Course Overview

CS 436: Spring 2018

Matthew Caesar

http://www.cs.illinois.edu/~caesar/cs436/

Motivating Scenario

- You are a VP Networking for fortune 500 company
- A data breach of critical assets occurs
- What do you do?

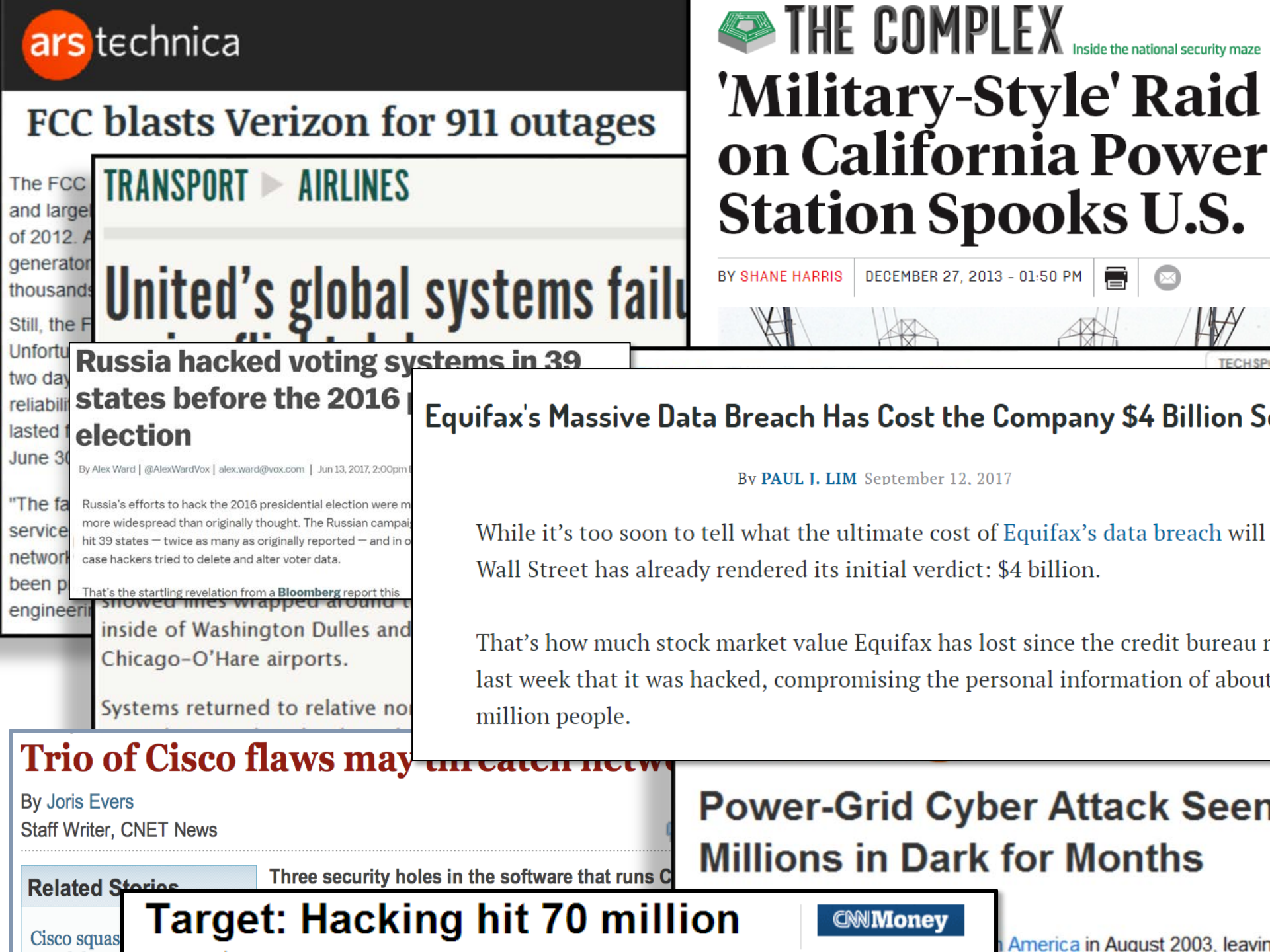
Example: Designing an Internet Service

- Imagine you are running Amazon, Yahoo, or Google.
- How do you design your software to handle massive load?
- Tens of thousands of clicks *a second*
- Extremely bursty demand
- Service must be up and running 24/7 with no (discernible) outages
- This is a hard problem!

Suppose someone asks you to… (real examples)

- "The latest OS software update caused half of our CDN to go down! We are losing millions of dollars of contracts by the hour! How can we get things back up as fast as possible?"
- "Customers of our North American ISP backbone are noticing random outages – diagnose and fix what's wrong"
- "Here's a bunch of routers and ethernet cable. Can you put together a network for us?"
- "Our web service is being hit with a DoS attack – defend us"
- What would you do?

This stuff is harder than it looks


ars technica
FCC blasts Verizon for 911 outages
The FCC
and large
of 2012. A
generator
thousands
Still, the F
Unfortu
two day
reliabili
lasted
June 3
"The fa
service
networ
been p
engineeri
TRANSPORT ▶ AIRLINES
United's global systems failu
inside of Washington Dulles and
Chicago-O'Hare airports.
Systems returned to relative no
THE COMPLEX
Inside the national security maze
'Military-Style' Raid on California Power Station Spooks U.S.
BY SHANE HARRIS
DECEMBER 27, 2013 - 01:50 PM
Russia hacked voting systems in 39 states before the 2016 election
By Alex Ward | @AlexWardVox | alex.ward@vox.com | Jun 13, 2017, 2:00pm
Russia's efforts to hack the 2016 presidential election were m
more widespread than originally thought. The Russian campai
hit 39 states — twice as many as originally reported — and in o
case hackers tried to delete and alter voter data.
That's the startling revelation from a Bloomberg report this
Equifax's Massive Data Breach Has Cost the Company $4 Billion S
By PAUL J. LIM September 12, 2017
While it's too soon to tell what the ultimate cost of Equifax's data breach will
Wall Street has already rendered its initial verdict: $4 billion.
That's how much stock market value Equifax has lost since the credit bureau
last week that it was hacked, compromising the personal information of about
million people.
Trio of Cisco flaws may
By Joris Evers
Staff Writer, CNET News
Related Stories
Three security holes in the software that runs C
Cisco squas
Power-Grid Cyber Attack Seen
Millions in Dark for Months
America in August 2003, leavin
Target: Hacking hit 70 million
CNNMoney

Theory vs. Practice

- It's important to understand the theory
  - Protocols, algorithms, techniques behind systems
- But it's also important to know how to apply the theory
  - Where the real challenges are
  - Best practices and rules of thumb
  - Build breadth and see the "big picture"
- This class covers both, together
  - Learning practice along with theory makes learning the theory more fun

Why a “lab” class?

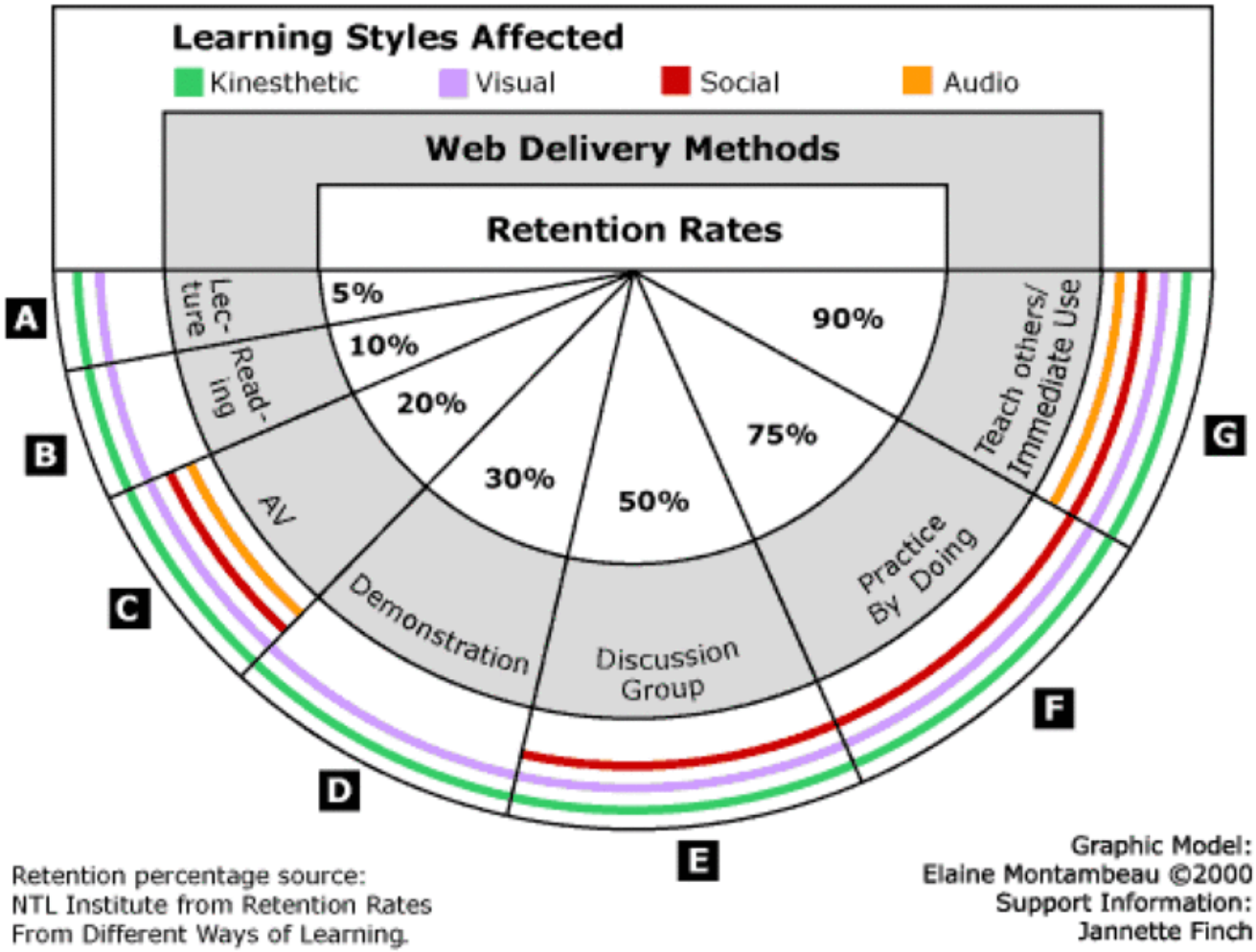
- Hands-on approach enables direct observation and experimentation
  - Builds deeper understanding of system design
  - Also makes learning the fundamentals more fun

- In this class you’ll work directly with real systems
  - We have an awesome lab environment for you
  - You’ll have direct access to code/systems widely used in many real-world production environments

So, how are we going to do this?

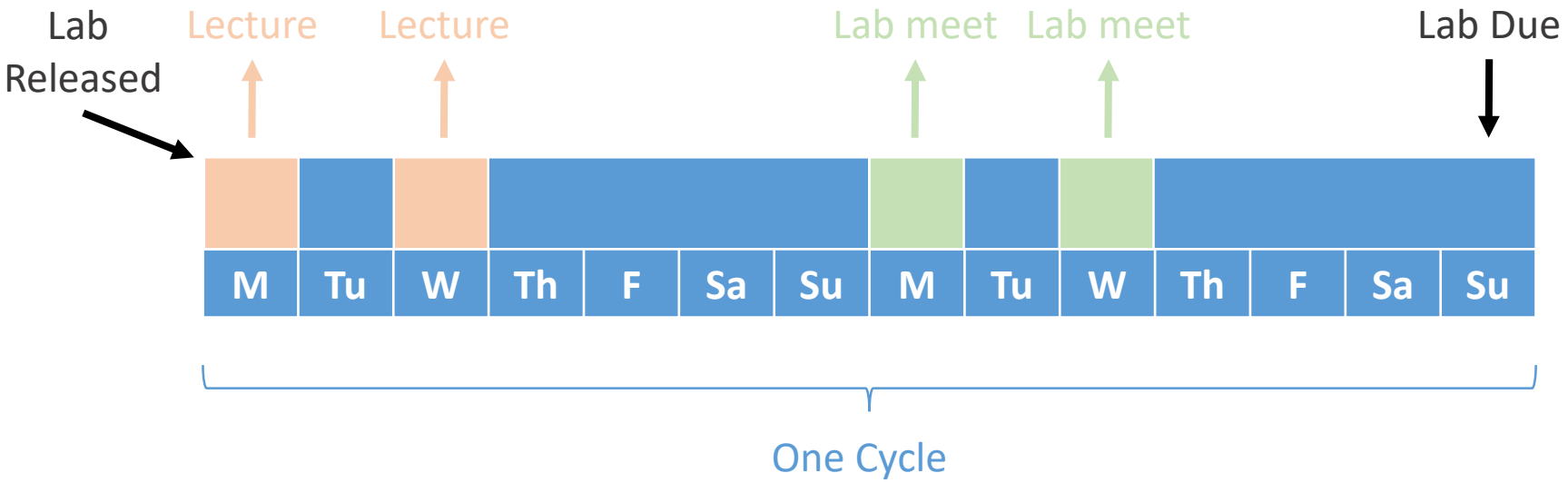
- Course is organized into a set of “cycles”
  - Each cycle composed of two lectures and two lab sessions
- Complete overview of modern network infrastructure
  - ISP and enterprise networking, data centers, clouds, streaming and online services, etc.
- Lab sessions are *not* MPs
  - Focus on education: goal is to “guide” you as much as possible, to walk you through the system’s design
  - Focus on interactivity: instructor and TAs will be present at the lab sessions to guide you
  - Focus on experimentation: we want you to explore–go off and have fun!

How do humans learn?



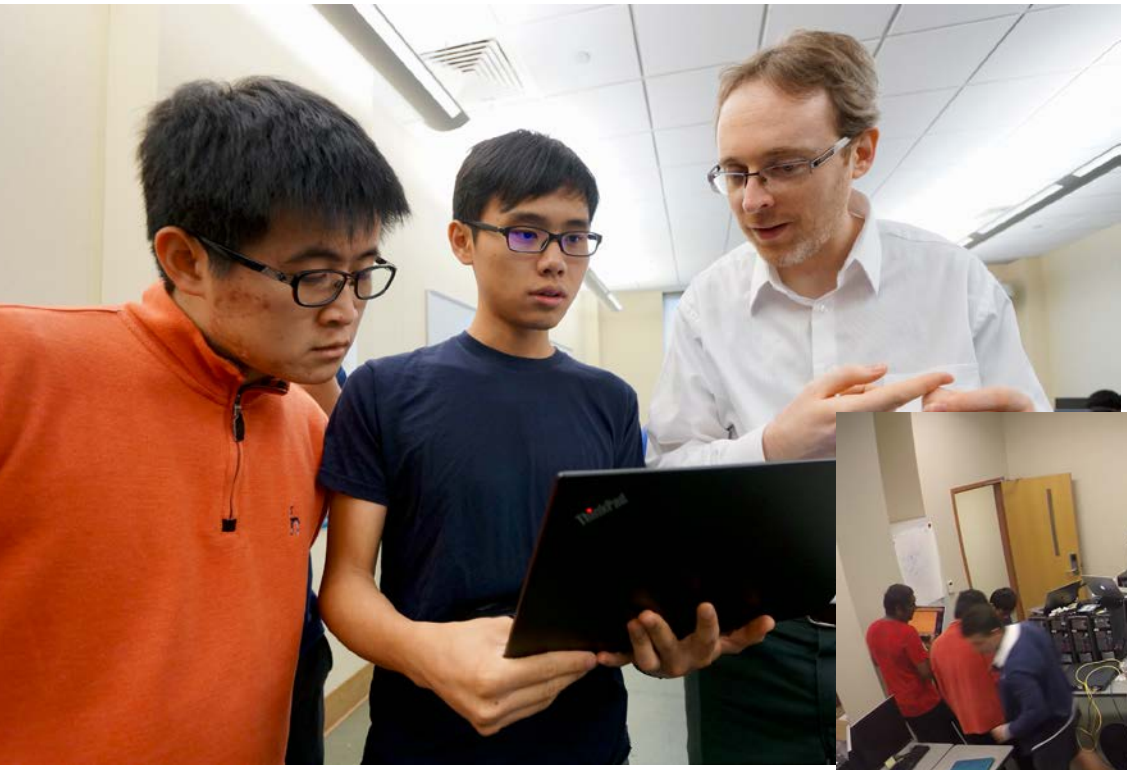
Cycle timeline

- There will be seven cycles, each comprising 2 weeks
- Lab released at start of cycle, due at end of cycle
- Labs designed to be started at beginning of cycle
  - Work on them throughout the cycle, not just lab days

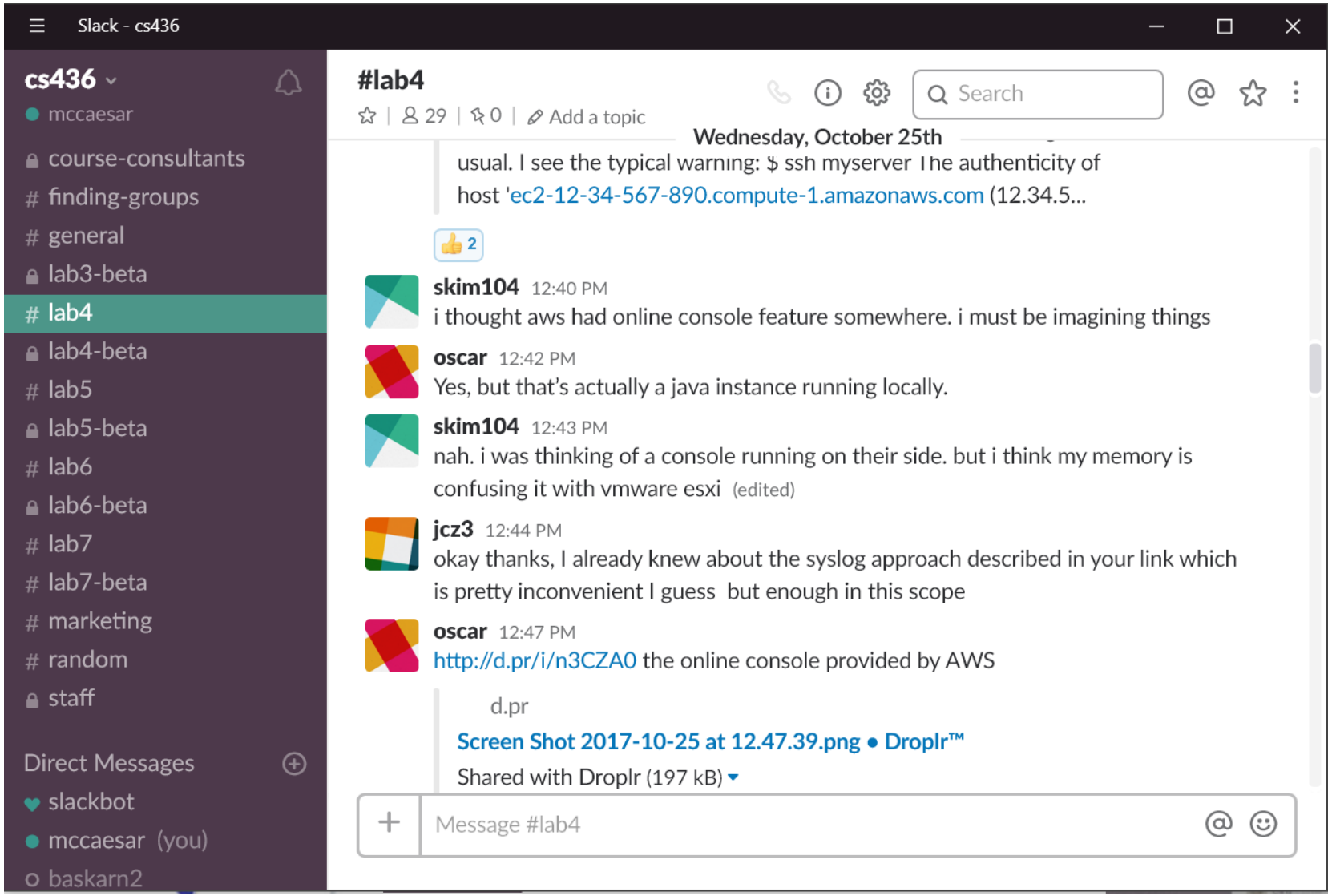


Real-Time Communication: Physical Labs

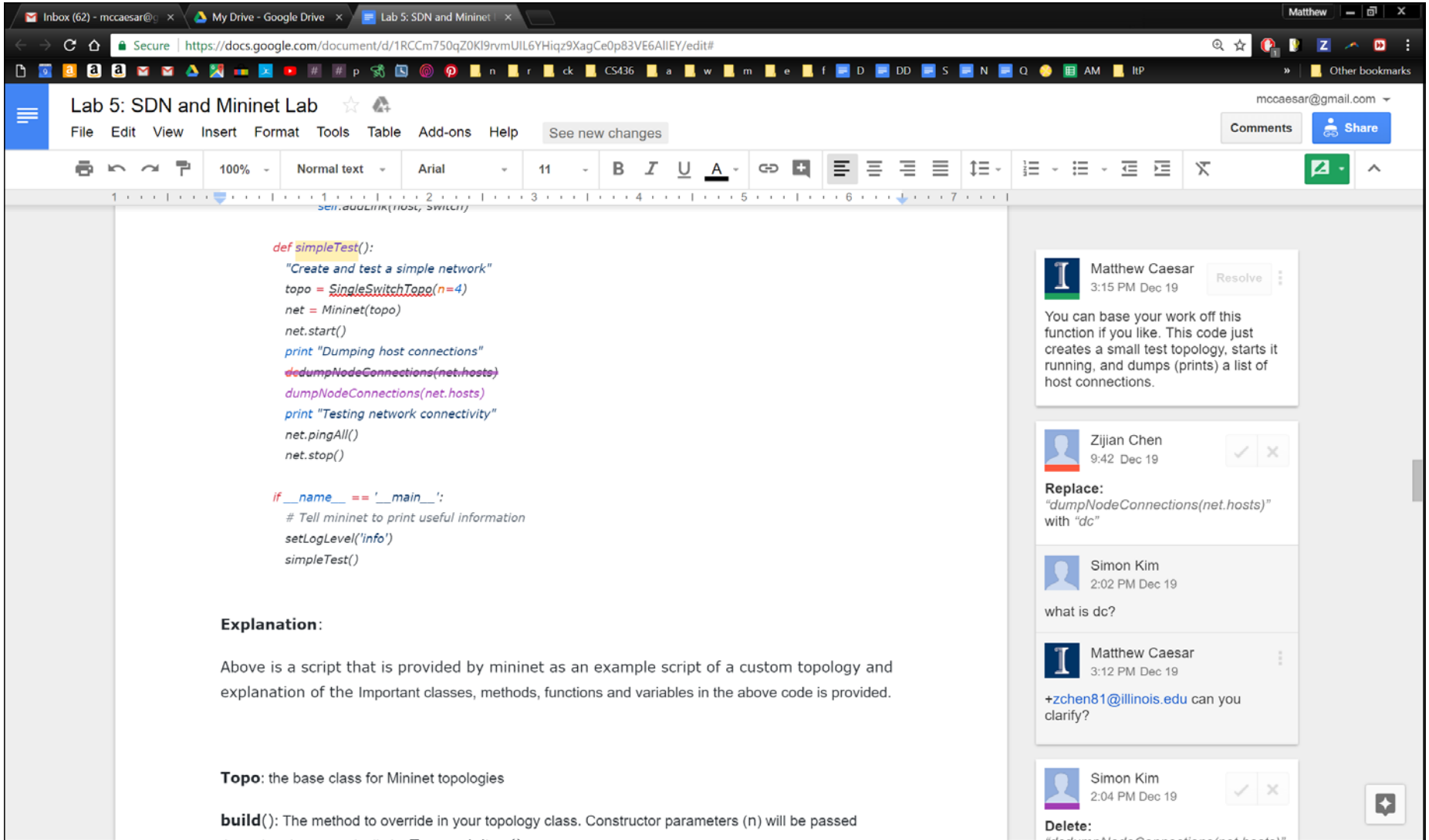
- Work with real things
- Talk to me/each other
- Interactive lecturing



Real-Time Communication: *Slack*



Real-Time Communication: *Living Labs*



Cycle	Topic	Description
1.	Internet and ISP Networking	Routing protocols, ISP systems, how to run your own ISP
2.	Enterprise Networking	Network defense, troubleshooting, and design
3.	Equipment and Devices	Router, switch, server, and network architectures
4.	Cloud Networking	Amazon Web Services, Cloud and service infrastructure, network programming
5.	Emerging Tech: SDN and NFV	Software-defined networking, network virtualization
6.	OS Network Stacks	Linux kernel internals, network stacks, service-customized OS
7.	Devops and Deployment	Network management and configuration, common network protocols and practices

About me: Matthew Caesar (Instructor)

- Expertise: networking, systems
- Faculty in CS department
  - PhD from UC Berkeley in 2007
- CSO at Veriflow Systems
- Industrial experience at AT&T Labs, Microsoft Research, HP, Nokia DSL; helped found a CDN company; ongoing partnerships/tech transfer with Cisco, DARPA, NSA, Boeing
- I like designing/building/deploying large-scale software systems that are grounded in strong theoretical principles
- Office: 3118 SC
- Email: caesar@Illinois.edu

This class

- Teaches networking/systems design through a hands-on approach
  - OS networking kernels, cloud services, firewall/router configuration, protocol stacks
- Uses modern implementations commonly used in industry
  - Cisco IOS, Linux networking stack, Amazon web services, etc.
- Leverages real-world scenarios commonly encountered in industry
  - Building networks, deploying cloud services, blocking an attack, implementing a new protocol

Grading Policy

7 Lab Assignments	75%
Course Excellence	25%

- Goal: I want you to learn
  - Grade measures completion of tasks and also how much you learned
  - If you know a lab already, talk to me

Examples of “Excellence”

- Take on a promotion
- Help improve a future lab
- Work through a lab early
- Research project
- Complete a lab and then go “a step beyond”
- Ask questions, pose ideas, answer questions

- I am looking for things that help you and help others
- I will suggest examples on slack throughout the semester

Administrative Details

Prerequisites

- Networking / Operating Systems Concepts
  - CS 241 or equivalent

- Programming
  - Shell scripting (sed/bash)
  - Java/Python/C++

Homework and Projects

- Write a report proving you did the lab
  - All code, answers to questions
  - Organize it to make it readable
- Email to the TA
- Late policy: 2% off per hour late

Academic Honesty

- Your work in this class must be your own.
- All infractions reported to the department
- If students are found to have collaborated excessively or to have cheated (e.g., by copying or sharing answers during an examination or sharing code for the project), all involved will at a minimum receive grades of 0 for the first infraction.
  - We will run a similarity-checking system on code and binaries
- Further infractions will result in failure in the course and/or recommendation for dismissal from the university.

CS 436: 

- Want an added challenge?

- Take on an additional CHALLENGE PROJECT
  - Get four units instead of three
  - Open to both grad and undergrad

- Come talk to me if you are interested in this option

Class Communications

- Web site: http://www.cs.illinois.edu/~caesar/cs436
  - Assignments, lecture slides, announcements
- Email list: cs-436@lists.illinois.edu
- Slack group: https://cs-436.slack.com
- Please cc our staff on any non-private emails sent directly to me (caesar@illinois.edu)

Emergency Preparedness

- Learn different ways to leave building
- Severe weather – get to low-level in middle of building
- Active shooter – run > hide > fight
  - Lock/barricade door
- Sign up for emergency text messages at emergency.illinois.edu
- More info: police.illinois.edu/safe

Any questions?

Cycle 1:
Internet and ISP Networking

CS 436: Spring 2018

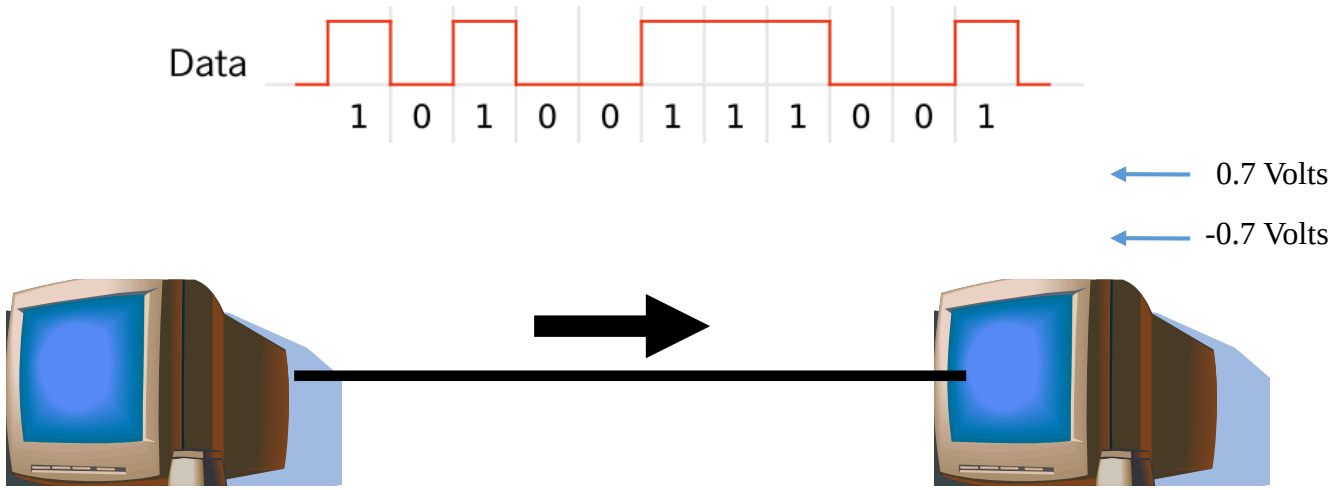
Matthew Caesar

http://www.cs.Illinois.edu/~caesar/cs436

Motivating Questions

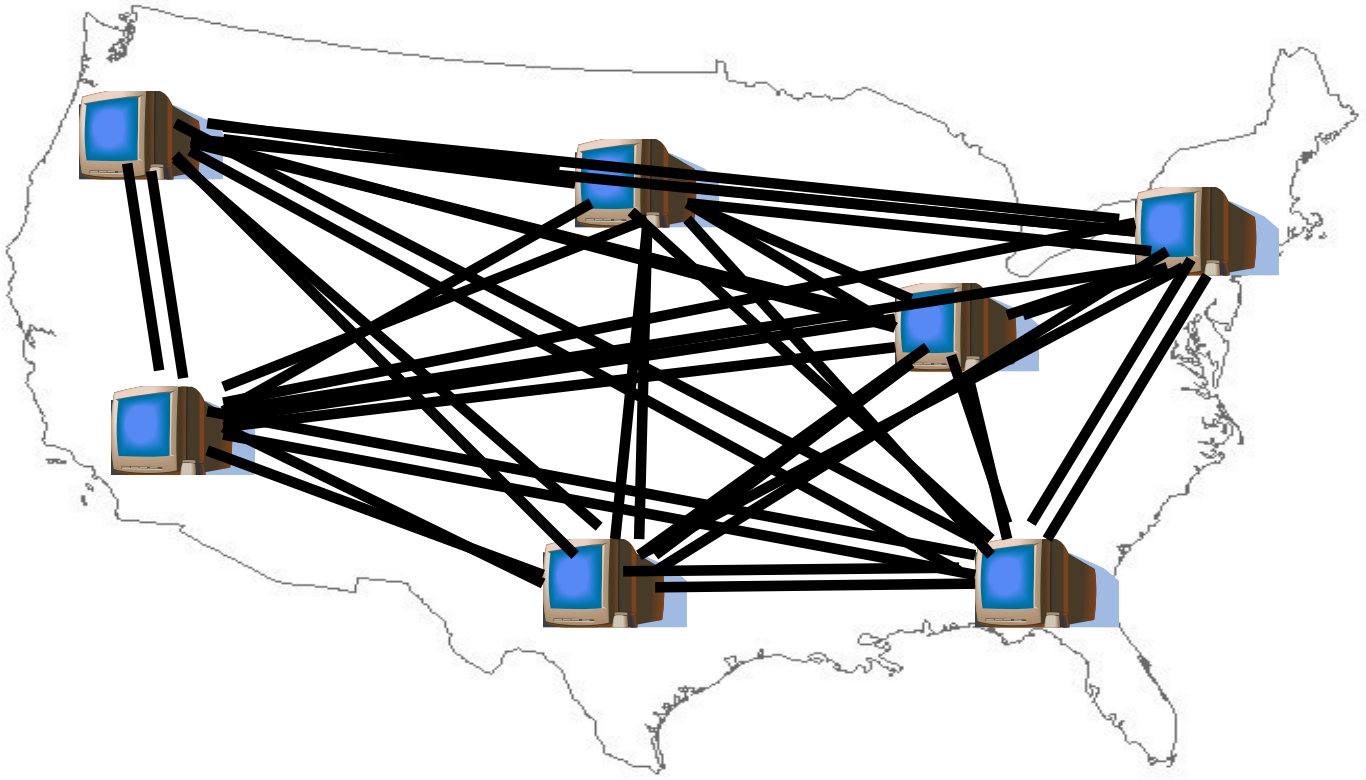
- One of our enterprise customers is undergoing a DoS attack – how can we protect them?
- We are entering the European market and need to construct an international IP backbone – can you propose a design for us?
- Customers of our North American ISP backbone are noticing random outages – diagnose and fix what's wrong.
- We have to peer our network with our biggest competitor – how do we protect ourselves from them?

How can Two Hosts Communicate?



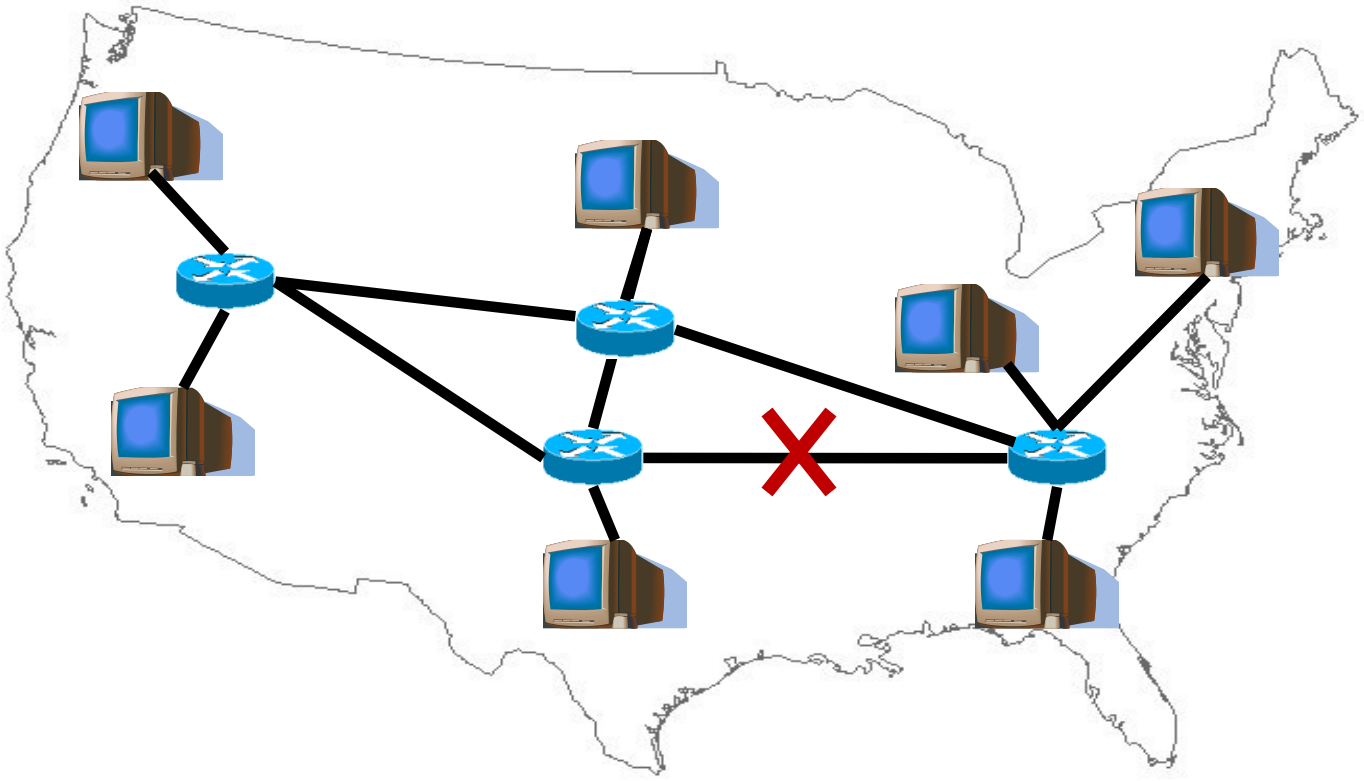
- Encode information on modulated “Carrier signal”
  - Phase, frequency, and amplitude modulation, and combinations thereof
  - Ethernet: self-clocking Manchester coding ensures one transition per clock
  - Technologies: copper, optical, wireless

How can many hosts communicate?



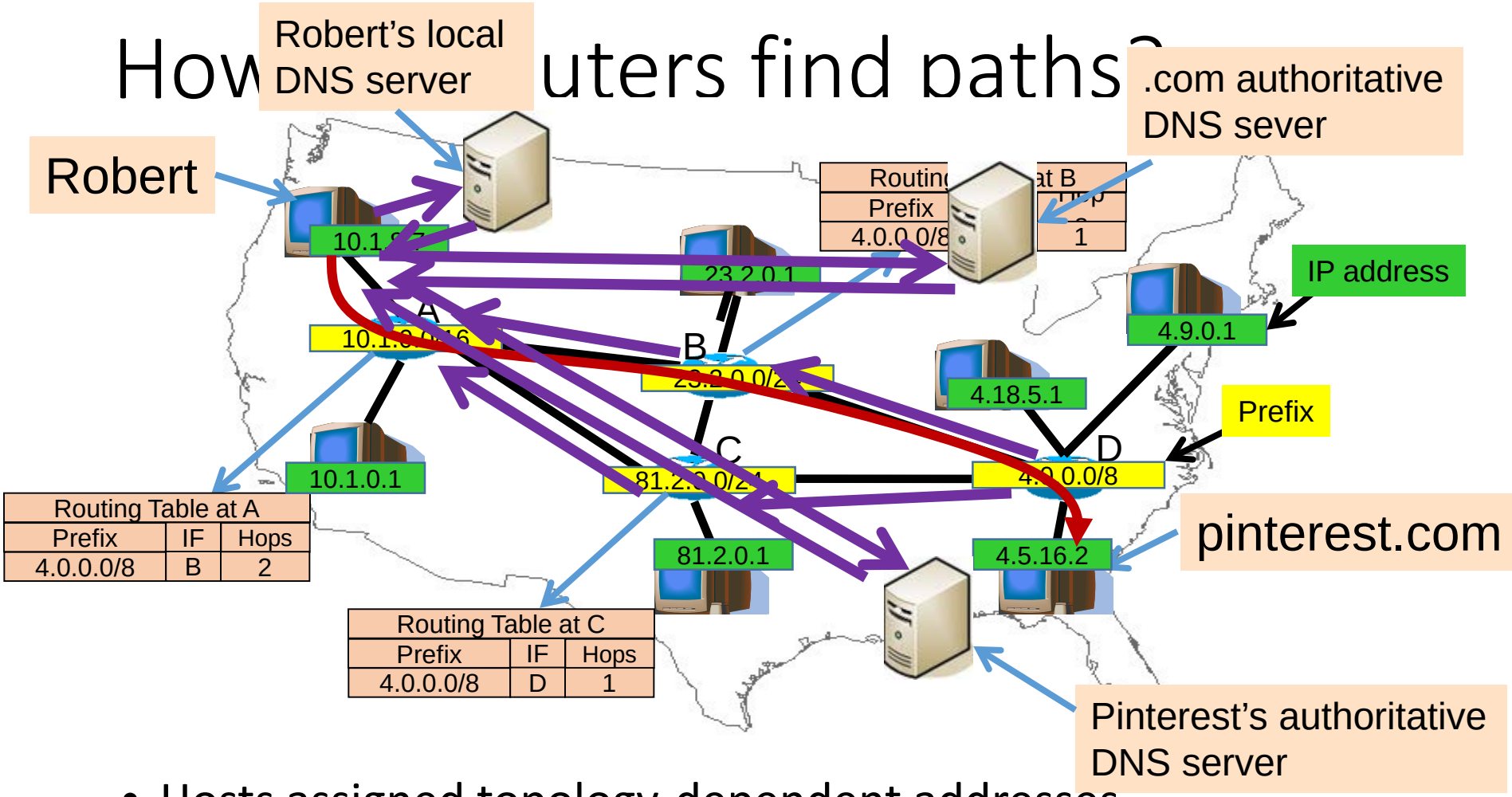
- Naïve approach: full mesh
- Problem:
  - Obviously doesn't scale to the 570,937,778+ hosts in the Internet

How can many hosts communicate?



- Better approach: Multiplex traffic with routers
- Goals: make network robust to failures and attack, maintain spare capacity, reduce operational costs
  - Introduces new challenges: What topology to use? How to find and look up paths? How to identify destinations?

How routers find paths?

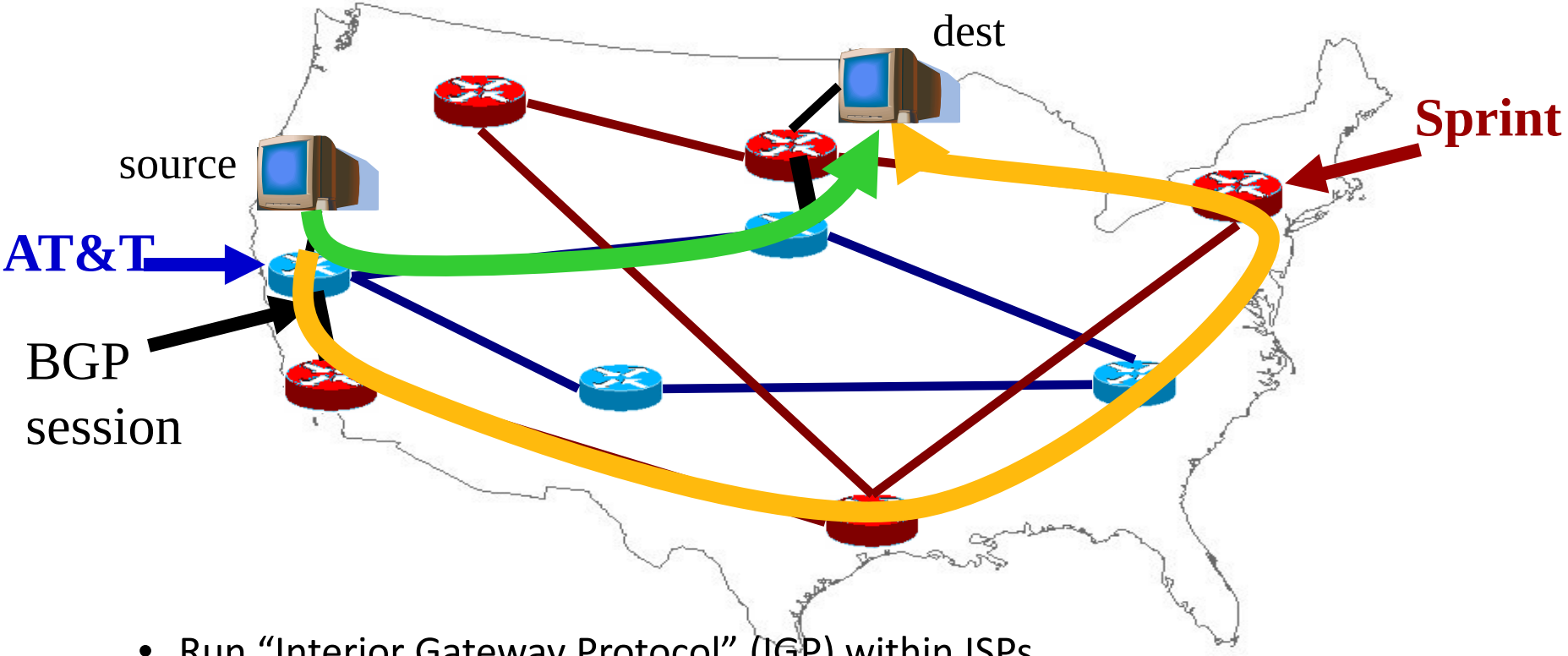


- Hosts assigned topology-dependent addresses
- Routers advertise address blocks ("prefixes")
- Routers compute "shortest" paths to prefixes
- Map IP addresses to names with DNS

Internet Routing

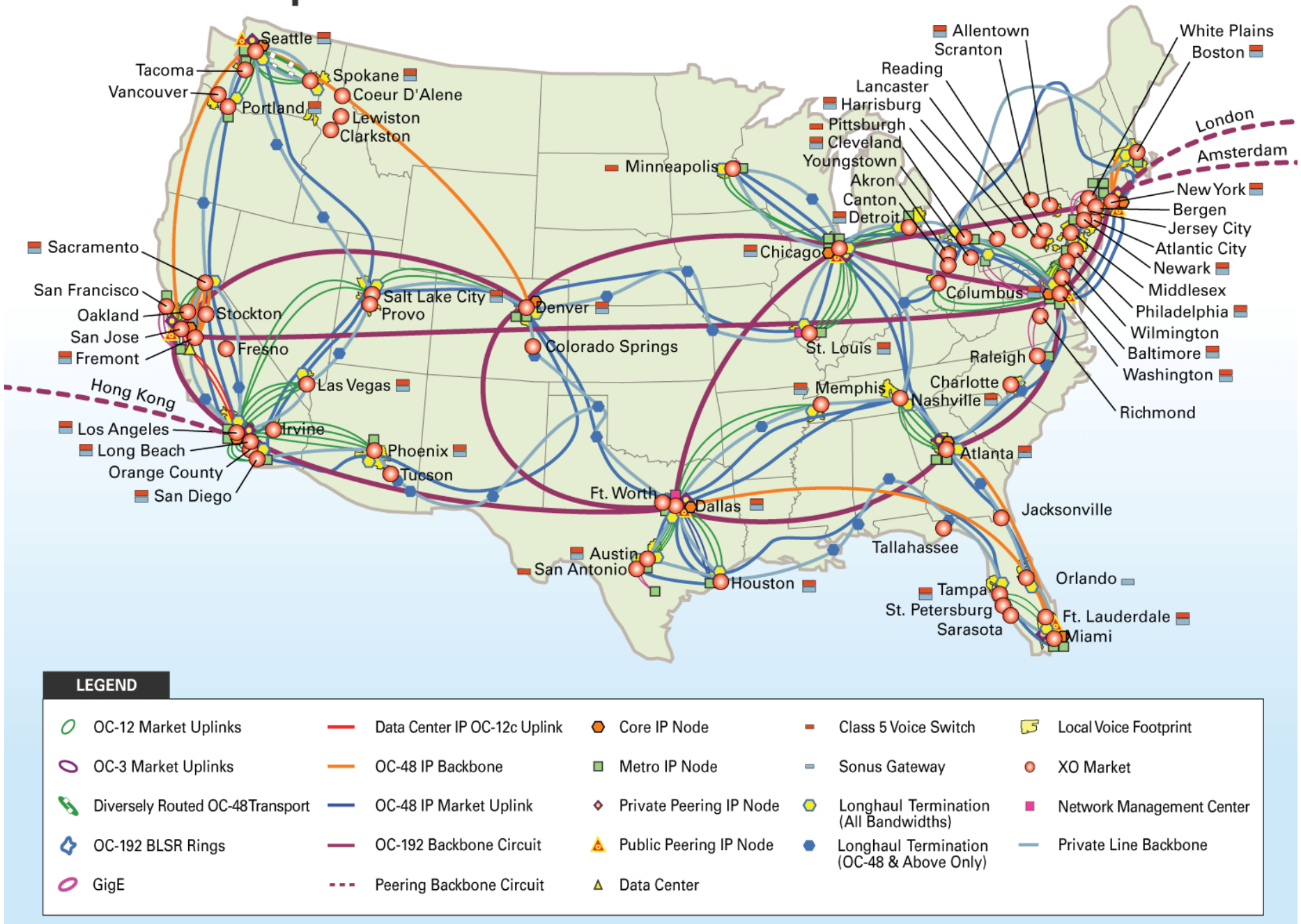
- Internet Routing works at two levels
- Each AS runs an intra-domain routing protocol that establishes routes within its domain
  - (AS -- region of network under a single administrative entity)
  - Link State, e.g., Open Shortest Path First (OSPF)
  - Distance Vector, e.g., Routing Information Protocol (RIP)
- ASes participate in an inter-domain routing protocol that establishes routes between domains
  - Path Vector, e.g., Border Gateway Protocol (BGP)

Intra- vs. Inter-domain routing



- Run “Interior Gateway Protocol” (IGP) within ISPs
  - OSPF, IS-IS, RIP
- Use “Border Gateway Protocol” (BGP) to connect ISPs
  - To reduce costs, peer at exchange points (AMS-IX, MAE-EAST)

Complete Network Assets : XO Communications

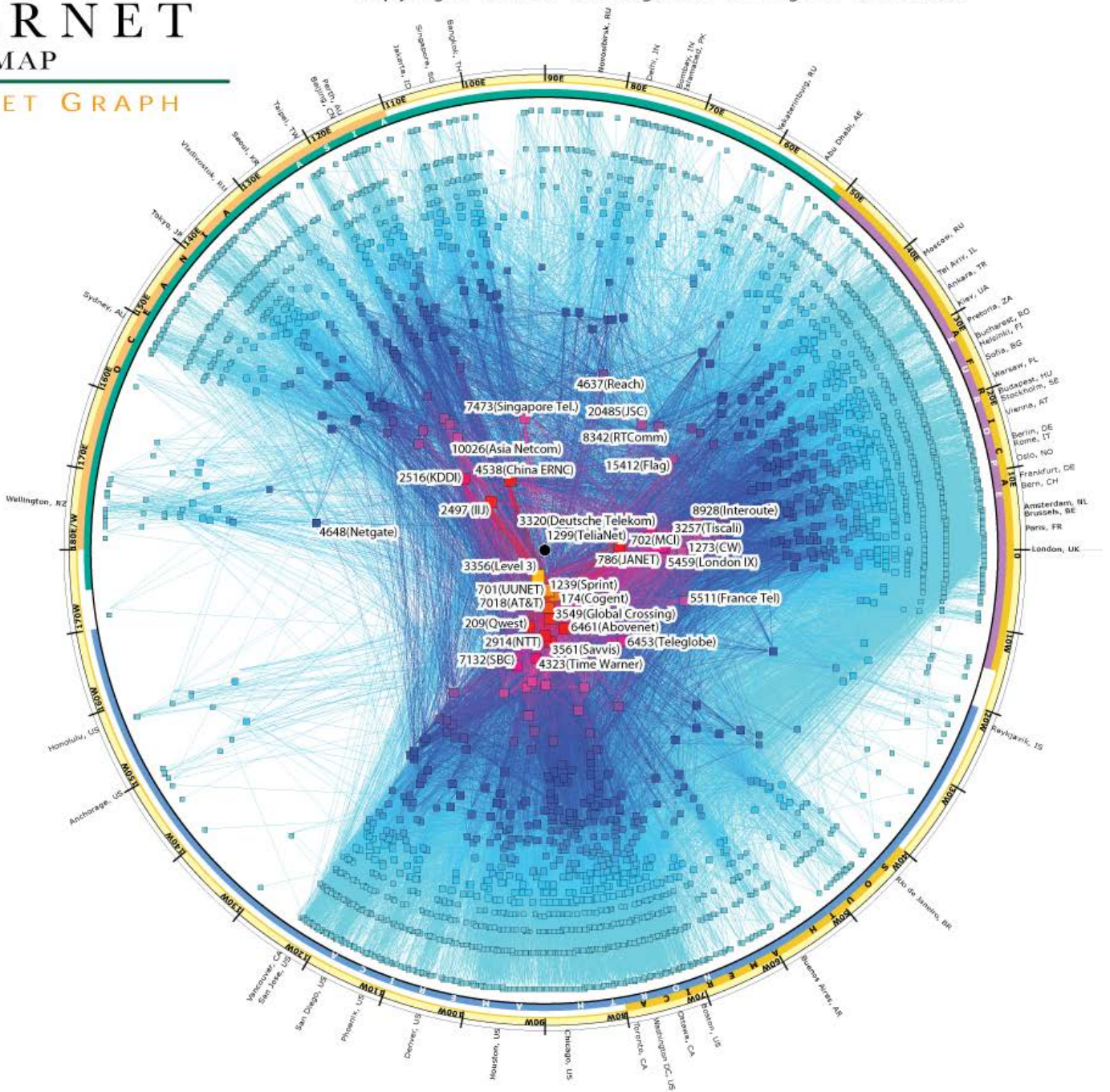


LEGEND

- OC-12 Market Uplinks
- OC-3 Market Uplinks
- Diversely Routed OC-48Transport
- OC-192 BLSR Rings
- GigE
- Data Center IP OC-12c Uplink
- OC-48 IP Backbone
- OC-48 IP Market Uplink
- OC-192 Backbone Circuit
- Peering Backbone Circuit
- Core IP Node
- Metro IP Node
- Private Peering IP Node
- Public Peering IP Node
- Data Center
- Class 5 Voice Switch
- Sonus Gateway
- Longhaul Termination (All Bandwidths)
- Longhaul Termination (OC-48 & Above Only)
- Local Voice Footprint
- XO Market
- Network Management Center
- Private Line Backbone

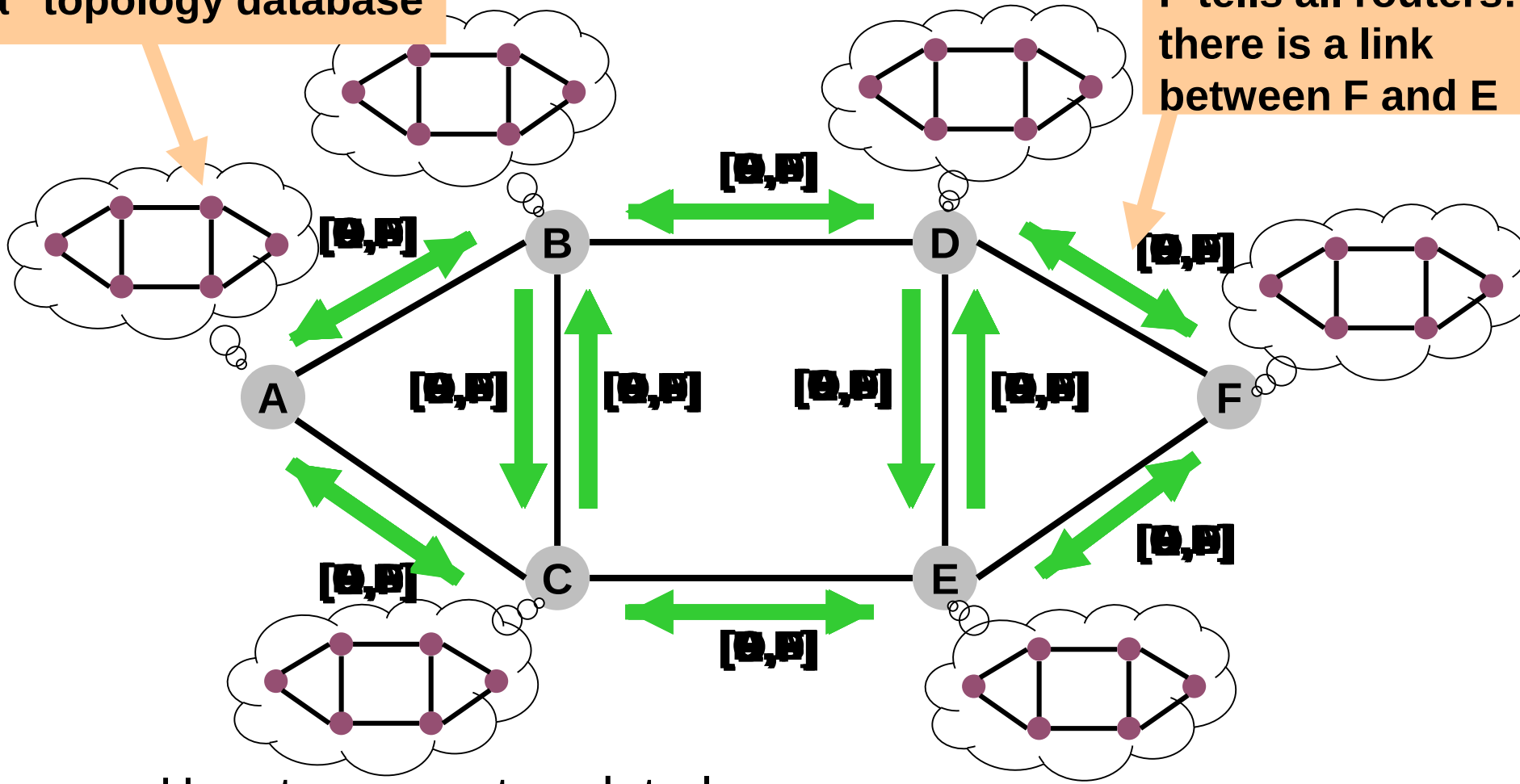
IPv4 INTERNET
TOPOLOGY MAP

AS-level INTERNET GRAPH



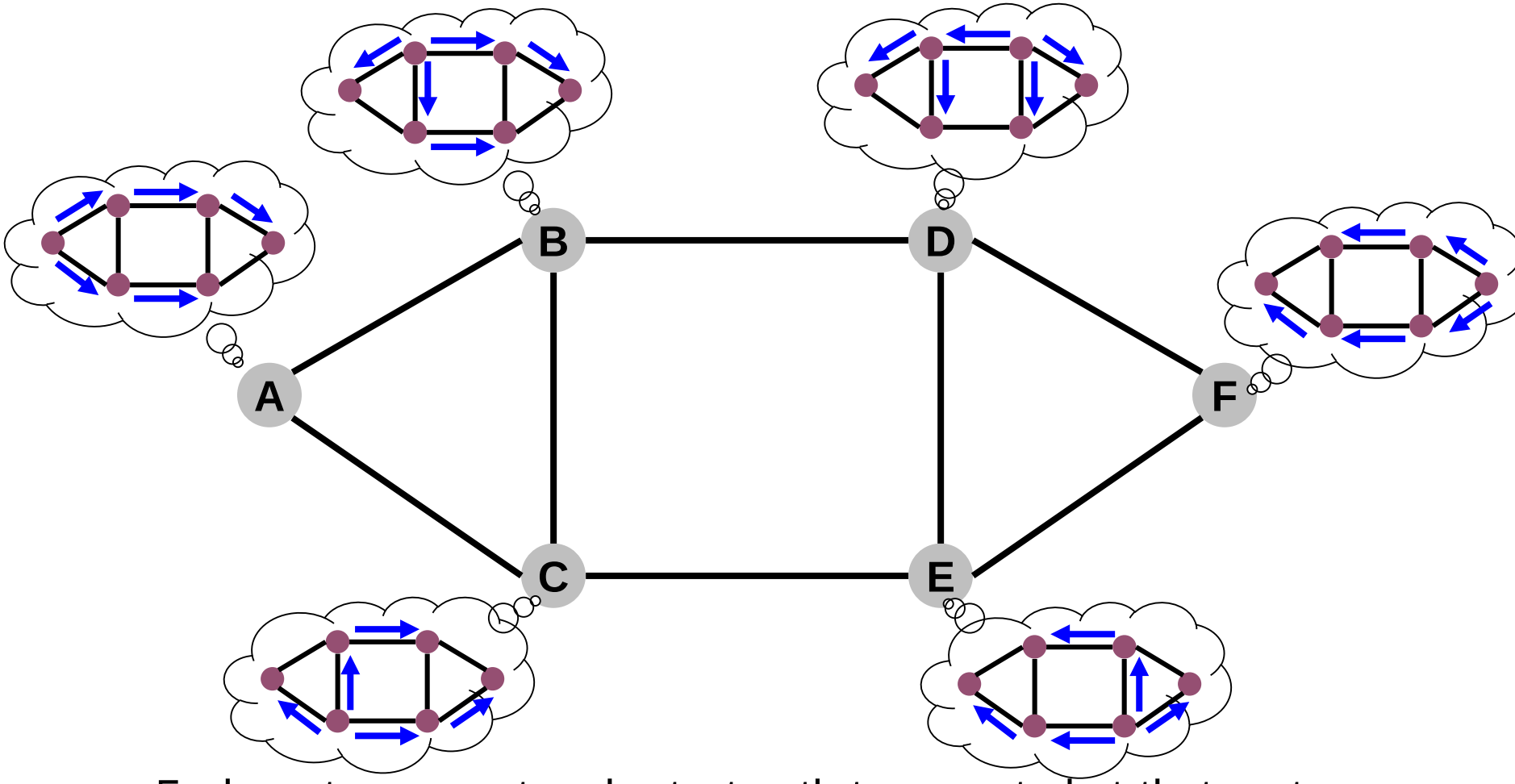
Link-State Routing

: update propagation



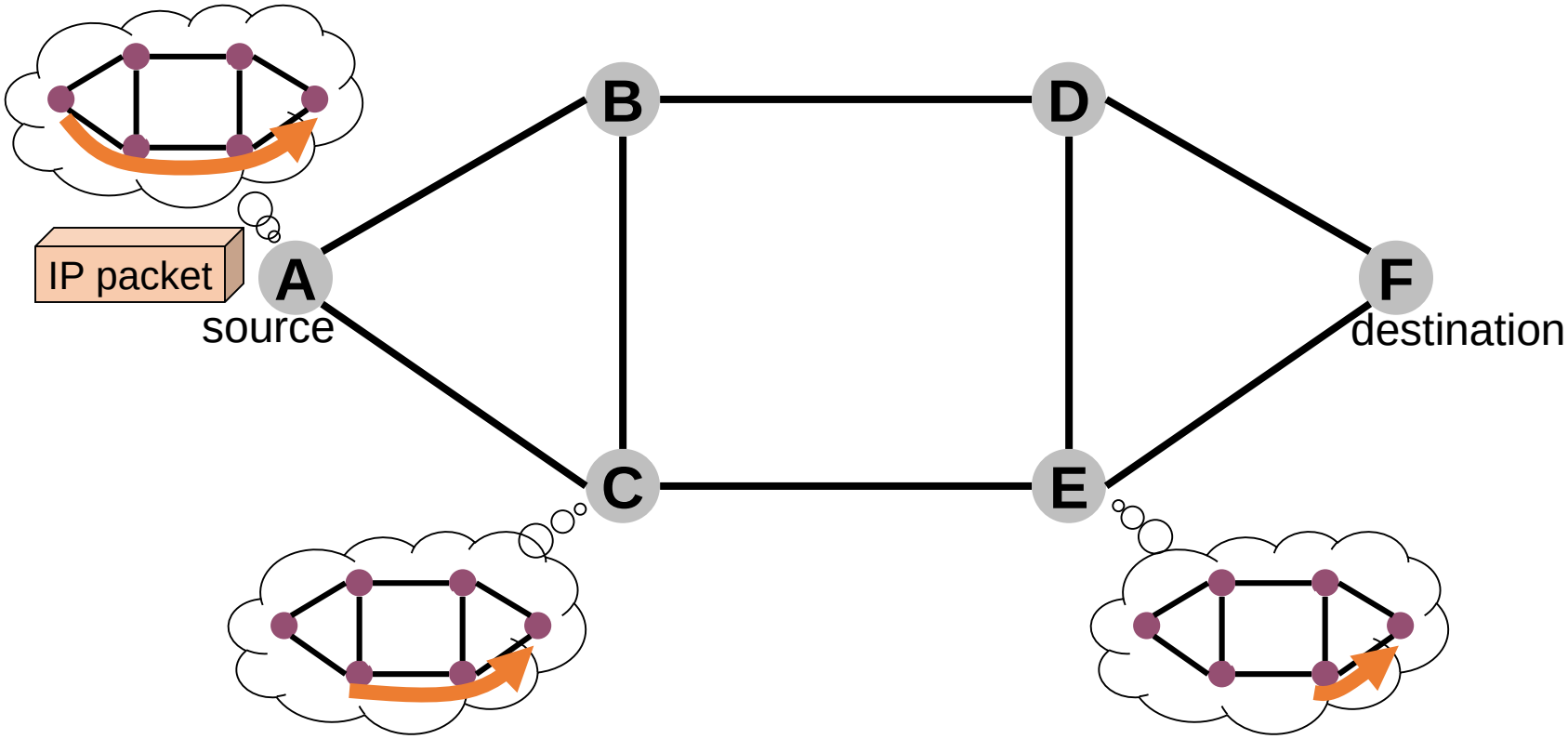
- How to prevent update loops:
- How to bring up new node:

Link state: route computation



- Each router computes shortest path tree, rooted at that router
- Determines next-hop to each dest, publish to forwarding table
- Operators can assign link costs to control path selection

Link-state: packet forwarding

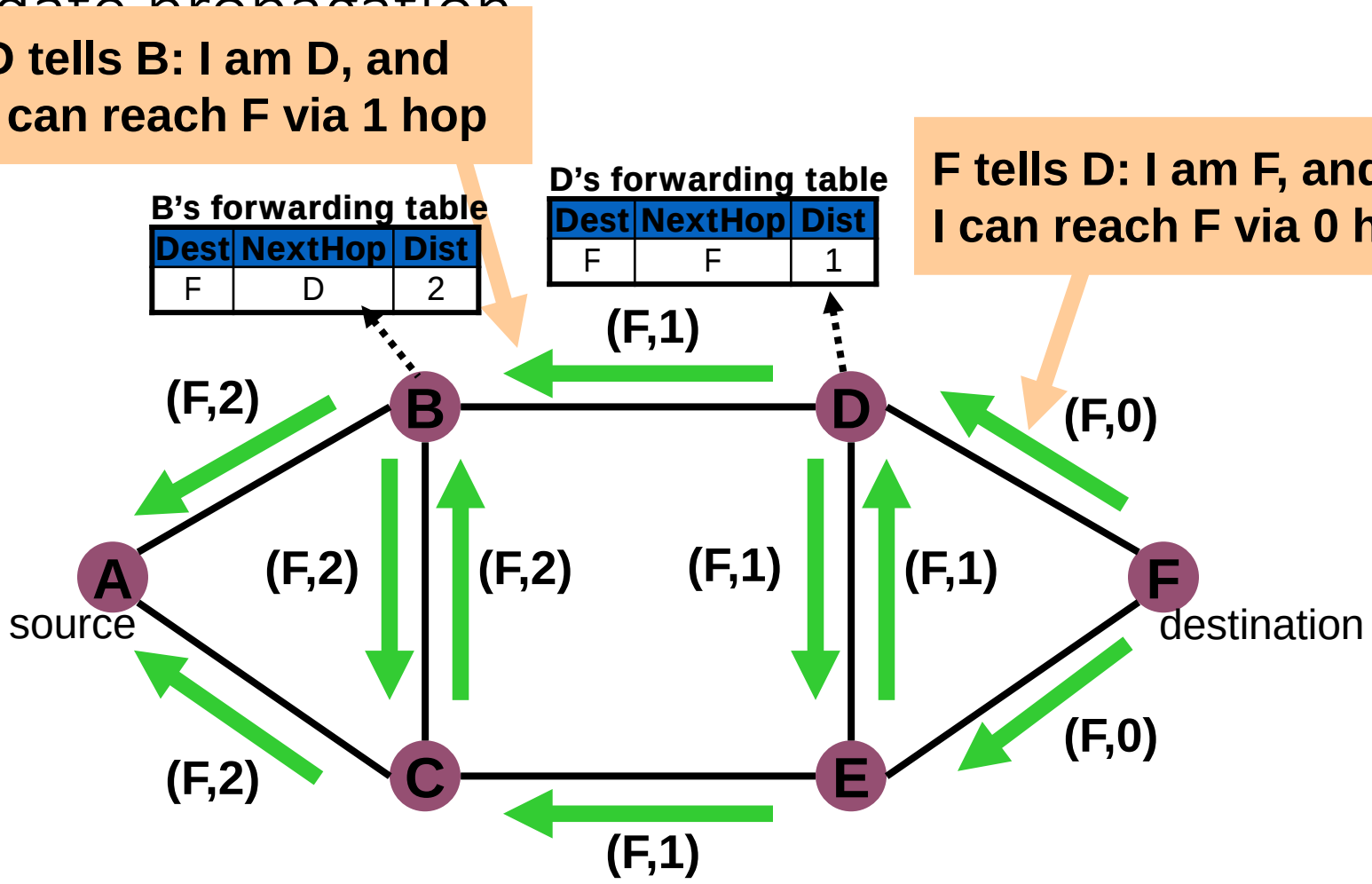


- In practice: shortest path precomputed, next-hops stored in forwarding table
- Downsides of link-state:
  - Lesser control on policy (certain routes can't be filtered), more cpu
  - Increased visibility (bad for privacy, but good for diagnostics)

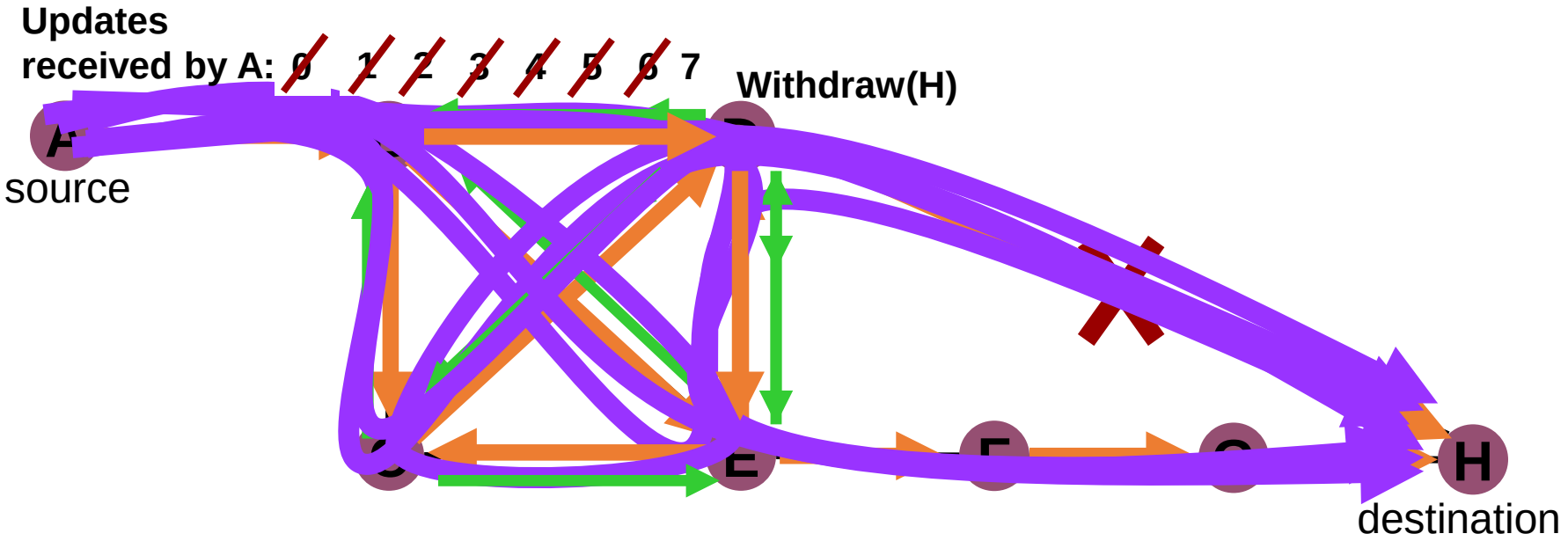
Distance Vector Routing

- Each router knows the links to its neighbors
  - Does *not* flood this information to the whole network
- Each router has provisional “shortest path” to every other router
  - E.g.: Router A: “I can get to router B with cost 11”
- Routers exchange this distance vector information with their neighboring routers
  - Vector because one entry per destination
- Routers look over the set of options offered by their neighbors and select the best one
- Iterative process converges to set of shortest paths

Distance vector: update propagation

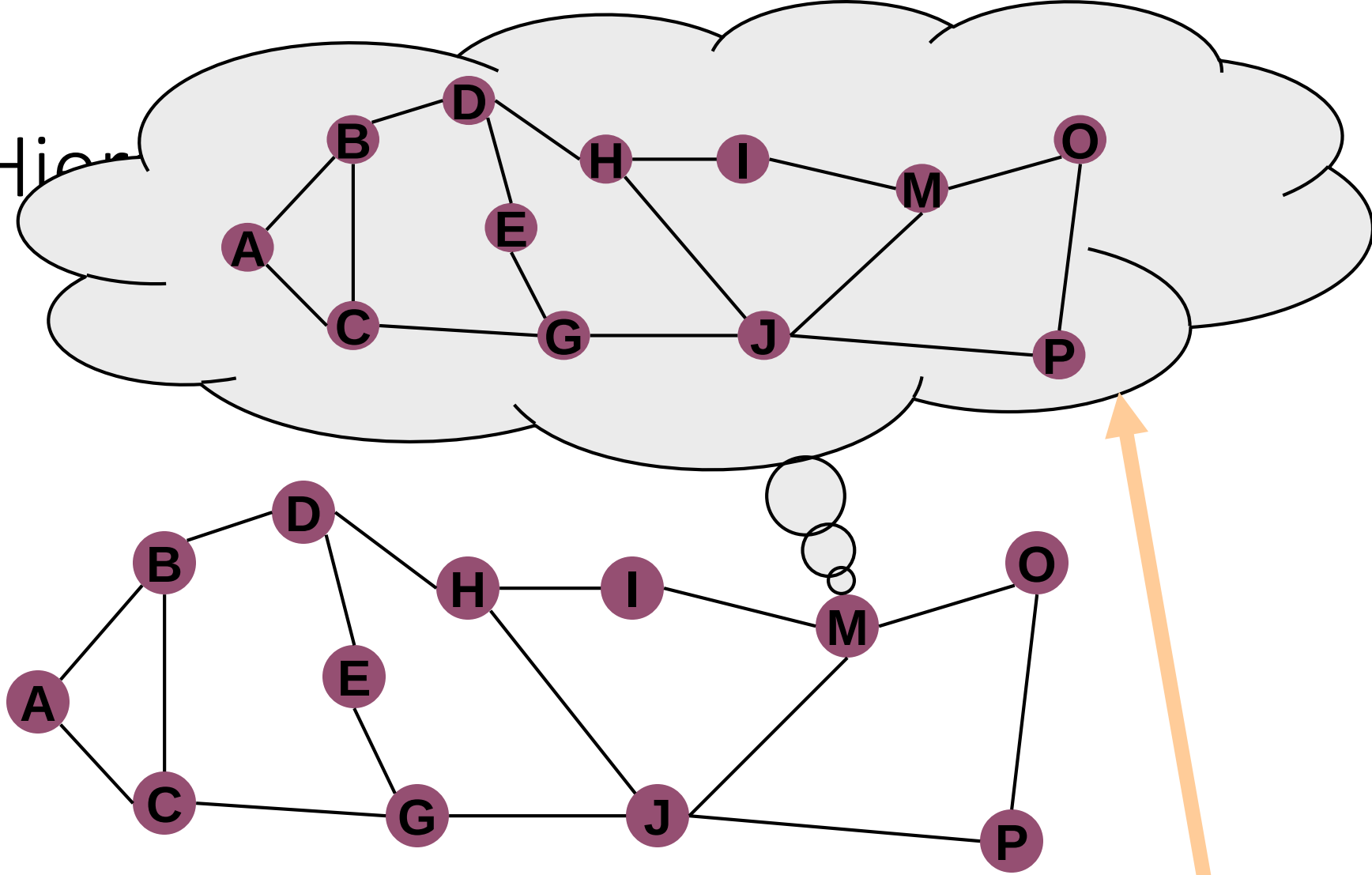


Distance vector: convergence



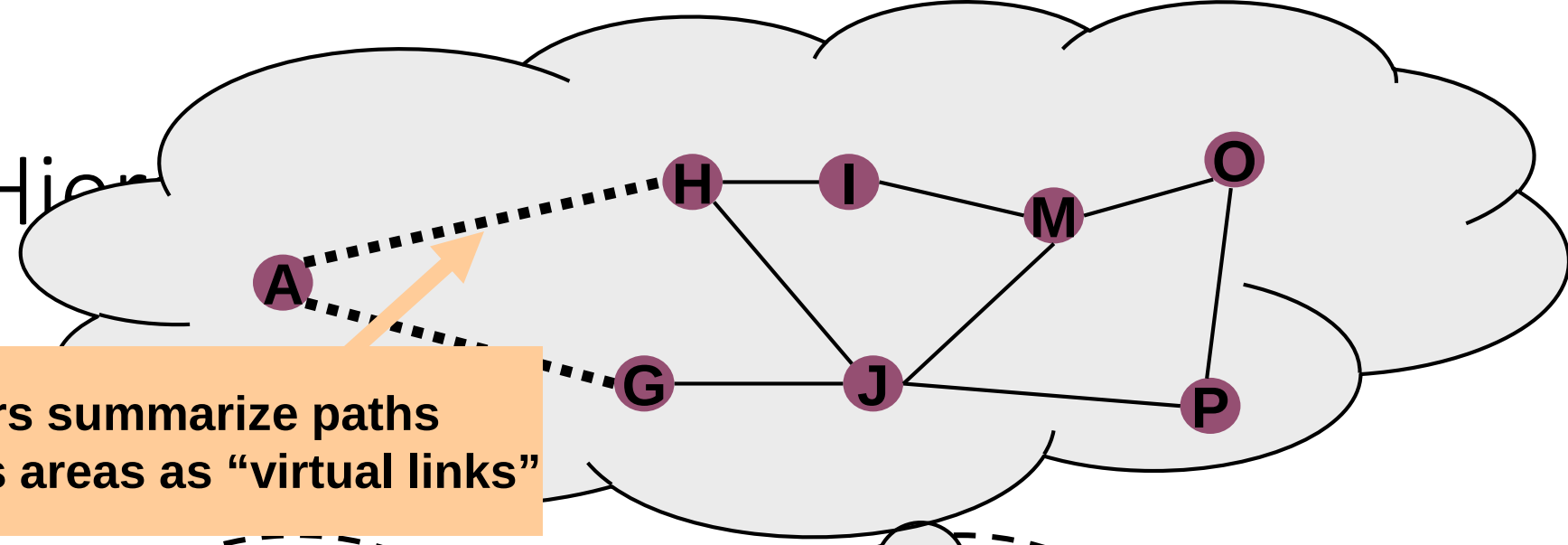
- How many updates would link-state require?
- Is link-state better or worse than distance vector?
- Which should be used for intra-domain routing? What about inter-domain routing?

Hier

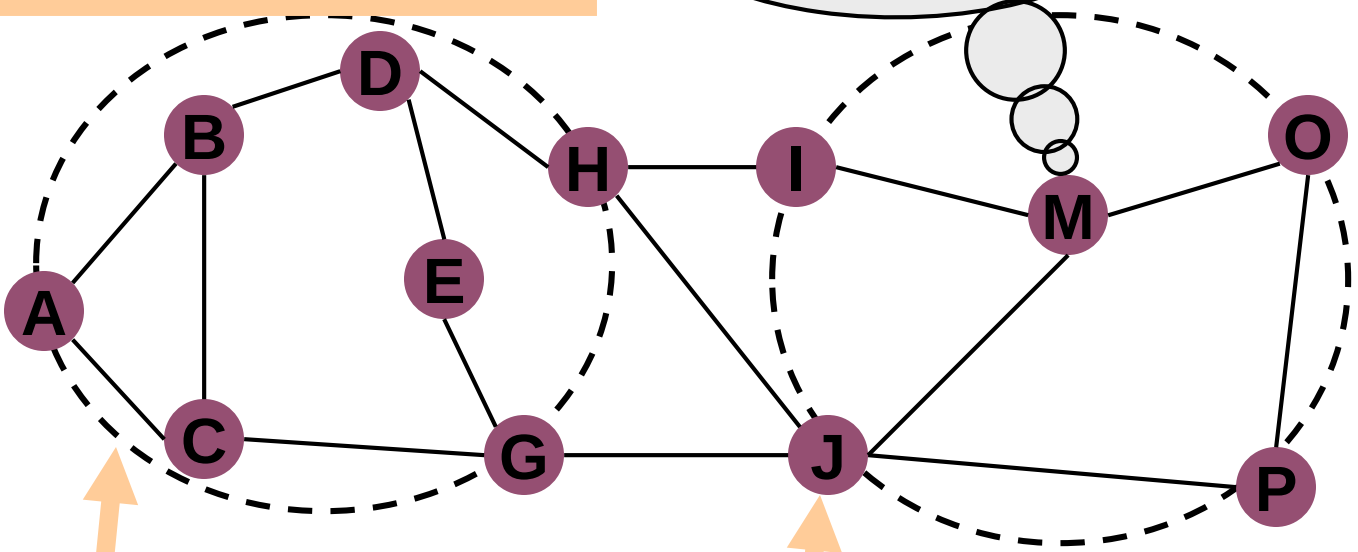


In regular link-state, routers maintain map of entire topology

Hier



Routers summarize paths across areas as “virtual links”



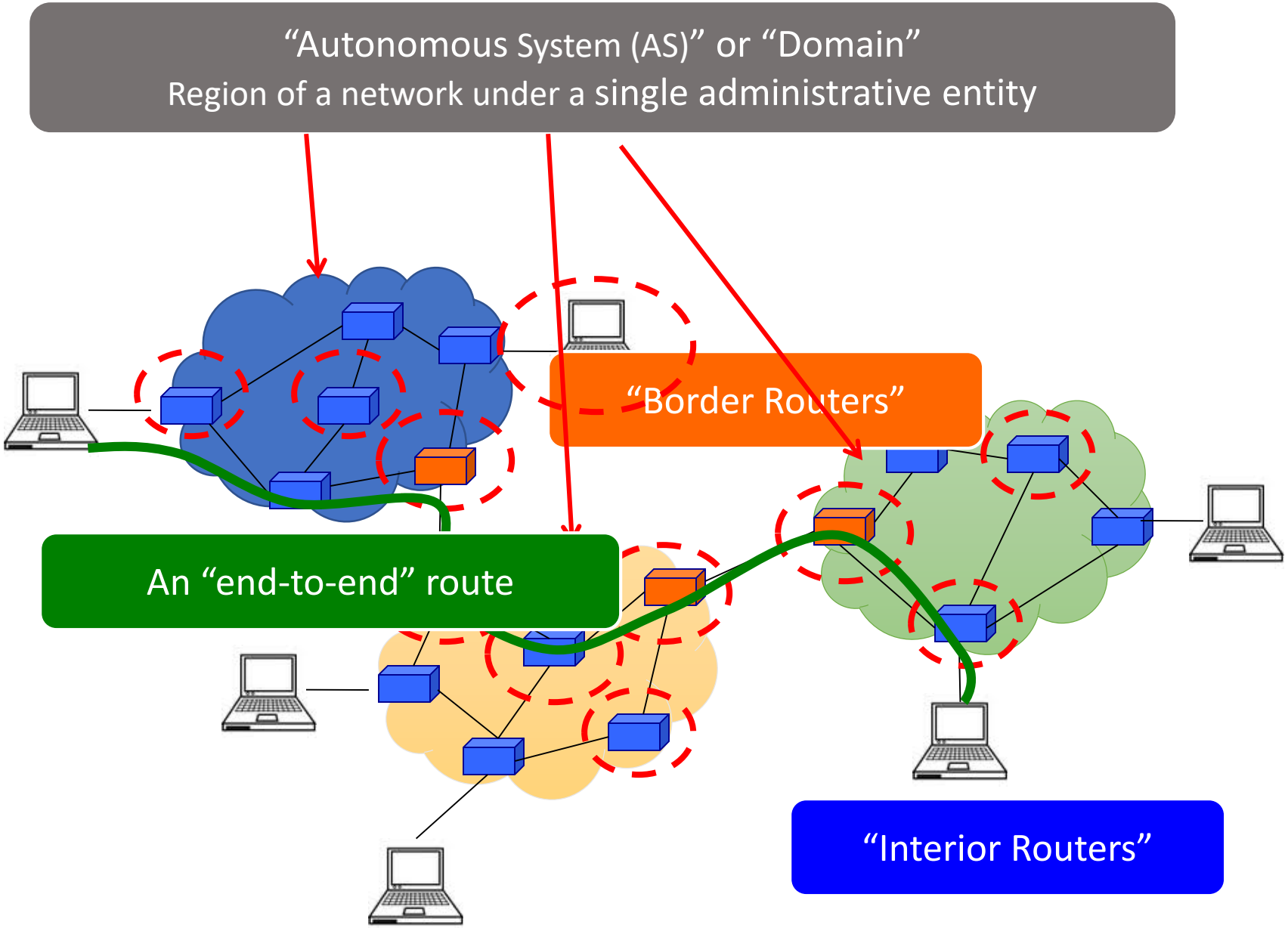
Aggregate groups of routers into “areas”

“Border routers” generate “summary LSPs” to reach other border routers

IP Routing: Interdomain

Internet Routing

- So far, only considered routing within a domain
- Many issues can be ignored in this setting because there is central administrative control over routers
  - Issues such as *autonomy*, *privacy*, *policy*
- **But the Internet is more than a single domain**


"Autonomous System (AS)" or "Domain"
Region of a network under a single administrative entity
"Border Routers"
An "end-to-end" route
"Interior Routers"

Autonomous Systems (AS)

- AS is a network under a single administrative control
  - currently over 30,000 ASes
  - Think AT&T, France Telecom, UCB, IBM, *etc.*
- ASes are sometimes called “domains”.
  - Hence, “interdomain routing”
- Each AS is assigned a unique identifier
  - 16 bit AS Number (ASN)

Administrative structure shapes Interdomain routing

- ASes want freedom to pick routes based on **policy**
  - *"My traffic can't be carried over my competitor's network"*
  - *"I don't want to carry A's traffic through my network"*
  - Not expressible as Internet-wide "shortest path"!
- ASes want **autonomy**
  - Want to choose their own internal routing protocol
  - Want to choose their own policy
- ASes want **privacy**
  - choice of network topology, routing policies, *etc.*

Choice of Routing Algorithm

Link State (LS) *vs.* Distance Vector (DV)?

- LS offers no privacy -- global sharing of all network information (neighbors, policies)
- LS limits autonomy -- need agreement on metric, algorithm
- DV is a decent starting point
  - per-destination advertisement gives providers a hook for finer-grained control over whether/which routes to advertise
  - but DV wasn't designed to implement policy

The "Border Gateway Protocol" (BGP) extends distance-vector ideas to accommodate policy

Shortest-path forwarding isn't enough

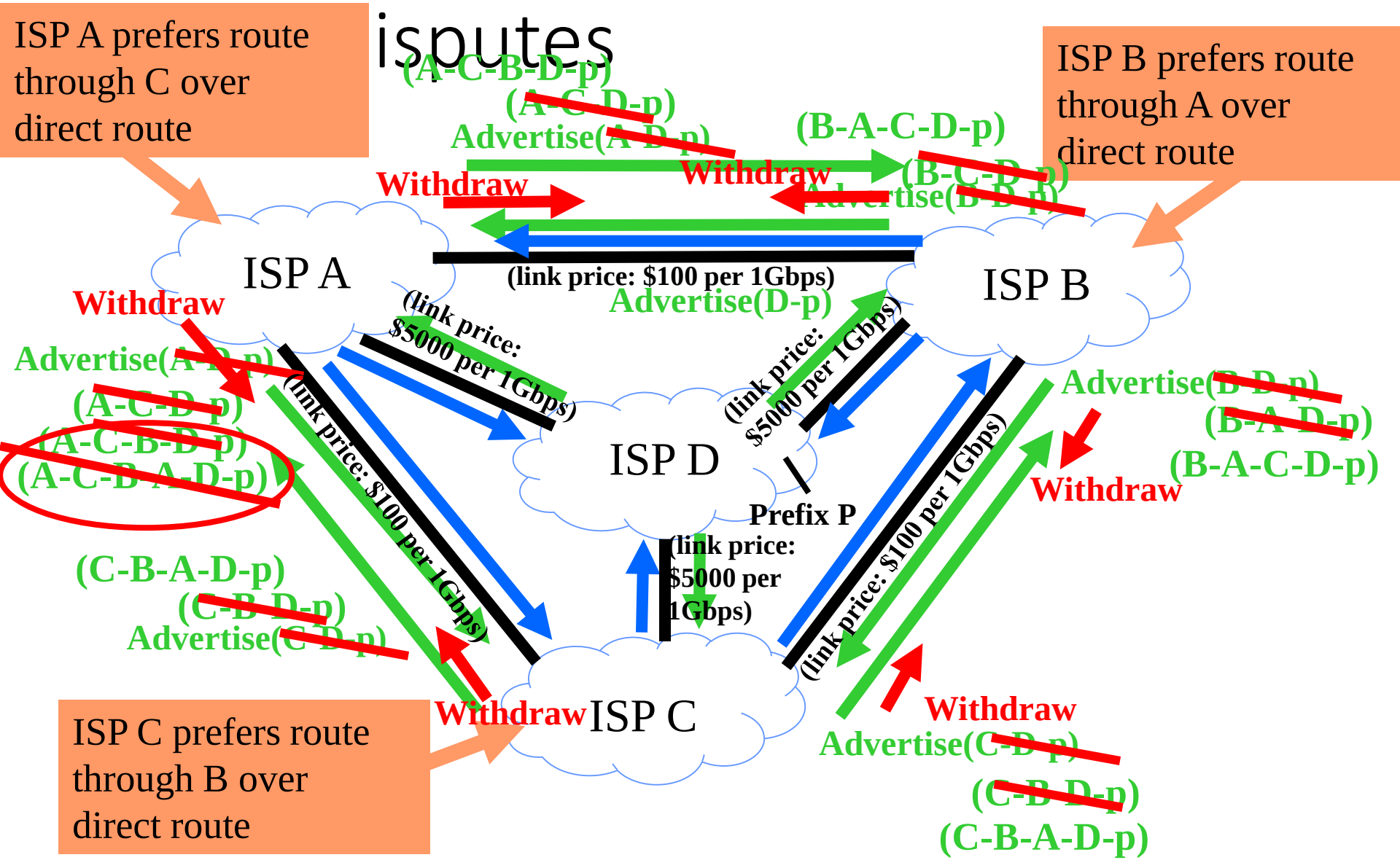
- In the real world, ISPs want to influence path selection
  - Load balance traffic, prefer cheaper paths, avoid untrusted routes, give preferential service, block reachability, limit external control over path selection decisions
- One trick: change the "cost" used to compute shortest paths
- Another trick: filter routes from being received from/advertised to certain neighbors

A few other inconvenient aspects

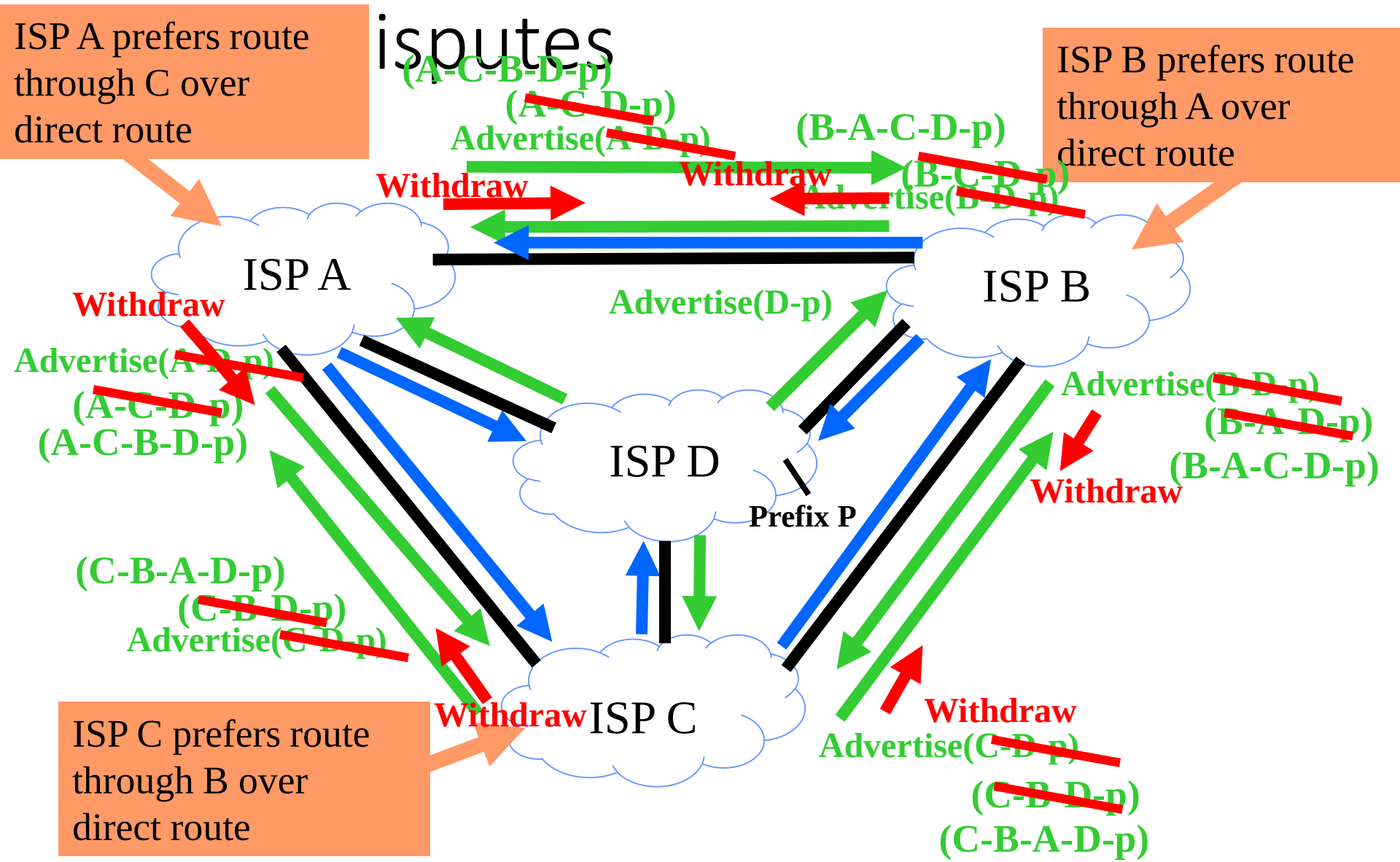
- What if we use a non-additive metric?
  - E.g., maximal capacity
- What if routers don't use the same metric?
  - I want low delay, you want low loss rate?
- What happens if nodes lie?

Can You Use Any Metric?

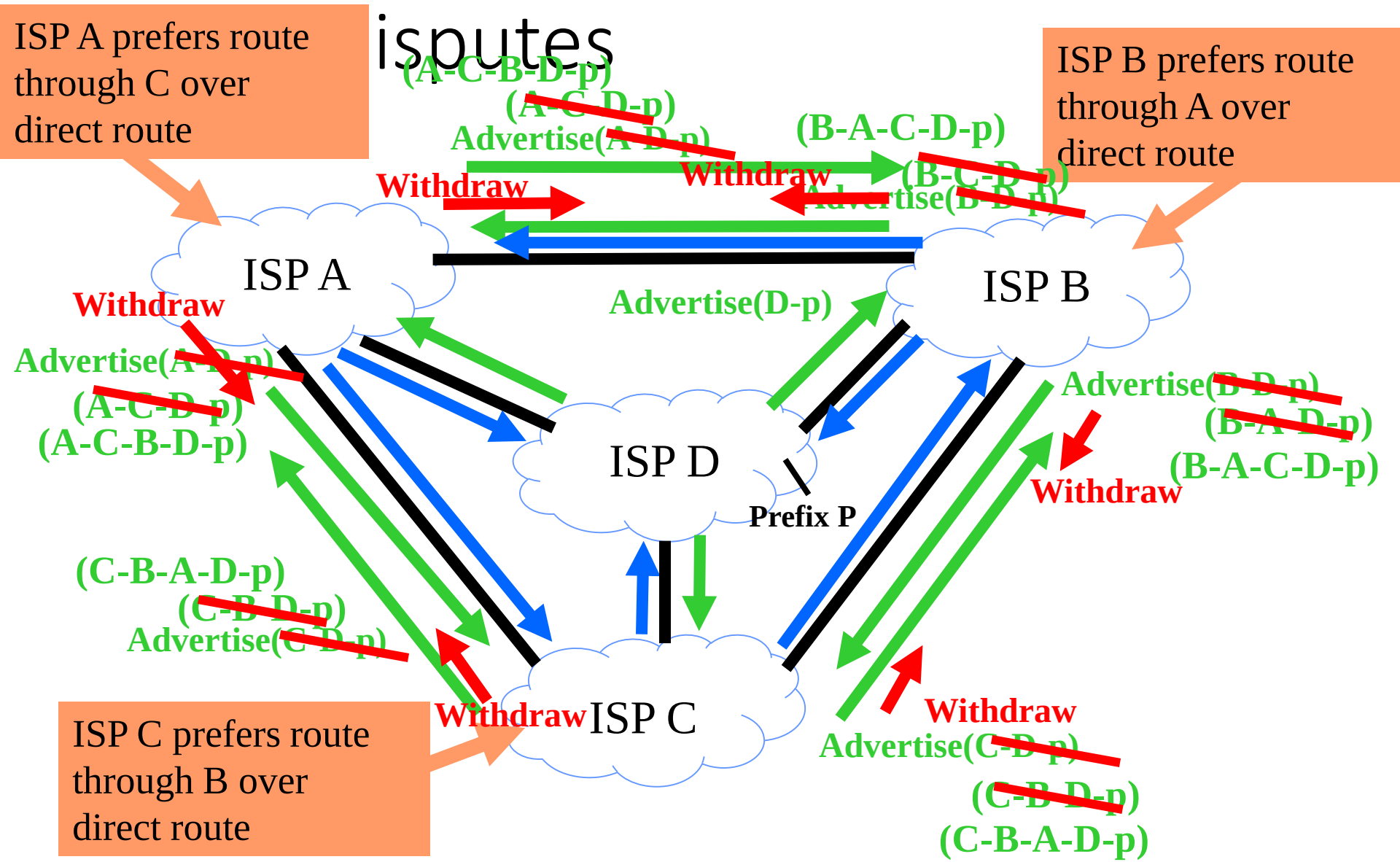
- I said that we can pick any metric. Really?
- What about maximizing capacity?


ISP A prefers route through C over direct route
isputes
(A-C-B-D-p)
(A-C-D-p)
Advertise(A-D-p)
Withdraw
(B-A-C-D-p)
Withdraw
(B-C-D-p)
Advertise(B-D-p)
ISP B prefers route through A over direct route
ISP A
(link price: $100 per 1Gbps)
ISP B
Withdraw
Advertise(D-p)
Advertise(A-D-p)
(A-C-D-p)
(A-C-B-D-p)
(A-C-B-A-D-p)
(link price: $5000 per 1Gbps)
(link price: $5000 per 1Gbps)
ISP D
Prefix P
(link price: $5000 per 1Gbps)
(link price: $100 per 1Gbps)
(link price: $100 per 1Gbps)
Advertise(B-D-p)
(B-A-D-p)
(B-A-C-D-p)
Withdraw
(C-B-A-D-p)
(C-B-D-p)
Advertise(C-D-p)
Withdraw
ISP C
ISP C prefers route through B over direct route
Withdraw
Advertise(C-D-p)
(C-B-D-p)
(C-B-A-D-p)

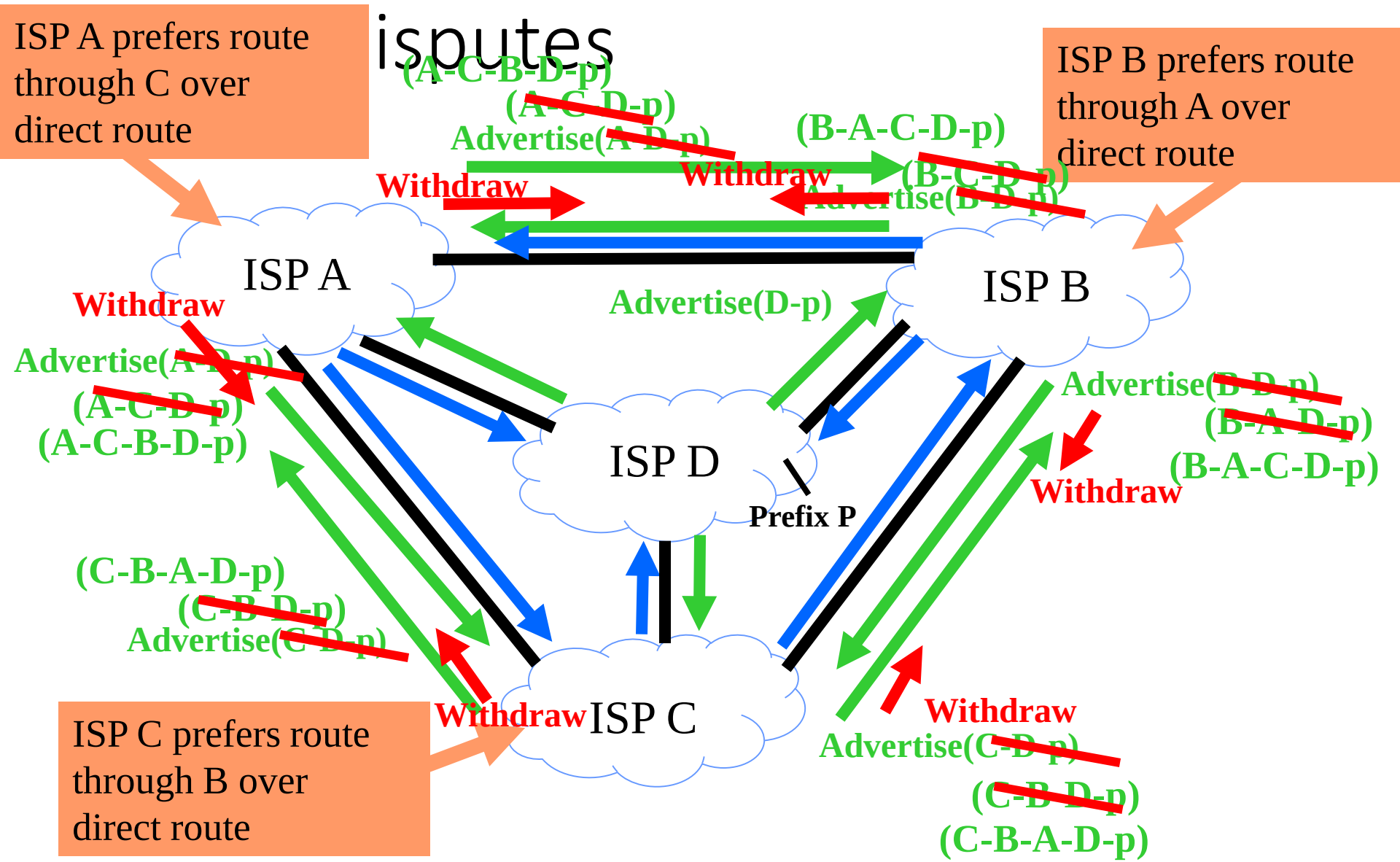
isputes


ISP A prefers route through C over direct route
(A-C-B-D-p)
(A-C-D-p)
Advertise(A-D-p)
Withdraw
(B-A-C-D-p)
(B-C-D-p)
Advertise(B-D-p)
Withdraw
ISP B prefers route through A over direct route
ISP A
ISP B
Withdraw
Advertise(A-D-p)
(A-C-D-p)
(A-C-B-D-p)
Advertise(D-p)
ISP D
Prefix P
Advertise(B-D-p)
(B-A-D-p)
(B-A-C-D-p)
Withdraw
(C-B-A-D-p)
(C-B-D-p)
Advertise(C-D-p)
Withdraw
ISP C
Withdraw
Advertise(C-D-p)
(C-B-D-p)
(C-B-A-D-p)
ISP C prefers route through B over direct route

isputes


ISP A prefers route through C over direct route
(A-C-B-D-p)
(A-C-D-p)
Advertise(A-D-p)
Withdraw
(B-A-C-D-p)
(B-C-D-p)
Advertise(B-D-p)
Withdraw
ISP B prefers route through A over direct route
ISP A
ISP B
Withdraw
Advertise(A-D-p)
(A-C-D-p)
(A-C-B-D-p)
Advertise(D-p)
ISP D
Prefix P
Advertise(B-D-p)
(B-A-D-p)
(B-A-C-D-p)
Withdraw
(C-B-A-D-p)
(C-B-D-p)
Advertise(C-D-p)
Withdraw
ISP C
Withdraw
Advertise(C-D-p)
(C-B-D-p)
(C-B-A-D-p)
ISP C prefers route through B over direct route

isputes


ISP A prefers route through C over direct route
(A-C-B-D-p)
(A-C-D-p)
Advertise(A-D-p)
Withdraw
(B-A-C-D-p)
(B-C-D-p)
Advertise(B-D-p)
Withdraw
ISP B prefers route through A over direct route
ISP A
ISP B
Withdraw
Advertise(A-D-p)
(A-C-D-p)
(A-C-B-D-p)
Advertise(D-p)
ISP D
Prefix P
Advertise(B-D-p)
(B-A-D-p)
(B-A-C-D-p)
Withdraw
(C-B-A-D-p)
(C-B-D-p)
Advertise(C-D-p)
Withdraw
ISP C
Withdraw
Advertise(C-D-p)
(C-B-D-p)
(C-B-A-D-p)
ISP C prefers route through B over direct route

No agreement on metrics?

- If the nodes choose their paths according to different criteria, then bad things might happen
- Example
  - Node A is minimizing latency
  - Node B is minimizing loss rate
  - Node C is minimizing price
- Any of those goals are fine, if globally adopted
  - Only a problem when nodes use different criteria
- Consider a routing algorithm where paths are described by delay, cost, loss

Must agree on loop-avoiding metric

- When all nodes minimize same metric
- And that metric increases around loops
- Then process is guaranteed to converge

Metrics (intradomain)

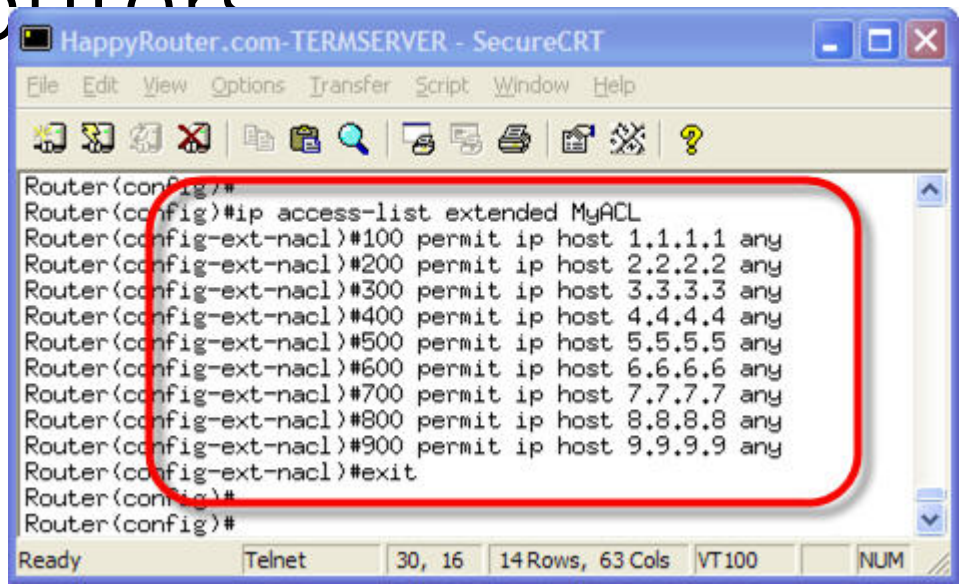
- Propagation delay
- Congestion
- Load balance
- Bandwidth (available, capacity, maximal, bbw)
- Price
- Reliability
- Loss rate
- Combinations of the above

In practice, operators set abstract “weights” (much like our costs); how exactly is a bit of a black art

What happens when routers lie?

- What if a router claims a 1-hop path to everywhere?
- All traffic from nearby routers gets sent there
- How can you tell if they are lying?
- Can this happen in real life?
  - It has, several times….

“Configuring” routers




```
R7(config)# route-map LOCALPREF permit 10
R7(config-route-map)# set local-preference 500
R7(config-route-map)# router bgp 67
R7(config-router)# neighbor 172.31.78.8 route-map LOCALPREF in
```

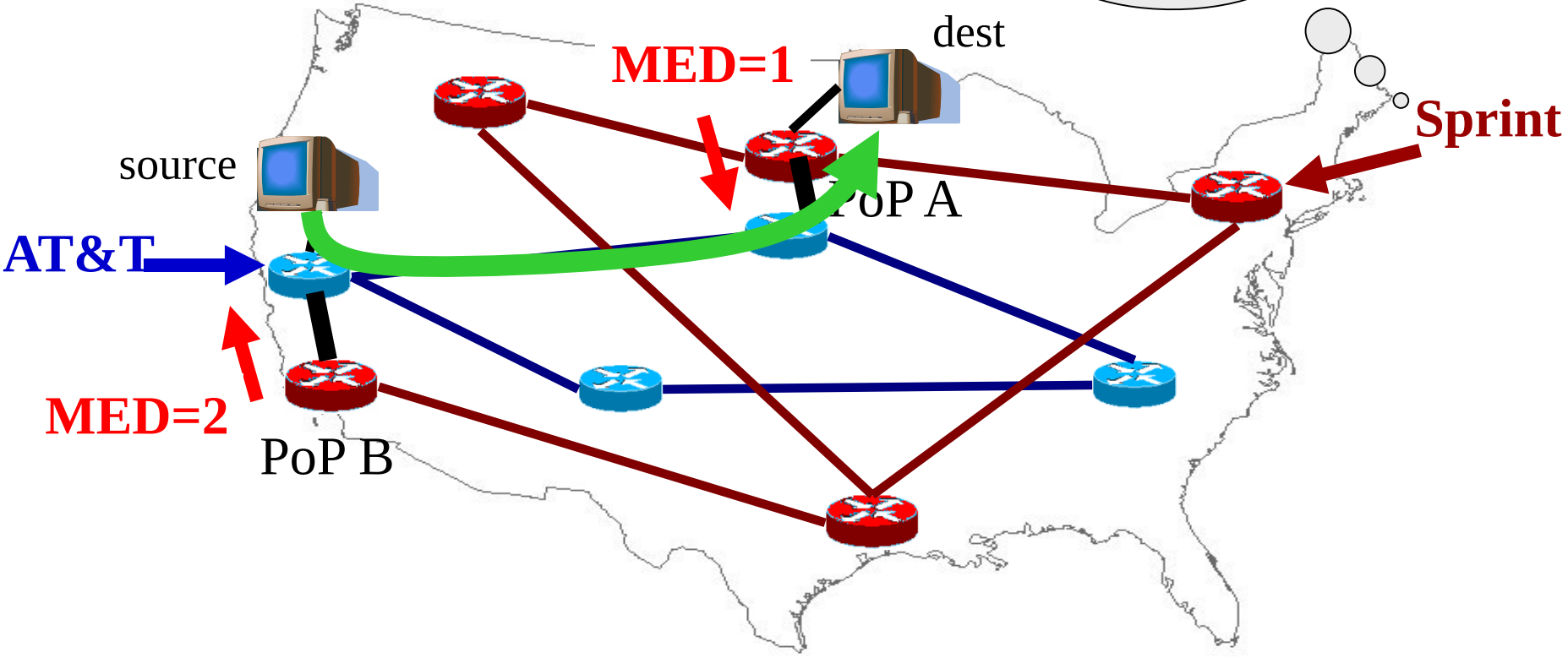
- “Log in” to router (vty)
- Issue commands to view state, instill policy

Changing the "cost" of paths

Step	Attribute	Controlled by local or neighbor AS?
1.	Highest LocalPref	local
2.	Lowest AS path length	neighbor
3.	Lowest origin type	neither
4.	Lowest MED	neighbor
5.	eBGP-learned over iBGP-learned	neither
6.	Lowest IGP cost to border router	local
7.	Lowest router ID (to break ties)	neither

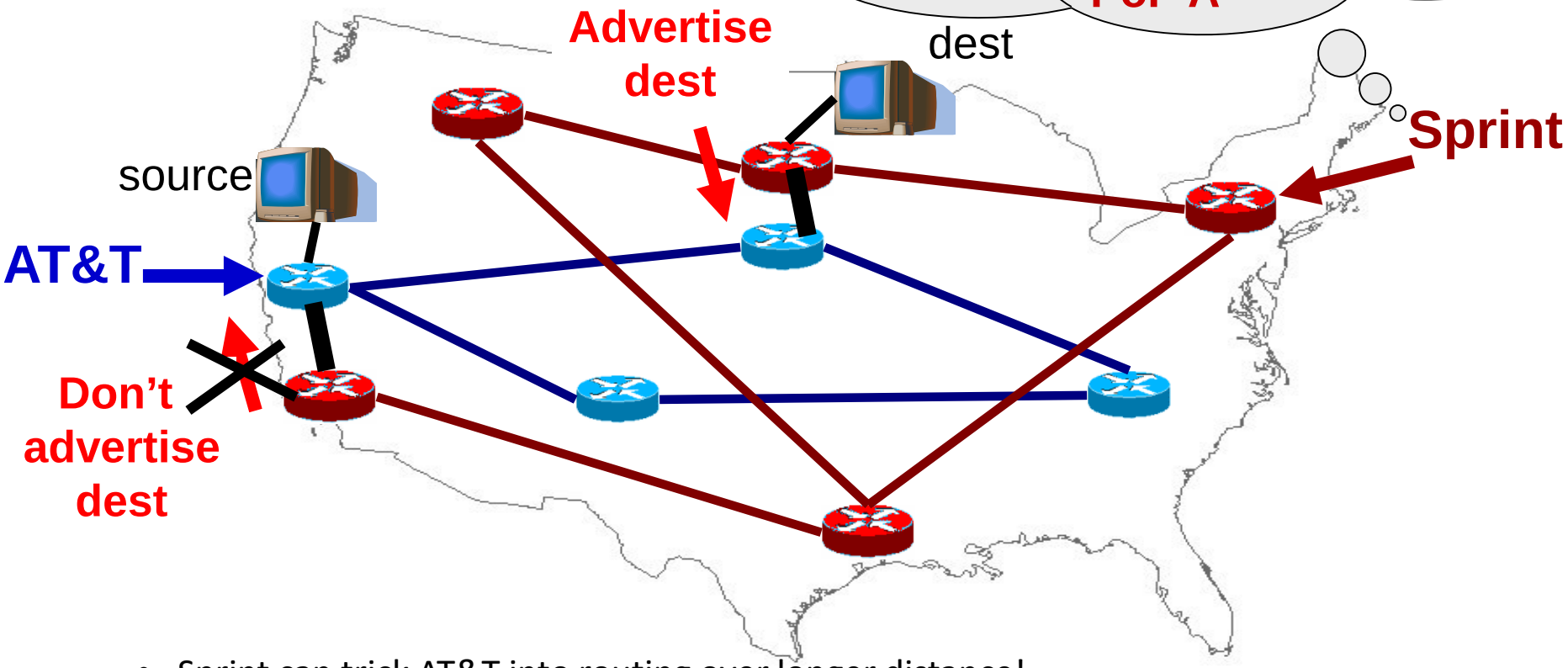
- ISPs have a lot of different kinds of policies
  - Could make cost a linear combination of different metrics
  - More expressive: have several "costs" per link
- Main idea: append "attributes" to updates
- Can set preferences (or filter the route) based on set of attributes contained in update
  - Hard-coded "decision process" orders importance of attributes
  - This process can be influenced by changing values of attributes

Example: Using MED t… ingresses



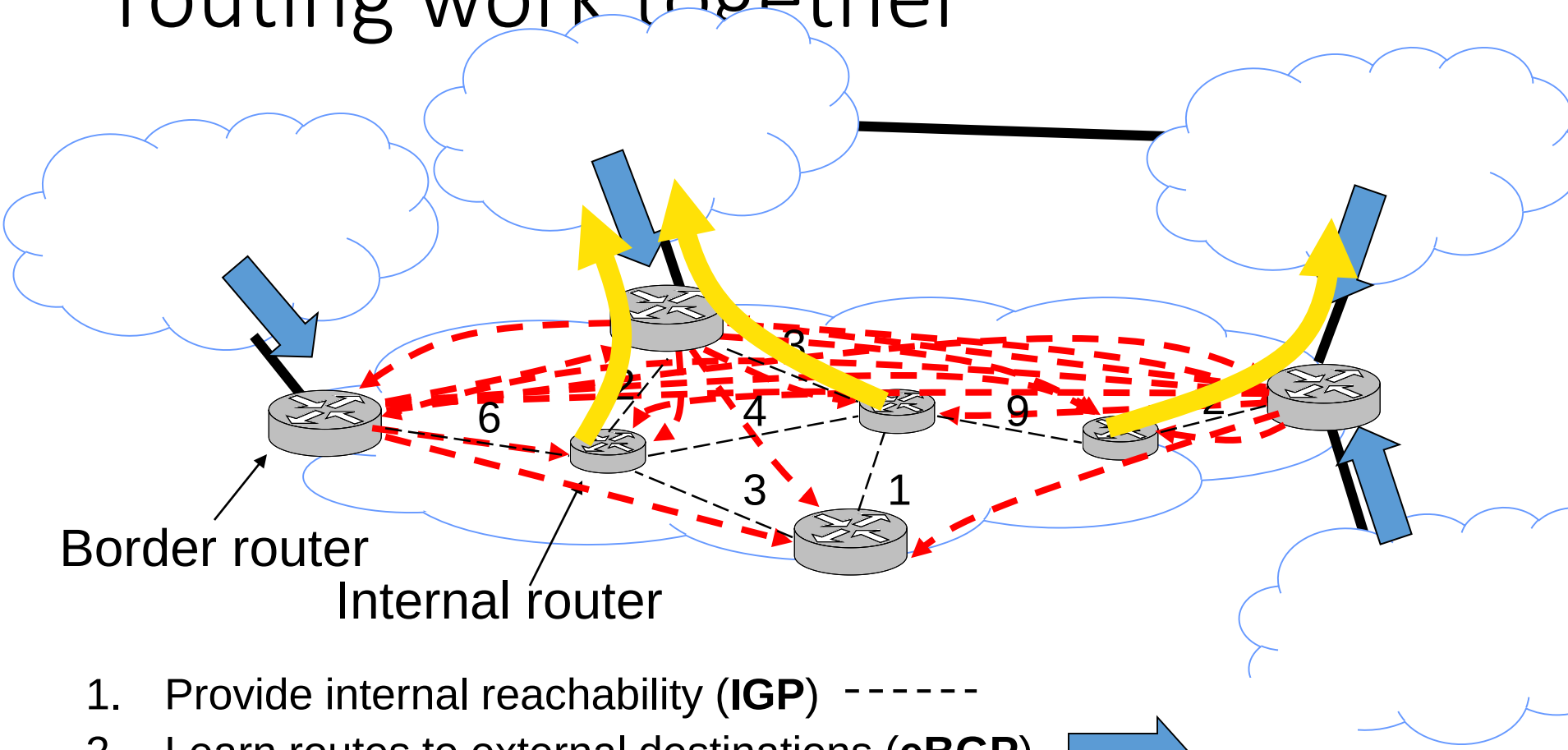
- MED: "multi-exit discriminator"
  - tell neighboring ISP which ingress peering points I prefer
  - Local ISP can choose to filter MED on import

Different peering po advertisements



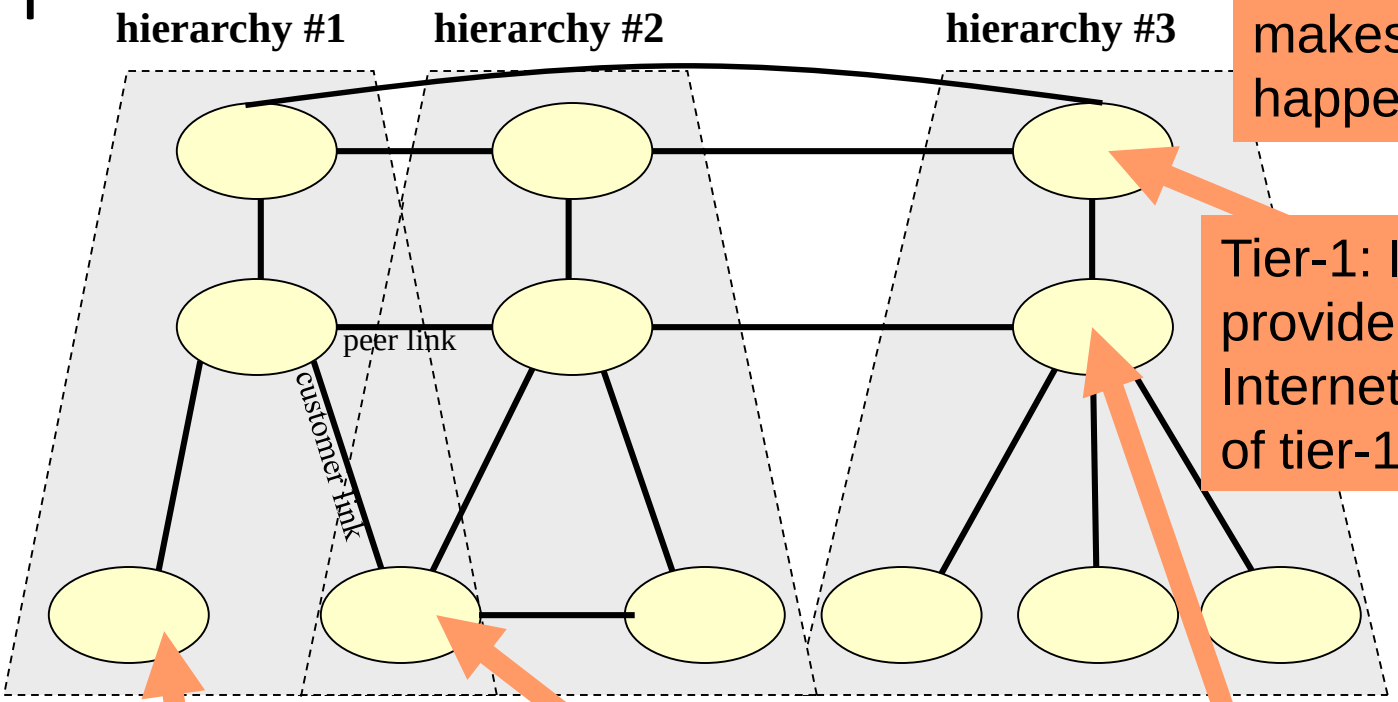
- Sprint can trick AT&T into routing over longer distance!
- Consistent export: make sure your neighbor is advertising the same set of prefixes at all peering points
- ISPs sometimes sign SLAs with consistent export clause

How inter- and intra- domain routing work together



1. Provide internal reachability (**IGP**) ------
2. Learn routes to external destinations (**eBGP**) ➡
3. Distribute externally learned routes internally (**iBGP**) - - ▸
4. Select closest egress (**IGP**) ------

Policies between ISPs: Types of ASes



Tier-1s must be connected in a full mesh (Why? Who makes sure that happens?)

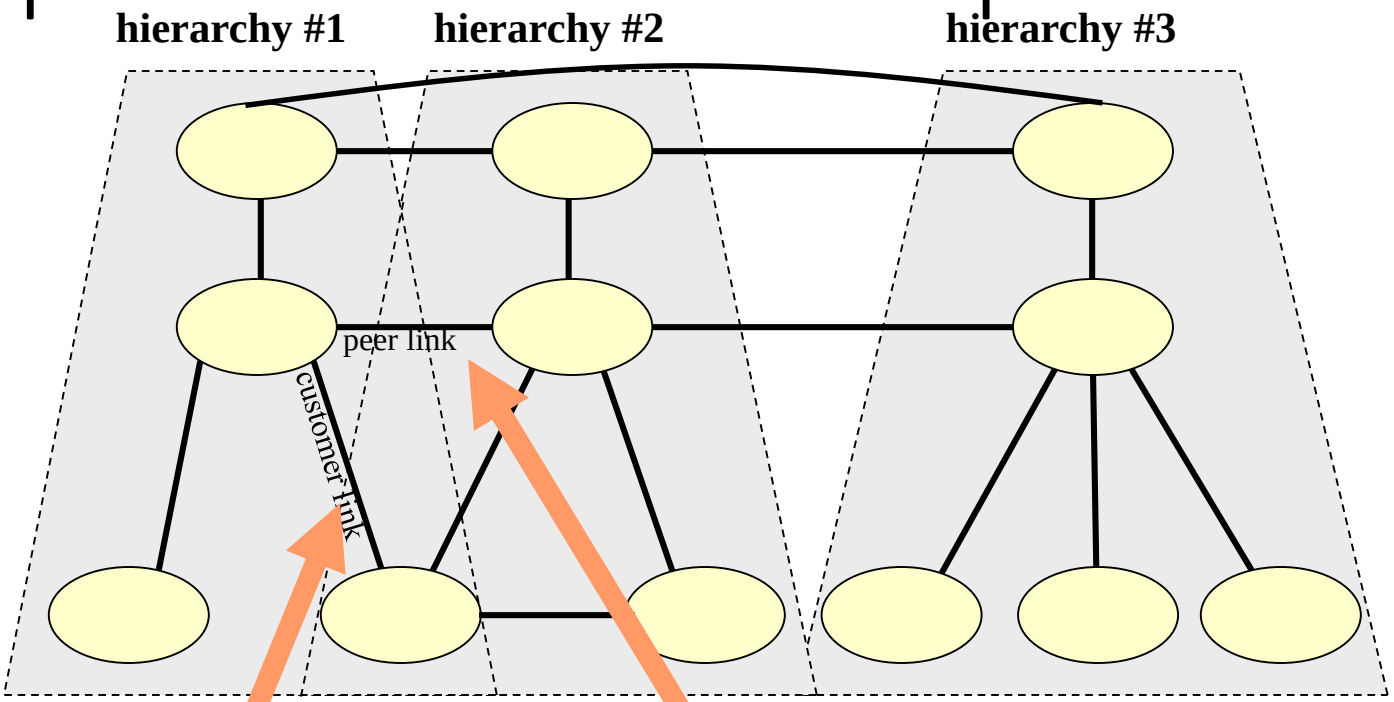
Tier-1: ISP with no providers (core of Internet is clique of tier-1s)

Stub: ISP with no customers

Multihomed: ISP with more than one provider

Transit: ISP that forward traffic between other ISPs

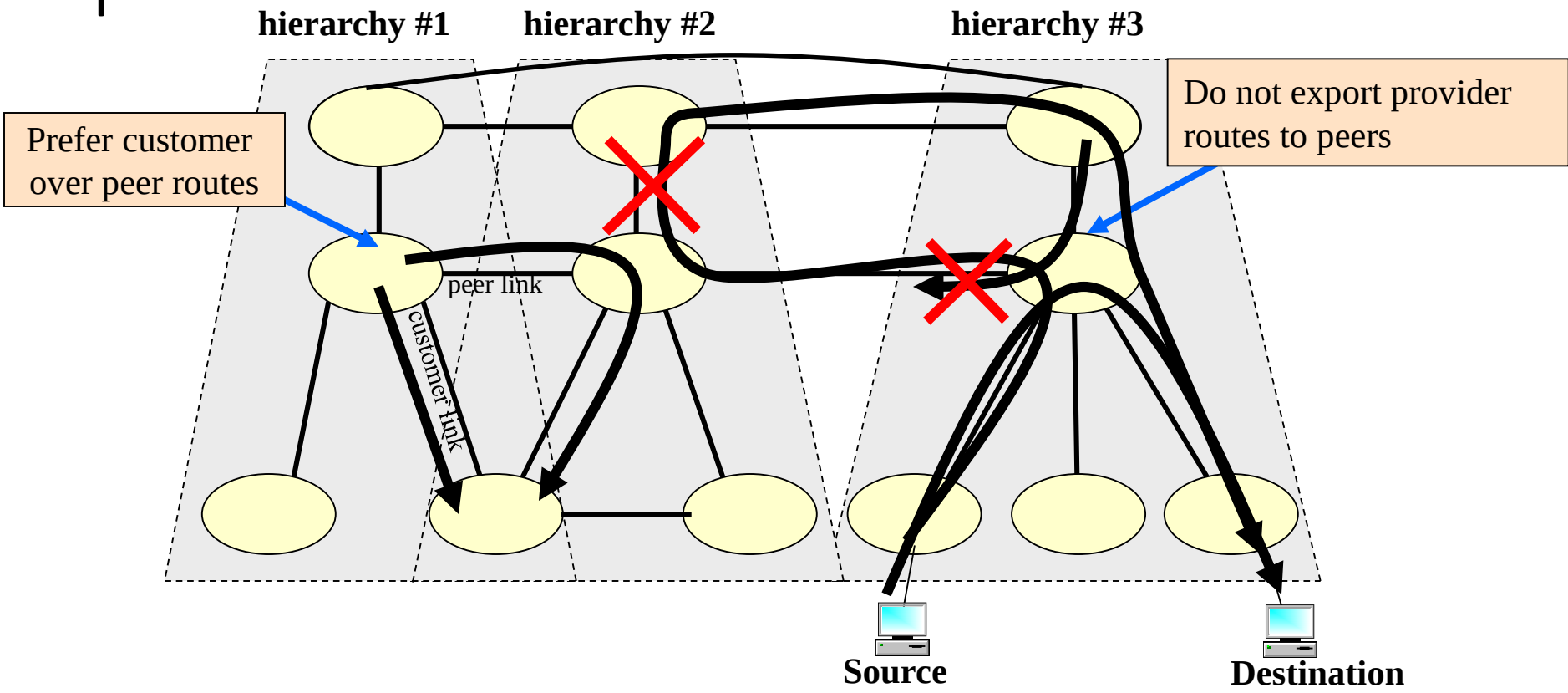
Policies between ISPs: Types of AS relationships



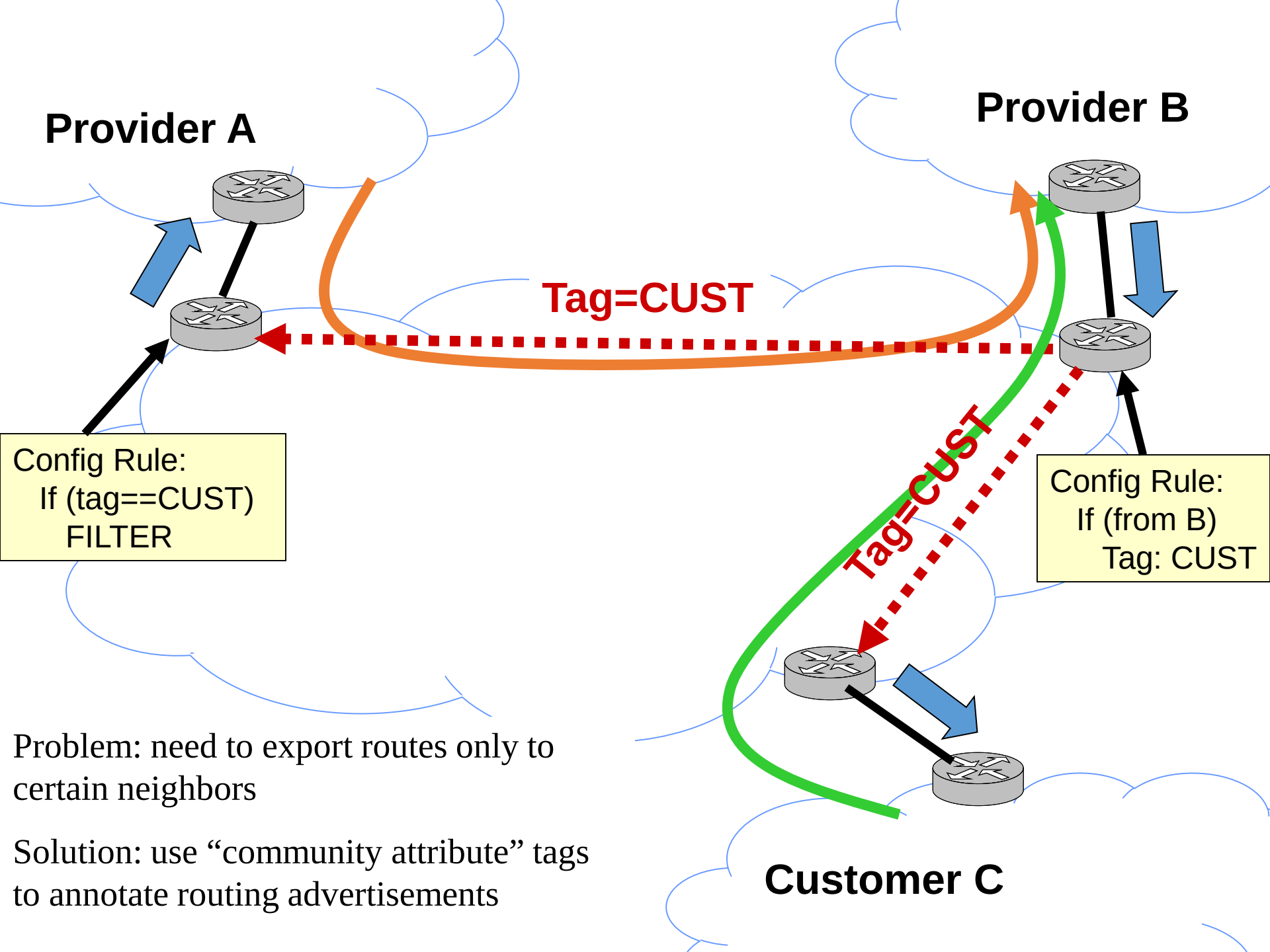
Provider-customer: customer pays provider money to transit traffic

Peer link: ISPs form link out of mutual benefit, typically no money is exchanged

AS relationships influence routing policies



- Example policies: peer, provider/customer
- Also trust issues, security, scalability, traffic engineering

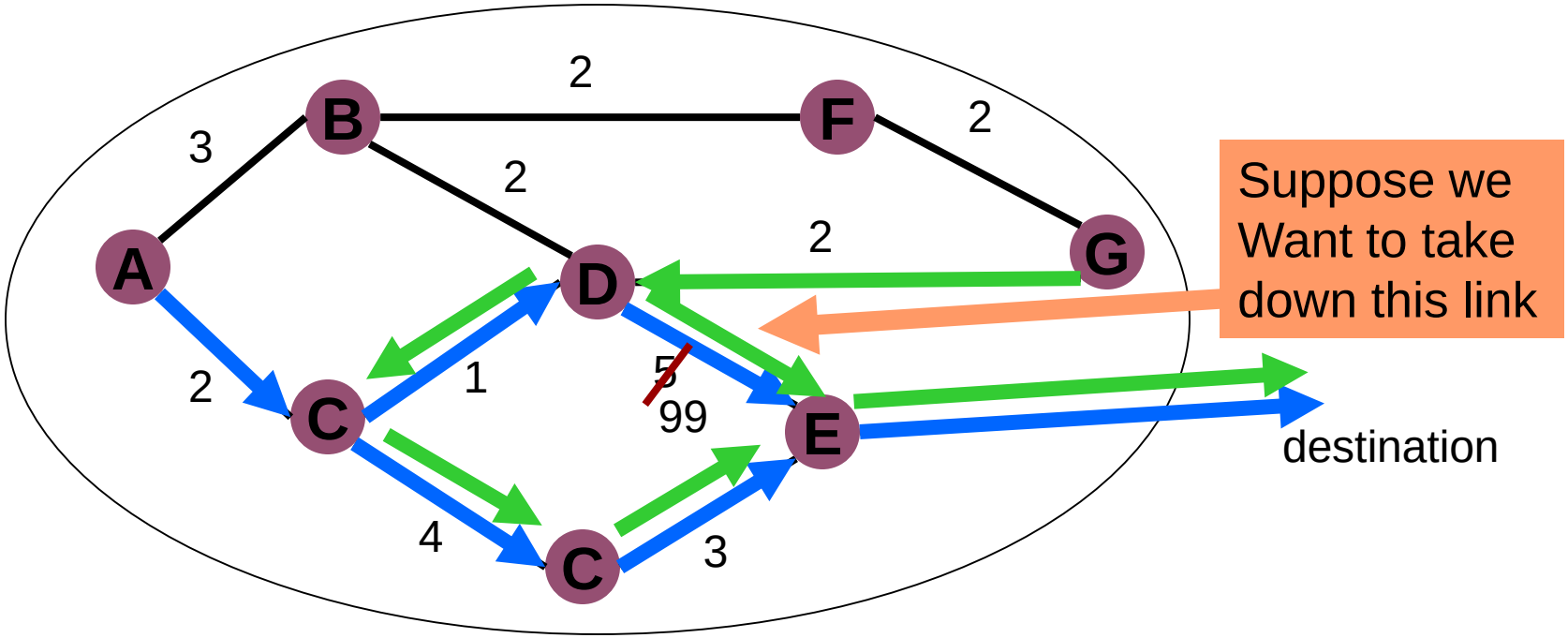


Problem: need to export routes only to certain neighbors

Solution: use “community attribute” tags to annotate routing advertisements

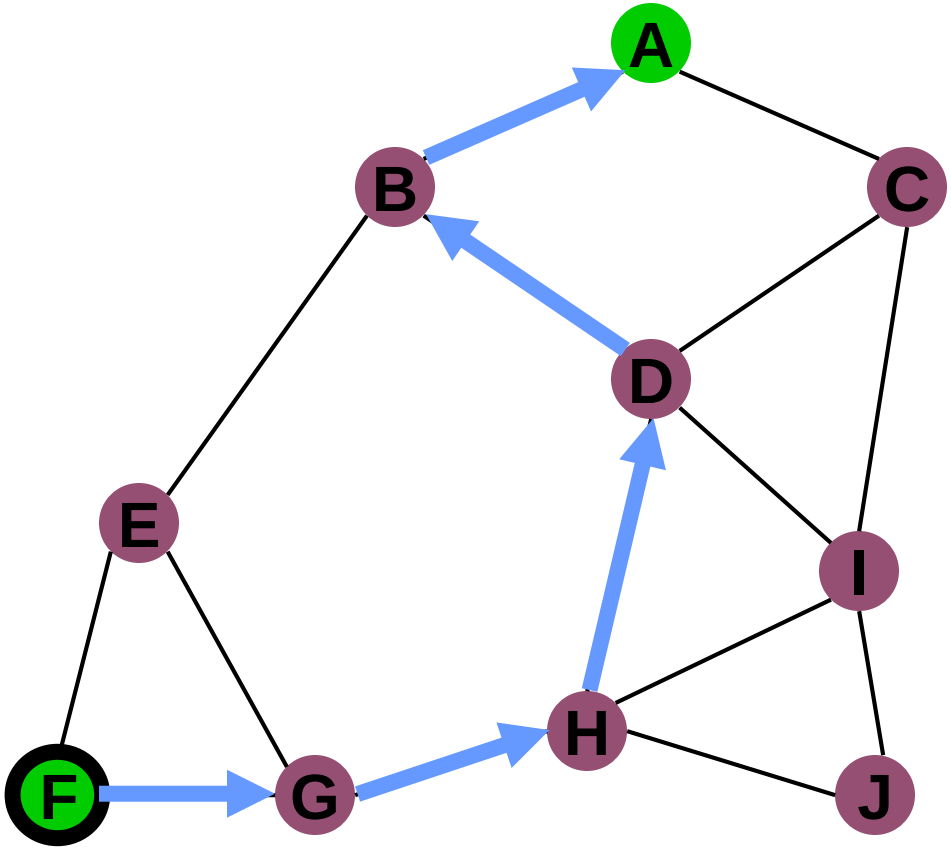
“Costing out” of equipment

- Increase cost of link to high value
  - Triggers immediate flooding of LSAs
- Leads to new shortest paths avoiding the link
  - While the link still exists to forward during convergence
- Then, can safely disconnect the link
  - New flooding of LSAs, but no influence on forwarding



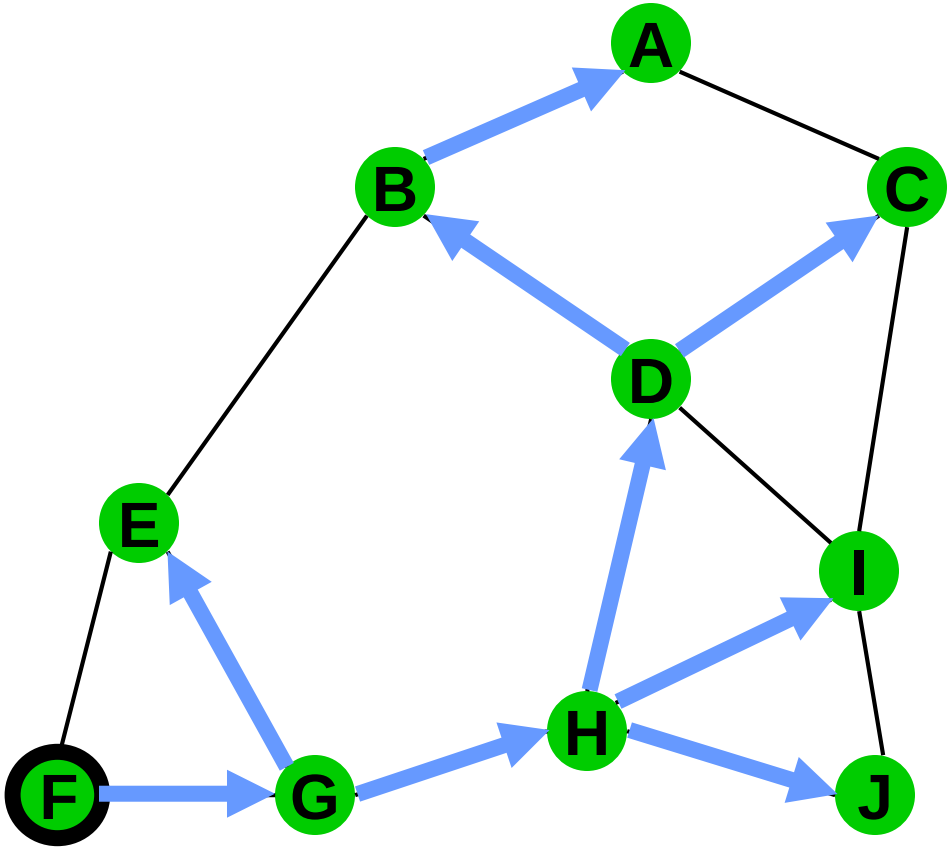
Delivery Models: Unicast vs Multicast

Delivery models



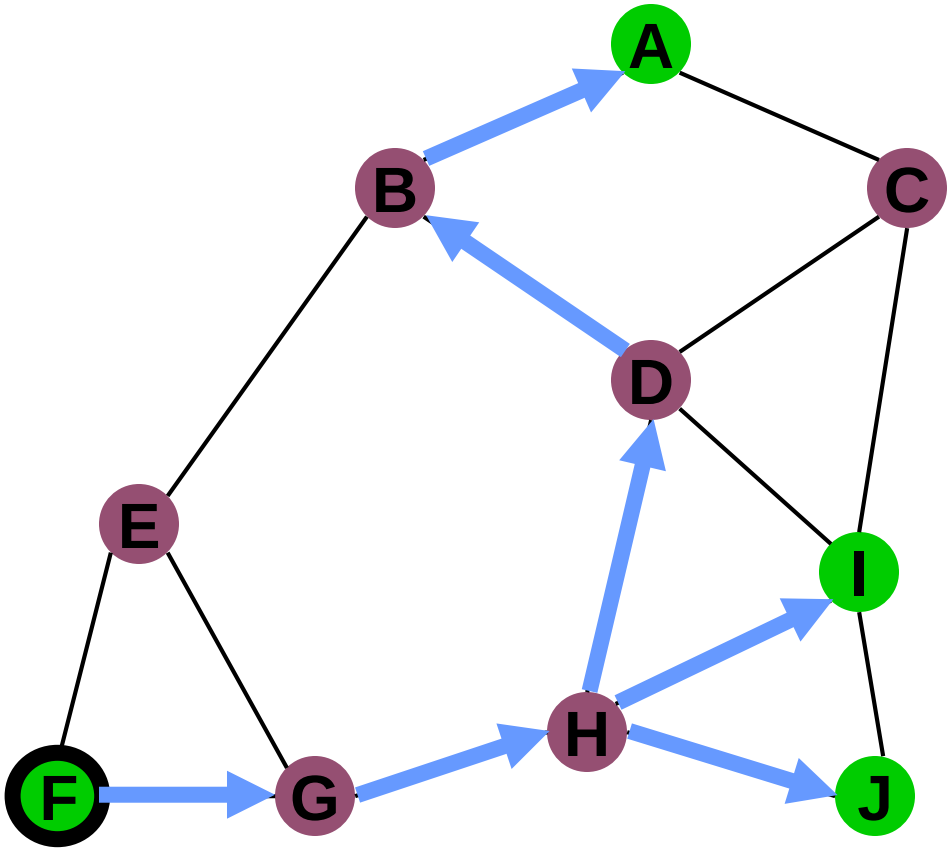
- Unicast
  - One source, one destination
  - Widely used (web, p2p, streaming, many other protocols)
- Broadcast
- Multicast
- Anycast

Delivery models



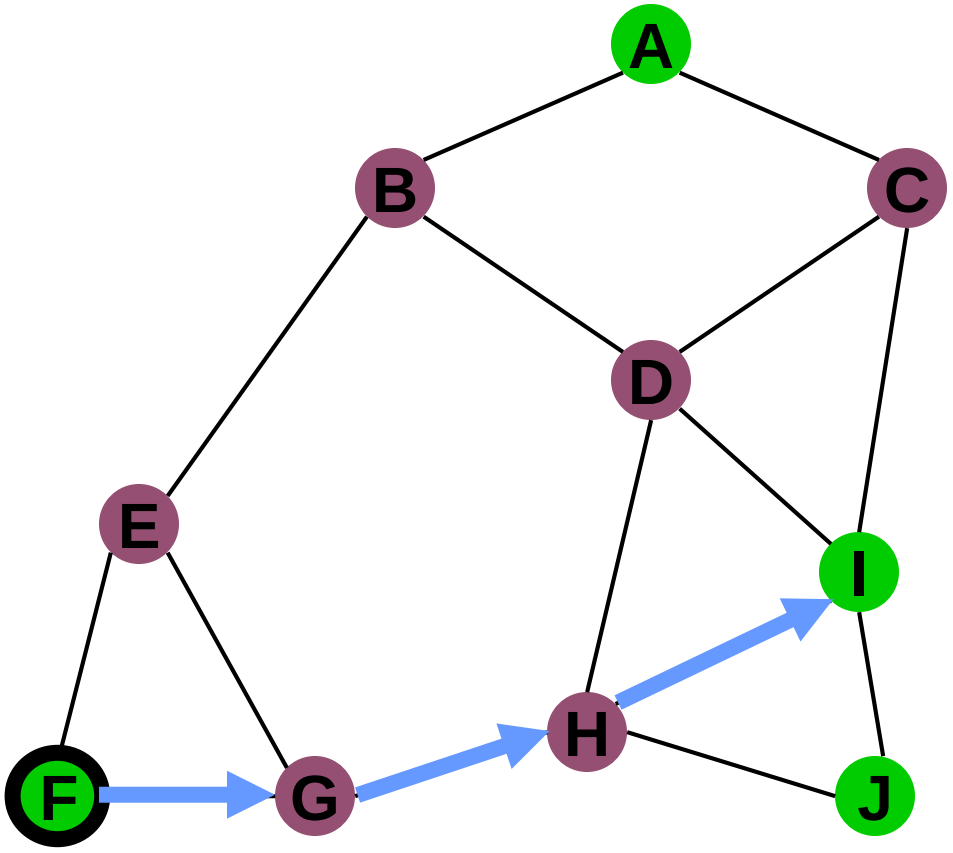
- Unicast
- Broadcast
  - One source, all destinations
  - Used to disseminate control information, perform service discovery
- Multicast
- Anycast

Delivery models



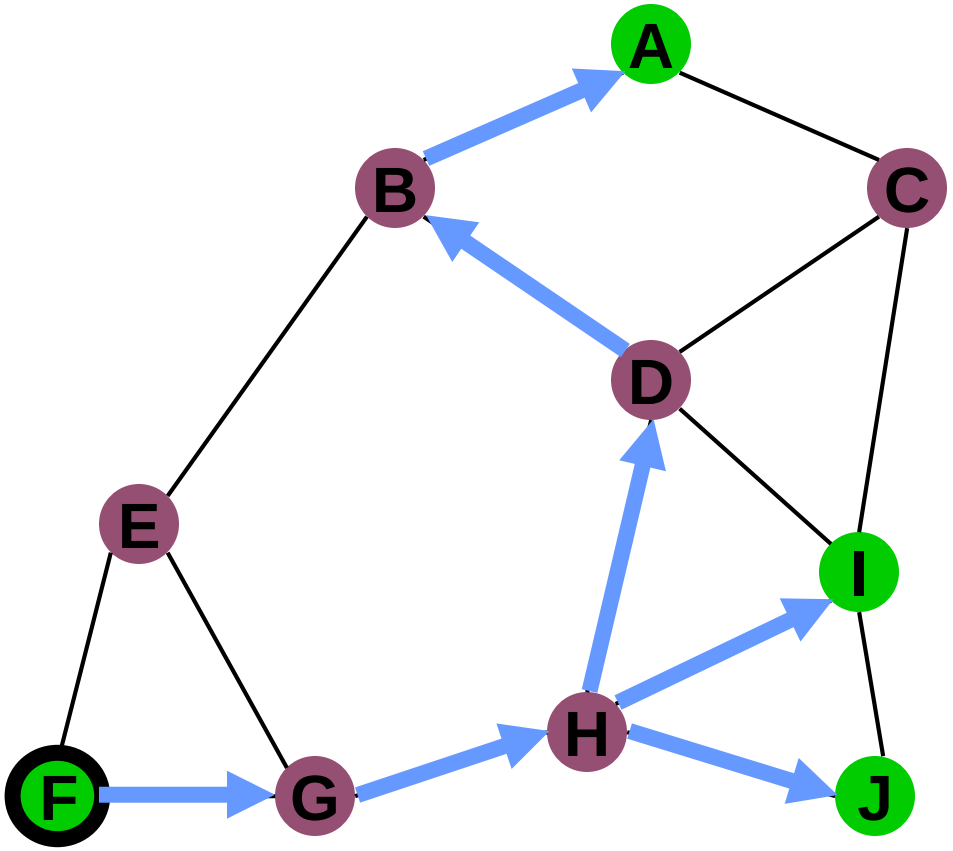
- Unicast
- Broadcast
- Multicast
  - One source, several (prespecified) destinations
  - Used within some ISP infrastructures for content delivery, overlay networks
- Anycast

Delivery models



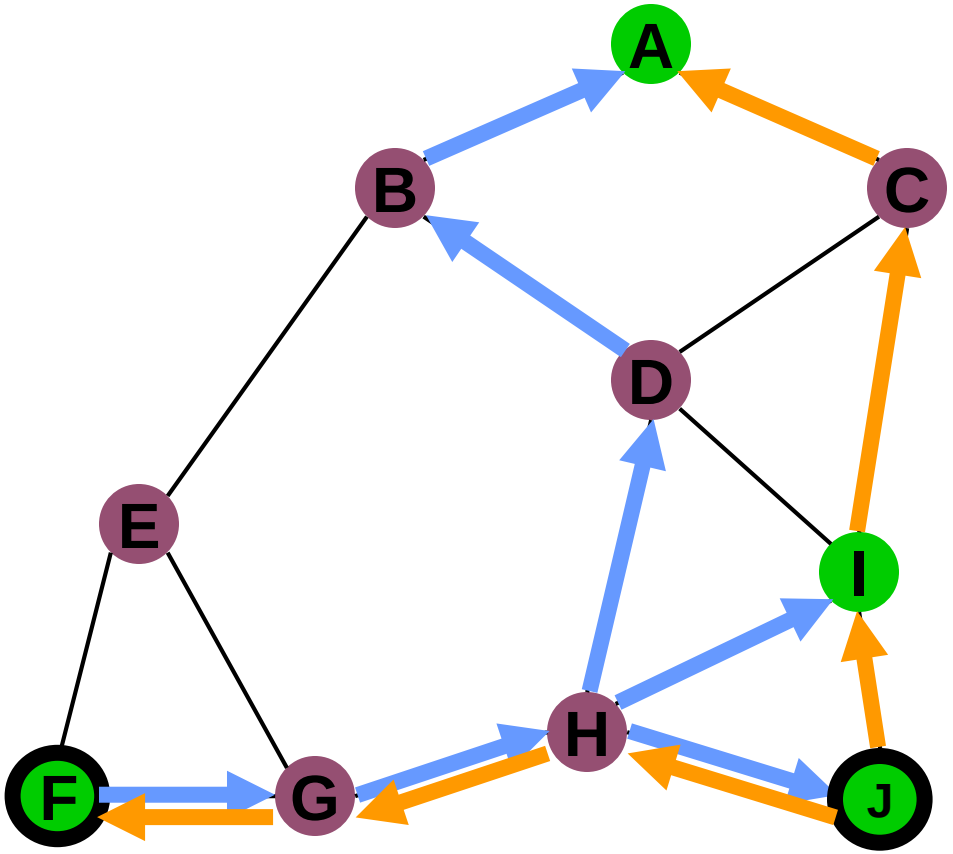
- Unicast
- Broadcast
- Multicast
- Anycast
  - One source, route to “best” destination
  - Used in DNS, content distribution

Source-specific trees



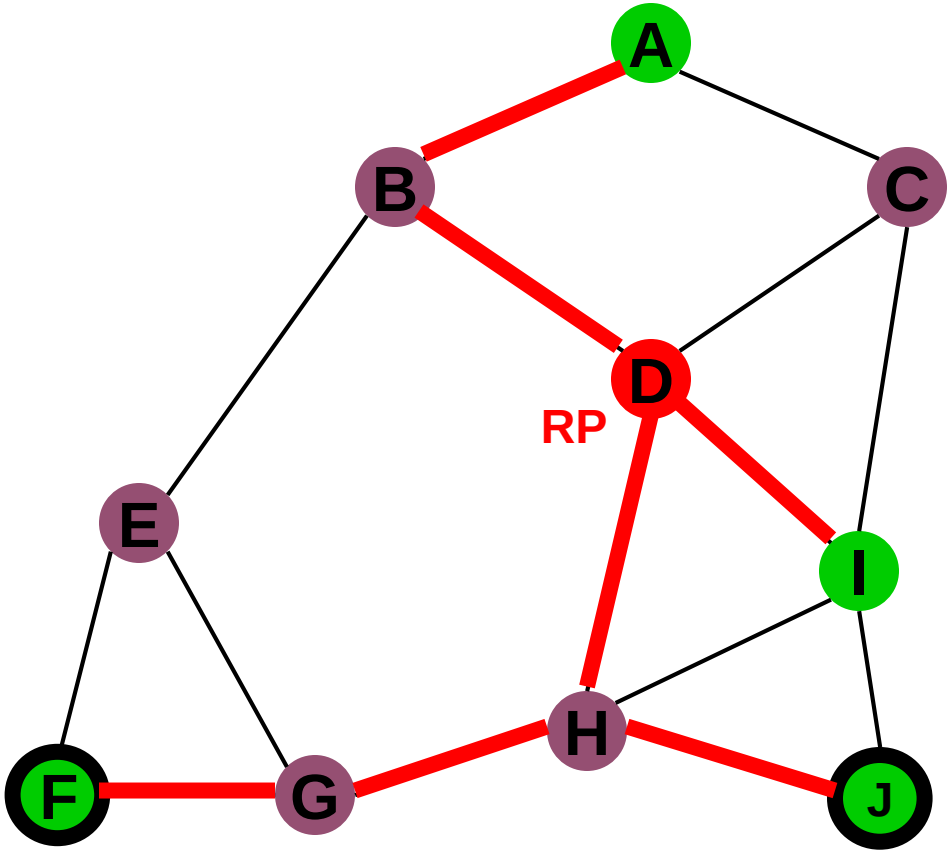
- Each source is the root of its own tree
- One tree per source
- Tree consists of shortest paths to each receiver

Source-specific trees



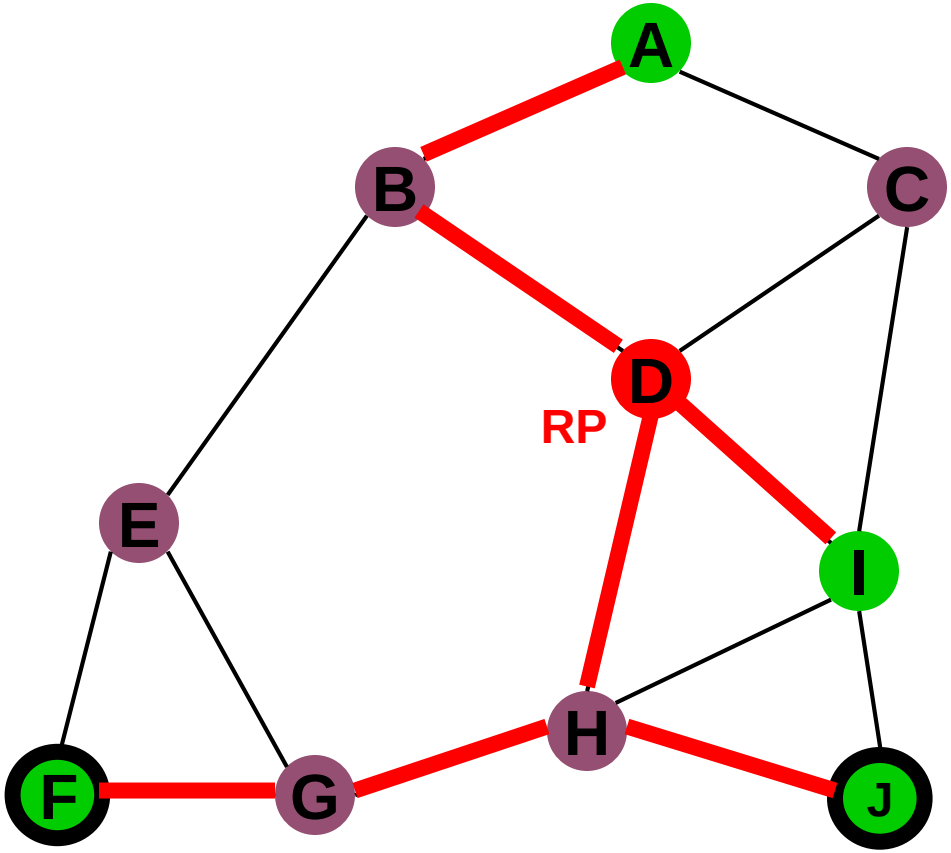
- Each source is the root of its own tree
- One tree per source
- Tree consists of shortest paths to each receiver

Shared Tree



- One tree used by all members of a group
- Rooted at “rendezvous point” (RP)
- Less state to maintain, but hard to pick a tree that’s “good” for everybody!

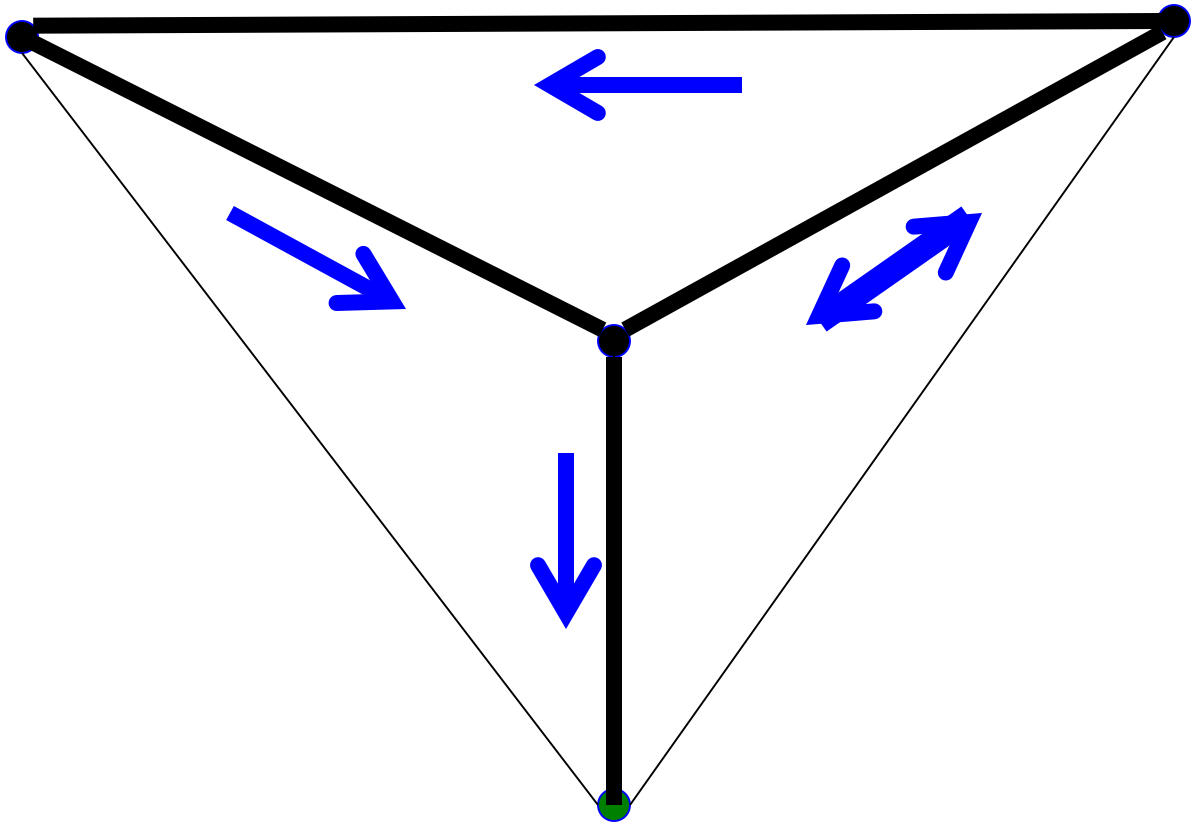
Shared Tree



- Ideally, find a "Steiner tree" minimum-weighted tree connecting **only** the multicast members
  - Unfortunately, this is NP-hard
- Instead, use heuristics
  - E.g., find a minimum spanning tree (much easier)

What Happens Here?

Problem: "cost" does not change around loop



Additive measures avoid this problem!

What Happens Here?