



# The Global Internet

\* Some slides are adapted from *Computer Networking: A Top-Down Approach*. J.F Kurose and K.W. Ross. All Rights Reserved.

Spring 2026

© CS 438 Staff, University of Illinois

1

# [ Learning Objectives ]

- Autonomous Systems
- Intradomain Routing
- Interdomain Routing
  - *BGP*



# [ Recap: Routing ]

- Provides paths between networks
- We know several designs already
  - link-state
  - distance vector
- But previous lectures assumed:
  - all routers have same routing metric
    - shortest (least-cost) path
  - no policy issues, no privacy issues

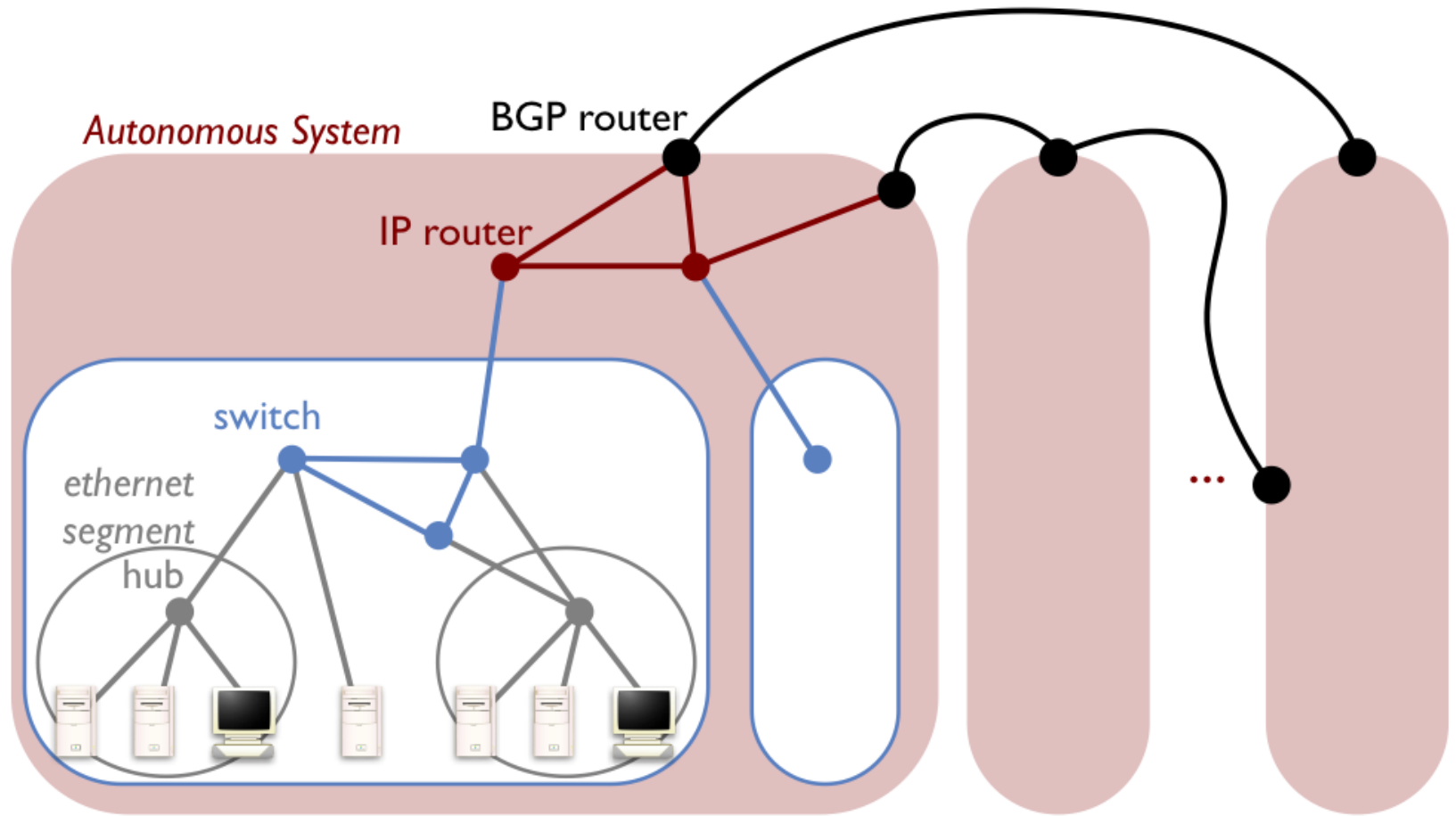


# [ Internet is more complicated..... ]

- Internet not just unstructured collection of networks
- Internet is comprised of a set of “autonomous systems” (ASes)
  - Independently run networks, some are commercial ISPs
  - Autonomy of control, e.g., companies and universities
  - Around 120,000 ASes (as of 2025)
- Enables **hierarchical** aggregation of routing information
- ASes are sometimes called “domains”
  - hence “intradomain” / “interdomain” routing



# [ Big Picture of the Internet ]



# [ Autonomous Systems ]

- Intradomain Routing (within an AS)
  - Performed using domain-specific algorithm
  - Selected by domain administrators
  - Allows heterogeneous *interior gateway protocols (IGP)*
- Interdomain Routing (between ASes)
  - Performed using standard global algorithm
  - Homogeneous *exterior gateway protocol (EGP)*
  - Main goal: reachability



# [ Intradomain Routing ]

- Routing Information Protocol (RIP)
  - From the early Internet
  - Part of the 1982 BSD Unix
  - Distance vector algorithm (Bellman-Ford)
  - Based on hop count (infinity set to 16 hops)
  - Obsolete (“Rest In Peace”)
- Intermediate System to Intermediate System (IS-IS)
  - Published in the 1990s
  - ISO Standard (ISO 10589)
  - Link state algorithm (Dijkstra’s)
  - Widely used within ISP backbones



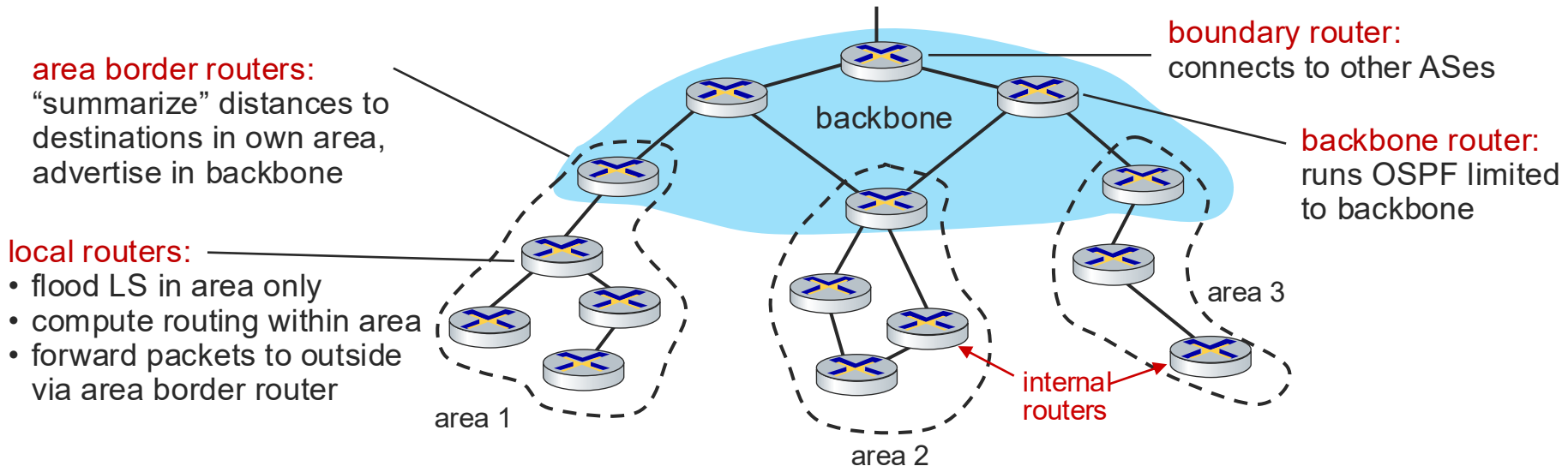
# Intradomain Routing

- Open Shortest Path First (OSPF)
  - Designed/published originally in the 1980s/1990s
  - Internet Standard (RFC 2328)
  - Link state algorithm (Dijkstra's)
    - Each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
    - Multiple link costs metrics possible: bandwidth, delay
    - Each router has full topology, uses Dijkstra's algorithm to compute forwarding table
  - Security
    - All OSPF messages authenticated (to prevent malicious intrusion)
  - Widely used in enterprise networks



# OSPF is hierarchical

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



# [ Intradomain Routing ]

- IS-IS vs. OSPF

	<b>IS-IS</b>	<b>OSPF</b>
Hierarchy	Flexible “L1”/“L2” hierarchy	Strict “Area 0” backbone
Scaling	Scales naturally	Scales well enough
Adoption	More complex initially, preferred by large ISPs	Simpler, IP-native, preferred by enterprises



# [ Inter-domain Routing ]

- General aspects
  - Very complex and difficult
  - Focuses on reachability rather than optimality
  - Must be loop free
  - Specify how reachability information should be exchanged
- Border Gateway Protocol (BGP)
  - Why does BGP exist?
  - Why does BGP look the way it does?
  - How does BGP work?
  - *Pay more attention to the “why” than the “how”*



# The “A” in AS really means Autonomous

- Want to choose their own internal routing protocol
  - Different algorithms and metrics
- Want freedom to route based on **policy**
  - “My traffic can’t be carried over my competitor’s network”
  - “I don’t want to carry transit traffic through my network”
  - Not expressible as Internet-wide “shortest path”!
- Want to keep their connections and policies **private**
  - Would reveal business relationships, network structure



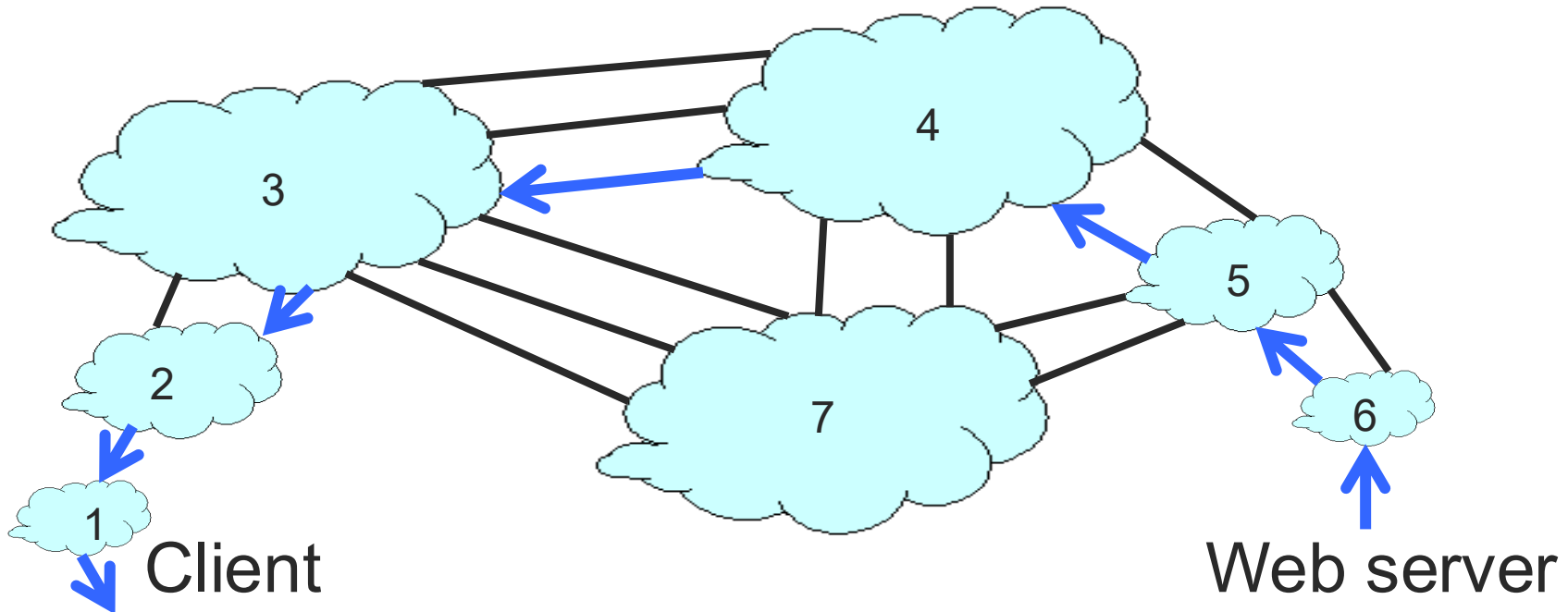
# [ ASes are Businesses ]

- Three kinds of common relationships between ASes
  - Customer, Provider, Peer
- Business implications
  - Customer pays provider, peers don't pay each other
- Policy implications
  - “When sending traffic, I prefer to route through customers over peers, and peers over providers”
  - “I don't carry traffic from one provider to another provider”



# [ AS-Level Topology ]

- Destinations are IP prefixes (e.g., 12.0.0.0/8)
- Nodes are ASes
- Links are connections & business relationships



# [ Autonomous Systems ]

## ■ Challenges

- Scale
  - Border router must be able to forward any packet destined anywhere on the Internet
- Autonomous routing in ASes
  - Impossible to calculate meaningful costs for paths that cross multiple ASes
- Trust
  - One AS must trust the advertised routes of other ASes

## ■ Goal

- Specify policies that lead to “good” paths (even if they are not optimal)



# [ Routing Choices ]

- Key issues are *policy* and *privacy*
- Challenges
  - No universal metric
  - AS-specific policy decisions
- Problems with link state
  - Metric used by routers not the same
    - Can't use shortest path
  - LS database too large — entire Internet
  - Flooding may expose internal topology and policies to other ASes



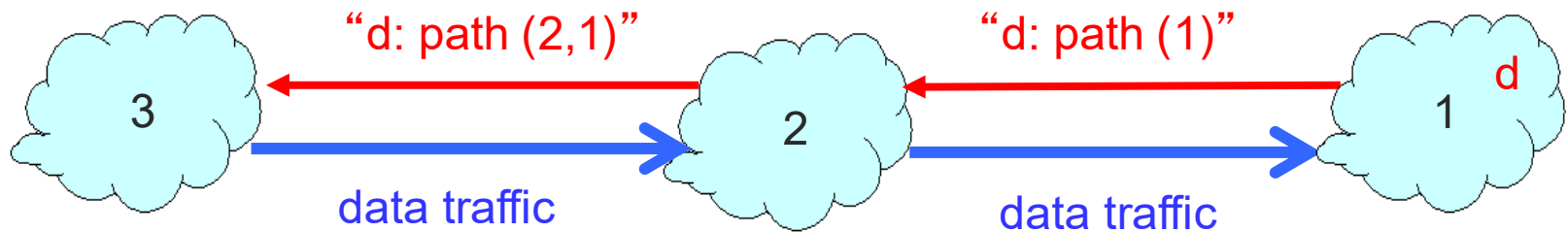
# [ Routing Choices ]

- Key issues are *policy* and *privacy*
- Challenges
  - No universal metric
  - AS-specific policy decisions
- Problems with distance-vector
  - Does not reveal any connectivity information
    - Susceptible to loops
  - Still uses shortest path
  - Slow to converge



# Solution: Path Vector Routing

- Extension of distance-vector routing
  - Support flexible routing policies
  - Faster loop detection (no count-to-infinity)
- Key idea: advertise the entire path
  - Path vector sends the entire path for each **dest d**
    - In contrast, distance vector sends distance metric per **dest d**



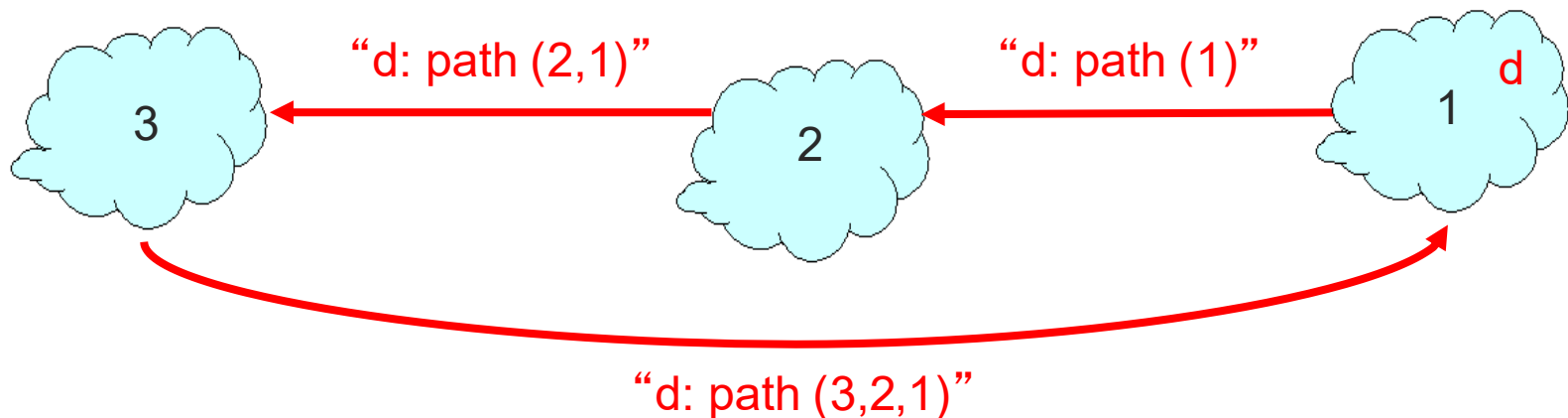
# [ Path Vectors ]

- Each routing update carries the entire path
- Loop detection
  - When AS gets route check if AS already in path
    - If yes, reject route
    - If no, add self and (possibly) advertise route further
- Advantage
  - Metrics are local
  - AS chooses path, protocol ensures no loops



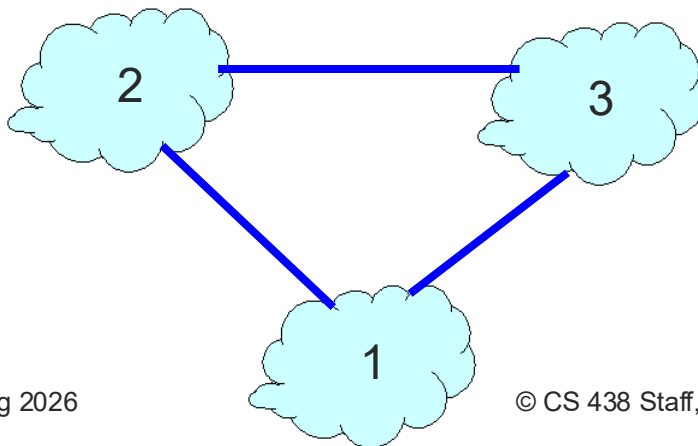
# [ Loop Detection ]

- Node can easily detect a loop
  - Look for its own node identifier in the path
  - e.g., node 1 sees itself in the path “3, 2, 1”
- Node can simply discard paths with loops
  - e.g., node 1 simply discards the advertisement



# Flexible Policies

- Each node can apply local policies
  - Path selection: Which path to use?
  - Path export: Which paths to advertise?
- Examples
  - Node 2 may prefer the path “2, 3, 1” over “2, 1”
  - Node 1 may not let node 3 hear the path “1, 2”



# [ BGP So Far ]

- Simple! Path-vector policy-based routing.
  - Class dismissed?
- Well, only conceptually...
- How does BGP work *exactly*?
- How does it work together with IGP (e.g., OSPF)?

