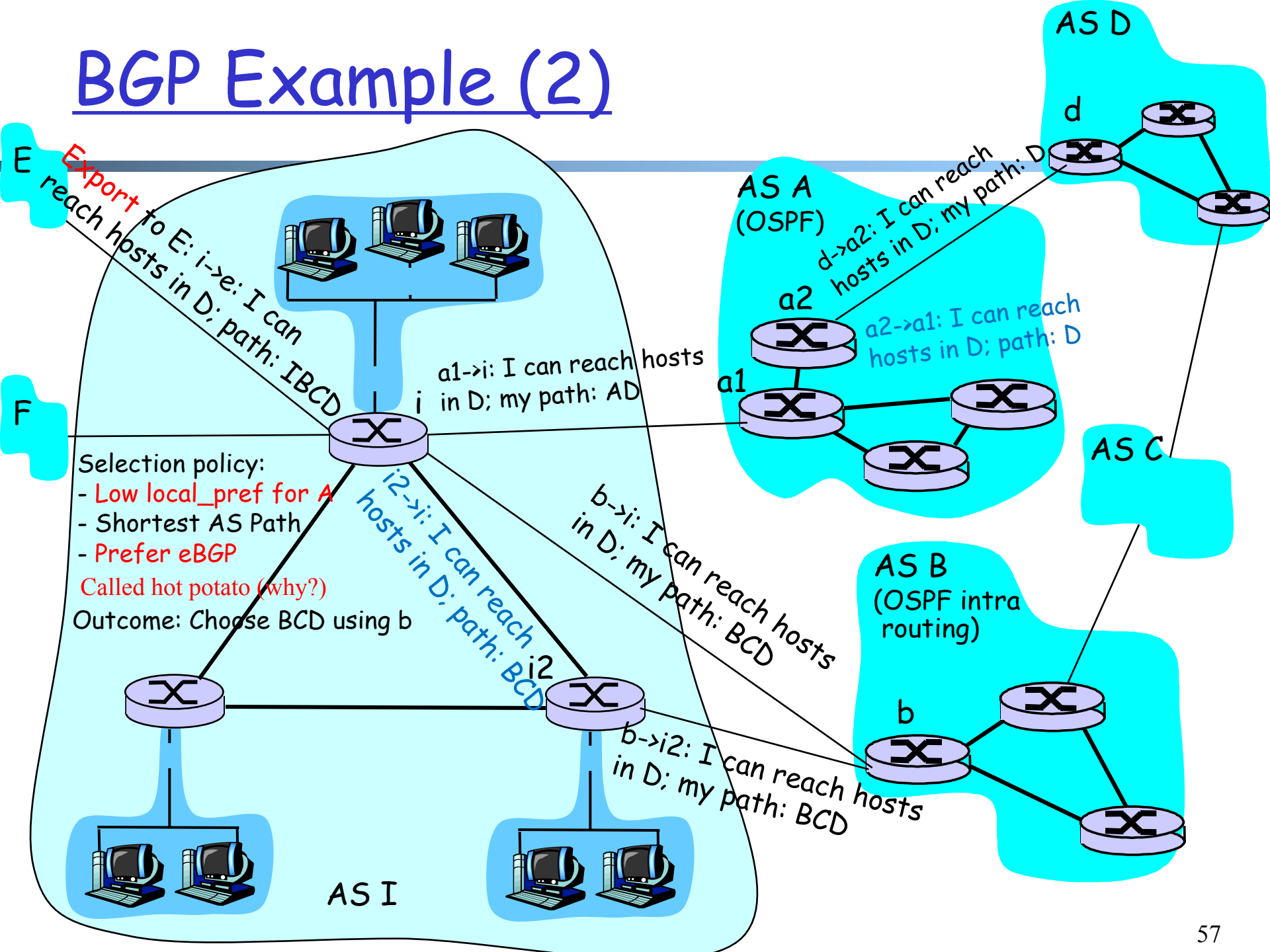
BGP Policy Routing Framework: Decision Components



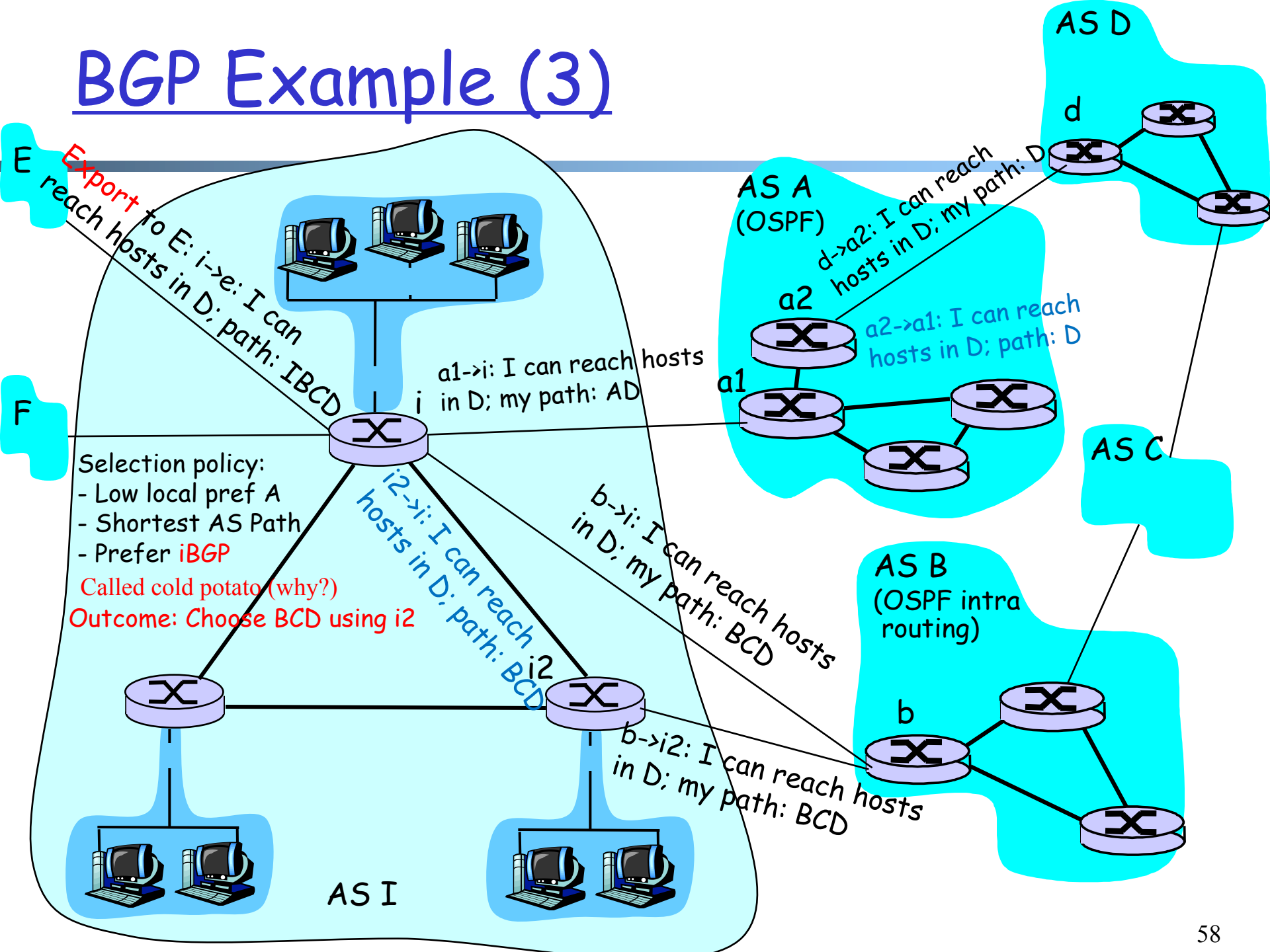
BGP Example (1)



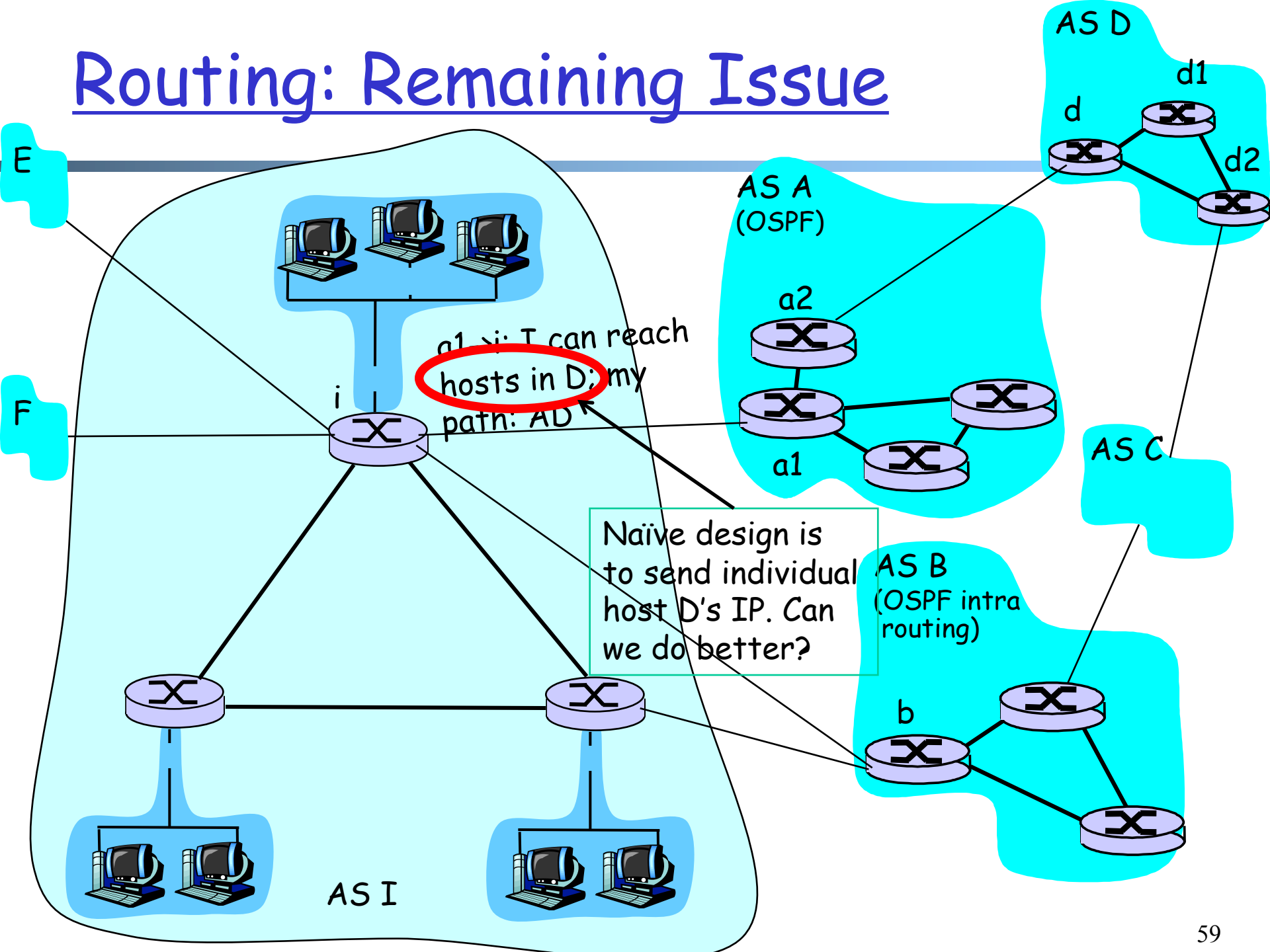
BGP Example (2)



BGP Example (3)



Routing: Remaining Issue



IP Addressing Scheme: Requirements

- Uniqueness: We need an address to uniquely identify each destination
- Aggregability : Routing scalability needs flexibility in aggregation of destination addresses
  - we want to aggregate as a large set of destinations as possible in BGP announcements
- Current: the unit of routing in the Internet is a classless interdomain routing (CIDR) address

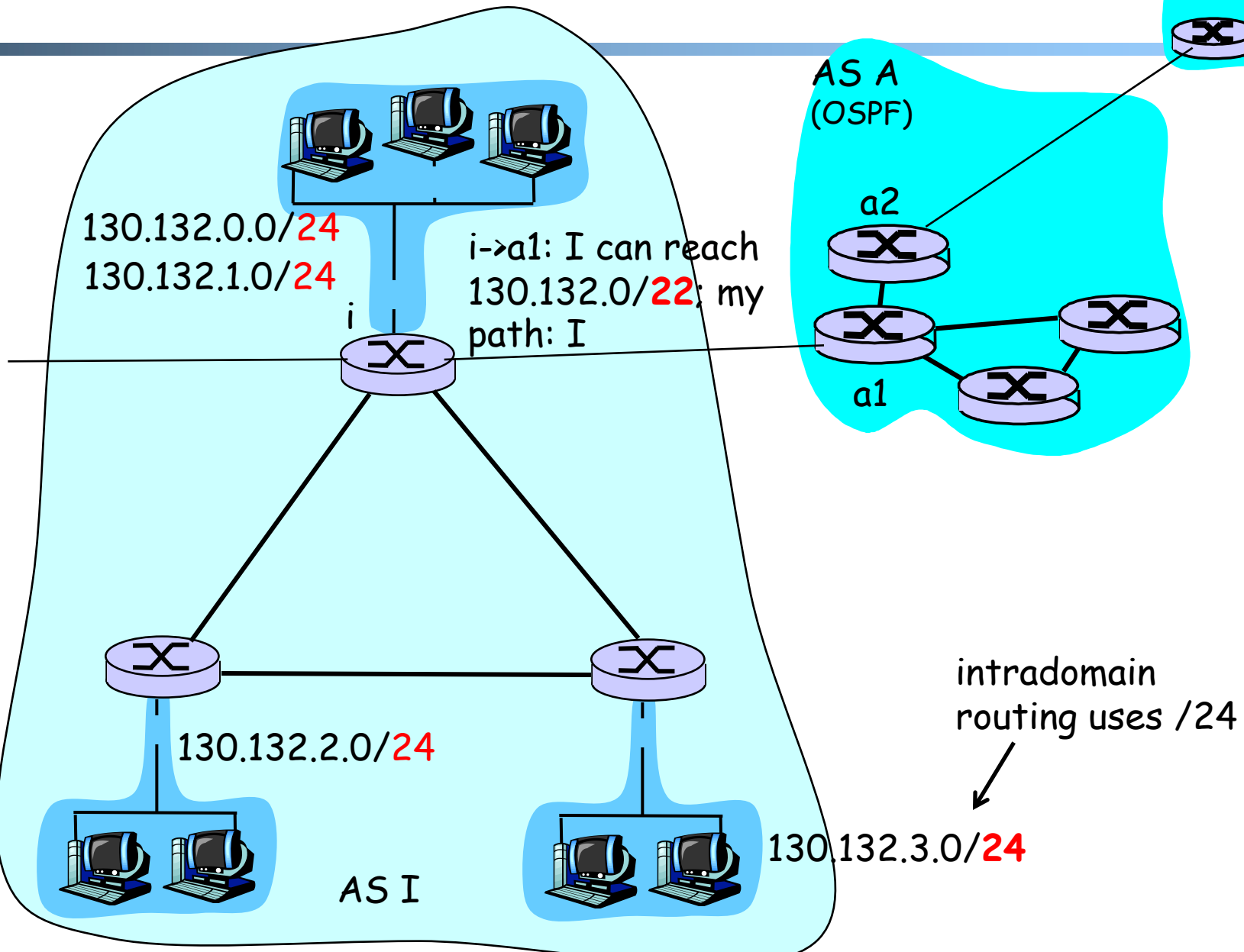
Classless InterDomain Routing (CIDR) Address: Aggregation

- A CIDR address partitions an IP address into two parts
  - A prefix representing the network portion, and the rest (host part)
  - address format: a.b.c.d/x, where x is # bits in network portion of address

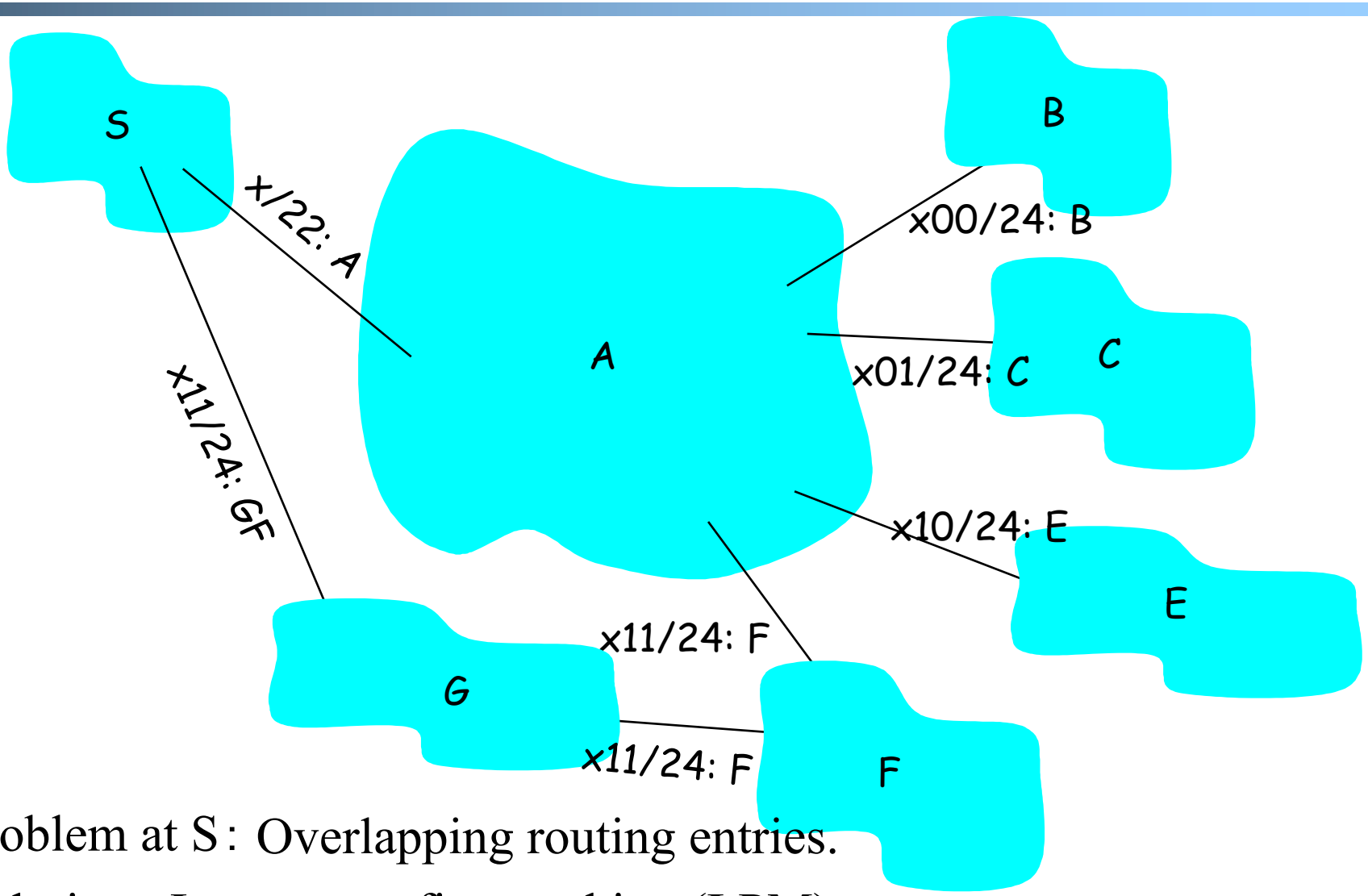
200.23.16.0/23

Some systems use mask (1's to indicate network bits), instead of the /x format

CIDR Aggregation in BGP



CIDR Aggregation in BGP



Problem at S : Overlapping routing entries.

Solution : Longest prefix matching (LPM)