

EECS 489

Computer Networks

Winter 2025

Mosharaf Chowdhury

Material with thanks to Aditya Akella, Sugih Jamin, Philip Levis, Sylvia Ratnasamy, Peter Steenkiste, and many other colleagues.

Logistics

- You're ONLY allowed **two double-sided 8.5in x 11in note sheet**
 - Basic calculators are allowed but you won't need it
- General guidelines from the midterm still apply

Format

- ~75-minute in-person exam
 - Tuesday, Apr. 29 from 1:30PM - 3:30PM
 - »Room assignments: **TBA**
 - SSD accommodations
 - »Tuesday, Apr. 29 from 10:30AM at LBME 1130
 - Other conflicts
 - »**Email us ASAP** to potentially start earlier on Tuesday, Apr. 29 from 10:30AM at LBME 1130
 - »**MUST be confirmed first**

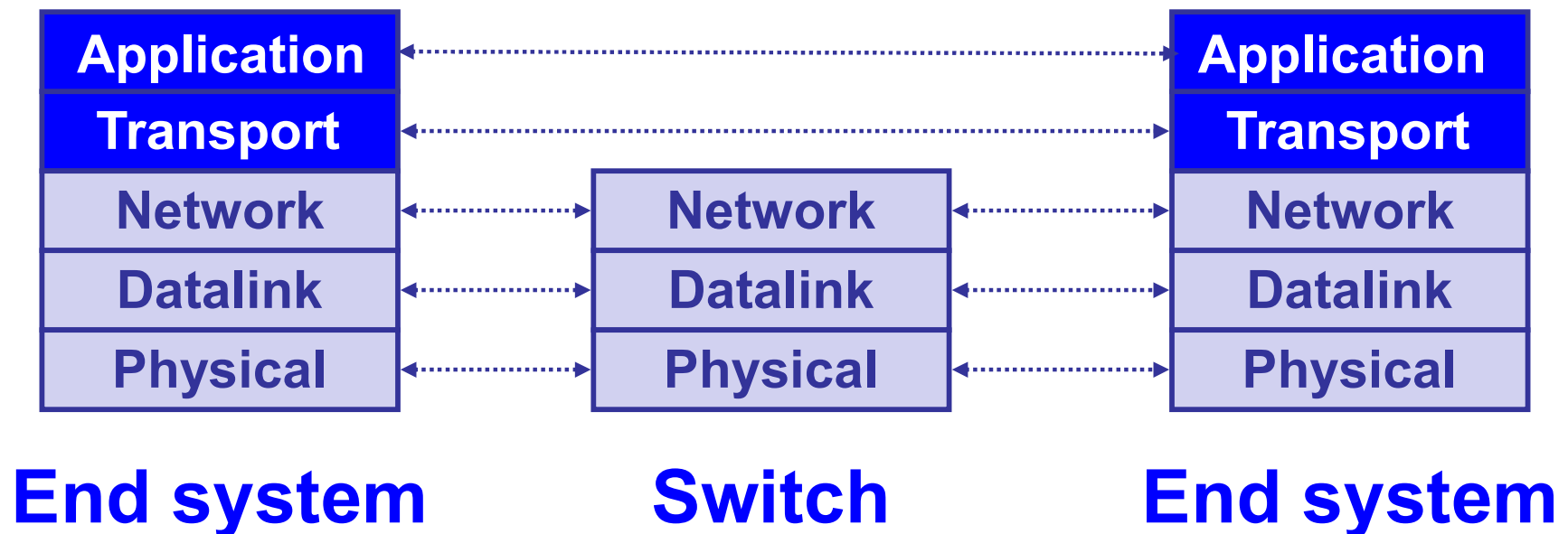
What's covered in the final?

- ▣ Lectures: 10—20
- ▣ Assignments: 2—4
- ▣ Discussion sections: 4—10

- ▣ What about content in the midterm? Is the final cumulative?
 - Since we are progressively going down the network stack, there will be questions that might require knowledge from the first half, but there will not be questions just on the first-half materials

The networking stack

- Lower three layers implemented everywhere
- Top two layers implemented only at hosts



Topics

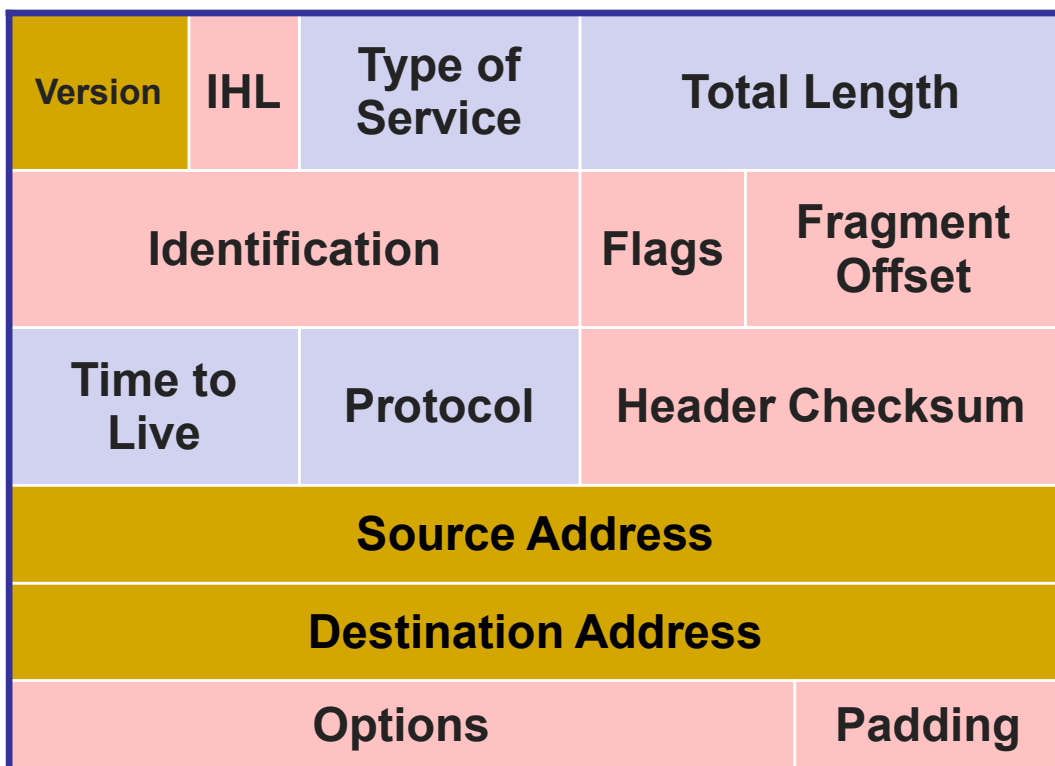
- Network layer (lectures 10–15)
 - Addressing
 - Data plane
 - Intra-domain routing
 - Inter-domain routing
- SDN and programmable networks (lecture 16)
- Link layer (lectures 17–19)
 - Ethernet
 - Wireless
- Datacenter networking (lecture 20)

What information do we need in the IP header?

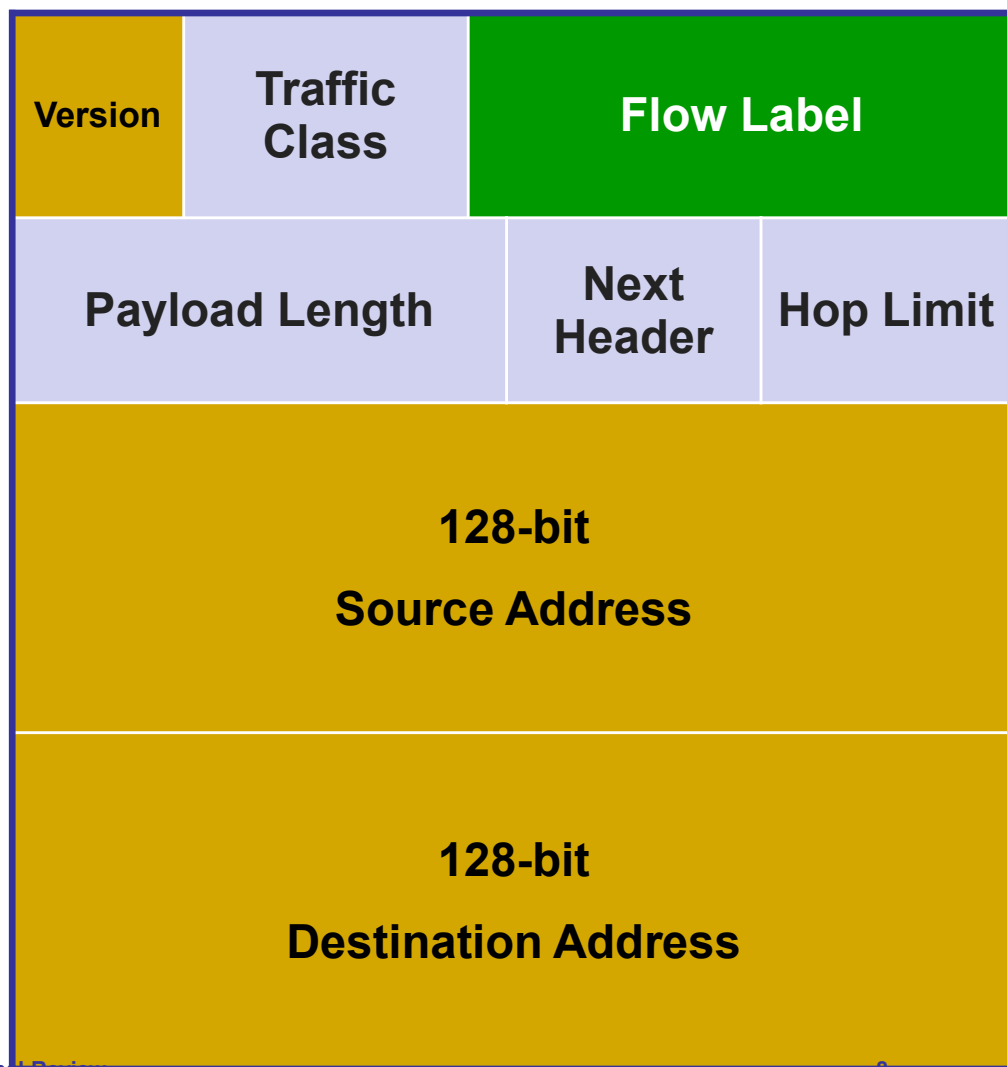
- ▣ Parse packet
 - IP version number (4 bits), packet length (16 bits)
- ▣ Carry packet to the destination
 - Destination's IP address (32 bits)
- ▣ Deal with problems along the way
 - Loops: TTL (8 bits)
 - Corruption: checksum (16 bits)
 - Packet too large: fragmentation fields (32 bits)

IPv4 and IPv6 header comparison

IPv4

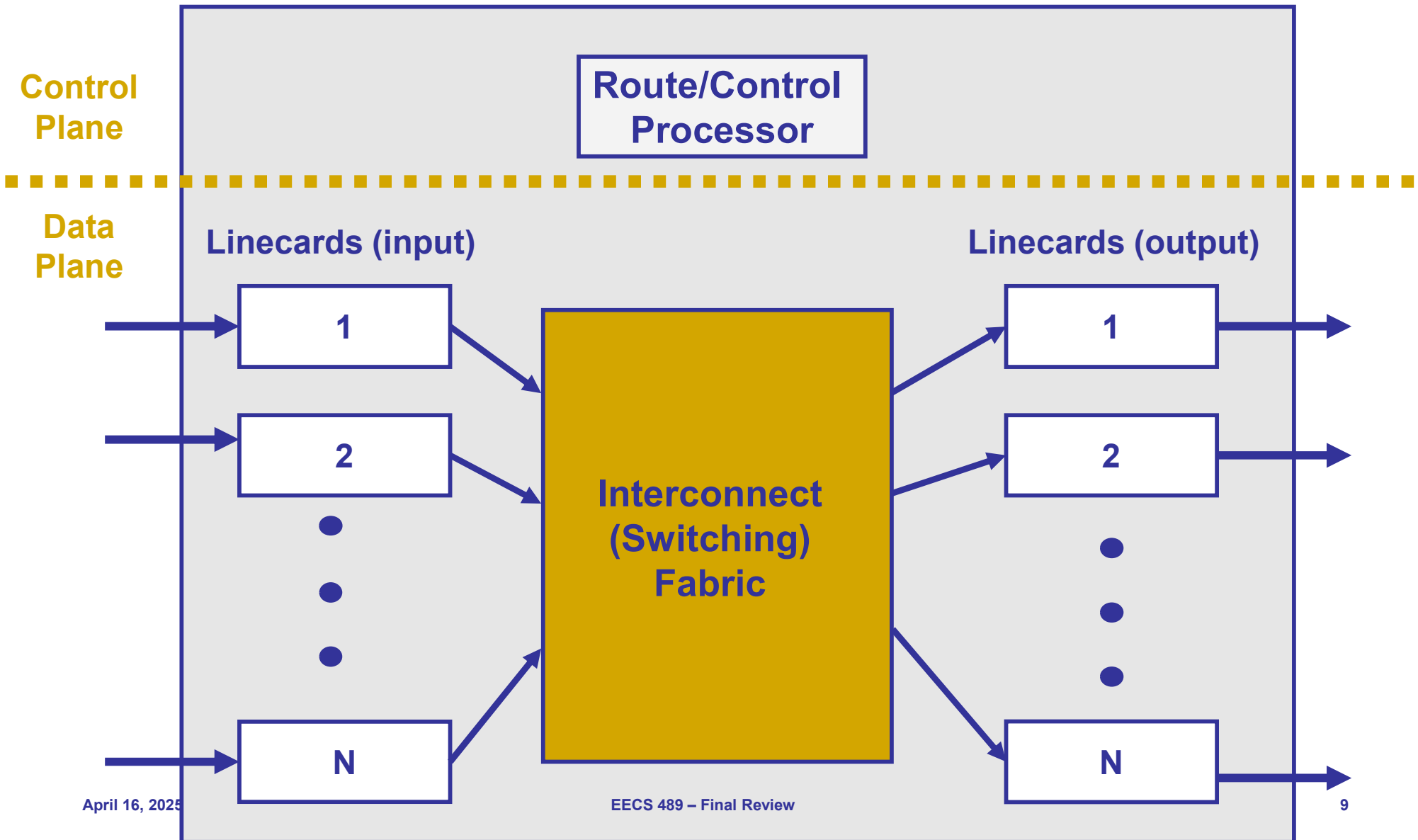


IPv6



- Field name kept from IPv4 to IPv6
- Fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6

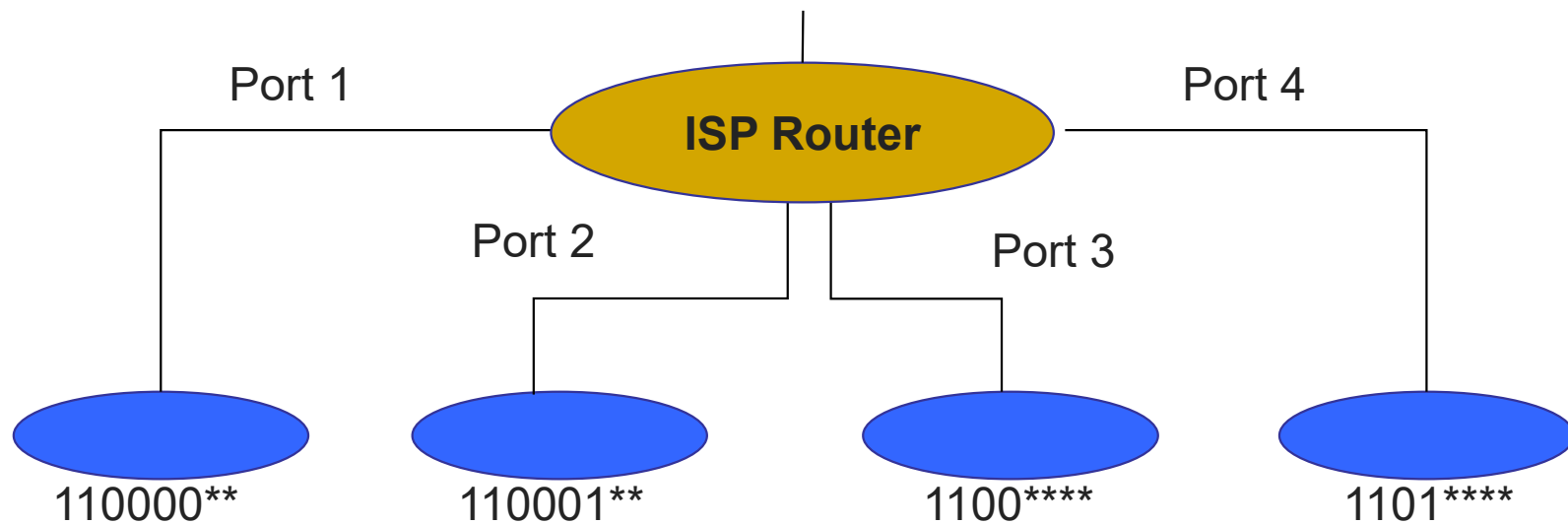
What's inside a router?



Input linecards

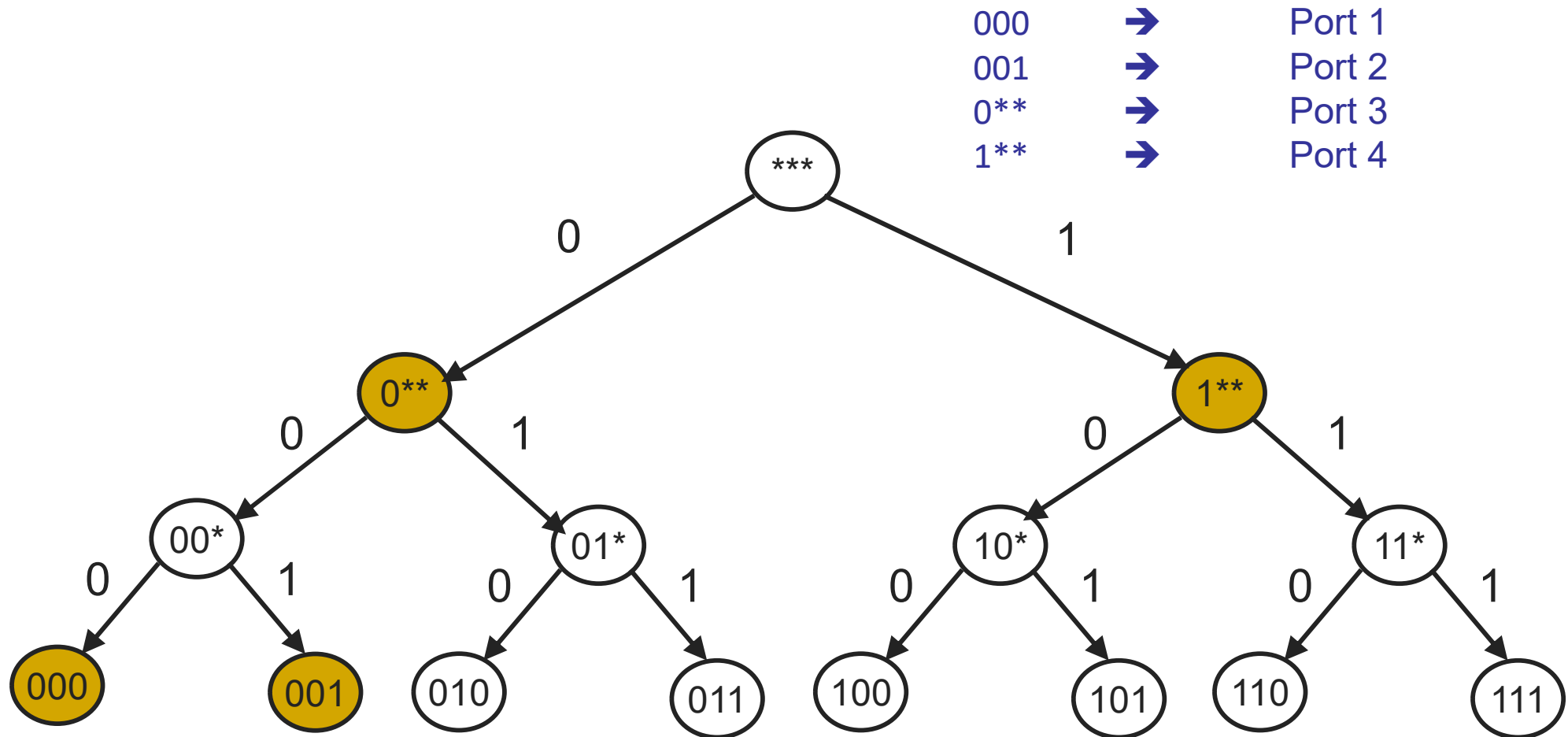
- Main challenge is processing speeds
- Tasks involved:
 - Update packet header (easy)
 - LPM lookup on destination address (harder)
- Mostly implemented with specialized hardware

Longest prefix matching



Send to the port with the longest prefix match

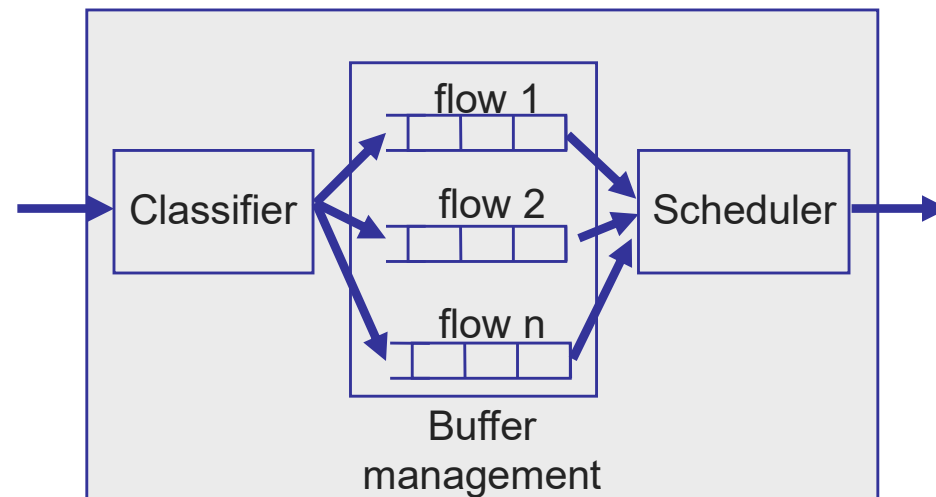
Tree structure



Record port associated with latest match, and only override when it matches another prefix during walk down tree

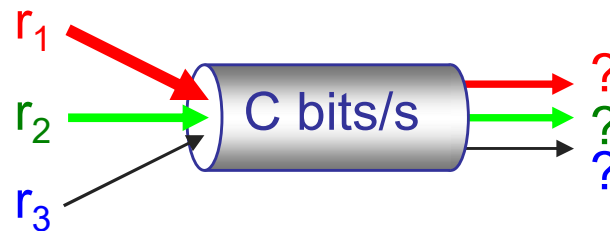
Output linecards

- ❑ **Packet classification**: map packets to flows
- ❑ **Buffer management**: decide when and which packet to drop
- ❑ **Scheduler**: decide when and which packet to transmit



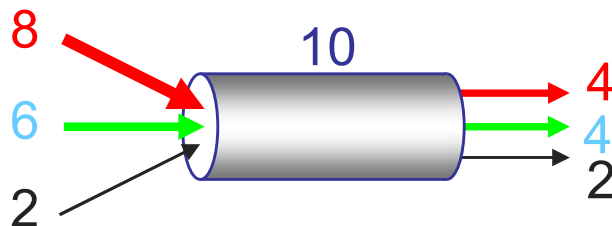
Max-Min fairness

- Given set of bandwidth demands r_i and total bandwidth C , max-min bandwidth allocations are:
 - $a_i = \min(f, r_i)$
 - where f is the unique value such that $\text{Sum}(a_i) = C$



Example

- $C = 10; r_1 = 8, r_2 = 6, r_3 = 2; N = 3$
- $C/3 = 3.33 \rightarrow$
 - r_3 needs only 2
 - » Can service all of r_3
 - Remove r_3 from the accounting: $C = C - r_3 = 8; N = 2$
- $C/2 = 4 \rightarrow$
 - Can't service all of r_1 or r_2
 - So hold them to the remaining fair share: $f = 4$



$$\begin{aligned} f &= 4: \\ \min(8, 4) &= 4 \\ \min(6, 4) &= 4 \\ \min(2, 4) &= 2 \end{aligned}$$

Max-Min fairness

- Given set of bandwidth demands r_i and total bandwidth C , max-min bandwidth allocations are:
 - $a_i = \min(f, r_i)$
 - where f is the unique value such that $\text{Sum}(a_i) = C$
- If you don't get full demand, no one gets more than you
- This is what round-robin service gives if all packets are the same size

Topics

- Network layer (lectures 10–15)
 - Addressing
 - Data plane
 - Intra-domain routing
 - Inter-domain routing
- SDN and programmable networks (lecture 16)
- Link layer (lectures 17–19)
 - Ethernet
 - Wireless
- Datacenter networking (lecture 20)

Routing: Local vs. global view

- *Local* routing state is the forwarding table in a single router
 - By itself, the state in a single router cannot be evaluated
 - It must be evaluated in terms of the global context
- *Global* state refers to the collection of forwarding tables in each of the routers
 - Global state determines which paths packets take

“Valid” routing state

- Global state is “valid” if it produces forwarding decisions that always deliver packets to their destinations
- Goal of routing protocols: compute valid state
 - How can we tell if routing state is valid?

Necessary and sufficient condition

- ❑ Global routing state is valid *if and only if*:
 - There are no dead ends (other than destination)
 - There are no loops
- ❑ A **dead end** is when there is no outgoing link (next-hop)
 - A packet arrives, but the forwarding decision does not yield any outgoing link
- ❑ A **loop** is when a packet cycles around the same set of nodes forever

Least-cost routes

- ❑ Least-cost routes provide an easy way to avoid loops
 - No reasonable cost metric is minimized by traversing a loop
- ❑ Least-cost paths form a spanning tree for each destination rooted at that destination

Intra-domain routing

- Link-state (LS) routing protocol
 - Dijkstra's algorithm
 - Broadcast neighbors' info to everyone
- Distance vector (DV) routing protocol
 - Bellman-Ford algorithm
 - Gossip to neighbors about everyone

Similarities between LS and DV routing

- Both are shortest-path based routing
 - Minimizing cost metric (link weights) a common optimization goal
 - » Routers share a common view as to what makes a path “good” and how to measure the “goodness” of a path
- Due to shared goal, commonly used inside an organization
 - RIP and OSPF are mostly used for **intra**-domain routing

Comparison of LS and DV routing

Messaging complexity

- LS: with N nodes, E links, $O(NE)$ messages sent
- DV: exchange between neighbors only

Speed of convergence

- LS: relatively fast
- DV: convergence time varies
 - Count-to-infinity** problem

Robustness: what happens if router malfunctions?

- LS:
 - Node can advertise incorrect **link** cost
 - Each node computes its *own* table
- DV:
 - Node can advertise incorrect **path** cost
 - Each node's table used by others (**errors propagate**)

Addressing is key to scalable inter-domain routing

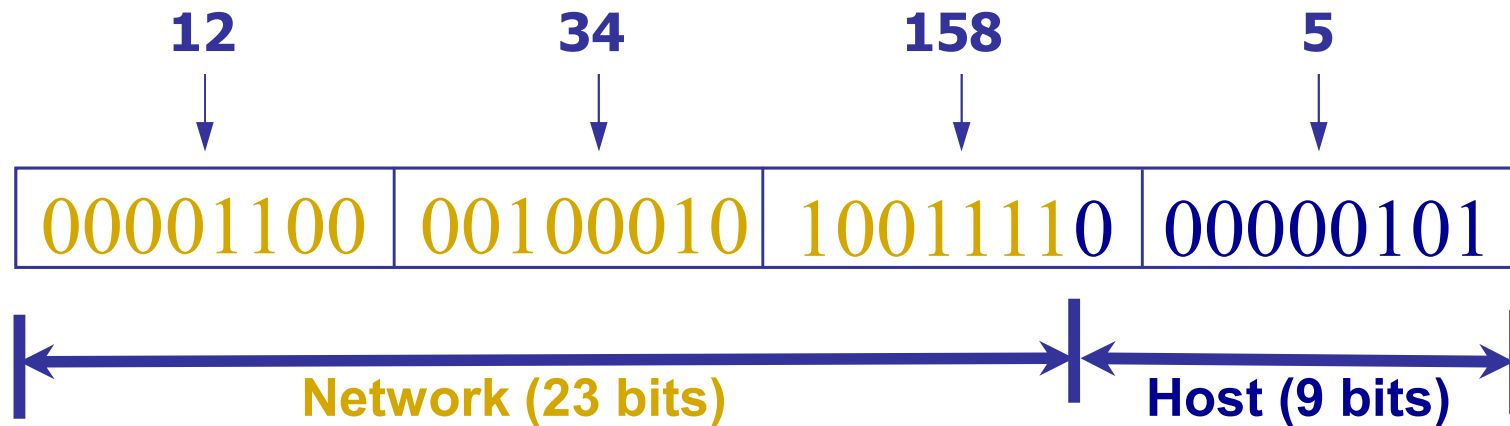
- ▣ Ability to aggregate addresses is crucial for
 - State: Small forwarding tables at routers
 - » Much less than the number of hosts
 - Churn: Limited rate of change in routing tables

CIDR: Classless inter-domain routing

- Flexible division between network and host addresses
- Offers a better tradeoff between size of the routing table and efficient use of the IP address space

Hierarchy in IP addressing

- 32 bits are partitioned into a prefix and suffix components
- Prefix is the **network** component; suffix is the **host** component

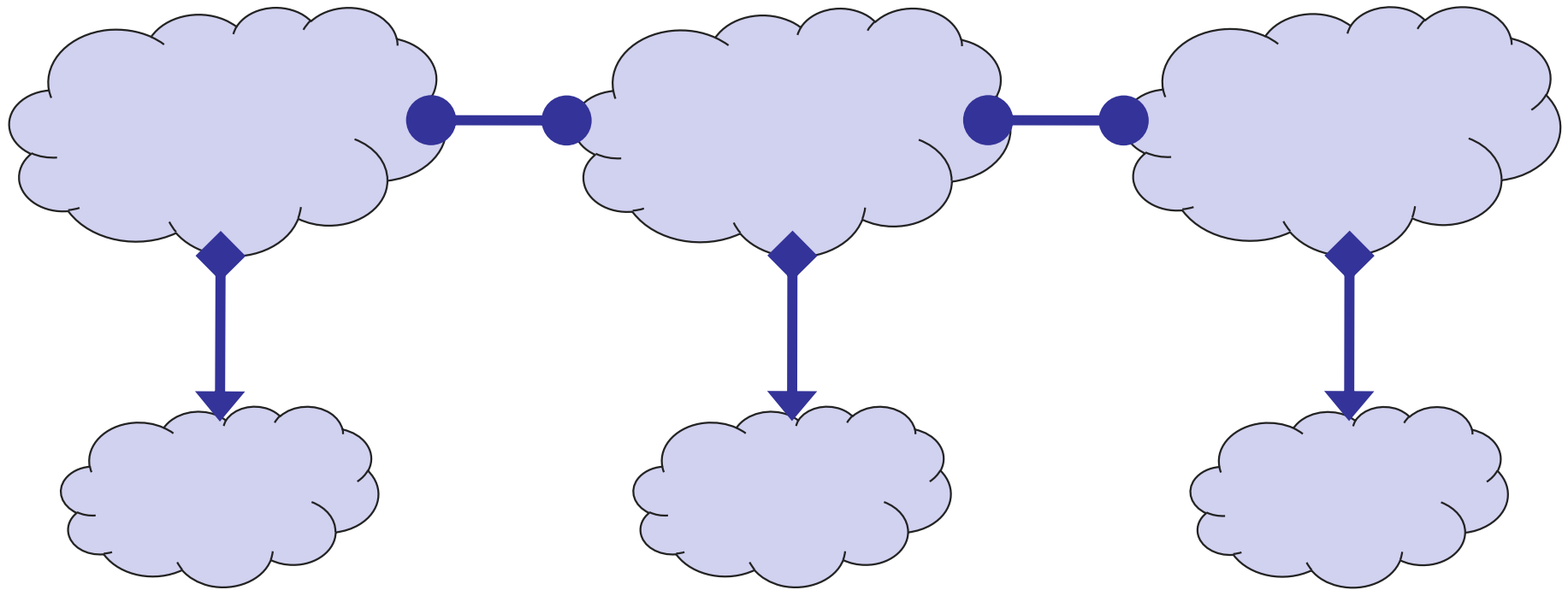


- Inter-domain routing operates on network prefix

Administrative structure shapes Inter-domain routing

- ▣ ASes want freedom to pick routes based on policy
- ▣ ASes want autonomy
- ▣ ASes want privacy

Business relationships



Relations between ASes

provider \longleftrightarrow customer
peer $\bullet\text{---}\bullet$ peer

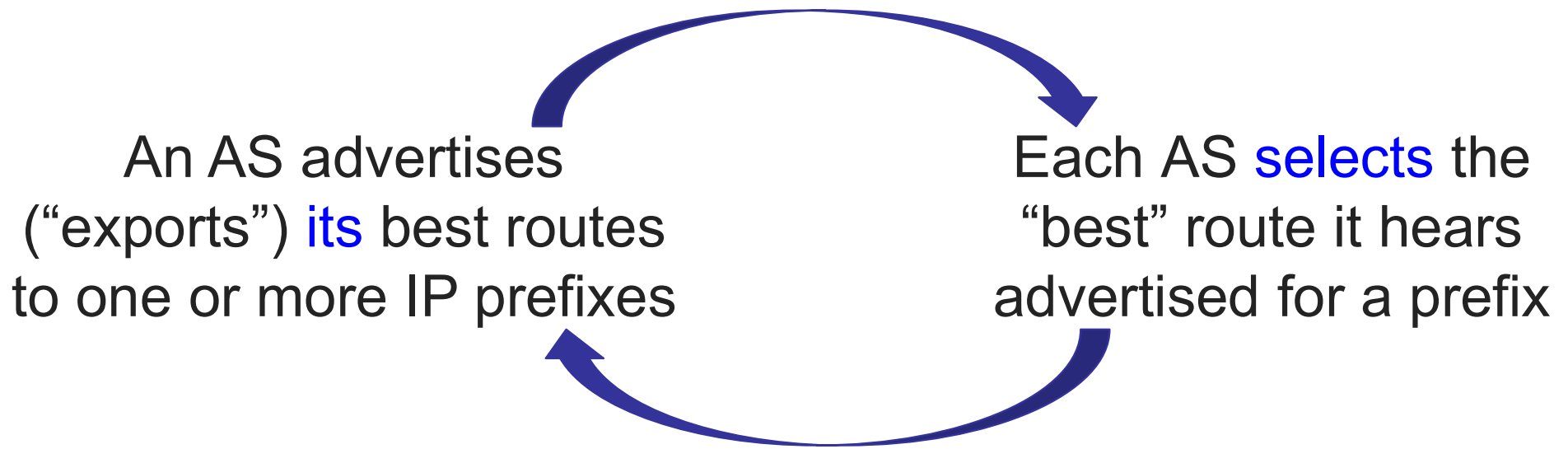
Business implications

- Customers pay provider
- Peers don't pay each other

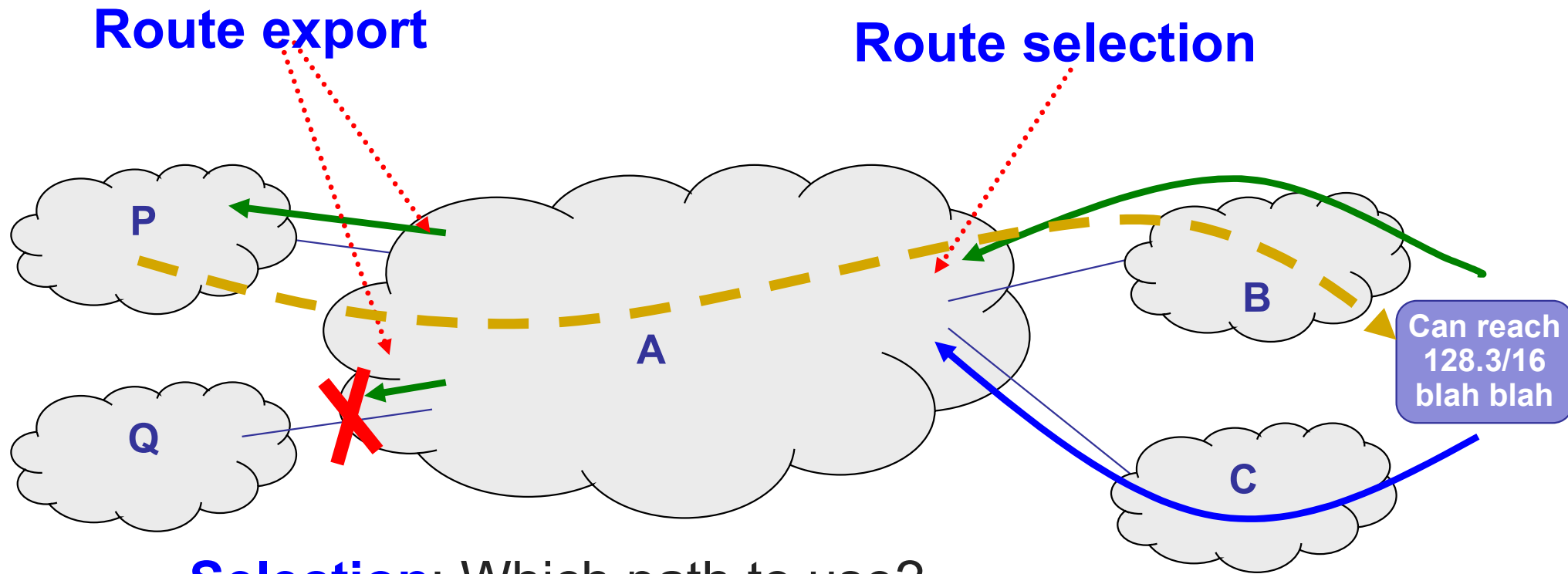
BGP inspired by Distance-Vector with four differences

- ❑ Shortest-path routes may not be picked to enforce policy
- ❑ Path-Vector routing to avoid loops
- ❑ Selective route advertisement may affect reachability
- ❑ Routes may be aggregated for scalability

BGP: Basic idea



Policy dictates how routes are “selected” and “exported”



- ❑ **Selection:** Which path to use?
 - Controls whether/how traffic leaves the network
- ❑ **Export:** Which path to advertise?
 - Controls whether/how traffic enters the network

Typical export policy

Destination prefix advertised by...	Export route to...
Customer	Everyone (providers, peers, other customers)
Peer	Customers
Provider	Customers

We'll refer to these as the “Gao-Rexford” rules (capture common – **but not required!** – practice)

Selection using attributes

- Rules for route selection in priority order

Priority	Rule	Remarks
1	LOCAL PREF	Pick highest LOCAL PREF
2	ASPATH	Pick shortest ASPATH length
3	MED	Lowest MED preferred
4	eBGP > iBGP	Did AS learn route via eBGP (preferred) or iBGP?
5	iBGP path	Lowest IGP cost to next hop (egress router)
6	Router ID	Smallest next-hop router's IP address as tie-breaker

eBGP, iBGP, and IGP

- ❑ **eBGP**: BGP sessions between border routers in different ASes
 - Learn routes to external destinations
- ❑ **iBGP**: BGP sessions between border routers and other routers within the same AS
 - Distribute externally learned routes internally
- ❑ **IGP**: “Interior Gateway Protocol” = Intra-domain routing protocol
 - Provide internal reachability via shortest path
 - E.g., OSPF, RIP

5-MINUTE BREAK!

Announcements

- ❑ Teaching evaluations
 - Due by **Apr 23**
 - 75% or higher completion rate will result in **+1 on the final grade for everyone** (we are around **10%**)
- ❑ By popular request, **additional sample final exams** will be released this week!
- ❑ NO office hours for me this week (SOSP)

- ❑ Attend **Vint Cerf** talk this Friday, 4/18 (130PM)
 - Chesebrough Auditorium, Chrysler Center

Topics

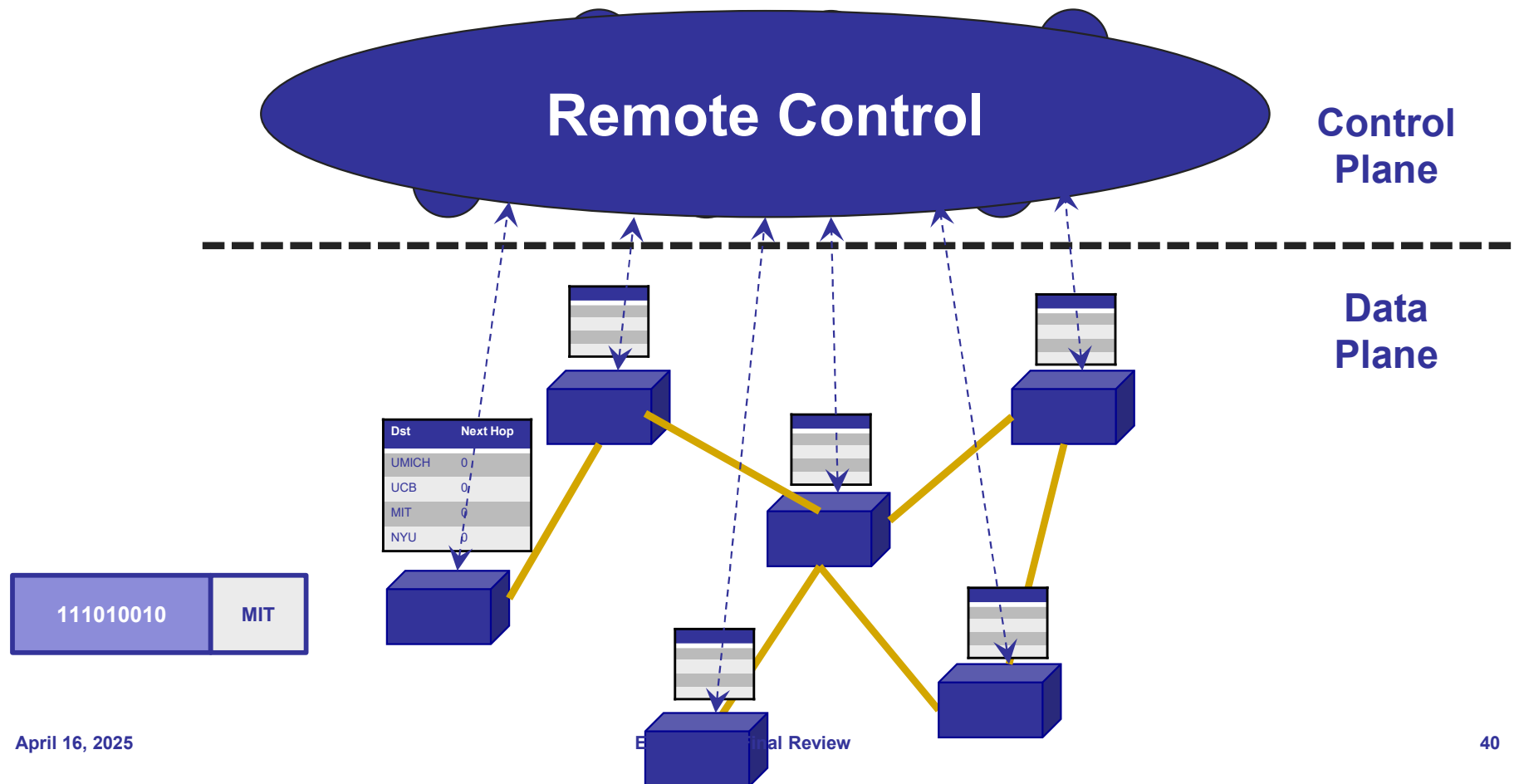
- Network layer (lectures 10–15)
 - Addressing
 - Data plane
 - Intra-domain routing
 - Inter-domain routing
- SDN and programmable networks (lecture 16)
- Link layer (lectures 17–19)
 - Ethernet
 - Wireless
- Datacenter networking (lecture 20)

Separate concerns with abstractions

- Be compatible with low-level hardware/software
 - Need an abstraction for general forwarding model
- Make decisions based on entire network
 - Need an abstraction for network state
- Compute configuration of each physical device
 - Need an abstraction that simplifies configuration

Logically centralized control plane

- A distinct (typically remote) controller interacts with local control agents (CAs)



SDN: Many challenges remain

- ❑ Hardening the control plane: dependable, reliable, performance-scalable, secure distributed system
 - **Robustness to failures**: leverage strong theory of reliable distributed system for control plane
 - **Security**: “baked in” from day one?
- ❑ Networks, protocols meeting mission-specific requirements
 - E.g., real-time, ultra-reliable, ultra-secure
- ❑ **Internet-scaling**

Fixed-function data plane

- ❑ Traditional switches are fixed-function
 - They can do whatever they can do at birth, but they cannot change!
 - Bottom-up design
- ❑ Even OpenFlow was designed to be a fixed protocol
 - With a fixed table format
 - Capable of doing limited things

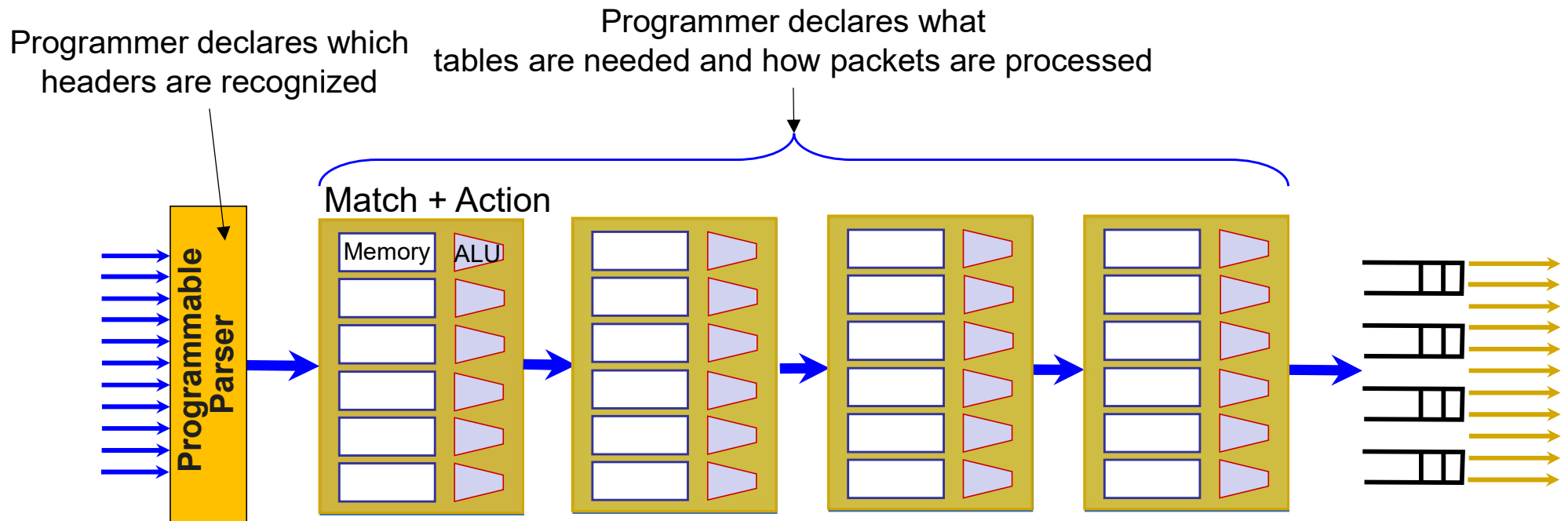
Programmable data plane

- What if we could tell switches exactly what we want?
 - What table to keep?
 - What rules to use?
 - What data to keep track of?
 - ...

Top-down workflow

- ▣ Precisely specify using a well-defined language
- ▣ Compile it down to run on a standardized hardware (e.g., using P4)
- ▣ Run at line speed

PISA: Protocol Independent Switch Architecture



All stages are identical – makes PISA a good “compiler target”

Topics

- Network layer (lectures 10–15)
 - Addressing
 - Data plane
 - Intra-domain routing
 - Inter-domain routing
- SDN and programmable networks (lecture 16)
- Link layer (lectures 17–19)
 - Ethernet
 - Wireless
- Datacenter networking (lecture 20)

Data link layer

- Provides four primary services
 - Framing
 - » Encapsulates network layer data
 - Link access
 - » Medium access control (MAC) protocol defines when to transmit frames
 - Reliable delivery
 - » Primarily for mediums with high error rates (e.g., wireless)
 - Error detection and correction

Point-to-point vs. broadcast medium

- **Point-to-point**: dedicated pairwise communication
 - E.g., long-distance fiber link
 - E.g., Point-to-point link b/n Ethernet switch and host
- **Broadcast**: shared wire or medium
 - Traditional Ethernet (pre ~2000)
 - 802.11 wireless LAN

Random access MAC protocols

- ❑ When node has packet to send
 - Transmit at full channel data rate **w/o** coordination
- ❑ Two or more transmitting nodes \Rightarrow collision
 - Data lost
- ❑ Random access MAC protocol specifies
 - How to detect and recover from collisions
- ❑ Examples
 - ALOHA and Slotted ALOHA
 - CSMA, CSMA/CD, CSMA/CA (wireless)

CSMA (Carrier Sense Multiple Access)

- ❑ CSMA: listen before transmit
 - If channel sensed idle: transmit entire frame
 - If channel sensed busy, defer transmission
- ❑ Human analogy: don't interrupt others!
- ❑ Does not eliminate all collisions
 - Why?

CSMA/CD (Collision Detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
 - Collisions detected within short time
 - Colliding transmissions aborted, reducing wastage
- Collision detection easy in wired (broadcast) LANs
 - Compare transmitted, received signals
- Collision detection difficult in wireless LANs

Limits on CSMA/CD network length



- ❑ Latency depends on physical length of link
 - Time to propagate a frame from one end to other
- ❑ Suppose A sends a frame at time t
 - And B sees an idle line at a time just before $t + d$
 - ... so B happily starts transmitting a frame
- ❑ B detects a collision, and sends jamming signal
 - But A cannot see collision until $t + 2d$

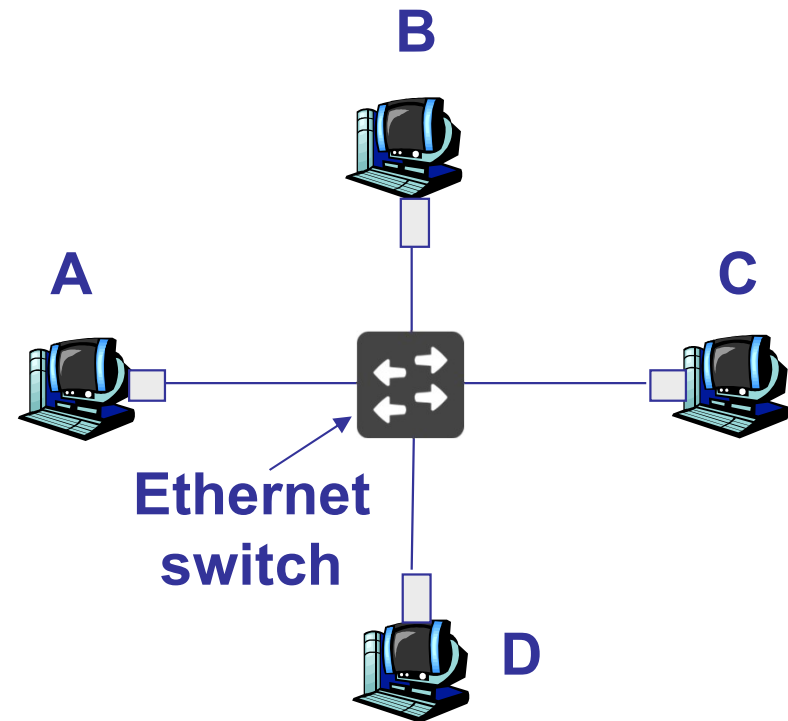
Limits on CSMA/CD network length



- ❑ A needs to wait for time **$2d$** to detect collision
 - So, A should keep transmitting during this period
 - AND keep an eye out for a possible collision
- ❑ Imposes restrictions; e.g., for 10 Mbps Ethernet
 - **Maximum length** of the wire: 2,500 meters
 - **Minimum length** of a frame: 512 bits (64 bytes)

Why switched Ethernet?

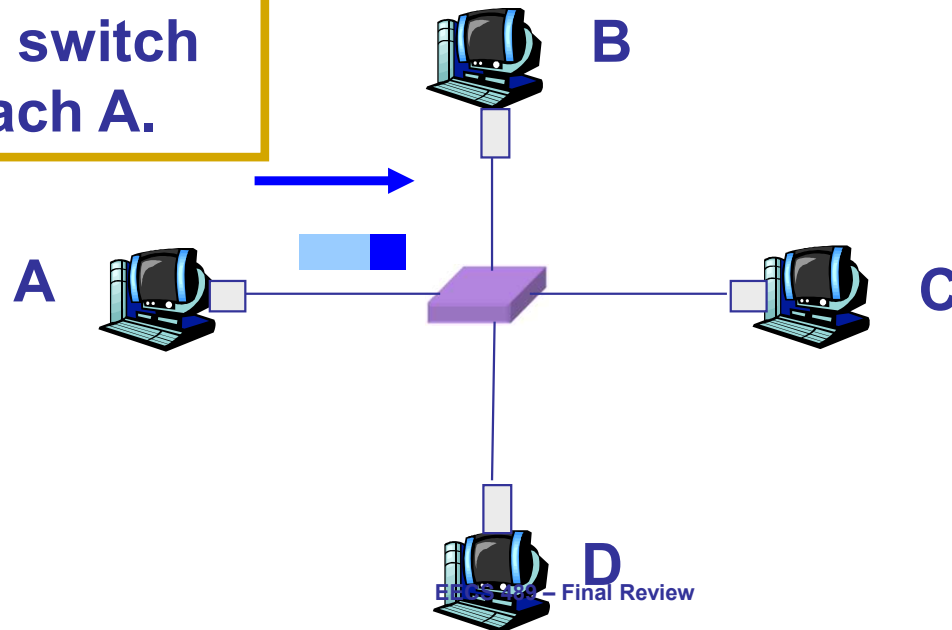
- Enables concurrent communication
 - Host A can talk to C, while B talks to D
 - No collisions and no need for CSMA/CD
 - No constraints on link lengths, etc.



Ethernet switches are “self learning”

- When a packet arrives:
 - Inspect source MAC address, associate with incoming port
 - Store mapping in the switch table
 - Use **time-to-live** field to eventually forget mapping

Packet tells switch how to reach A.



ARP and DHCP

- Link layer discovery protocols
 - ARP → Address Resolution Protocol
 - DHCP → Dynamic Host Configuration Protocol
 - Confined to a single local-area network (LAN)
 - Rely on broadcast capability

Key ideas in both ARP and DHCP

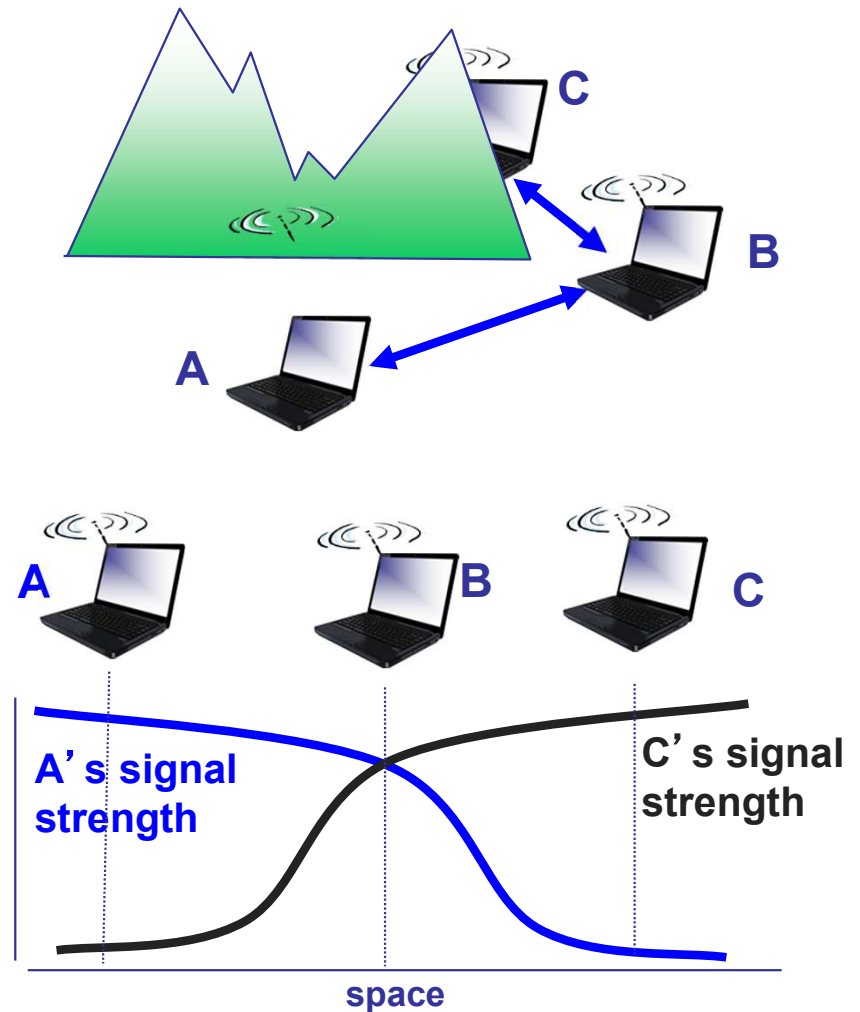
- ❑ **Broadcasting**: Can use broadcast to make contact
 - Scalable because of limited size
- ❑ **Caching**: remember the past for a while
 - Store the information you learn to reduce overhead
- ❑ **Soft state**: eventually forget the past
 - Associate a time-to-live field with the information
 - ... and either refresh or discard the information
 - Key for robustness in the face of unpredictable change

Wireless link characteristics

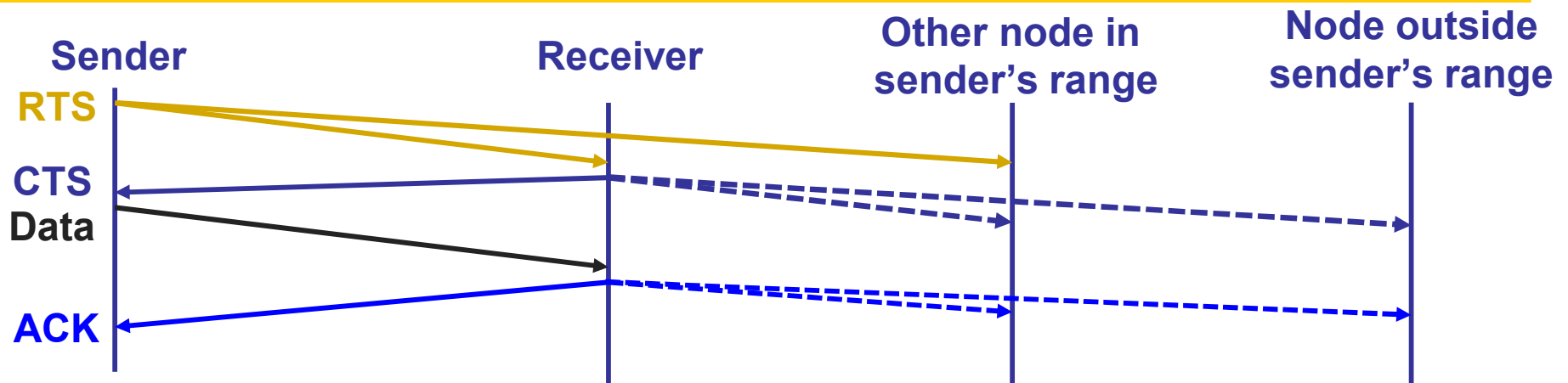
- Three important differences from wired link ...
 - **Decreased signal strength**: Radio signal attenuates as it propagates through matter (path loss)
 - **Multipath propagation**: Radio signal reflects off objects ground, arriving at destination at slightly different times
 - **Interference from other sources**: Standardized wireless network frequencies (e.g., 2.4 GHz) shared by other devices (e.g., phone); devices (motors) interfere as well

Hidden terminal problem

- B, A hear each other
- B, C hear each other
- A, C can not hear each other
- Hence, A, C are unaware of their interference at B



CSMA/CA



| Before every data transmission

- Sender sends a Request to Send (RTS) frame with the length of transmission and the destination
- Receiver respond with a Clear to Send (CTS) frame
- Sender sends data
- Receiver sends an ACK

| If sender doesn't get a CTS back, it assumes collision

Topics

- | Network layer (lectures 10–15)
 - Addressing
 - Data plane
 - Intra-domain routing
 - Inter-domain routing
- | SDN and programmable networks (lecture 16)
- | Link layer (lectures 17–19)
 - Ethernet
 - Wireless
- | Datacenter networking (lecture 20)

Datacenter applications

- | Common theme: **parallelism**
 - Applications decomposed into tasks
 - Running in parallel on different machines
- | Two common paradigms
 - Partition-Aggregate
 - Map-Reduce

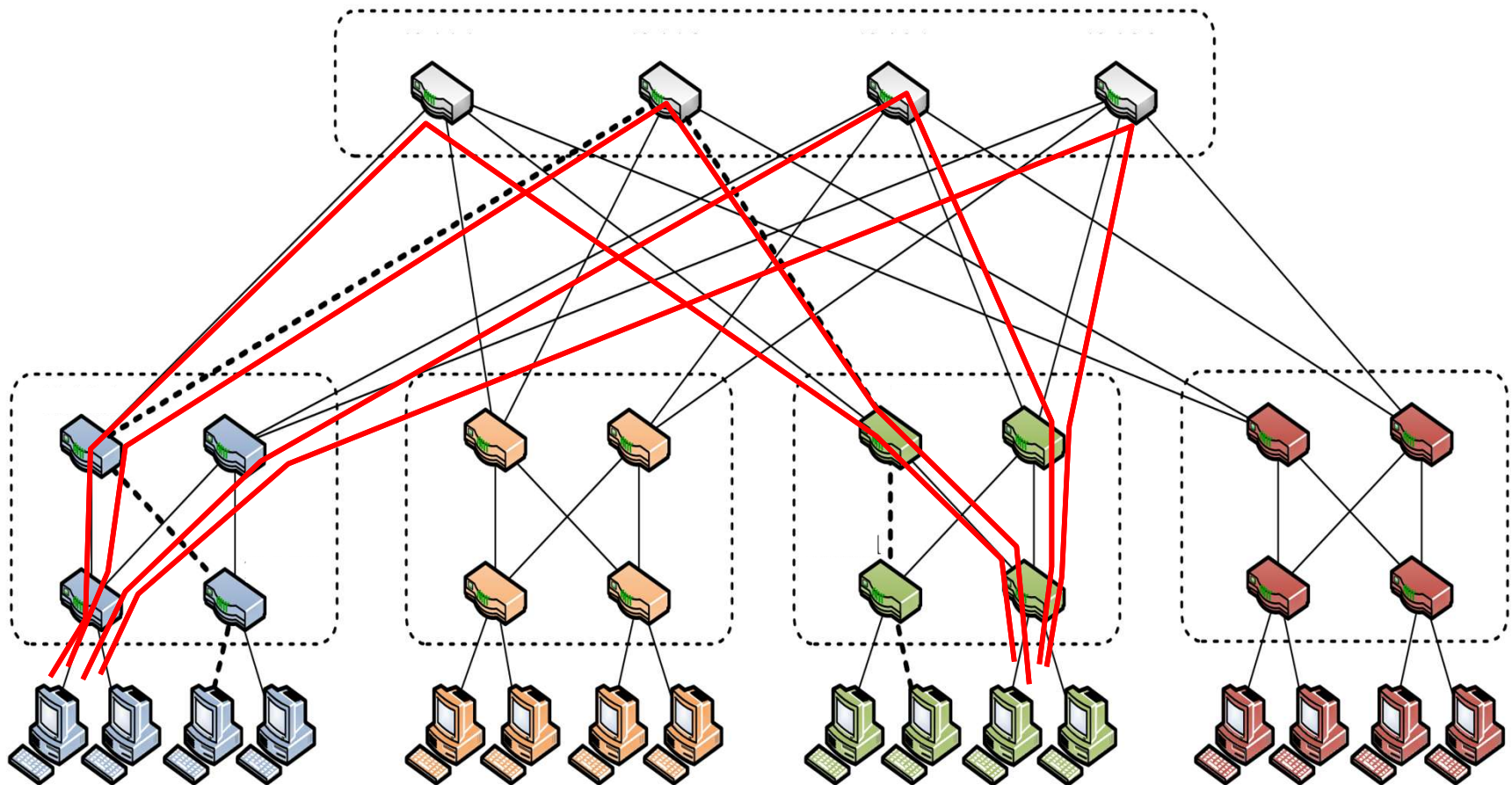
Datacenter traffic characteristics

- | Two key characteristics
 - Most flows are small
 - Most bytes come from large flows
- | Applications want
 - High bandwidth (large flows)
 - Low latency (small flows)

Datacenter networking stack

- | Networking in modern datacenters
 - L2/L3 design
 - » Addressing / routing / forwarding in the Fat-Tree
 - L4 design
 - » Transport protocol design (w/ Fat-Tree)
 - L7 design
 - » Exploiting application-level information (w/ Fat-Tree)

Using multiple paths well



L2/L3 highlights

- | Load balancing while forwarding
 - Per-packet
 - Per-flow
- | Hard-coded addressing or via indirection
- | Modified LS/DV or source routing

L4 highlights

- | Tension between high throughput and low latency requirements
 - Deep queues vs shallow queues
- | DCTCP
 - React early, quickly, and with certainty using ECN
 - React in **proportion to the extent of congestion**, not its presence

L7 highlights

- | What do applications care about?
 - Flow completion time (FCT)
 - Coflow completion time (CCT)
 - » A coflow is a collection of flows with a shared application-level objective
 - We should strive to optimize as close an objective as possible to the application

Summary

| THANK YOU SO MUCH!!!