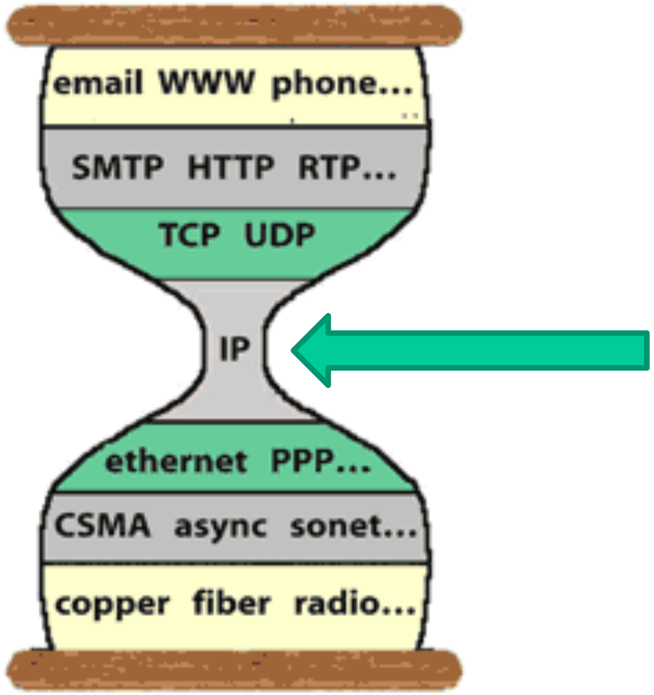
Lecture 13: Routing Protocols



5.1 Introduction

5.2 Routing protocols

- Link state
- Distance vector

5.3 Intra-AS routing in the Internet: OSPF

5.4 Routing among the ISPs: BGP

Internet routing: what we have learned so far

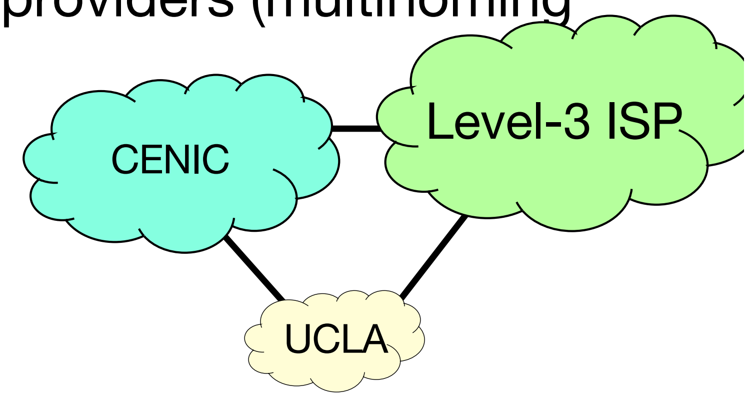
- Shortest path *algorithms*
  - Link-state (Dijkstra): **each node** computes its shortest paths to all other nodes using *the topology map*
  - Distance Vector (Bellman-Ford): **each node** computes its shortest paths to all other nodes based on its *neighbors' distance to all destinations*
- Routing protocols
  - Link-state protocol: each node's updates (link state) flood to the entire network
  - Distance-vector protocol: each node's updates (distance vector) is sent to its direct neighbors

What else a routing protocol must also do

- Monitor link and neighbor nodes status
- Once a failure is detected, send routing update to inform the rest of the network of the changes
- Mitigate potential packet losses in routing update delivery
  - Link state: explicitly tell **everyone** link is down
  - Distance vector: explicitly tell **neighbors** D(v) has changed

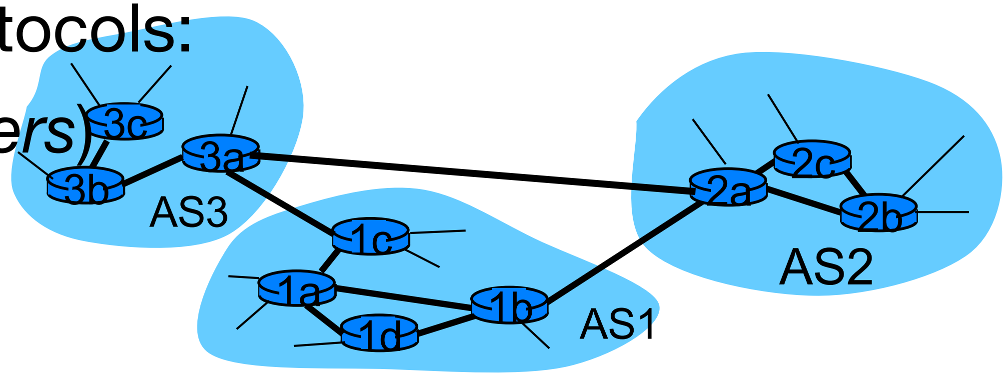
Routing in the Global Internet

- **Within an administrative domain**: all routers faithfully run the *same* routing protocol
  - A common goal: Find the *best* paths to all destinations based on delay, loss, bandwidth, or other *shared* measures
- **Global Internet**: interconnection of a large number of *Autonomous Systems (AS)*
  - Each AS is assigned a unique autonomous system number (ASN)
  - Stub AS: end user networks (corporations, campuses)
    - A stub may connects to multiple service providers (multihoming
  - Transit AS: Internet service provider
    - They may also offer connectivity to user networks (which are not ASes)



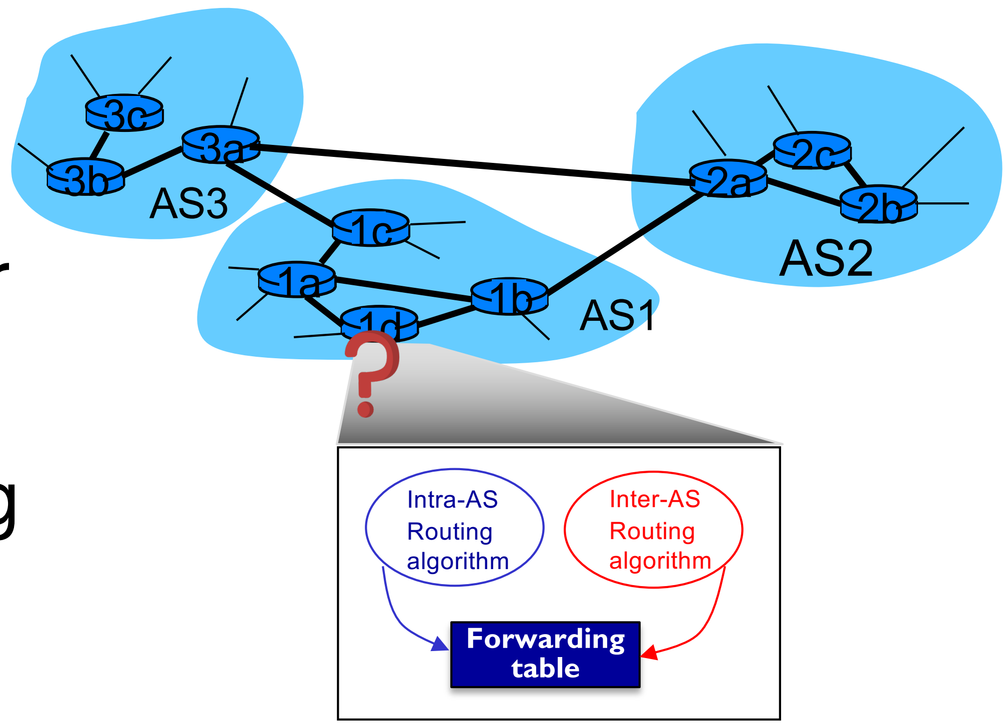
Internet routing: 2-level hierarchy

- Intra-AS (within a campus, within an ISP)
  - *Intra*-Domain Routing protocols: RIP, **OSPF** (*and a few others*)
- Inter-AS (between ISPs, between stub and transit ASes)
  - *Inter*-Domain Routing protocol: **BGP** (*the only one*)



Internet routing: 2-level hierarchy

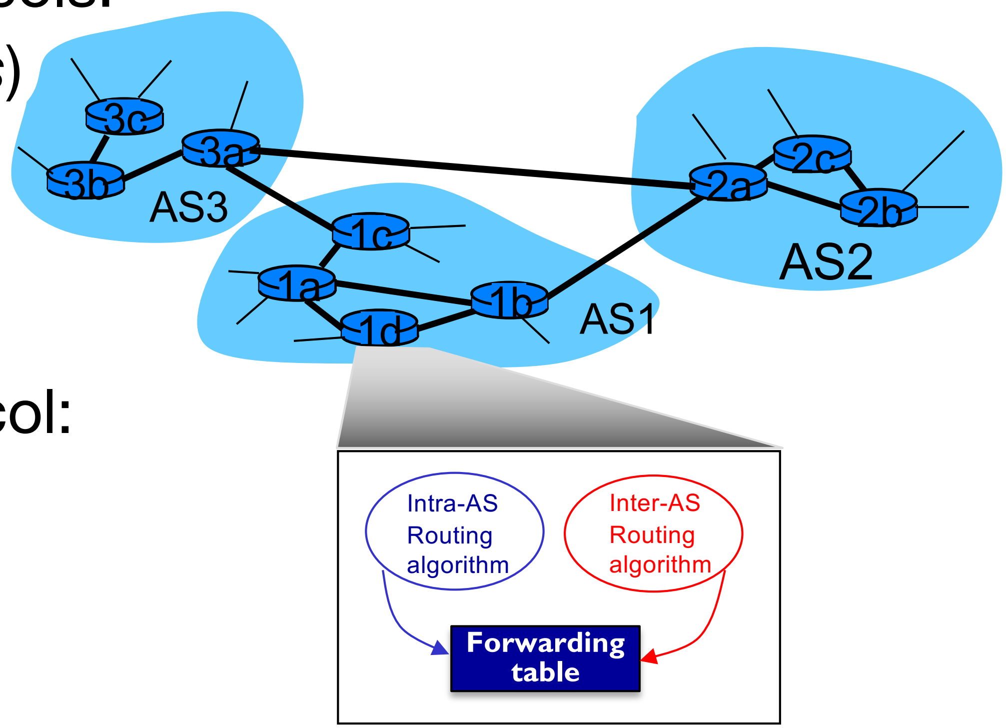
- Suppose router in AS1 receives datagram destined outside of AS1
- Router should forward packet to gateway router in AS1, but which one?
- AS1 inter-domain routing must:
  - learn which destinations reachable through AS2, which through AS3
  - propagate this reachability info to all routers in AS1



Internet routing: 2-level hierarchy

important

- Intra-AS (within a campus, within an ISP)
  - *Intra*-Domain Routing protocols: RIP, **OSPF** (*and a few others*)
- Inter-AS (between ISPs, between stub and transit ASes)
  - *Inter*-Domain Routing protocol: **BGP** (*the only one*)
- intra- and inter-AS routing protocols jointly fill in each router's forwarding table
  - intra-AS sets entries for internal destinations
  - inter-AS & intra-AS sets entries for external destinations



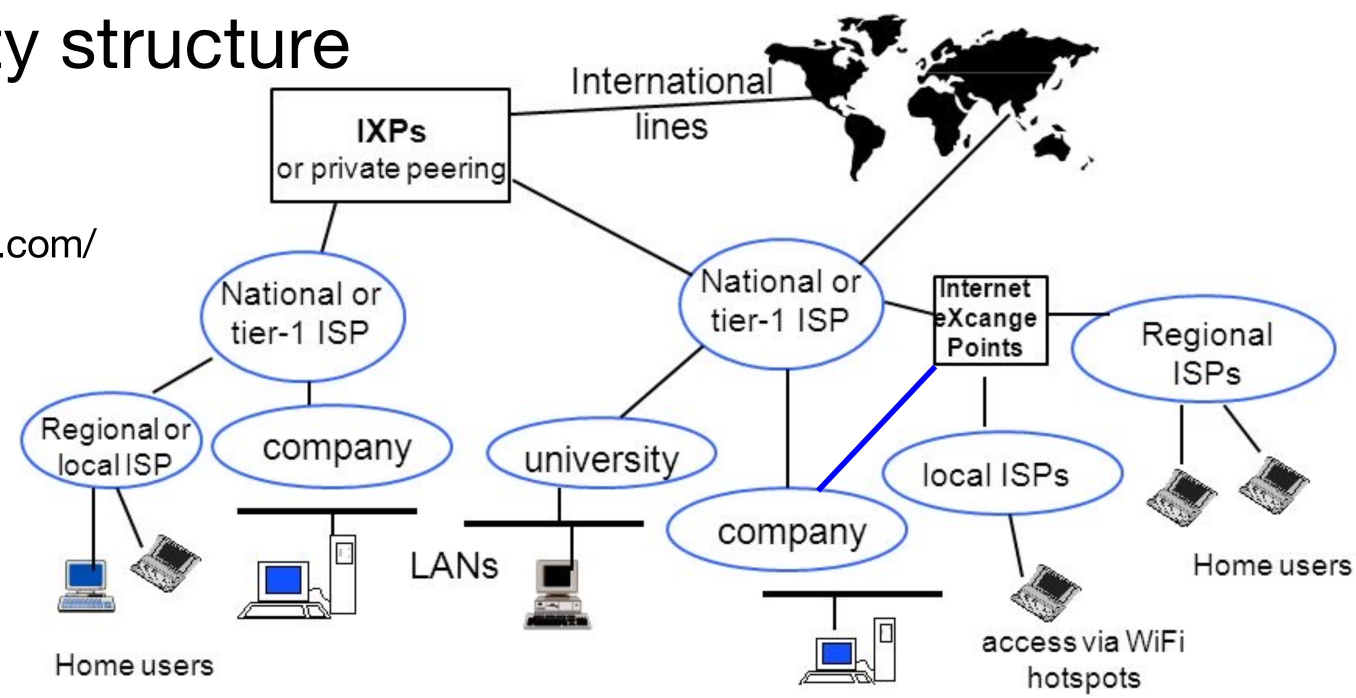
Border Gateway Protocol: BGP
Interior Gateway Protocol: OSPF, etc

Routing protocols in use today

- Open Shortest Path First (OSPF)
  - use link-state computation ⇒ the protocol delivers the complete topology map to all the nodes in the network
- Border Gateway Protocol (BGP)
  - path-vector (≈ distance-vector) ⇒ the protocol delivers one's reachability to (not necessarily all) destinations to direct neighbors

Internet connectivity structure

Internet eXchange Point (IXP)
https://www.internetexchangemap.com/



Why different Intra- and Inter-AS routing ?

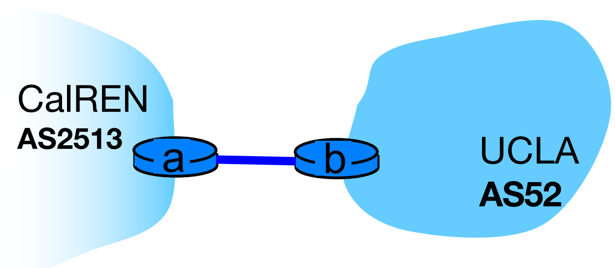
- Policy:
  - Inter-AS: admin wants control over how its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed
- Scale:
  - hierarchical routing saves table size, reduced update traffic
- Performance:
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance

BGP: Border Gateway Protocol (next lecture)

BGP provides each AS a means to:

1. Advertise its own IP address prefixes to the rest of the Internet
2. Obtain IP address prefix reachability info from neighboring ASes
3. Propagate the reachability info to all routers *internal to the AS*.
4. Determine “good” routes to use for learned reachability to destination prefix and policy

- Performing the above 4 tasks
- propagating (partial) prefix reachability info to (some of) the neighbors

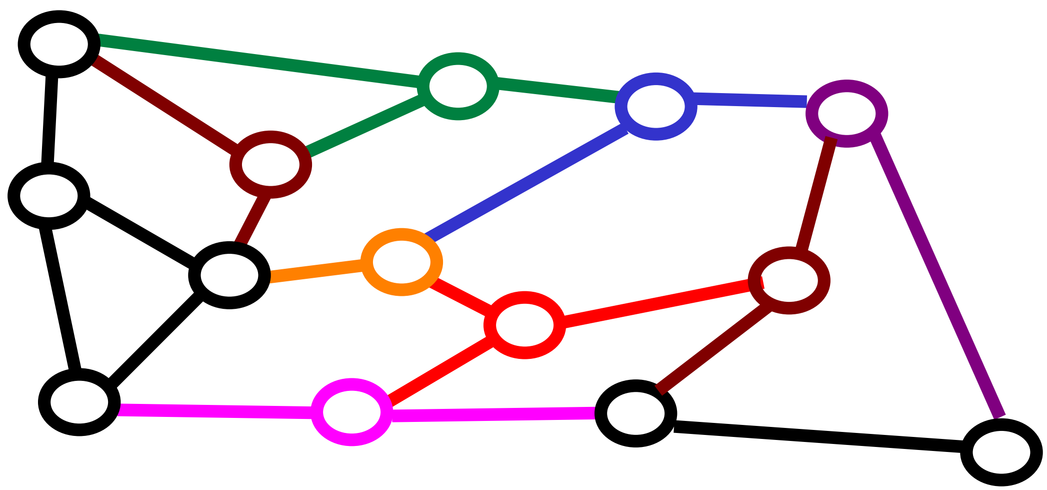


OSPF (Open Shortest Path First)

- Each node knows its directly connected neighbors & the link cost
- Each node sends a *Hello msg* to each neighbor *periodically*
  - monitor link and neighbor nodes status
- Each node broadcasts link-state advertisement (*routing update*) to *the entire network*
  - Periodically, or
  - when the status of any neighbor/link changes
- FYI: OSFP msgs are sent over raw IP packet
  - protocol ID = 89
    - *Implication?*

Building a complete network graph using Link State info from each node

- Every node broadcasts its local piece of the topology graph
- collecting all the pieces from all the nodes, one can put together the complete graph

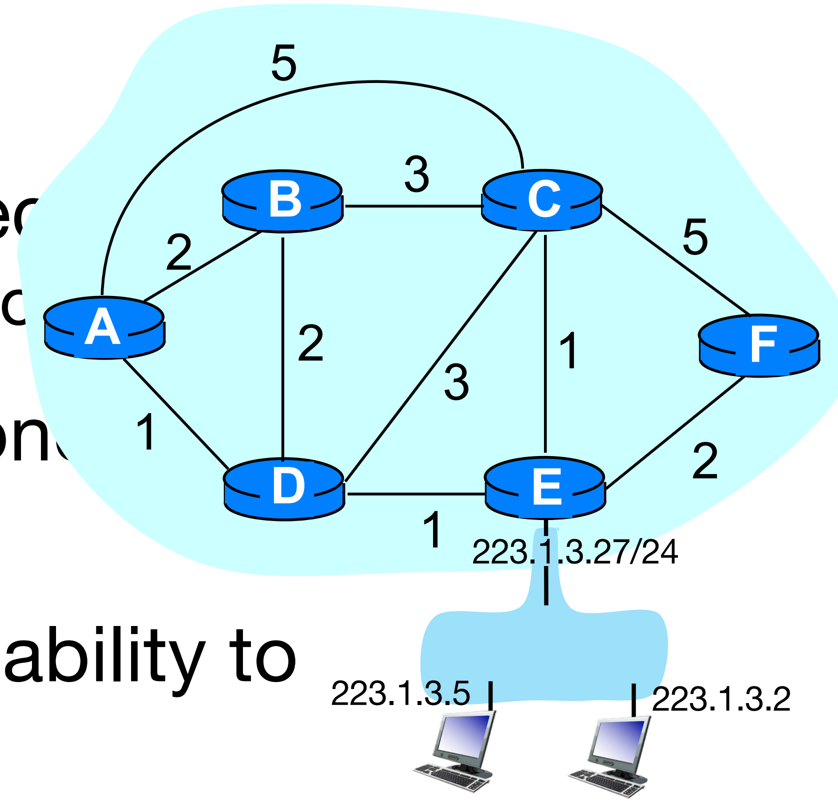


Then each node carries out its <u>own</u> routing calculation *independently*

A side note: exactly what to reach?

(routers versus prefixes)

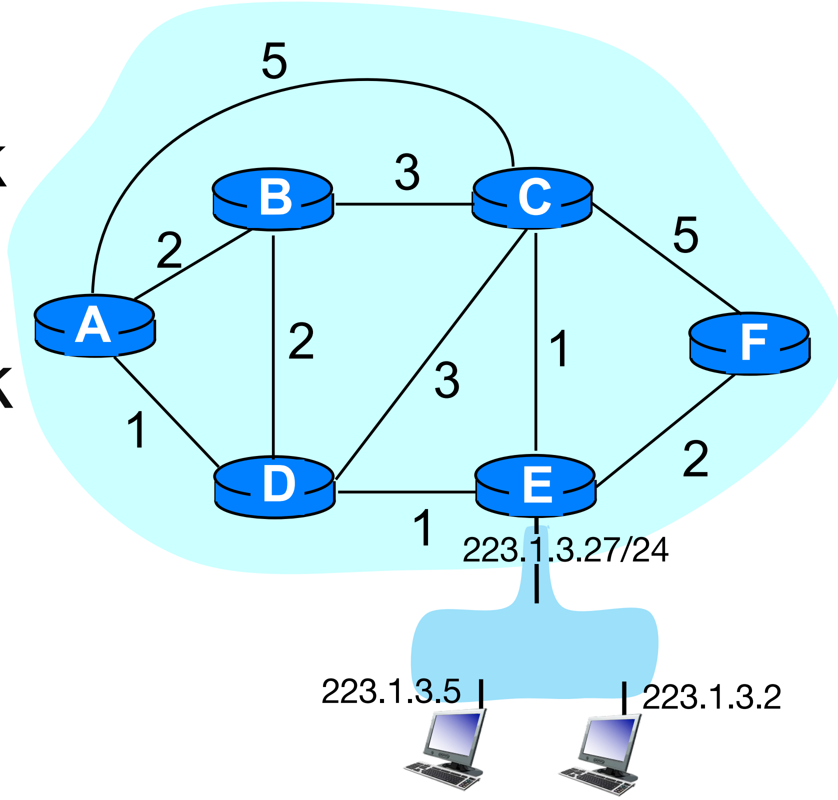
- Each subnet is allocated a spec
  address block (= address prefix, c
- Each subnet is connected to on
  more routers
- Ultimate goal of routing: reachability to all prefixes
- Link-state routing: figure out how to reach the router which can reach the destination (prefix)



Link-State Protocol

Assuming every node knows the network topology graph with link cost

- Each node advertises local link cost to every other node
- Need reliable flooding
- The three basic elements
  - SeqNo, timer, ACK
- Is SeqNo alone enough?
  - Assign a sequence # to each piece of data: ***uniquely identifies individual packet***



Unique ID: Router ID + SeqNo

Link-State Advertisement (LSA)

- ID of the node that created the LSA (LinkState ID)
- Sequence number for this LSA message
- A list of direct neighbors, with link cost to each
  - one entry per neighbor router
- LinkState age: the LSA's lifetime

Approximation to real LSA format (not exact)

Router ID: either manually configured, otherwise default to the highest IP address of the router

<table>
<tr><td>LS age (TTL)</td><td>LS type</td></tr>
<tr><td colspan="2">Link State ID</td></tr>
<tr><td colspan="2">Advertising Router</td></tr>
<tr><td colspan="2">LS sequence number</td></tr>
<tr><td>LS checksum</td><td>length</td></tr>
<tr><td colspan="2">List of direct neighbors,
link cost to each of them</td></tr>
</table>

router ID of this LSA's originating router

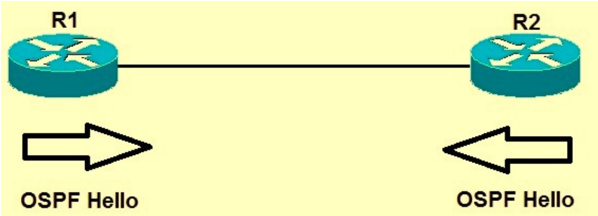
could be the same, depends on LS type

content

Each LSA can be uniquely identified by the combination of [LinkState ID, SeqNo]

How OSPF Works

- When neighboring routers discover each other for the first time: Exchange link-state database

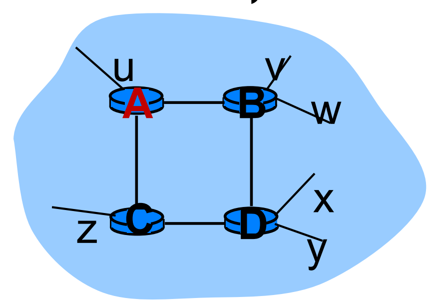


- Link failure detection
  - Neighbor nodes send HELLO msg to each other periodically
    - Default frequency: every 10 seconds
  - No HELLO message for long enough time → **failure detected** → send updated Link State Advertisement
    - Default RouterDeadInterval: 40 seconds
- In the absence of failure: send LSA every 30 minutes

Link-State Routing Daemon

A routing daemon running at each router:

- Send periodic HELLO msgs to neighbors,
- Generates LSA either periodically or event-driven, each carries an <u>increasing sequence #</u>

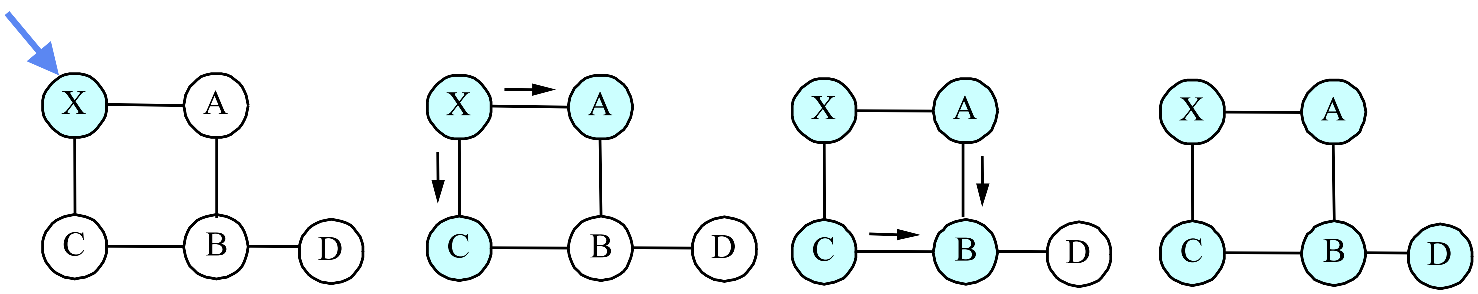


- Upon receiving a *new* LSA, a router ***R***
  - replay it (intact) to all other neighbors
  - process the LSP to update ***R***'s topology graph
  - compute shortest paths
- Each router stores most recent LSA from all others
  - decrement the TTL of stored LSAs
  - discard a LSA when its TTL=0

Reliable Flooding of LSA updates

important

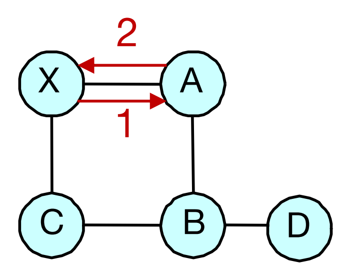
- Each node replays a received new LSA to all neighbor nodes except the one that sent it
- Receive ACK from neighbor, otherwise retransmit the LSA
  - Deliver all LSAs reliably across each hop
  - use the Link-State-ID and SEQ# in a LSA to detect duplicates



Resulting *Directed* Graph

- Link cost can be asymmetric
- Router stores LSAs in LSDB
  - Then derives the directed graph
- Start Dijkstra with the the direct graph
  - Generate FIB
  - If table update, then recompute

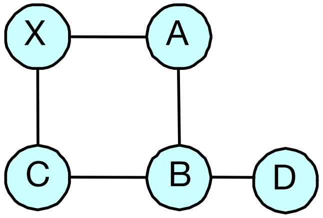
source **sink**	**X**	**A**	**B**	**C**	**D**
X		2		1	
A	1		1		
B		2		1	2
C	1		1		
D			2		



What if router X crashed?

FYI

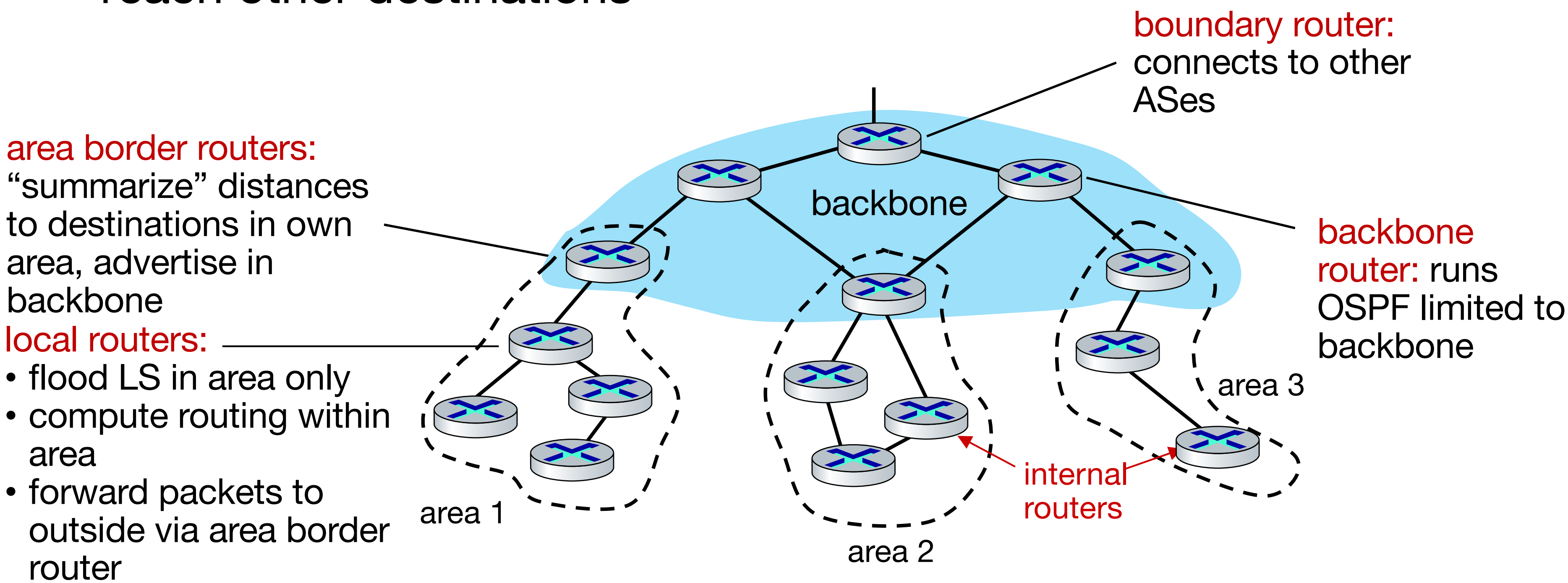
- No Hello msg for a long time (~40s)
  - A and C send new LSA
  - Reliable flooding among [A, B, C, D]
- What would happen if X came online?
  - Hello to discover, reliably flood LSA
  - Wait, what's my last sequence number???
    - LSDB typically on RAM for performance reason
    - Lower SeqNo, stale LSA
    - Same SeqNo, duplicate
- Rebuilding LSDB from neighbors



Q: Why TCP does not need to do this?

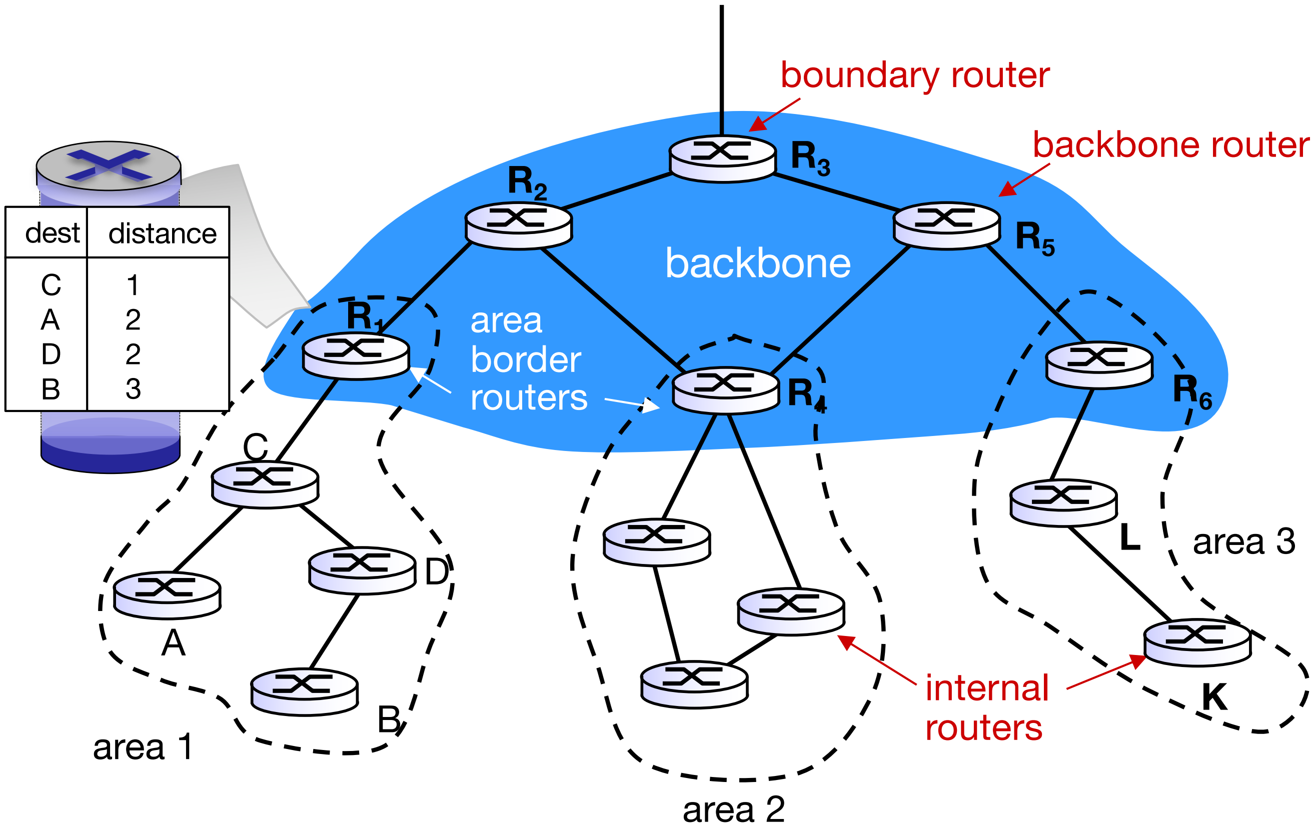
OSPF for a single, big AS domain: Hierarchical OSPF

- Two-level hierarchy: local area, backbone.
  - LSAs flooded only in area, or backbone
  - each node has detailed area topology; only knows direction to reach other destinations



Area summary is also LSA, containing tuples of [netaddr, netmask, cost]

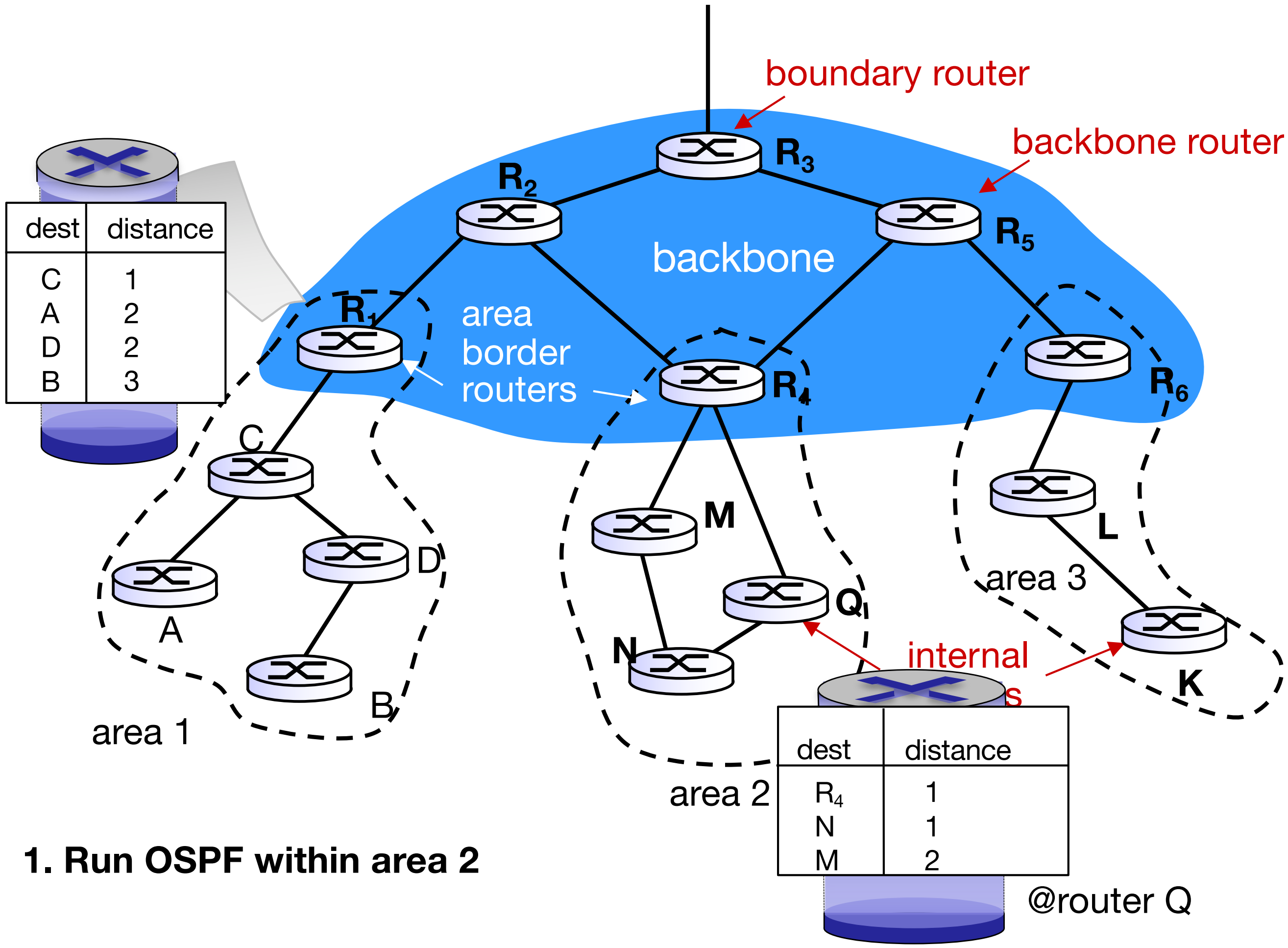
Hierarchical OSPF Example



dest	distance
C	1
A	2
D	2
B	3

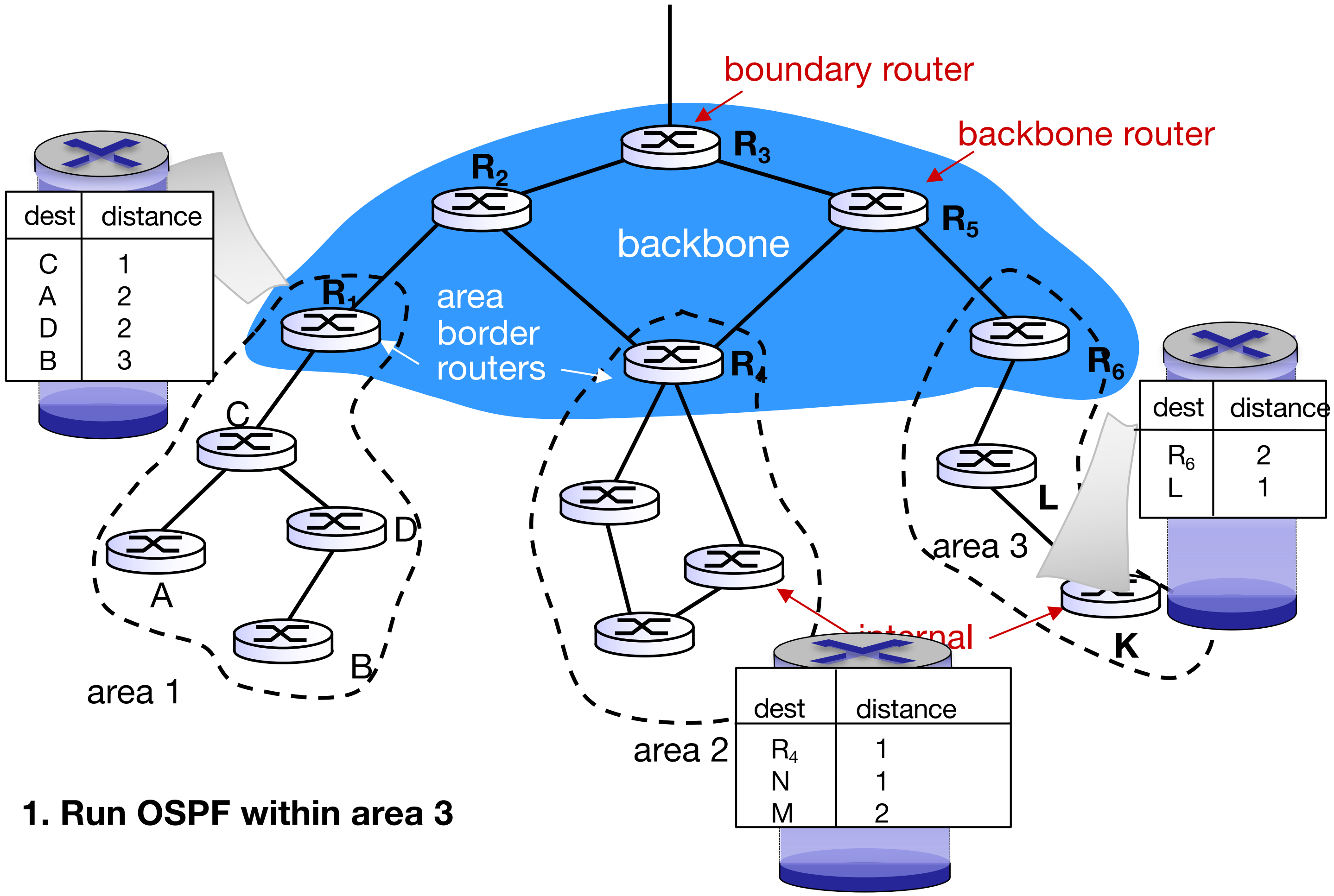
1. Run OSPF within area 1

Hierarchical OSPF Example



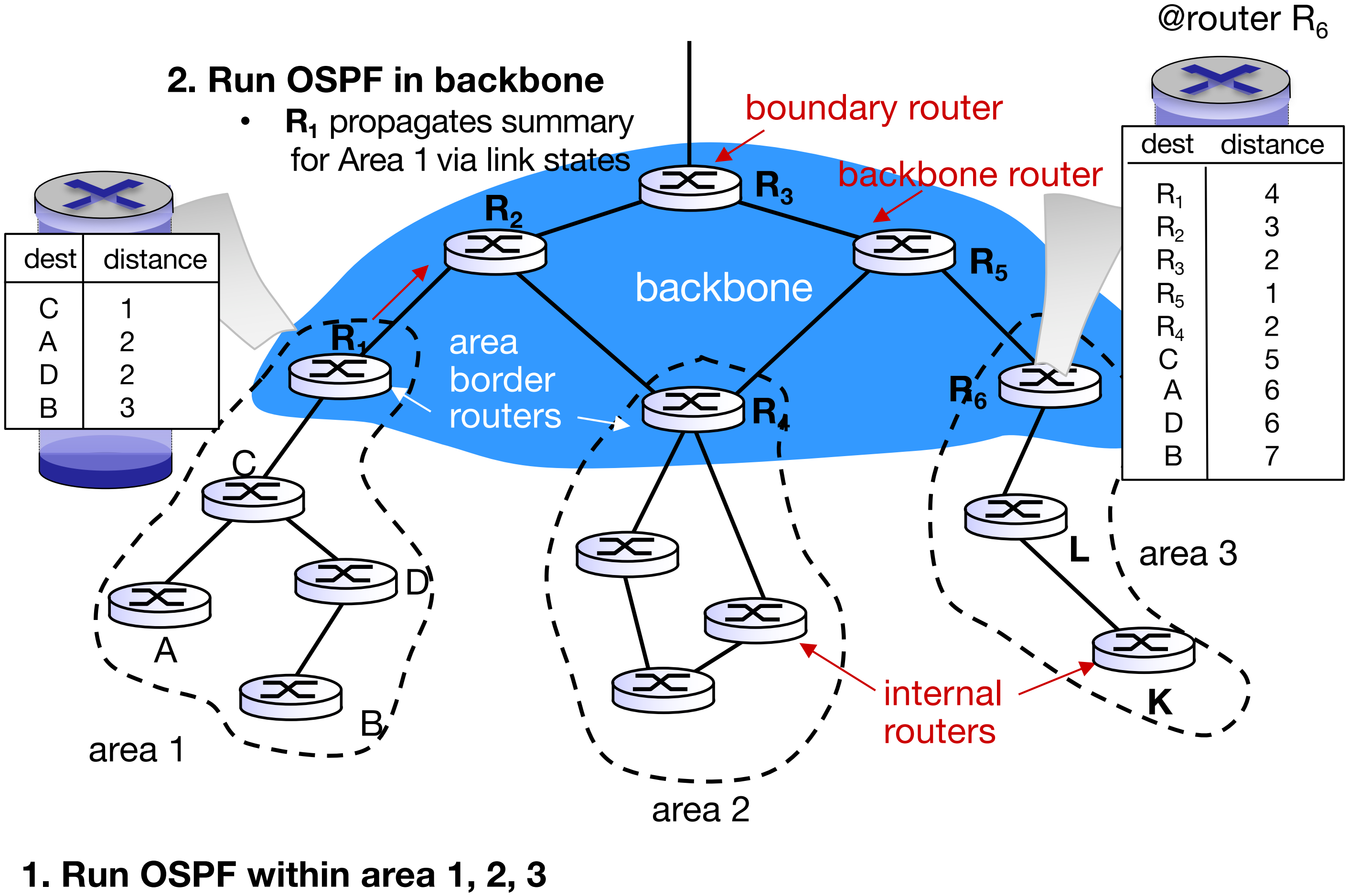
1. Run OSPF within area 2

Hierarchical OSPF Example

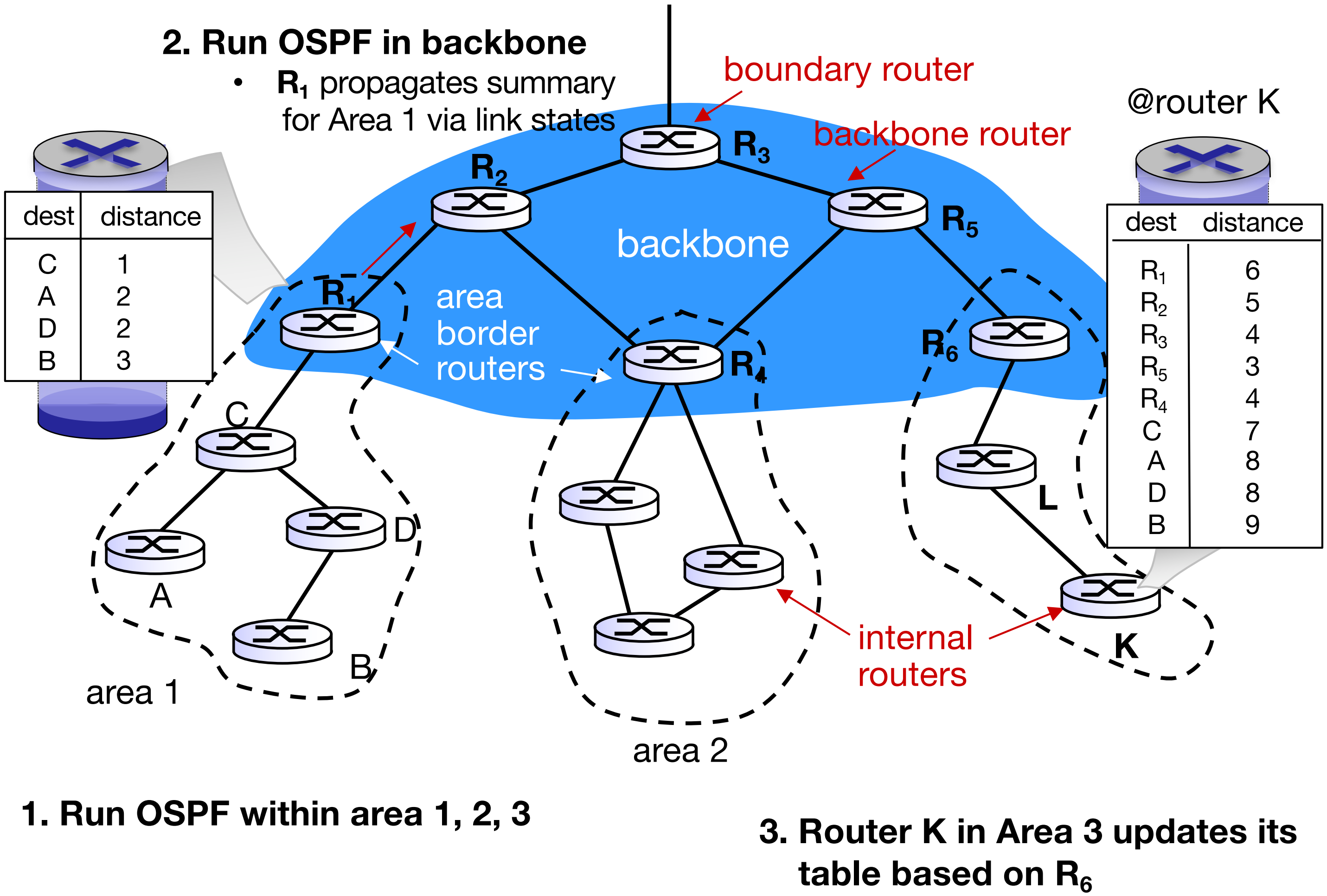


1. Run OSPF within area 3

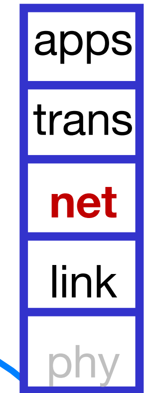
Hierarchical OSPF Example

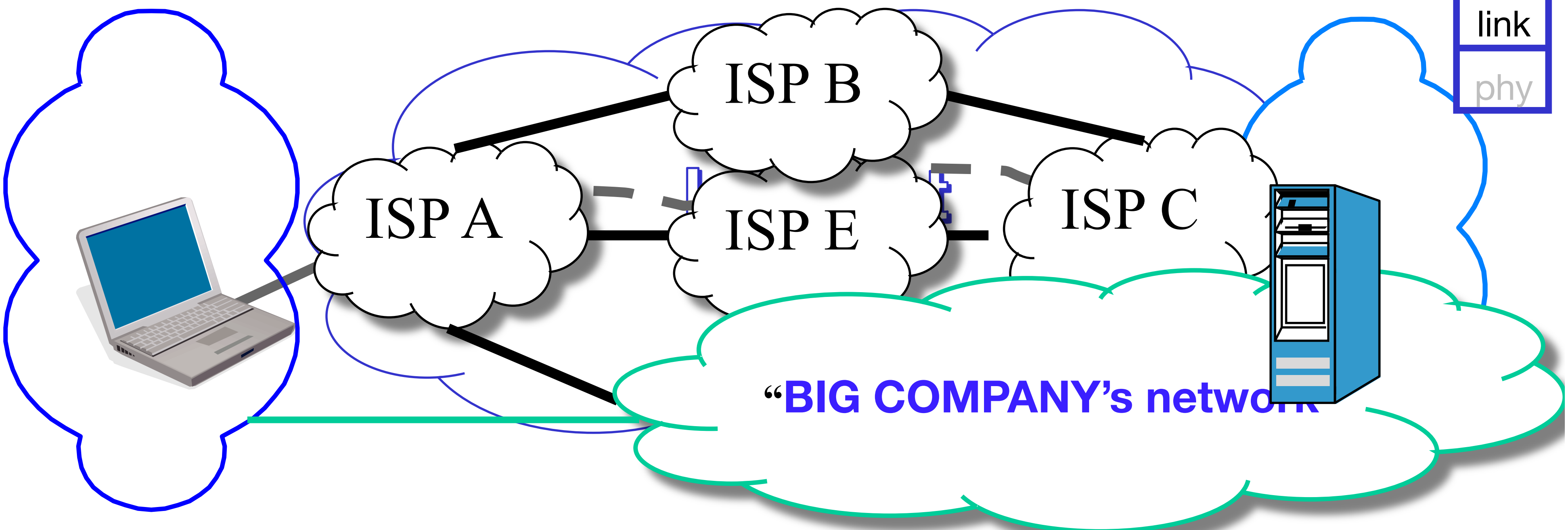


Hierarchical OSPF Example



Big picture: network Layer View





network protocol's job: forward packets to their destination hosts

- A really difficult task: the Internet is large, run by *a very large number of different parties*
  - connection from your laptop to cs118.org website:
    WiFi → campus backbone → local ISP → backbone ISP → Cloudflare

A clarification of terminology: **AS, Institution, Network, DNS Domain**

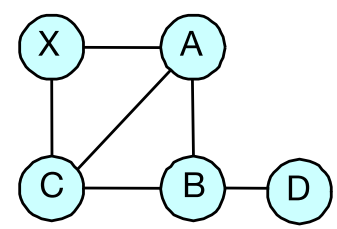
- An AS is not equivalent to a single institution
  - Some institutions own multiple autonomous systems
  - Many institutions do not have their own AS number
- An AS is not equivalent to a block of IP addresses (prefix)
  - Many institutions use multiple (non-contiguous) prefixes
    - UCLA: 131.179/16, 128.97/16, 149.142/16, 164.67/16, 169.232/16
  - Many institutions use a small portion of a larger address block belonging to their ISPs
- An AS is not equivalent to a DNS domain
  - Commonality: an AS or a DNS domain is under a single administrative authority
  - Difference: AS—a unit of topology; DNS domain: independent from network topological connectivity
    - A company can have multiple domain names (att.net, att.com)
    - A DNS domain may not correspond to any AS

Summary

- Two levels of routing: BGP and OSFP
- BGP deals with ASes
  - Announcing self AS prefix out
  - Propagate learned prefixes to internal routers
  - Propagate “appropriate” learned prefixes to neighbor Ases
- OSPF (as IGP) directly runs over IP
  - Periodic Hello msg to discover neighbors
  - Link State Advertisement describing local link states
  - Reliable LSA flooding
    - [Router ID + SeqNo], retransmission timer, ACK
    - Replays LSA to all possible downstreams
  - Hierarchical to reduce LSDB size
    - Focus on local topology

Practice Question 1

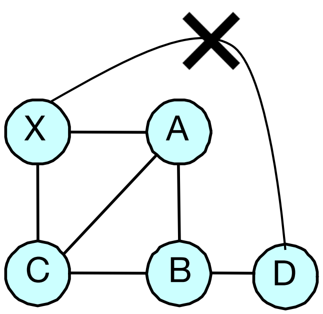
- Assuming X sends a LSA and no packet loss, all links are symmetric and same bandwidth with negligible propagation delay
- Before all LSDBs are synchronized, how many duplicate LSAs will be detected?
- Phase 1: X→ A, C
- Phase 2
  - C→A, B (without knowing A)
  - A→C, B (without knowing C)
  - A, B, C detect the duplicates
  - A or C, depends on who arrives first



- Phase 3: B→D

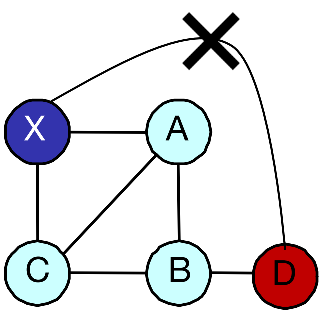
Practice Question 2

- No packet loss, all links are symmetric and same bandwidth with negligible propagation delay
- X-D link is broken
- Before LSDBs are synchronized, how many duplicate LSAs will be detected?



Practice Question 2 (contd.)

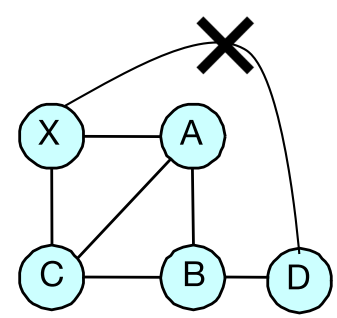
- X, D sends out LSP updates
- Phase 1
  - X→A, C {X, A, C} (the set of flooded node)
  - D→B {D, B}
- Phase 2
  - A→C, B {X, A, B, C}
  - C→A, B {X, A, B, C}
  - A, B, C detect the duplicates
  - A or C, depends on whose replay arrives first
  - B→A, C {A, B, C, D}



- Phase 3
  - A→C, X {X, A, B, C, D}
  - C→A, X {X, A, B, C, D}
  - X, A, C detect the duplicates
  - Same, depends on whose replay arrives first
  - B→D {X, A, B, C, D}

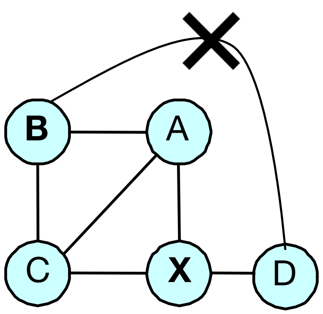
Practice Question 3

- No packet loss, all links are symmetric with equal cost, and same bandwidth with negligible propagation delay
- X-D link is broken
- After LSDBs are synchronized, whose routing table has changed? (Breaking the tie by ASCII value (A<B))
- X, D changed routing table



Practice Question 4

- No packet loss, all links are symmetric with equal cost, and same bandwidth with negligible propagation delay
- B-D link is broken
- After LSDBs are synchronized, whose routing table has changed? (Breaking the tie by ASCII value (A<B))
- B and D for sure
- A, C also change their routing tables
  - A→B, A→C, ***A→X→D***, A→X
  - C→A, C→B, ***C→X→D***, C→X
  - X→A, X→A→B, X→C, X→D



Q: Will A and C notify others about the table change?

Practice Question 5

- No packet loss, all links are symmetric with equal cost, and same bandwidth with negligible propagation delay
- B-D link is broken
- OSPF and RIP (FYI, a DV-based routing protocol), which sends more routing updates?
- OSPF: 2 LSAs reliably flooded
- RIP: who has changed routing table?
  - A, B, C, D