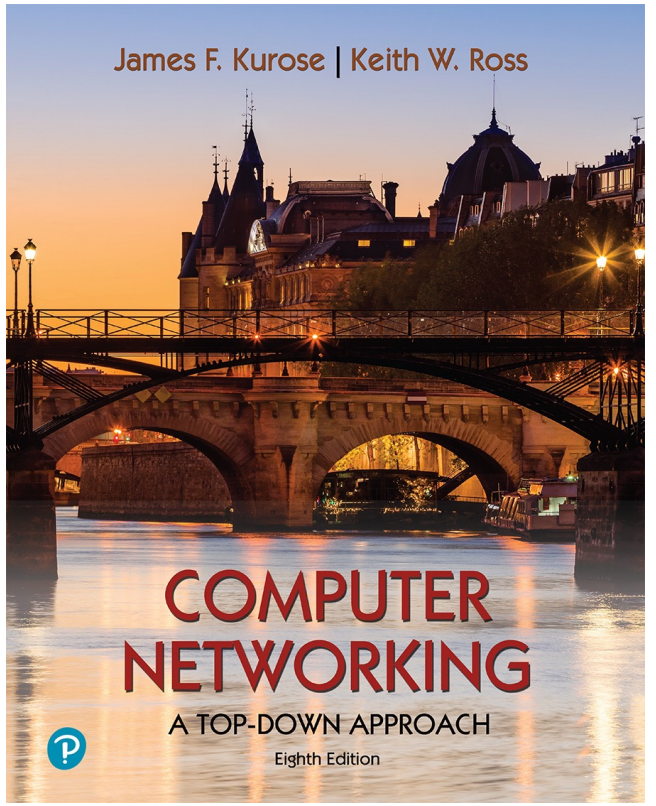
Chapter 5
Network Layer: Control Plane

Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

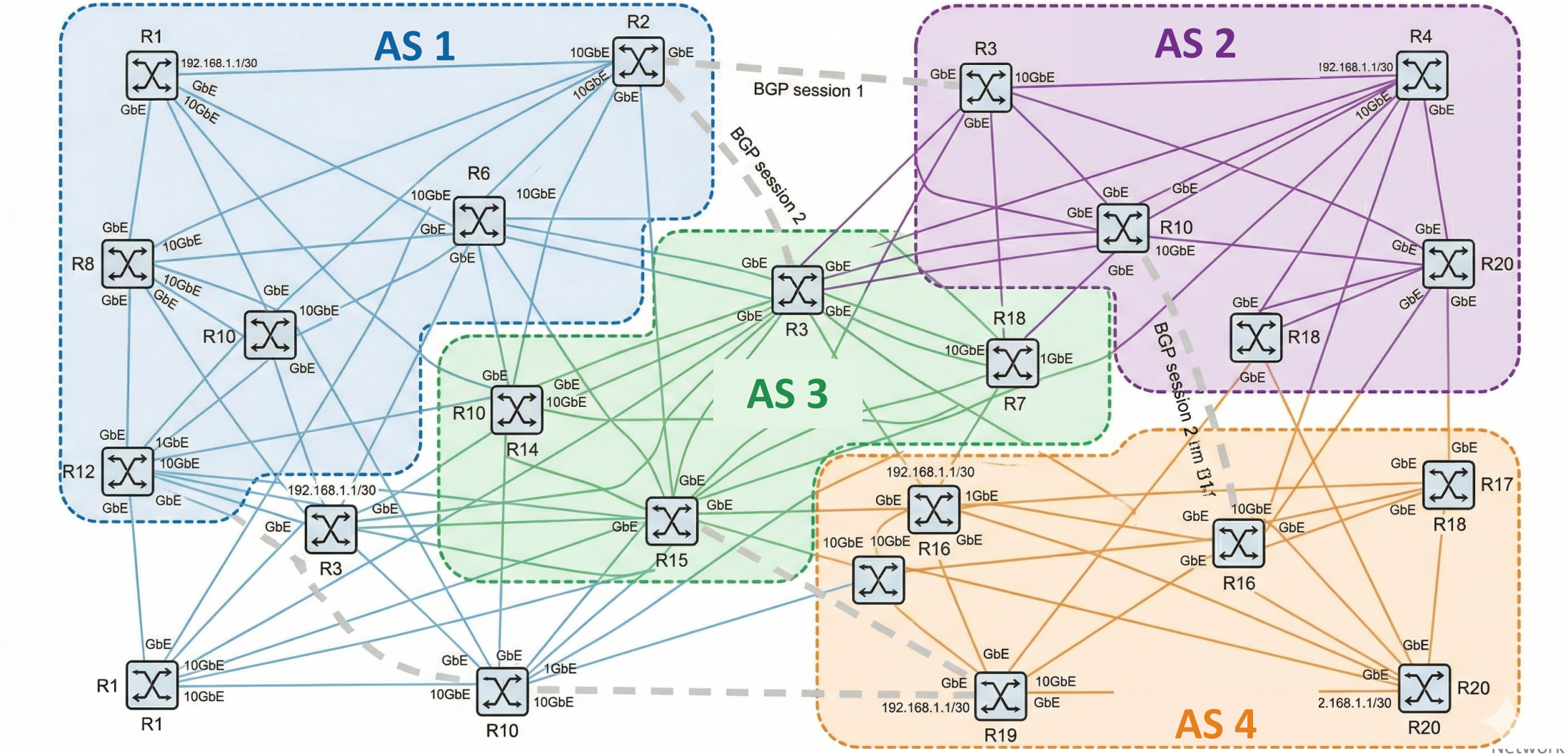
Adapted from the slides of the book's authors

Network layer: “control plane” roadmap

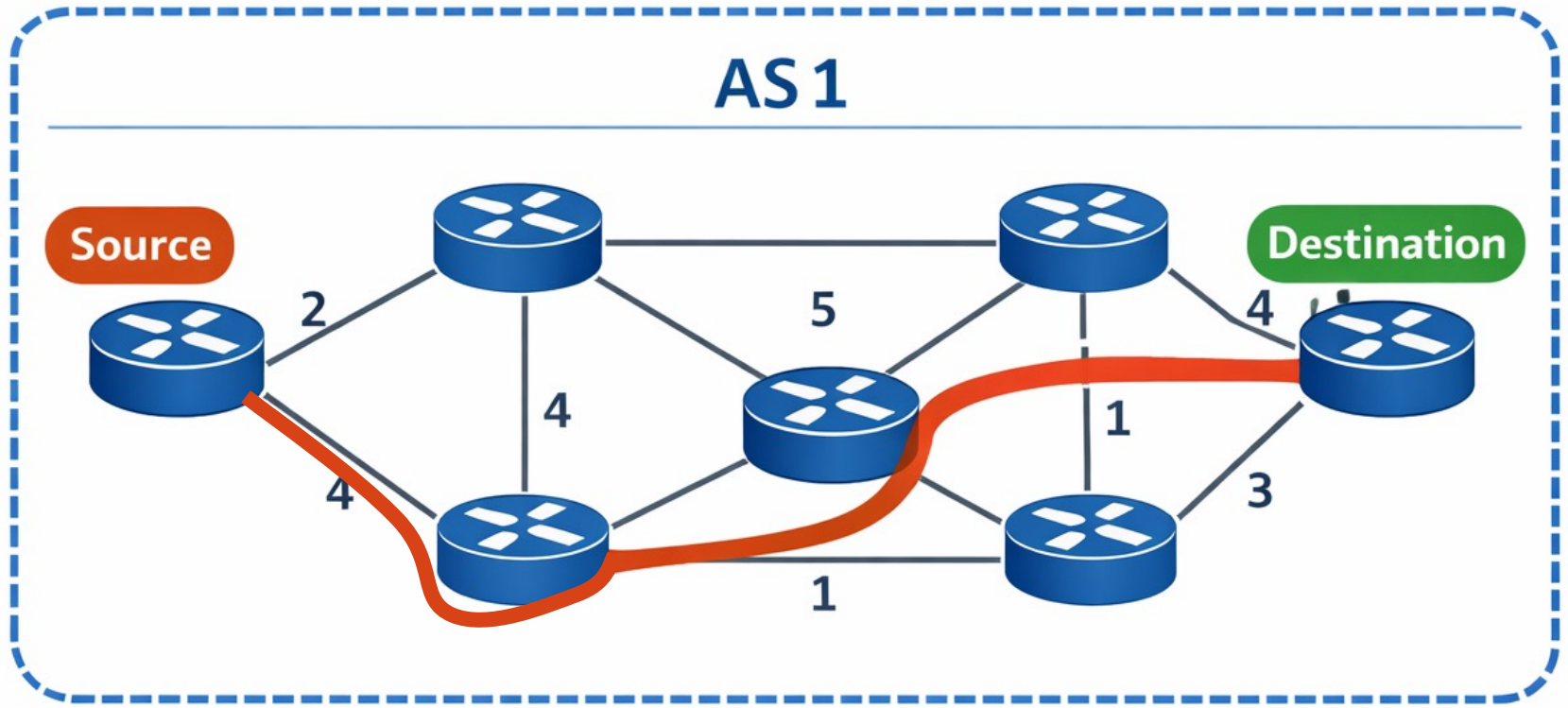
- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

- Measurement

Intra-AS routing and Inter-AS routing



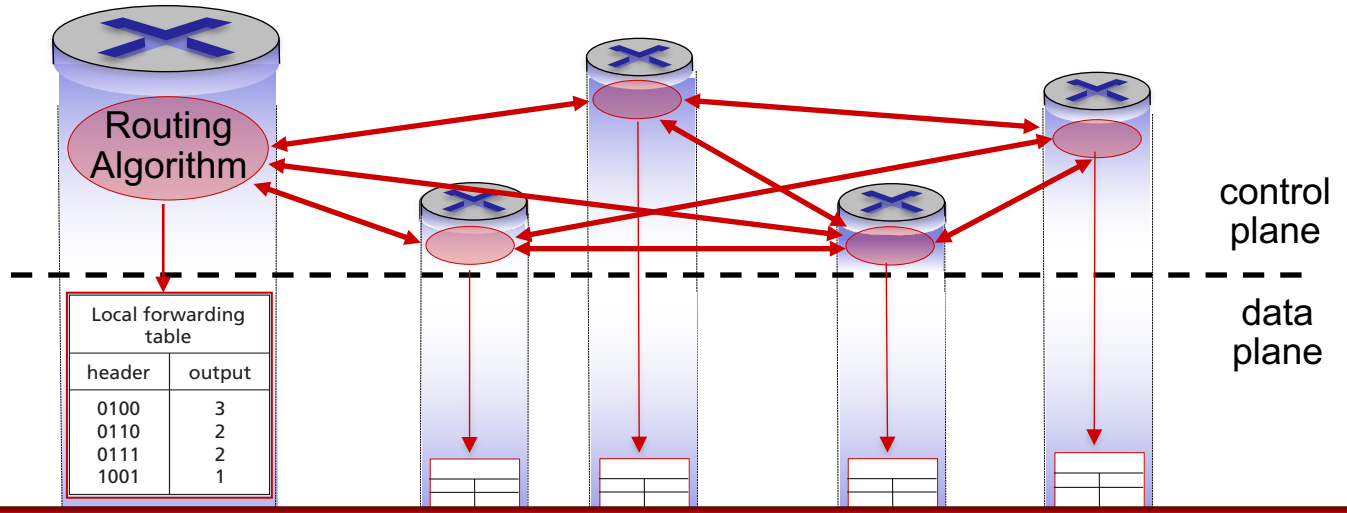
Intra-AS Routing Algorithm



Goal: find the shortest path to destination

Per-router control plane

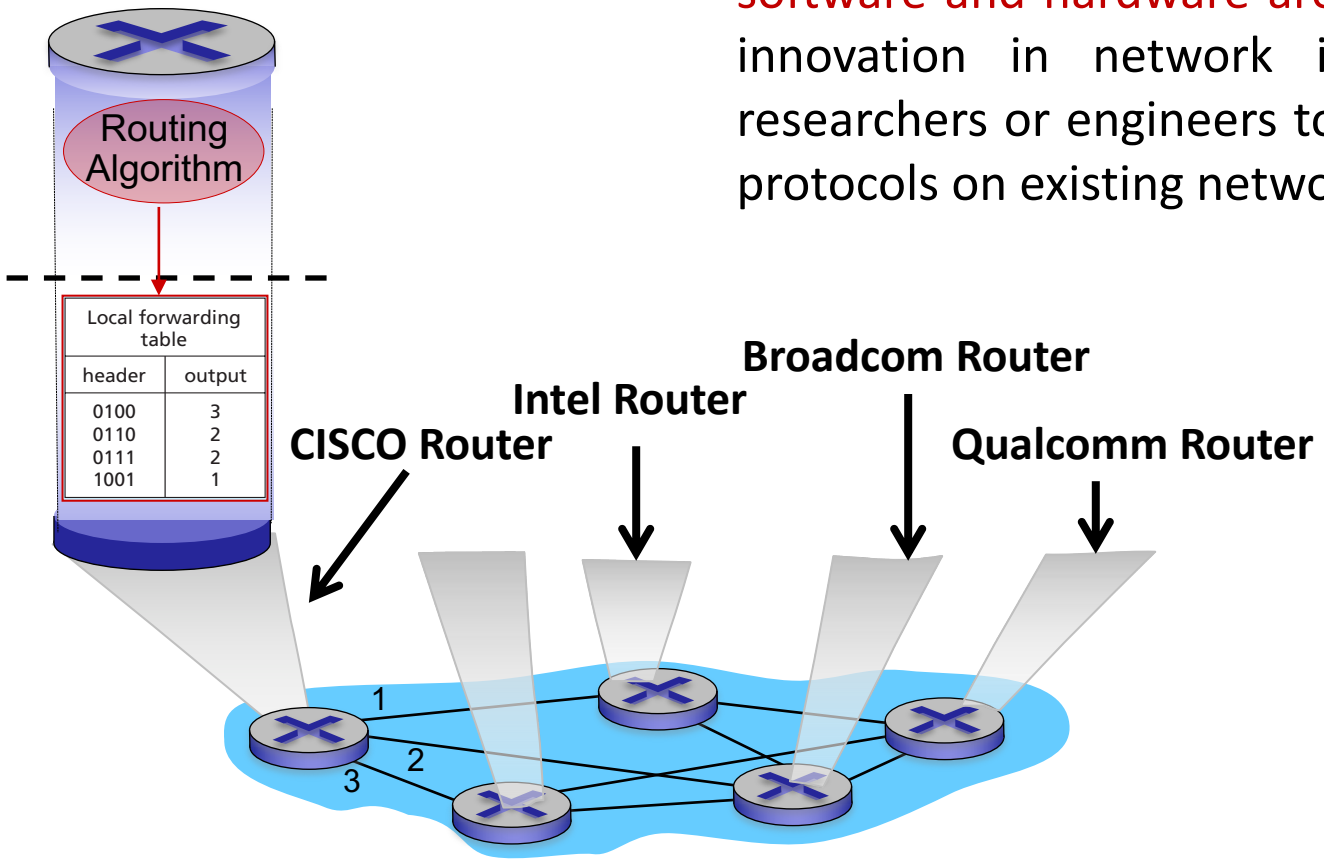
Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



What's the problem here?

Black-Box Routers

The operating system and the hardware are both blackbox

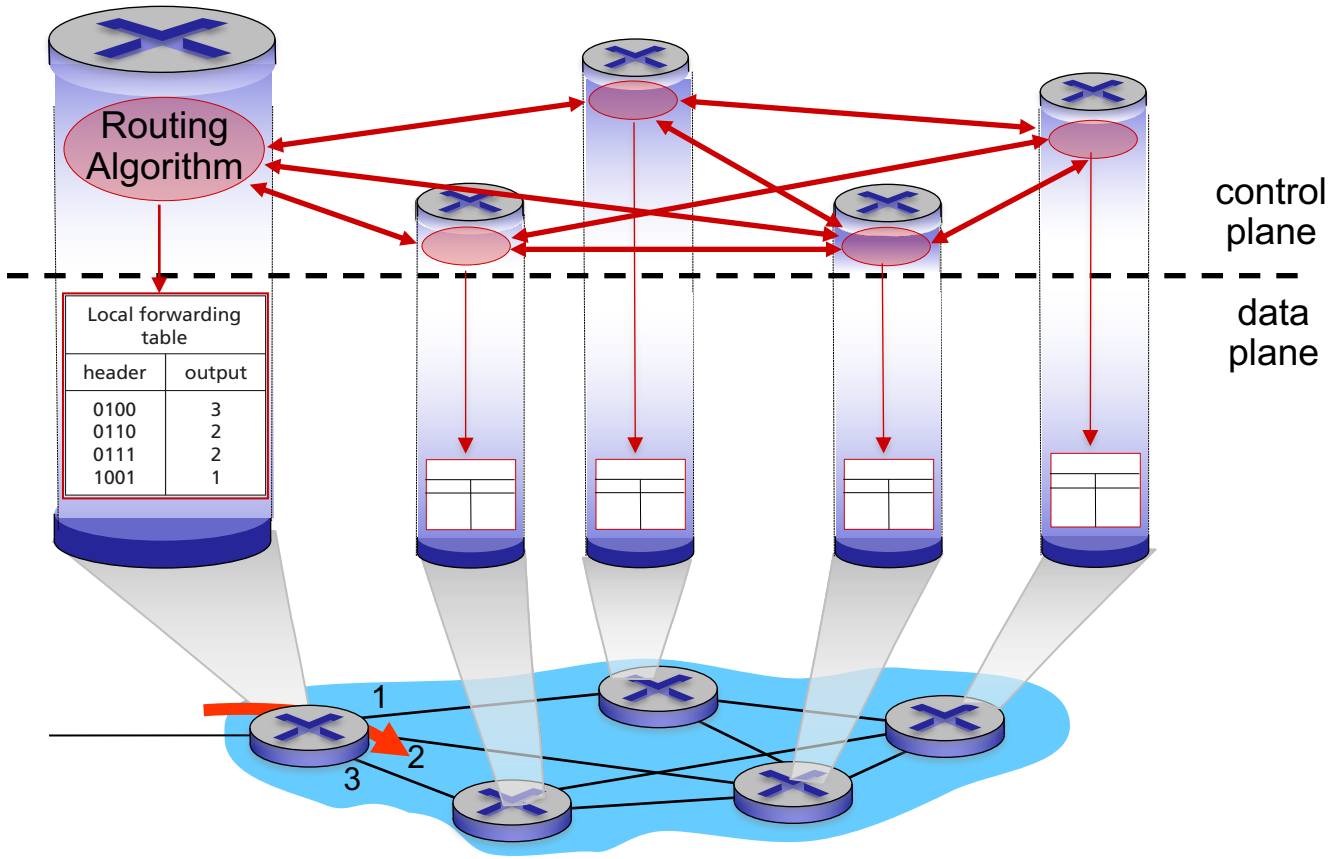


Severe Vendor Lock-in: Traditional devices are closed ecosystems where software and hardware are tightly coupled and sold together. This stifles innovation in network infrastructure, making it very difficult for researchers or engineers to test and validate new routing mechanisms or protocols on existing networks.

Extreme rigidity and high operational costs: Network protocols are strictly hardcoded into the proprietary hardware's operating system.

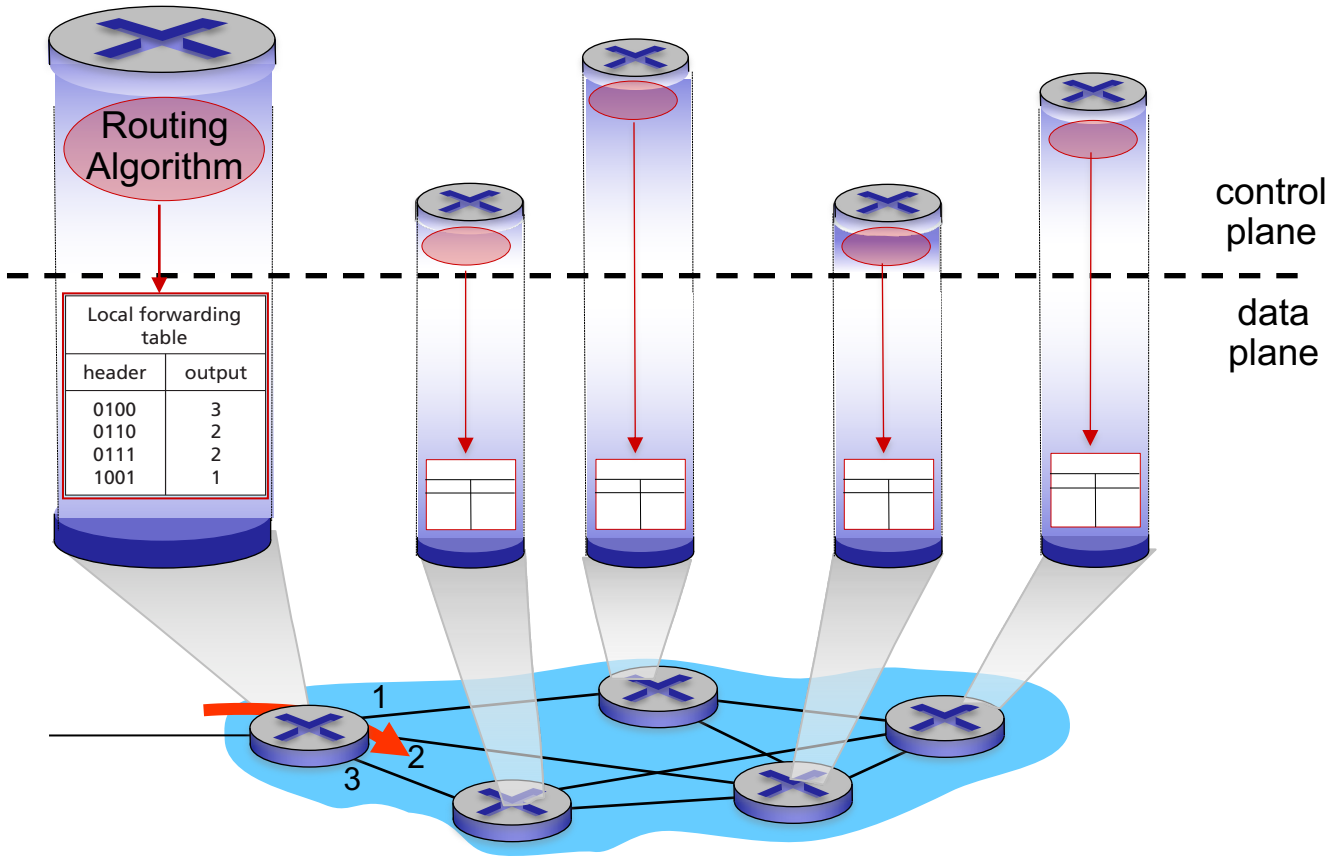
Distributed Algorithm

Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



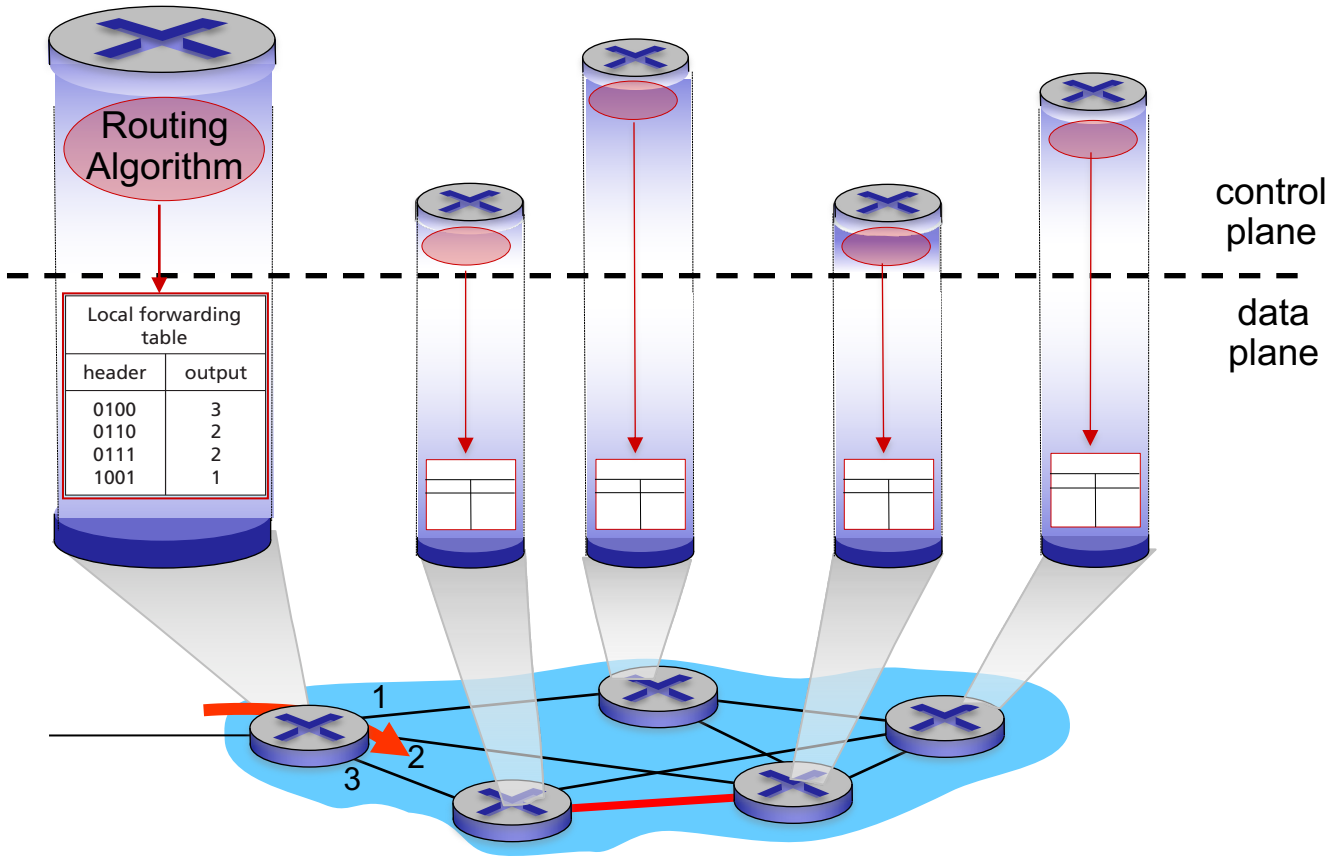
Distributed Algorithm

Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



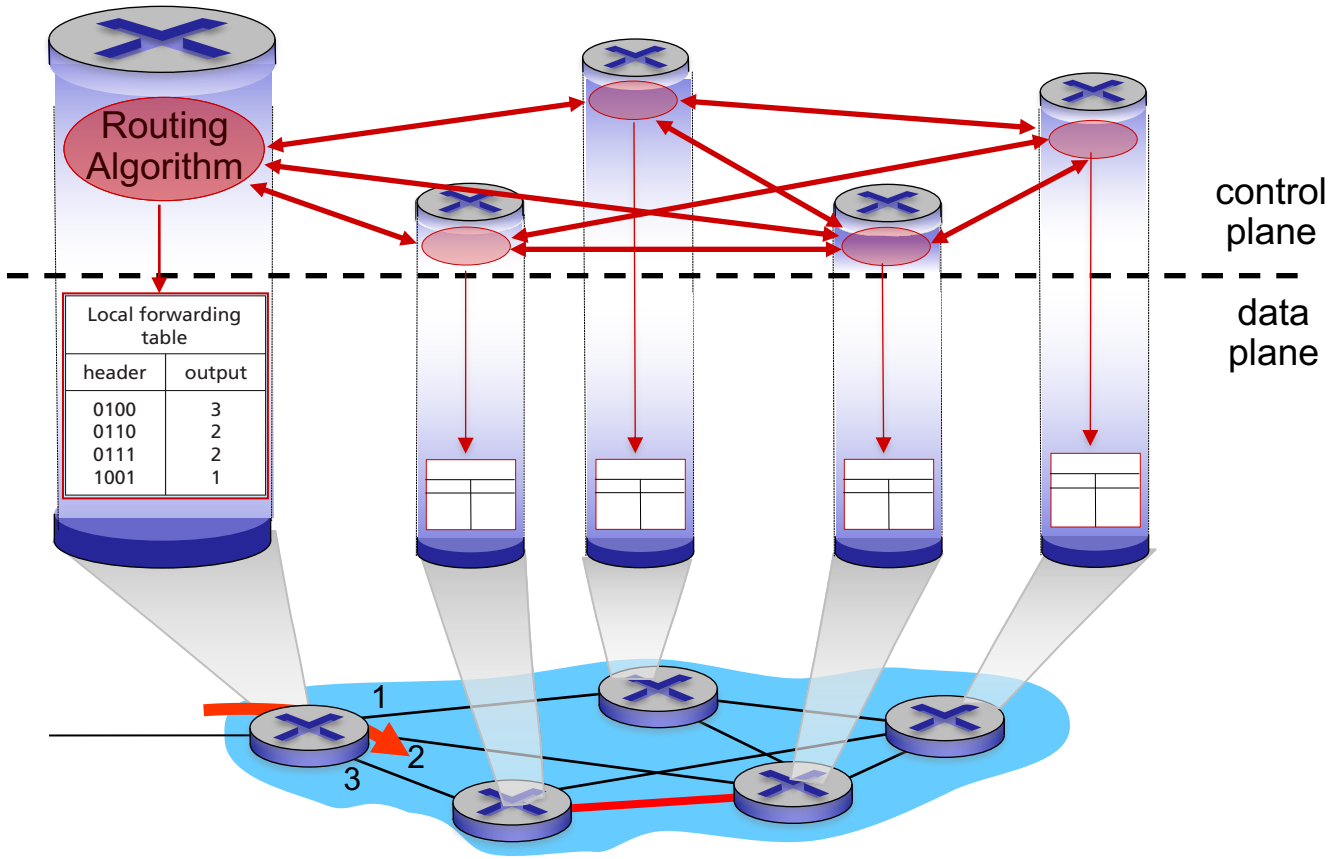
Distributed Algorithm

Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



Distributed Algorithm

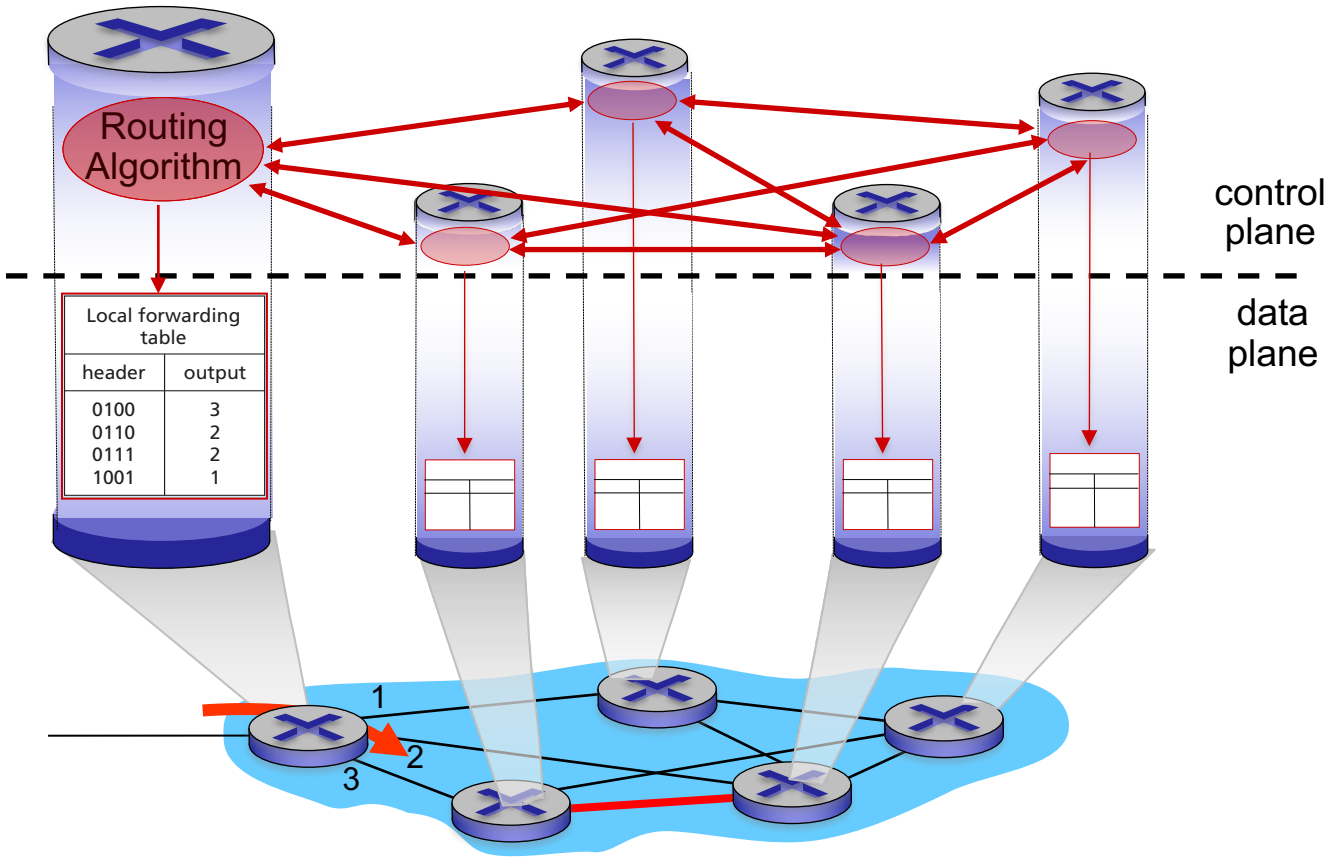
Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



Slow convergence: the bad news needs propagate hop by hop across the entire network. All devices must receive it before recalculating their routing tables

Rich Policy, not just Cost

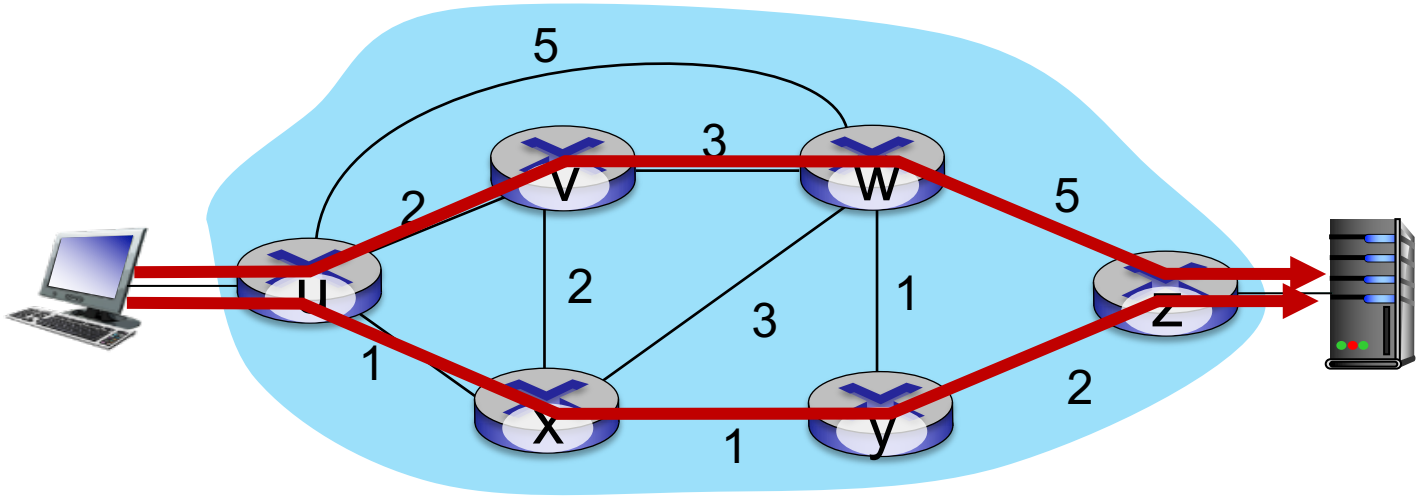
Network goals are often **more than shortest-path routing**



Operators may enforce policies such as:

- steer traffic through a **firewall** or middlebox
- block certain **users or hosts** from specific services
- give higher priority to **latency-sensitive applications**
- force some flows to **avoid certain links**
- isolate traffic across **tenants or groups**

Traffic engineering: difficult with traditional routing

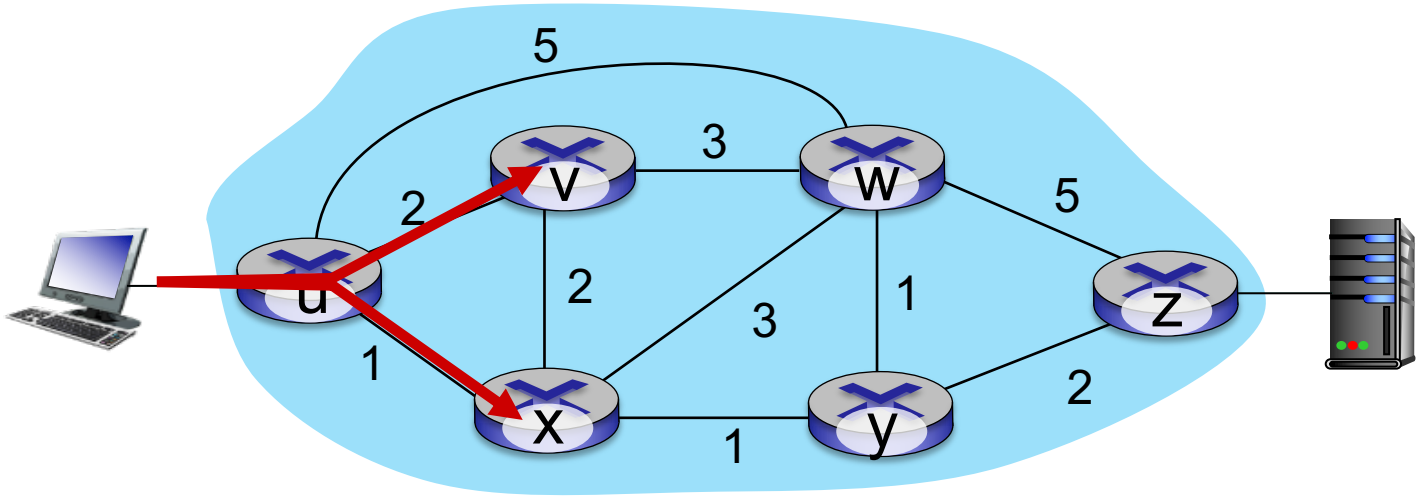


Q: what if network operator wants u-to-z traffic to flow along *uvwz*, rather than *uxyz*?

A: need to re-define link weights so traffic routing algorithm computes routes accordingly (or need a new routing algorithm)!

link weights are only control "knobs": not much control!

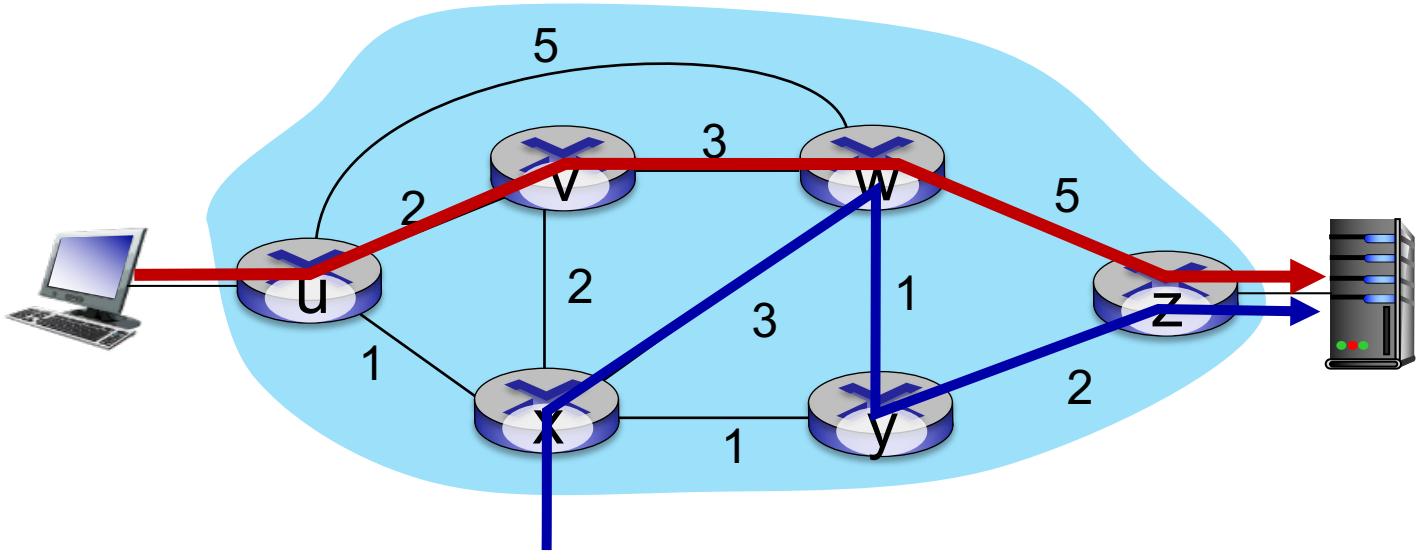
Traffic engineering: difficult with traditional routing



Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?

A: can't do it (or need a new routing algorithm)

Traffic engineering: difficult with traditional routing



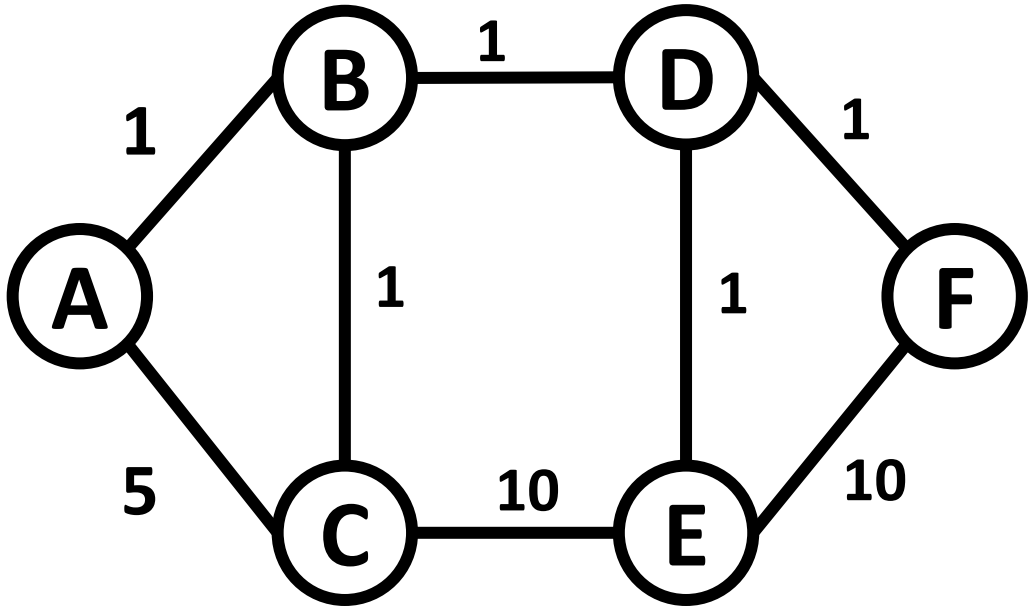
Q: what if w wants to route blue and red traffic differently from w to z?

A: can't do it (with destination-based forwarding, and LS, DV routing)

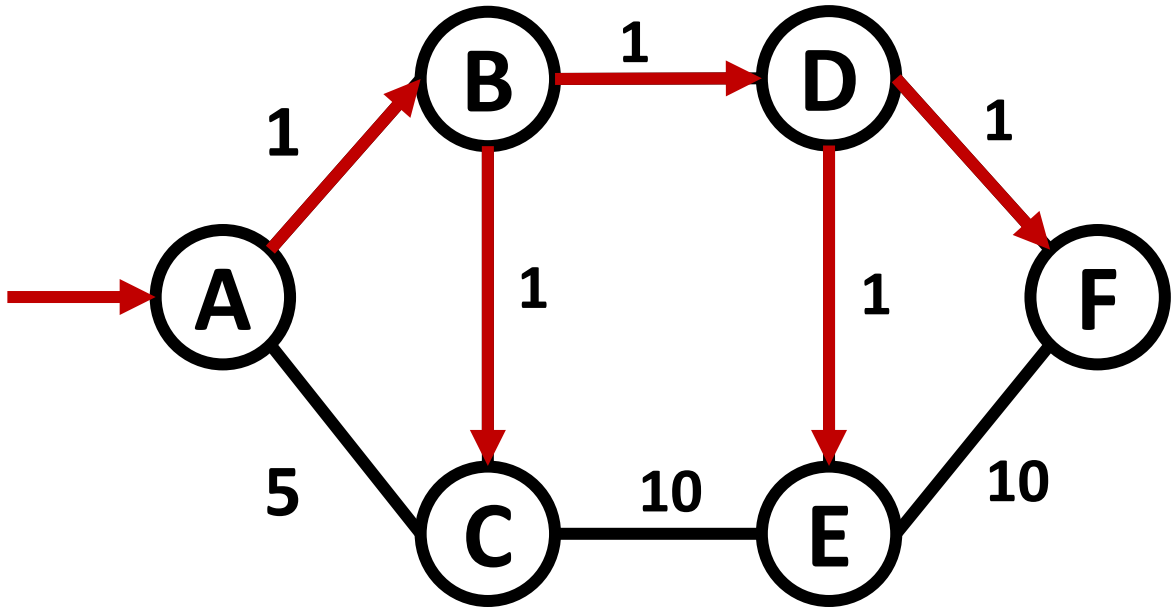
We learned in Chapter 4 that generalized forwarding and SDN can be used to achieve *any* routing desired

Topology-Driven not Traffic-Driven

The routing is purely based on the graph topology



Topology-Driven not Traffic-Driven



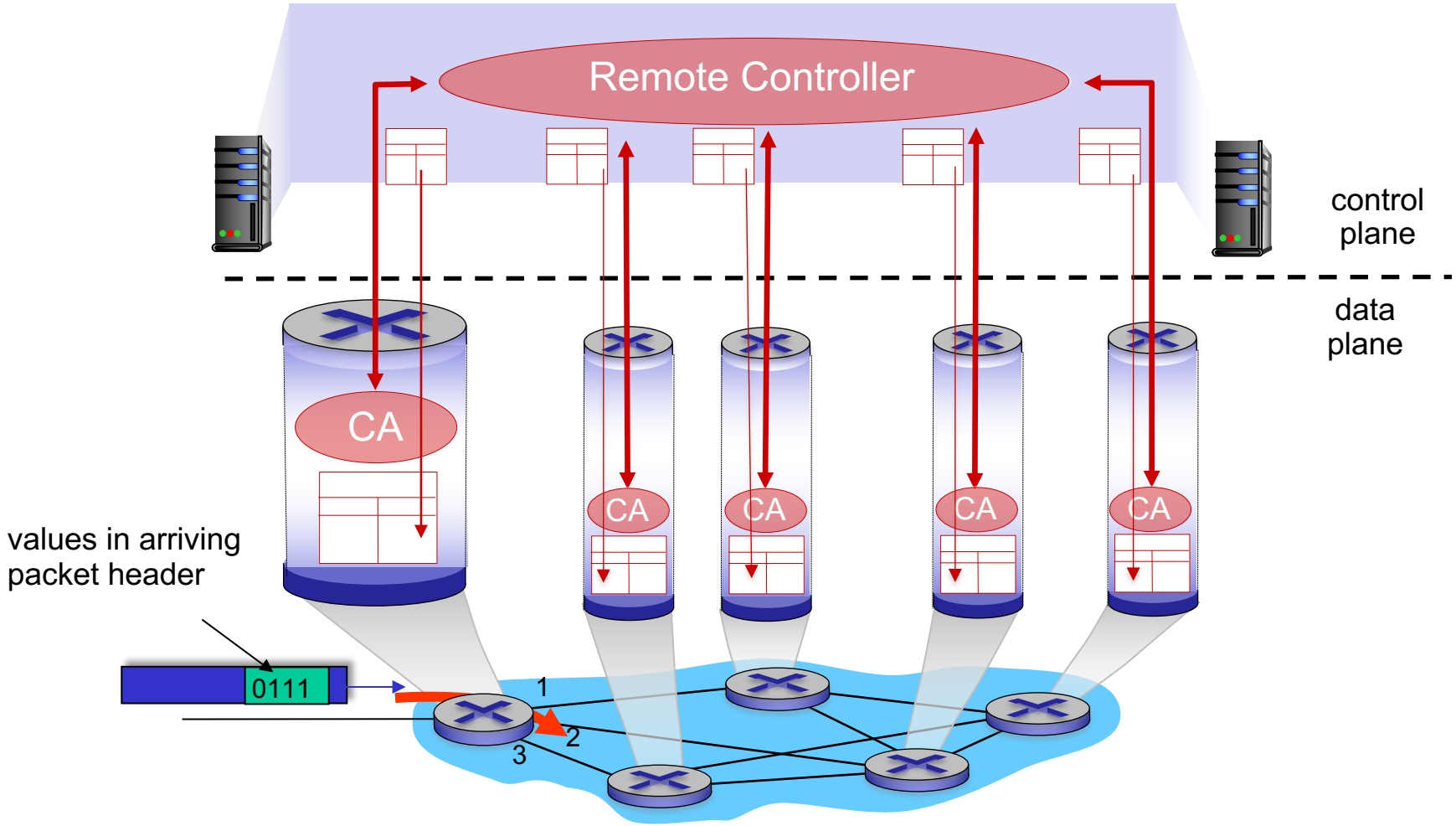
The routing is purely based on the graph topology

The routing algorithm cannot react to traffic distribution

The traffic from A needs to go through B to reach every other nodes of the network

Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers

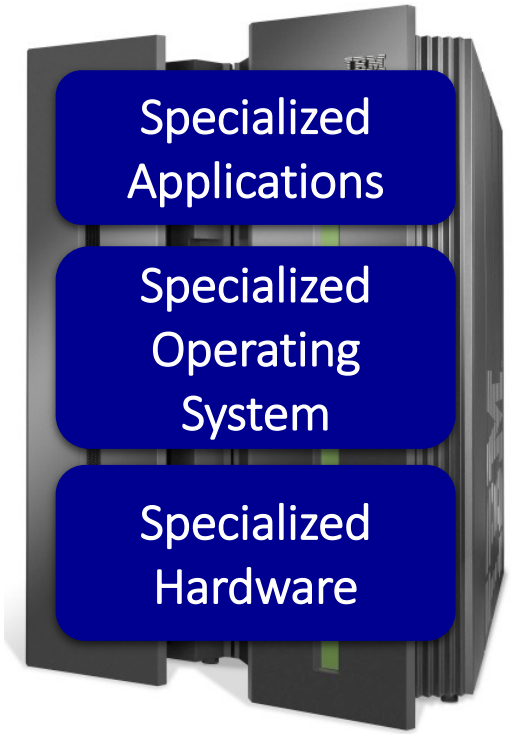


Software defined networking (SDN)

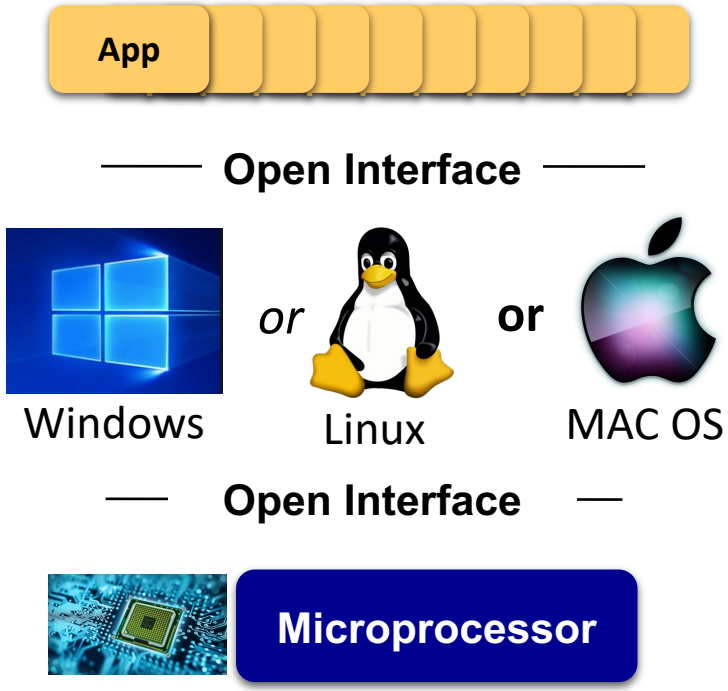
Why a *logically centralized* control plane?

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding (recall OpenFlow API) allows "programming" routers
  - centralized "programming" easier: compute tables centrally and distribute
  - distributed "programming" more difficult: compute tables as result of distributed algorithm (protocol) implemented in each-and-every router
- open (non-proprietary) implementation of control plane
  - foster innovation: let 1000 flowers bloom

SDN analogy: mainframe to PC revolution



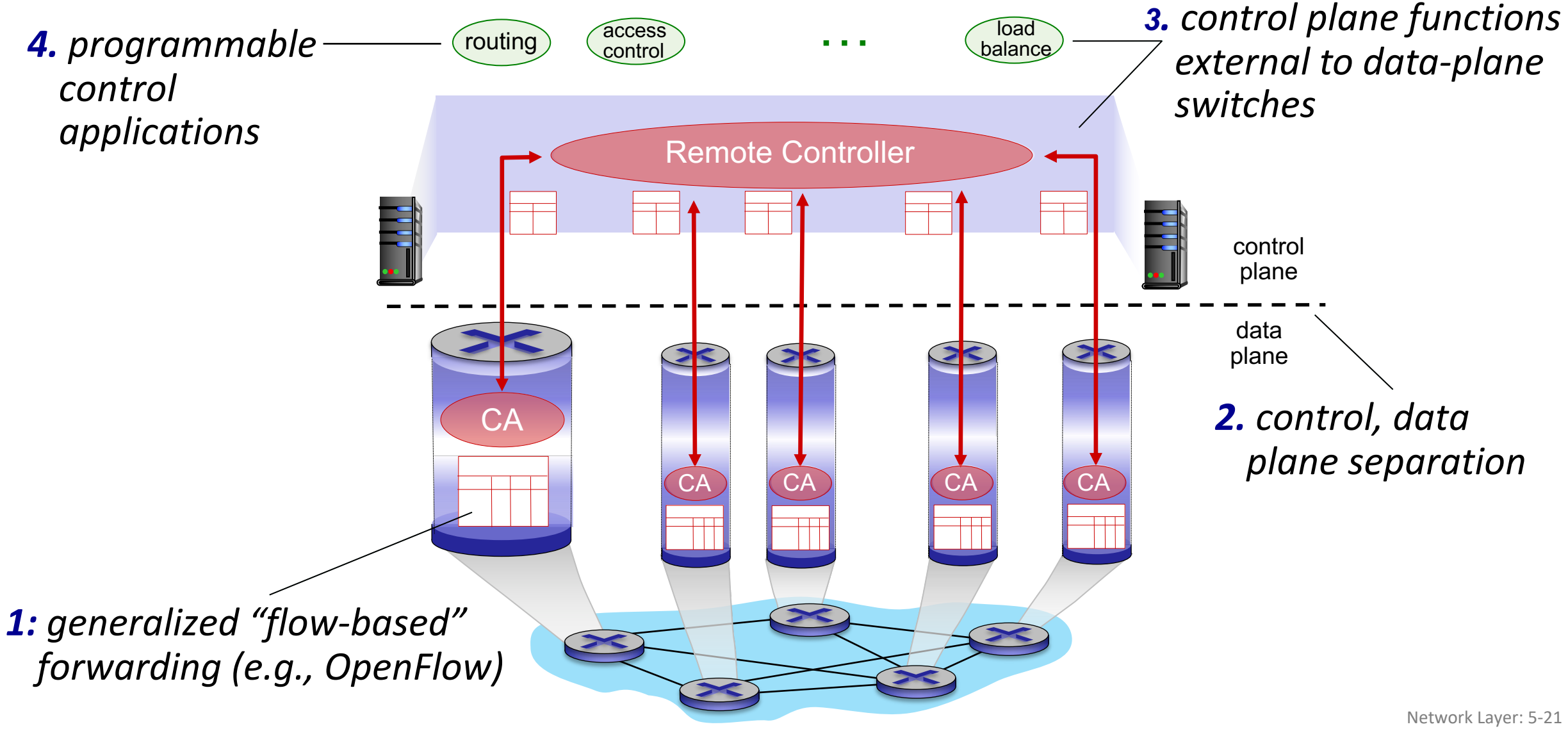
Vertically integrated
Closed, proprietary
Slow innovation
Small industry



Horizontal
Open interfaces
Rapid innovation
Huge industry

* Slide courtesy: N. McKeown

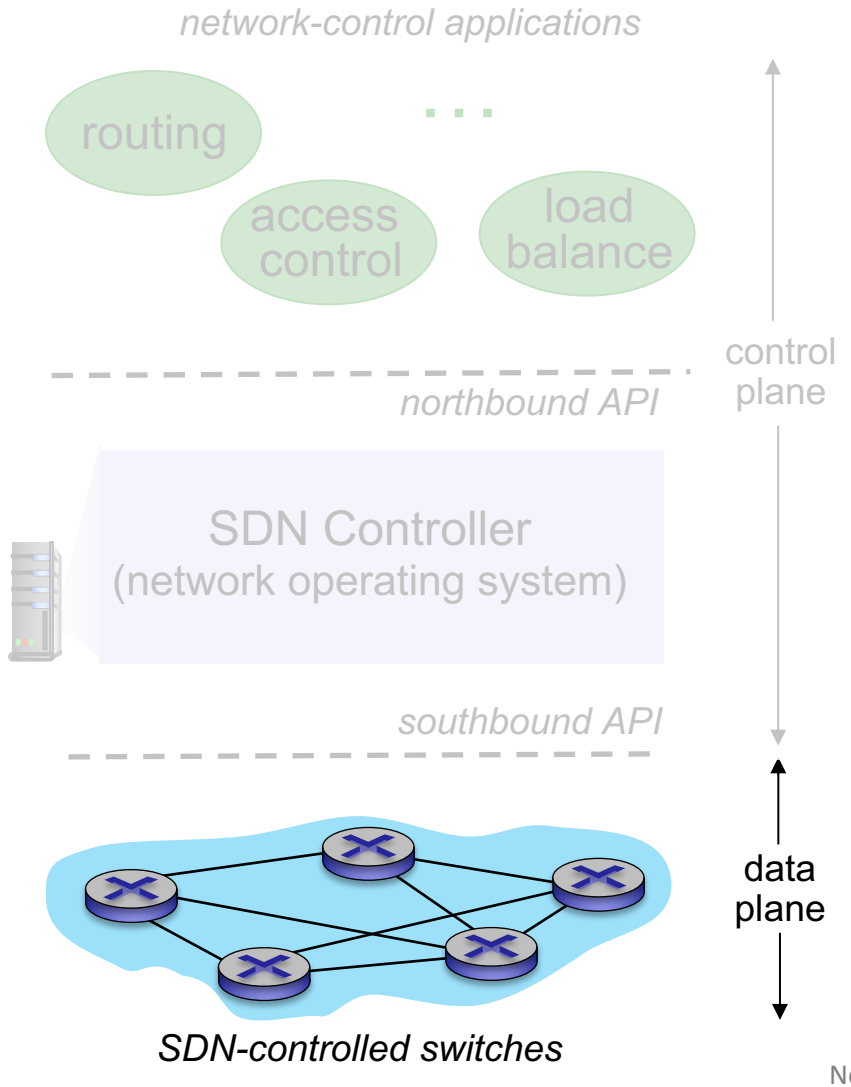
Software defined networking (SDN)



Software defined networking (SDN)

Data-plane switches:

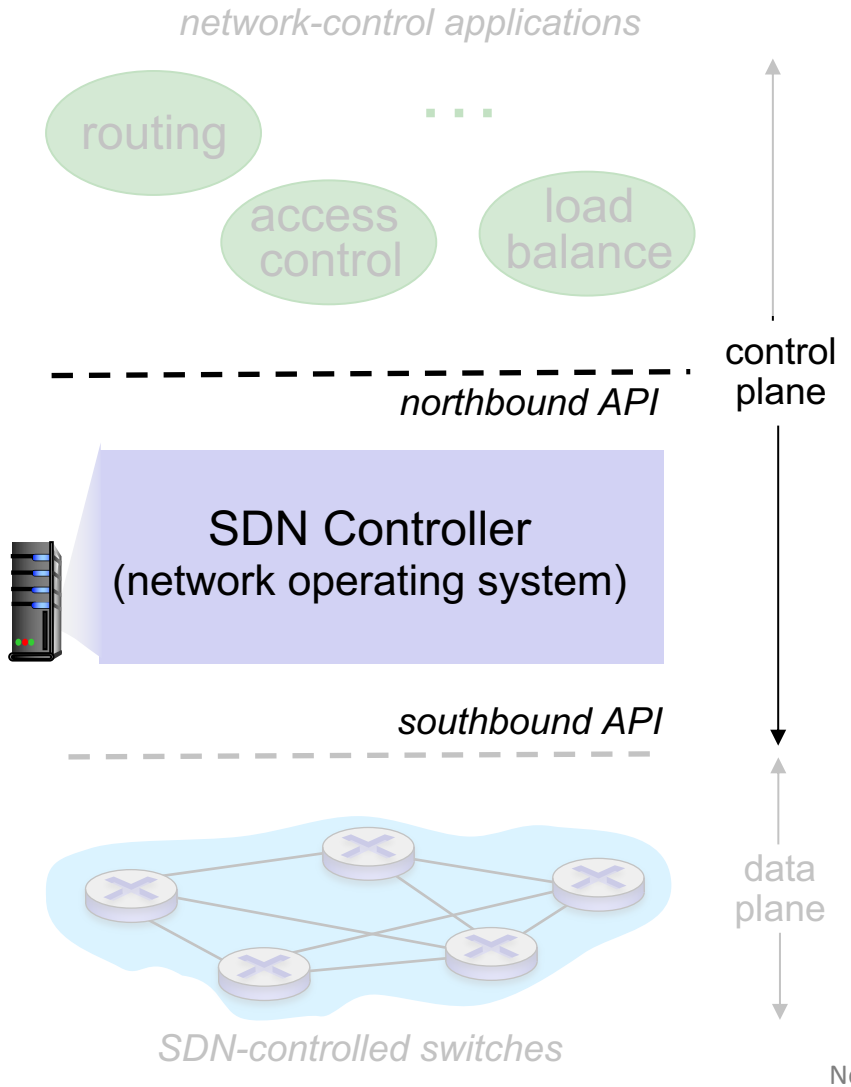
- fast, simple, commodity switches implementing generalized data-plane forwarding in hardware
- flow (forwarding) table computed, installed under controller supervision
- API for table-based switch control (e.g., OpenFlow)
  - defines what is controllable, what is not
- protocol for communicating with controller (e.g., OpenFlow)



Software defined networking (SDN)

SDN controller (network OS):

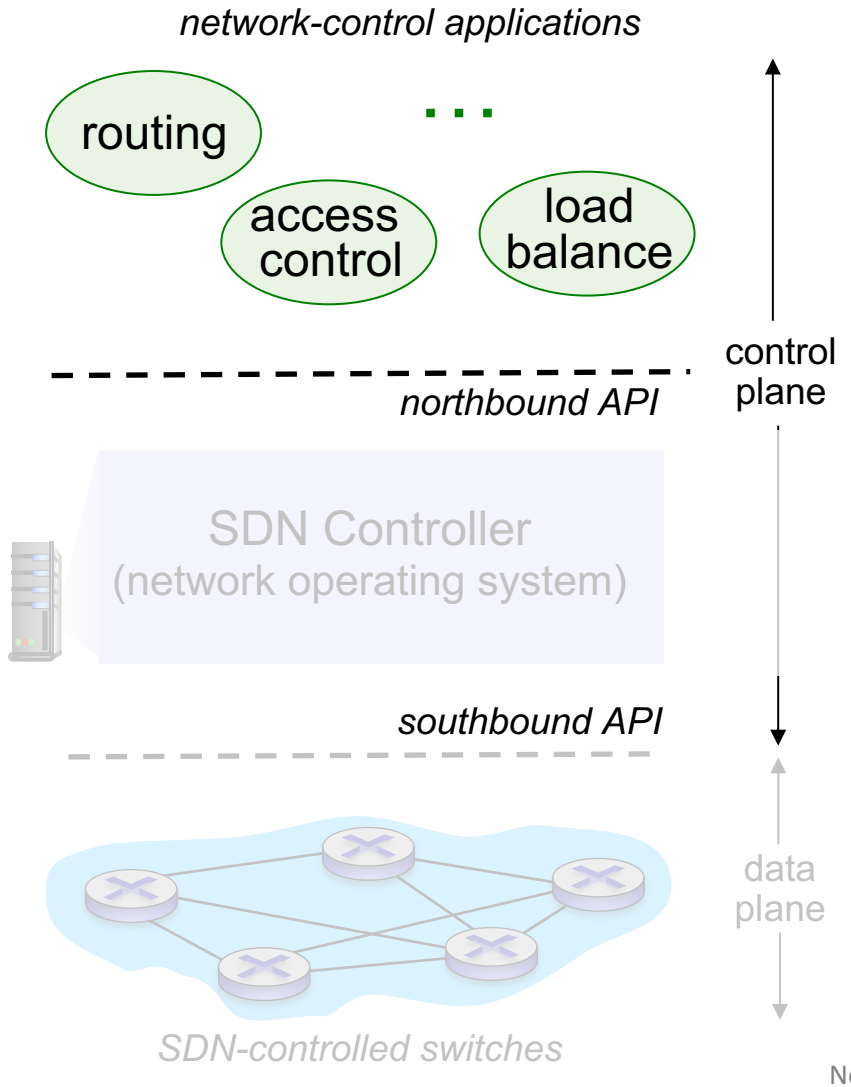
- maintain network state information
- interacts with network control applications “above” via northbound API
- interacts with network switches “below” via southbound API
- implemented as distributed system for performance, scalability, fault-tolerance, robustness



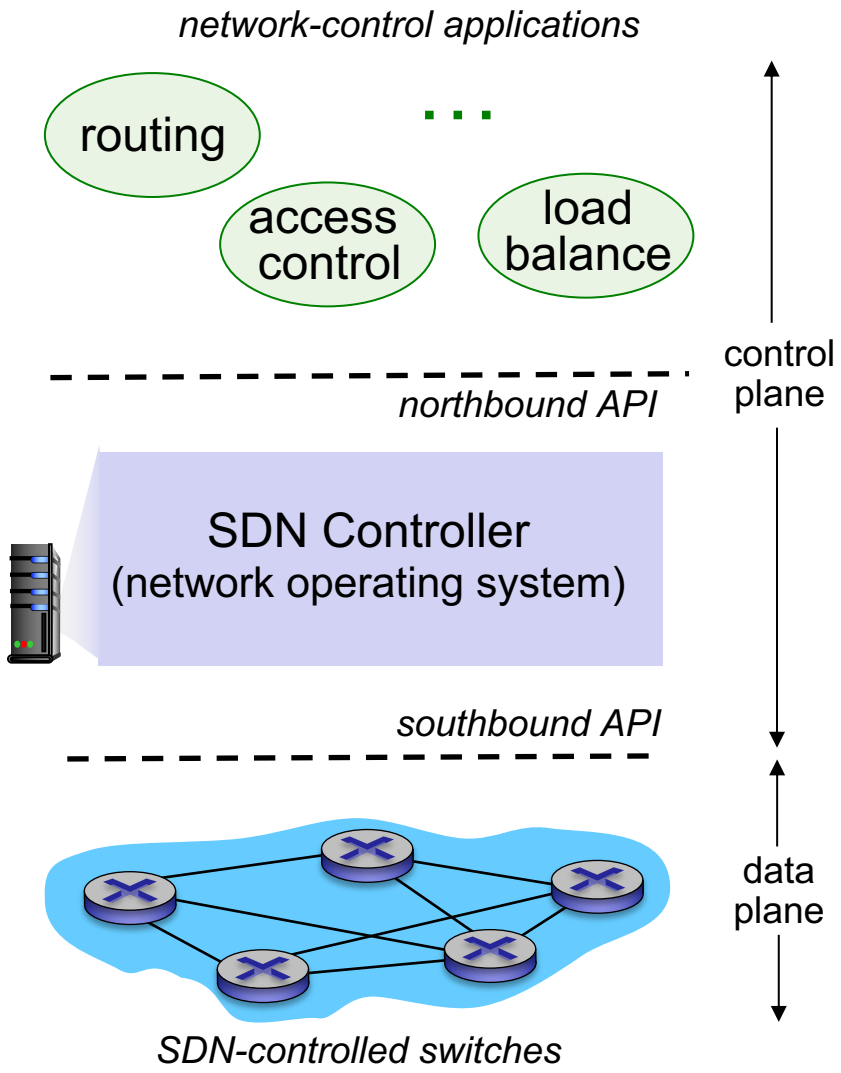
Software defined networking (SDN)

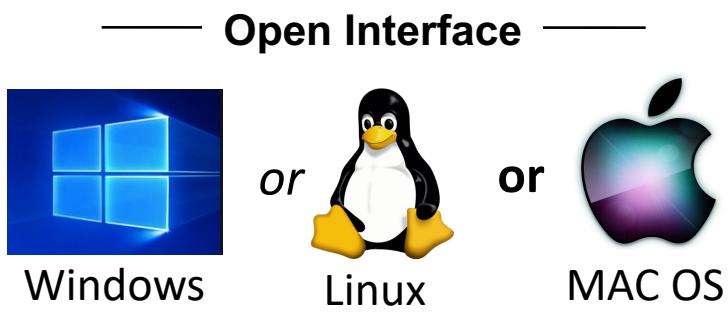
network-control apps:

- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled:* can be provided by 3rd party: distinct from routing vendor, or SDN controller



Software defined networking (SDN)



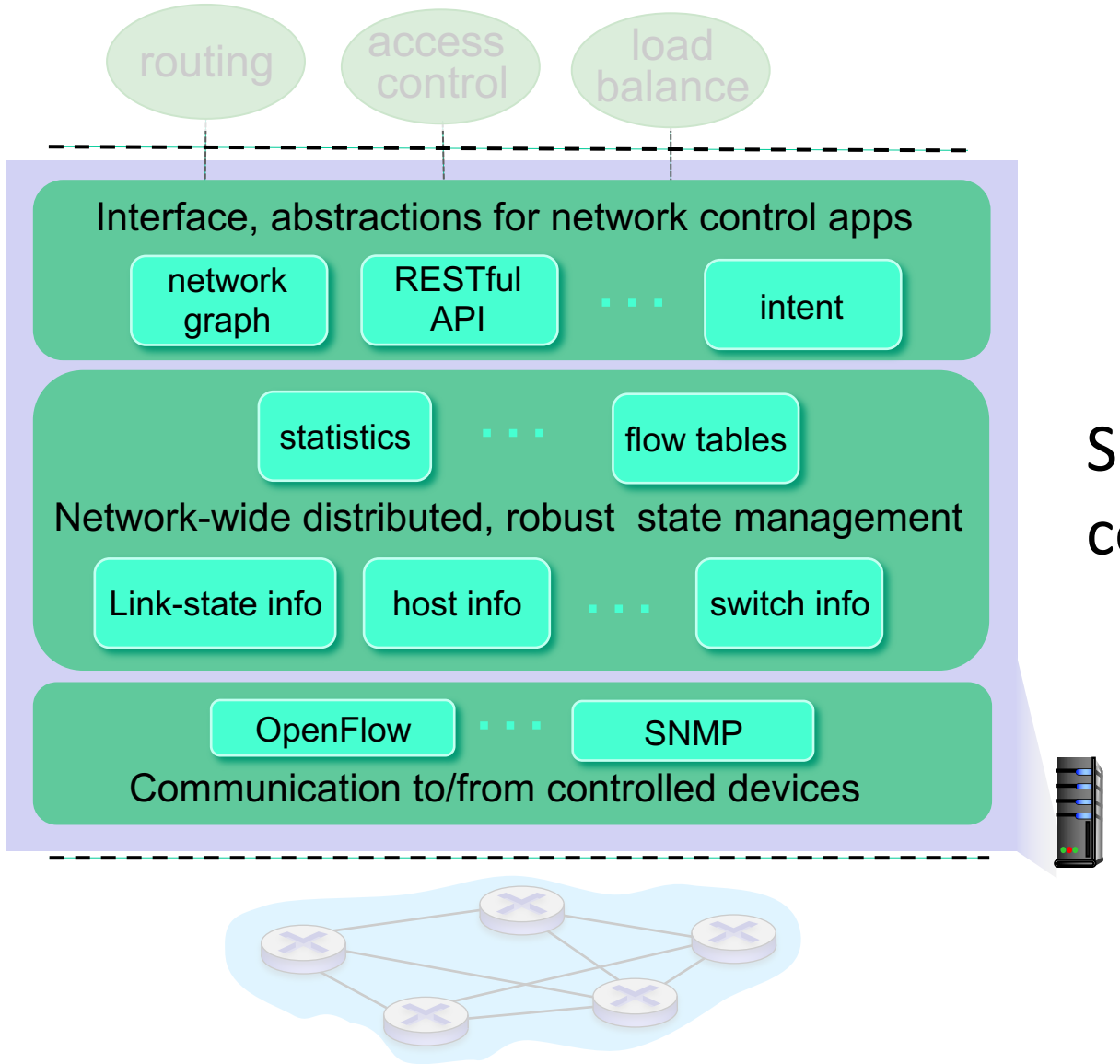


Components of SDN controller

interface layer to network control apps: abstractions API

network-wide state management : state of networks links, switches, services: a *distributed database*

communication: communicate between SDN controller and controlled switches

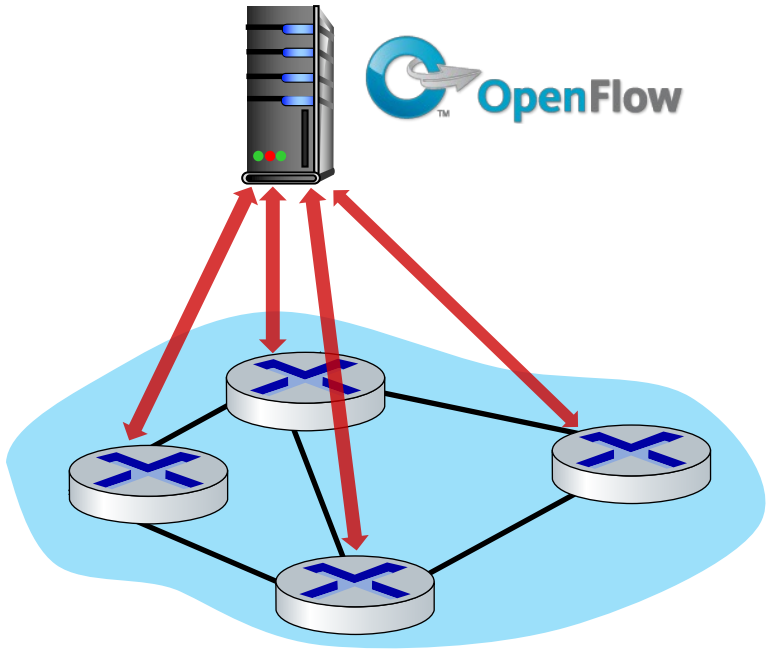


SDN controller

OpenFlow protocol

- operates between controller, switch
- TCP used to exchange messages
  - optional encryption
- three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc.)
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions

OpenFlow Controller

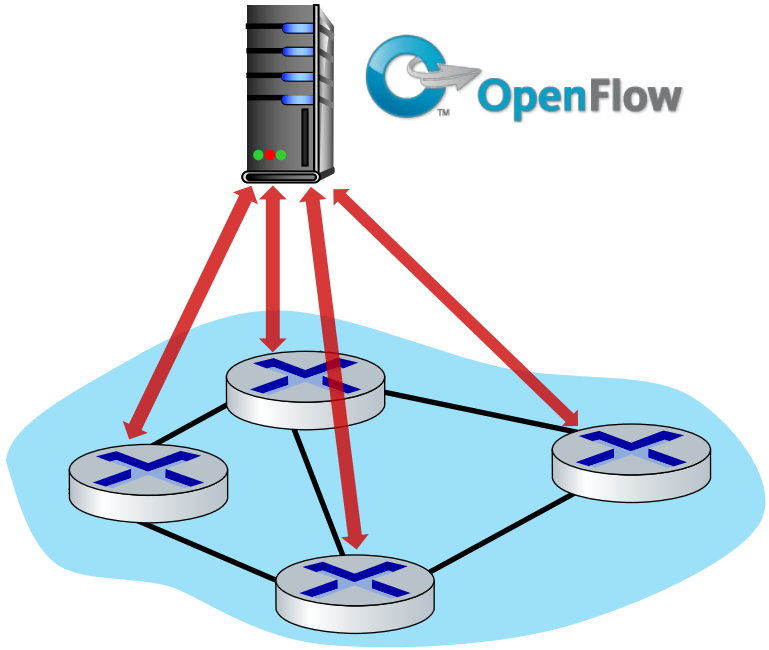


OpenFlow: controller-to-switch messages

Key controller-to-switch messages

- *features:* controller queries switch features, switch replies
- *configure:* controller queries/sets switch configuration parameters
- *modify-state:* add, delete, modify flow entries in the OpenFlow tables
- *packet-out:* controller can send this packet out of specific switch port

OpenFlow Controller

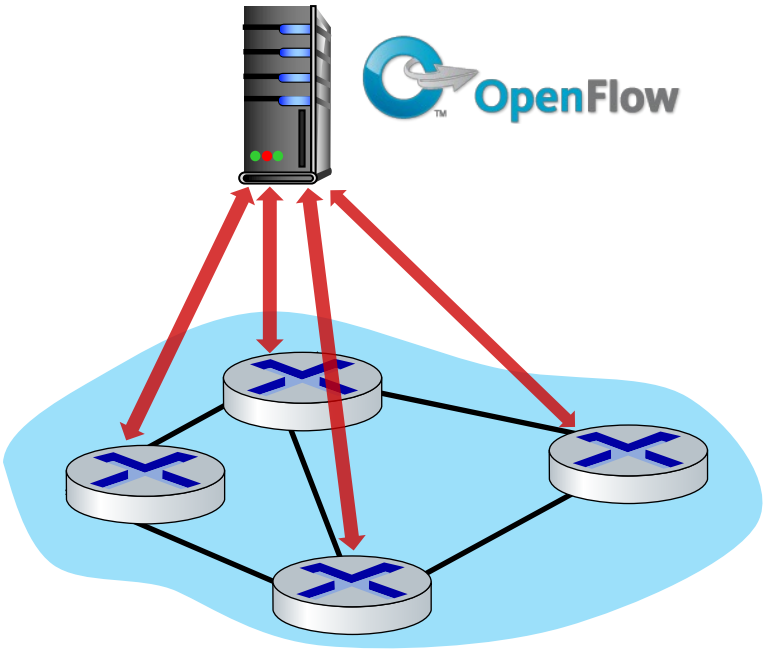


OpenFlow: switch-to-controller messages

Key switch-to-controller messages

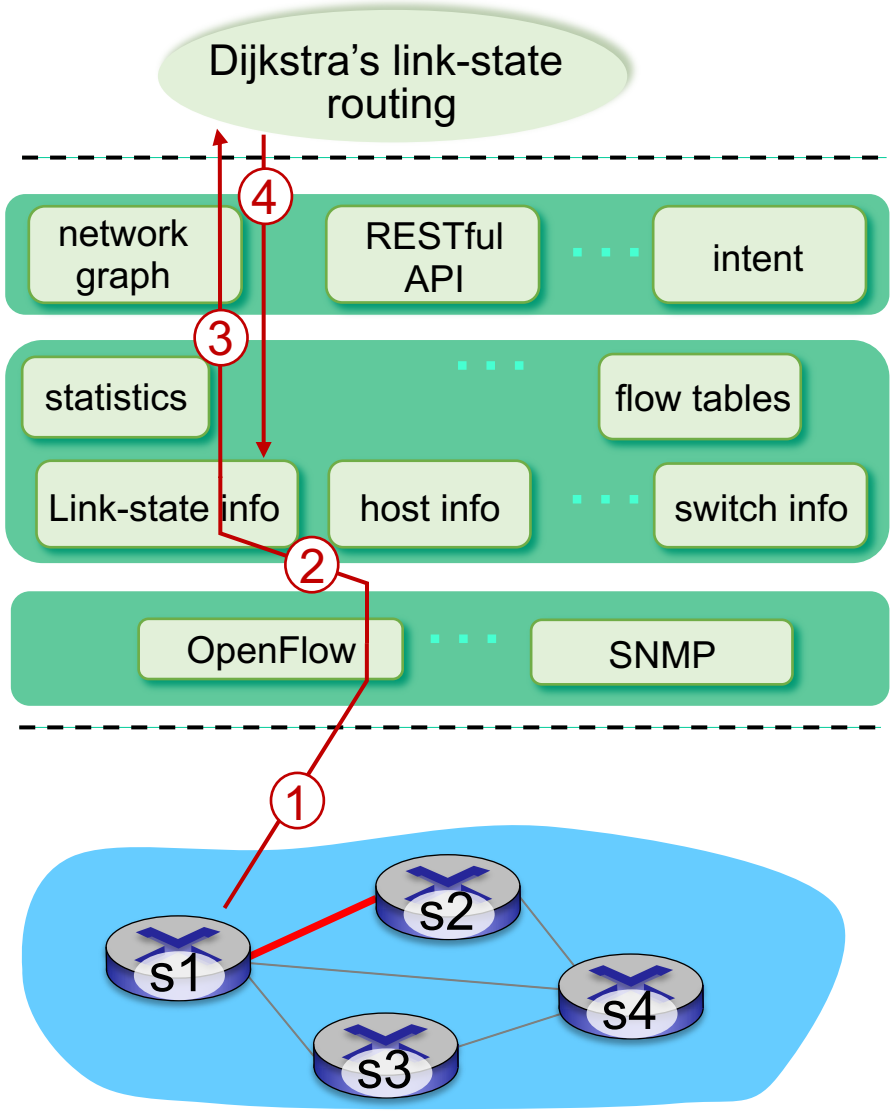
- *packet-in:* transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed:* flow table entry deleted at switch
- *port status:* inform controller of a change on a port.

OpenFlow Controller



Fortunately, network operators don't "program" switches by creating/sending OpenFlow messages directly. Instead use higher-level abstraction at controller

SDN: control/data plane interaction example



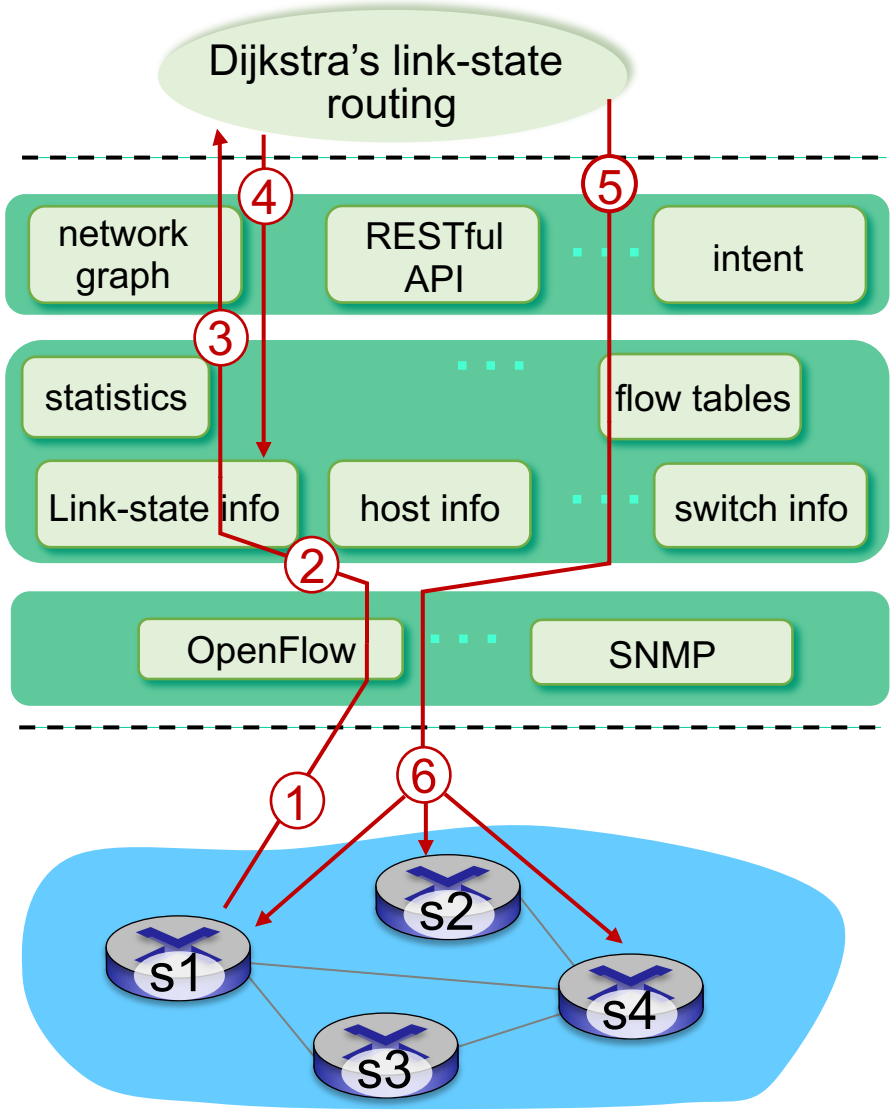
① S1, experiencing link failure uses OpenFlow port status message to notify controller

② SDN controller receives OpenFlow message, updates link status info

③ Dijkstra's routing algorithm application has previously registered to be called when ever link status changes. It is called.

④ Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes

SDN: control/data plane interaction example



⑤ link state routing app interacts with flow-table-computation component in SDN controller, which computes new flow tables needed

⑥ controller uses OpenFlow to install new tables in switches that need updating

SDN: selected challenges

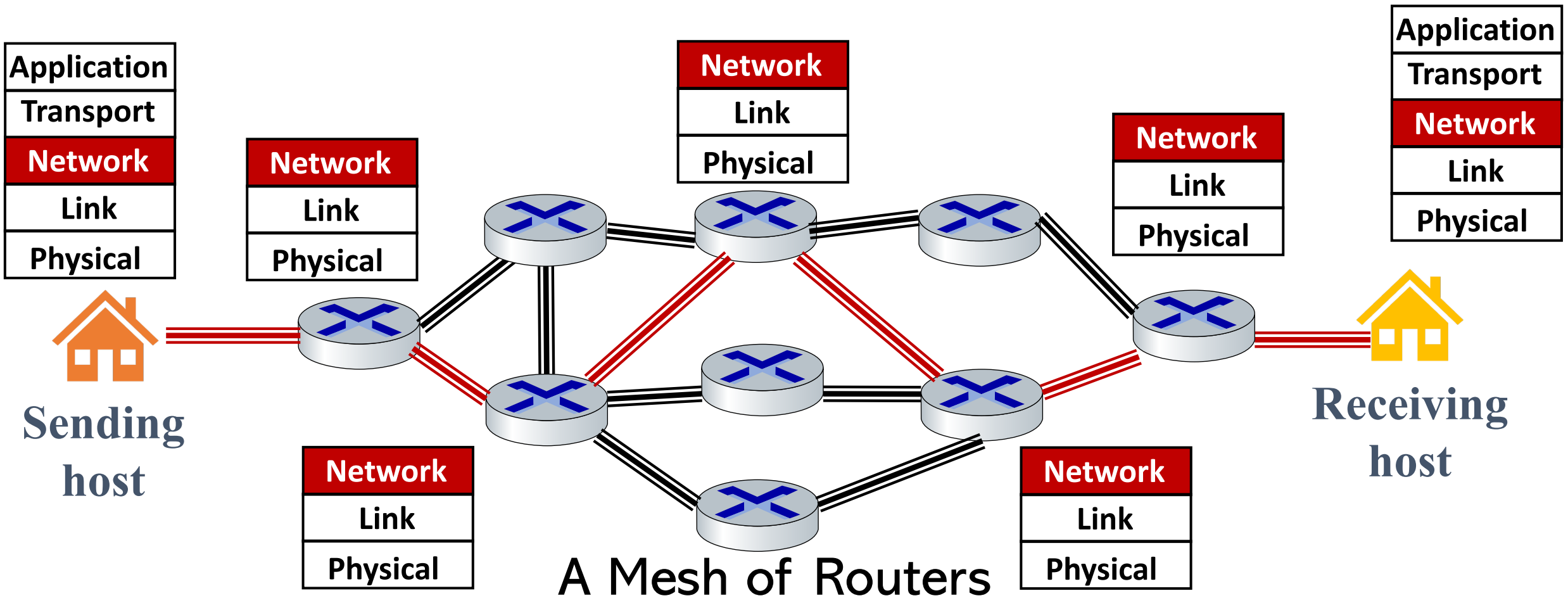
- hardening the control plane: dependable, reliable, performance-scalable, secure distributed system
  - robustness to failures: leverage strong theory of reliable distributed system for control plane
  - dependability, security: “baked in” from day one?
- networks, protocols meeting mission-specific requirements
  - e.g., real-time, ultra-reliable, ultra-secure
- Internet-scaling: beyond a single AS
- SDN critical in 5G cellular networks

Network layer: “control plane” roadmap

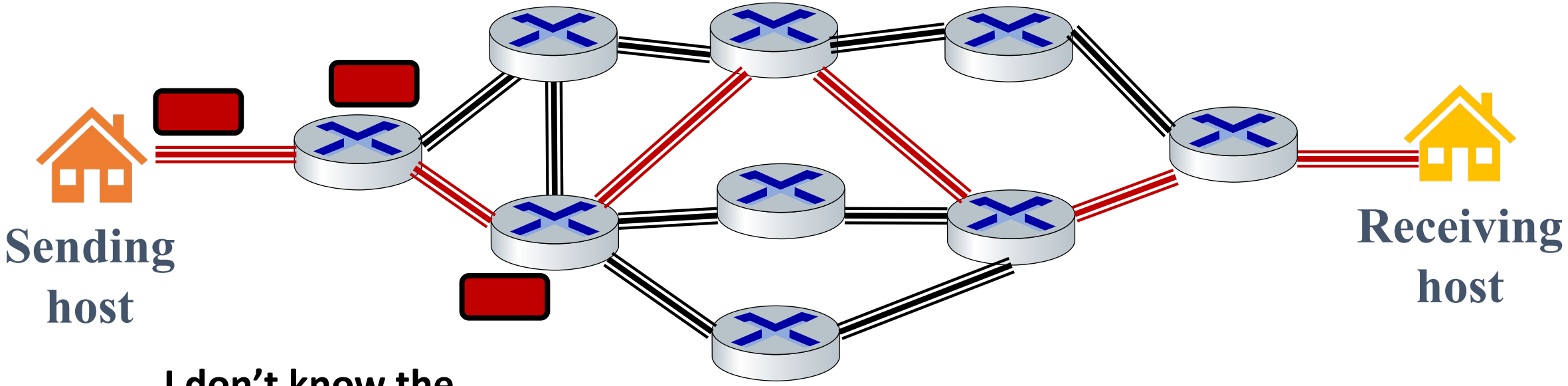
- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

- Measurement

End-to-end VS edge to router



End-to-end VS edge to router

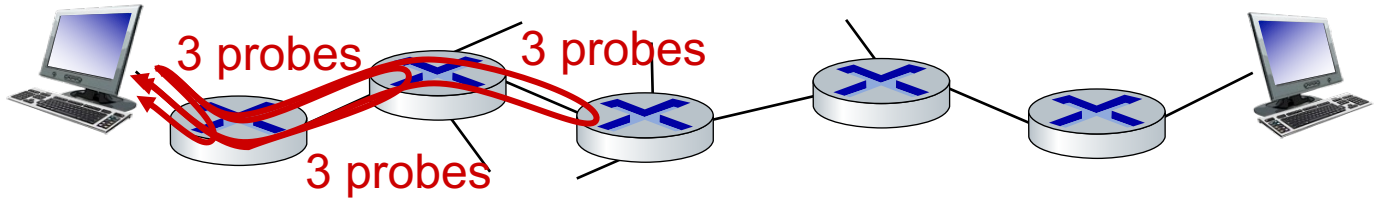


ICMP: internet control message protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP messages carried in IP datagrams
- *ICMP message:* type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP



- source sends sets of UDP segments to destination
  - 1st set has TTL =1, 2nd set has TTL=2, etc.
- datagram in *n*th set arrives to nth router:
  - router discards datagram and sends source ICMP message (type 11, code 0)
  - ICMP message possibly includes name of router & IP address
- when ICMP message arrives at source: record RTTs

stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops

Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

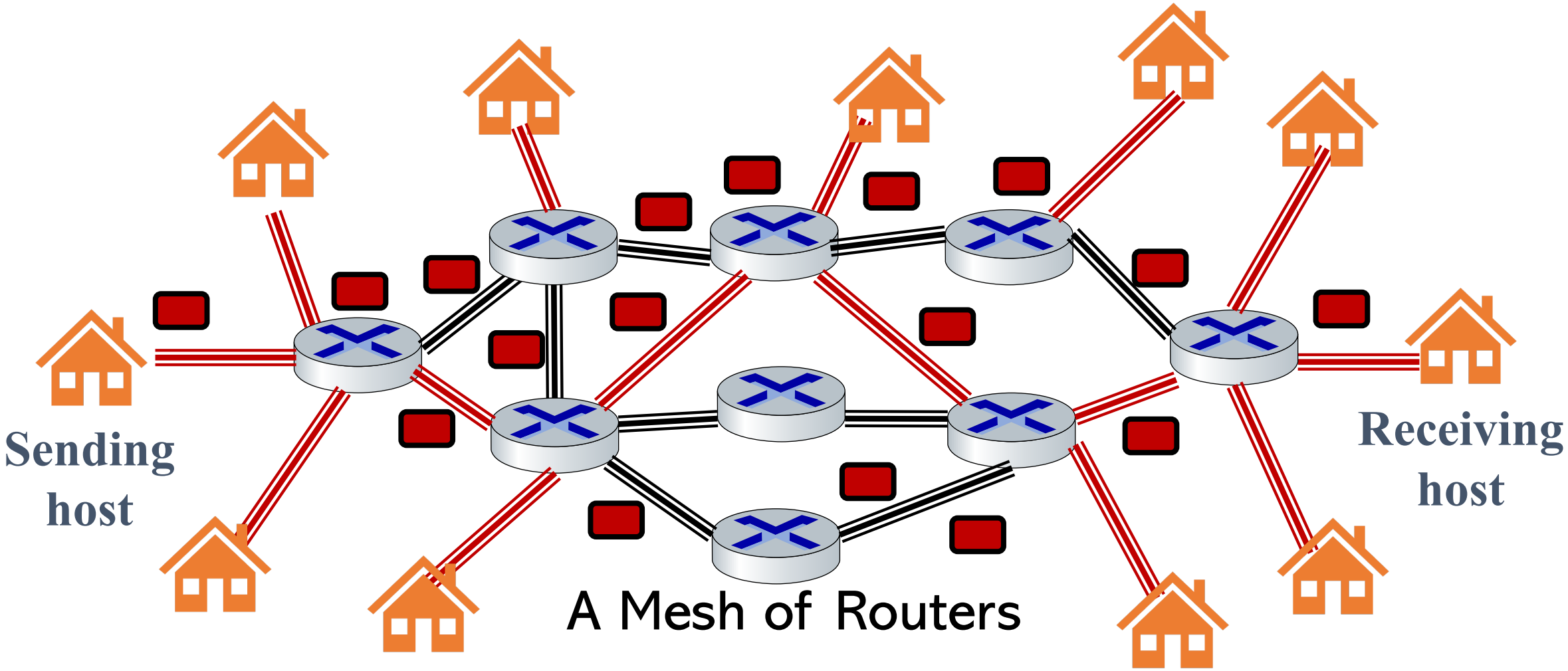
- Measurement

Why Measure the Network?

- Scientific discovery
  - Characterizing traffic, topology, performance
  - Understanding protocol performance and dynamics

- Network operation
  - Billing customers
  - Detecting, diagnosing, and fixing problem
  - Planning outlay of new equipment

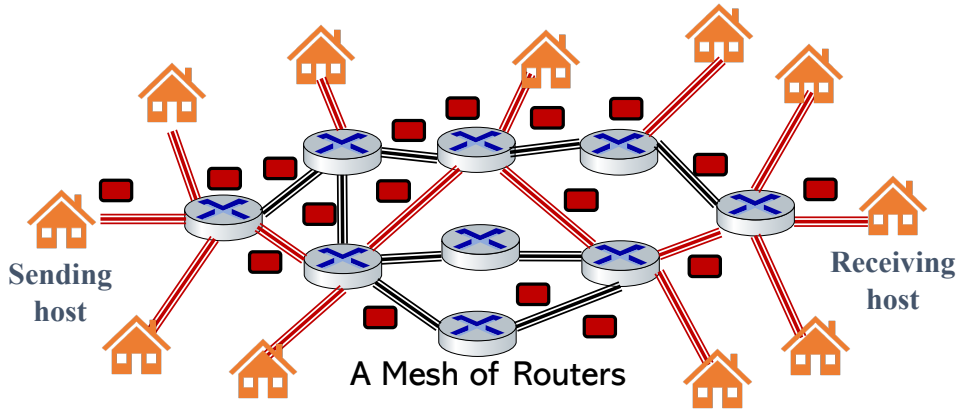
Where to Measure: Different Vantage Points



Where to Measure: Different Vantage Points

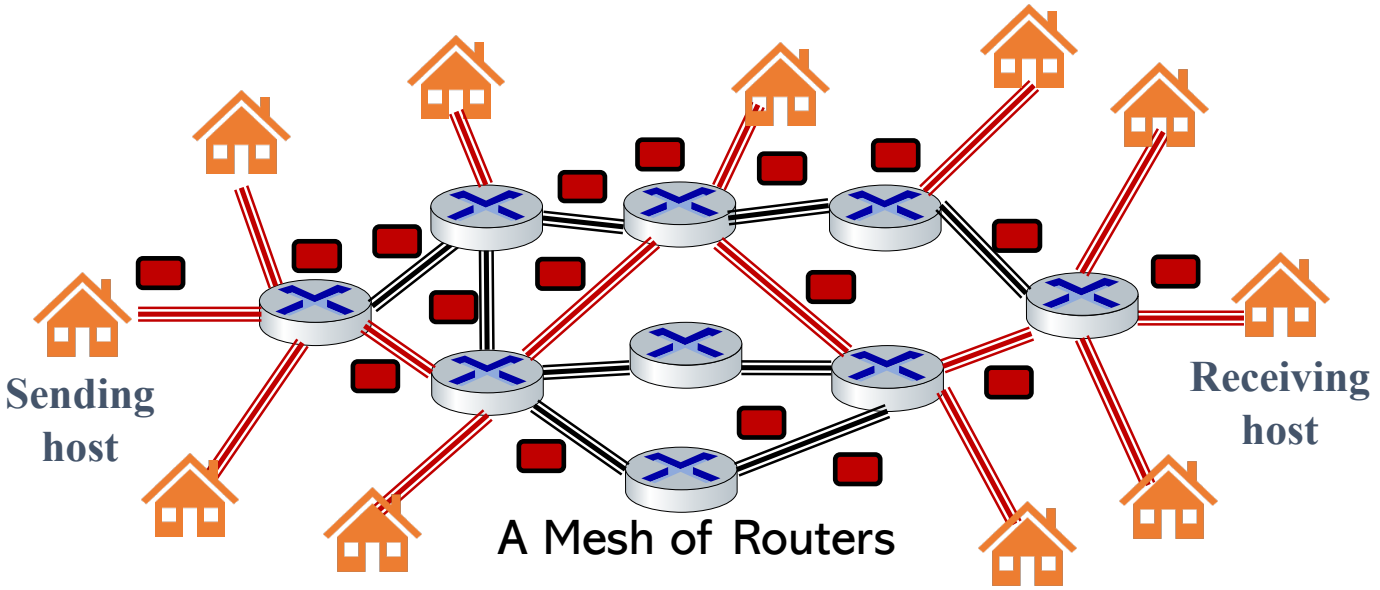
- End-host measurement
  - Injecting test traffic
  - Ping Traceroute
- Pros
  - Easy to implement
- Cons
  - Only end-to-end
  - Lack of insights to the network

- In-network measurement
  - router / switch / middlebox
  - Wireshark, NetFlow
- Pros
  - Deep visibility into network
- Cons
  - Hard to deploy
  - Requiring help from ISP



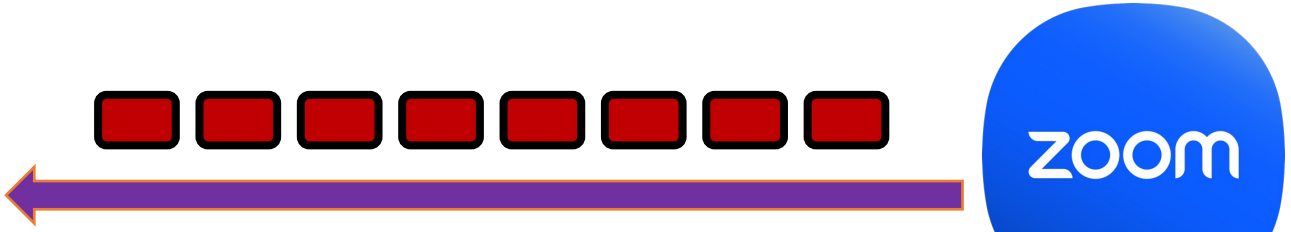
Packet level measurement

- Definition
  - Passively collecting IP packets on one or more links
  - Recording IP, TCP/UDP, or application layer traces
- Scope
  - Fine-grain information about user behavior
  - Characterizing traffic and diagnosing problems



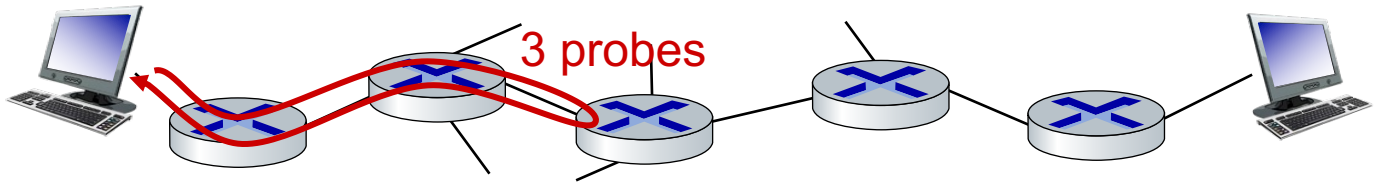
Application Level Measurement

- Diverse applications running atop of Internet
  - Youtube, Zoom, Web, ChatGPT
- Different App cares about different things
  - Video quality (resolution), Video stalls, Page loading time
- The performance of Apps varies across network conditions



Some Common Network Measurement Tools

- Traceroute
  - Per-hop (router) RTT
- Ping
  - RTT between two end-hosts (edge devices)
- Iperf
  - Throughput between two end-hosts
- tcpdump / Wireshark
  - Record every packet one interface transmits



Network layer: Summary

we've learned a lot!

- approaches to network control plane
  - per-router control (traditional)
  - logically centralized control (software defined networking)
- traditional routing algorithms
  - implementation in Internet: OSPF , BGP
- SDN controllers
  - implementation in practice: ODL, ONOS
- Internet Control Message Protocol
- network measurement

next stop: link layer!

Network layer, control plane: Done!

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

- network measurement