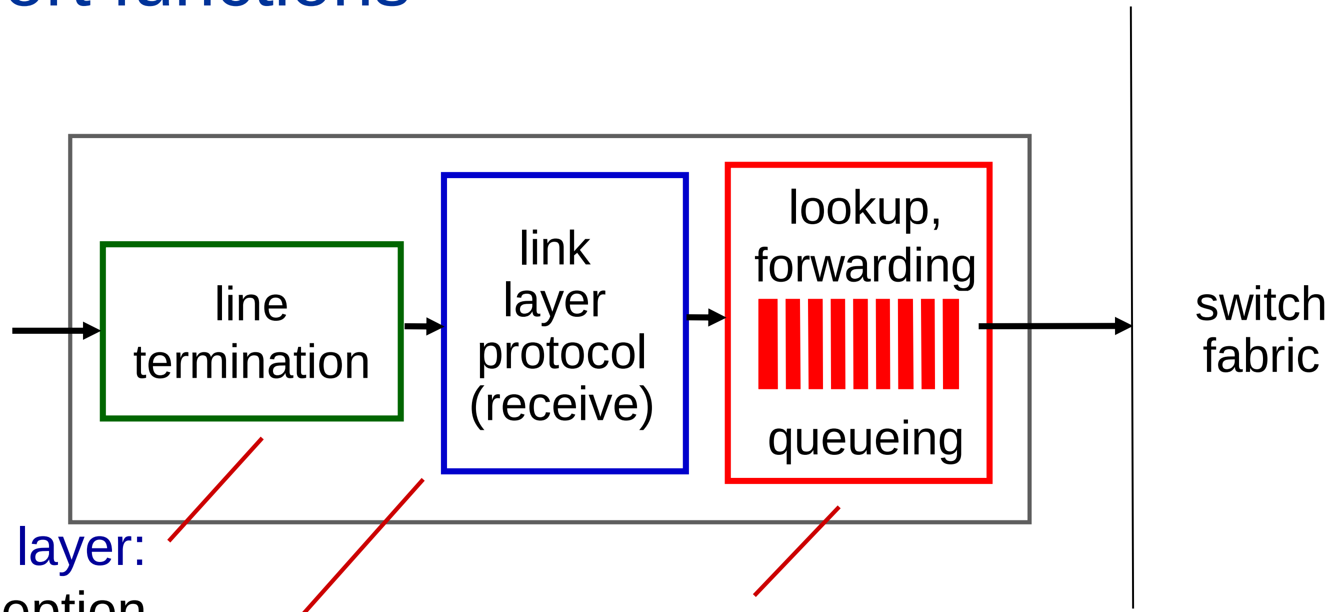
Packet-switching networks
Outline

- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What's inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

Input port functions

(Recall)



physical layer:
bit-level reception

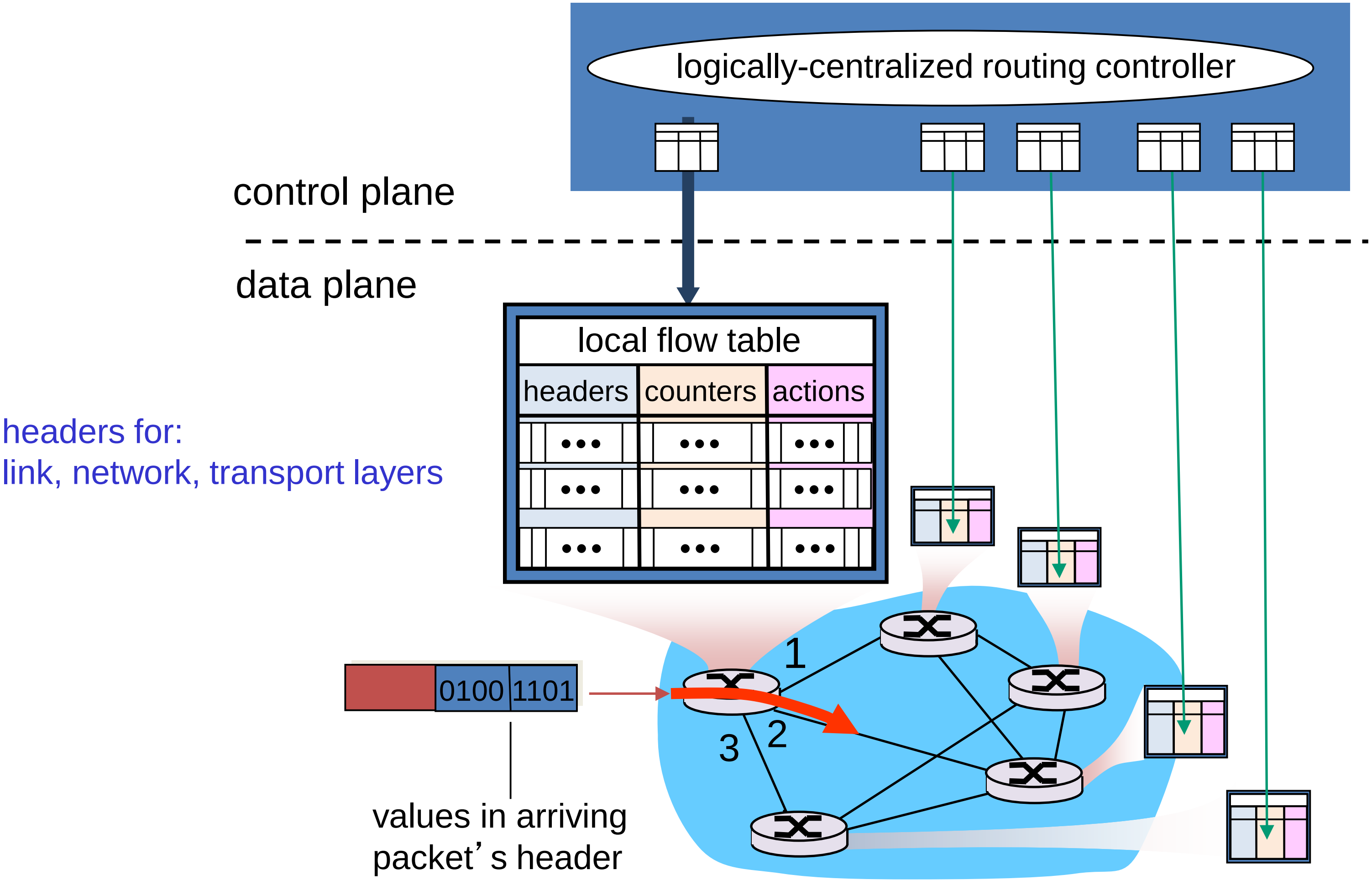
data link layer:
e.g., Ethernet

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- *destination-based forwarding:* forward based only on destination IP address (traditional)
- *generalized forwarding:* forward based on any set of header field values

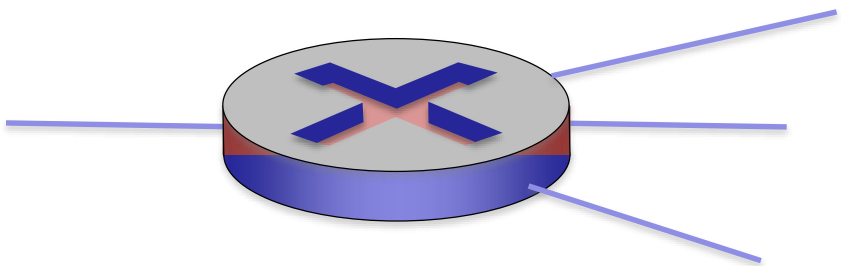
Generalized Forwarding and SDN

Each router contains a *flow table* that is computed and distributed by a *logically centralized* routing controller



OpenFlow data plane abstraction

- *flow*: defined by header fields (for link, network, transport layers)
- generalized forwarding: simple packet-handling rules
  - *Pattern:* match values in packet header fields
  - *Actions: for matched packet:* drop, forward, modify the packet or send it to controller
  - *Priority*: disambiguate overlapping patterns
  - *Counters:* #bytes and #packets



Flow table in a router (computed and distributed by controller) define router's match+action rules

* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow: Flow Table Entries

Rule	Action	Stats

Packet + byte counters

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline
5. Modify Fields

Switch Port	VLAN ID	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP Prot	TCP sport	TCP dport

Link layer

Network layer

Transport layer

Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	port3

layer 2 frames with dest. MAC address 22:A7:23:11:E1:02 should be forwarded to output port 3

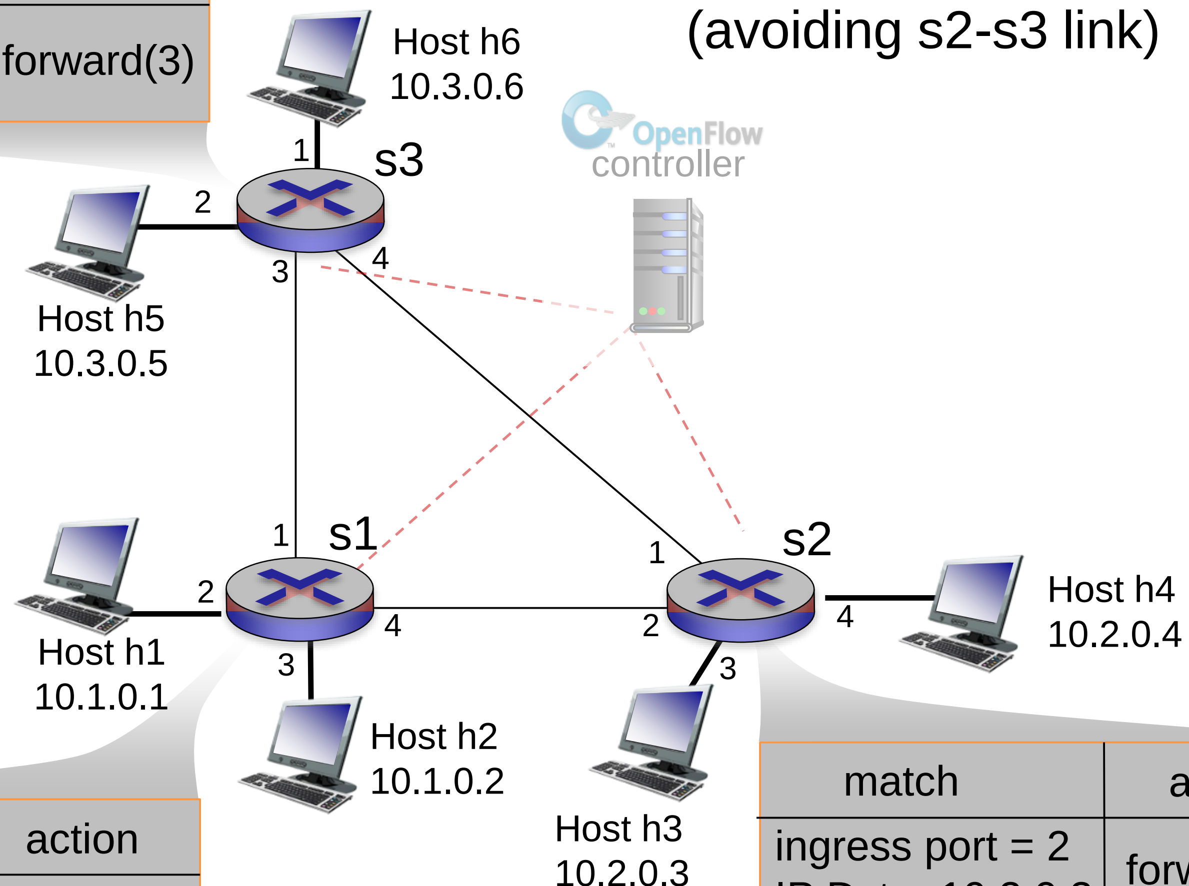
OpenFlow abstraction

- *match + action:* unifies different kinds of devices

- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

OpenFlow example

Example: datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2 (avoiding s2-s3 link)



match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)

match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

Question: How do forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

Answer: by the control plane (Recall: Interplay between routing and forwarding).

Packet-switching networks
Outline

- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What's inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

Routing Algorithm classification

Global or decentralized information?

Global:

- all routers have complete topology, link cost info
- "link state" protocols

Decentralized:

- router knows physically-connected neighbors, link costs to neighbors
- iterative process of computation, exchange of info with neighbors
- "distance vector" protocols

Static or dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

Hierarchical Routing

scale: with billions of destinations:

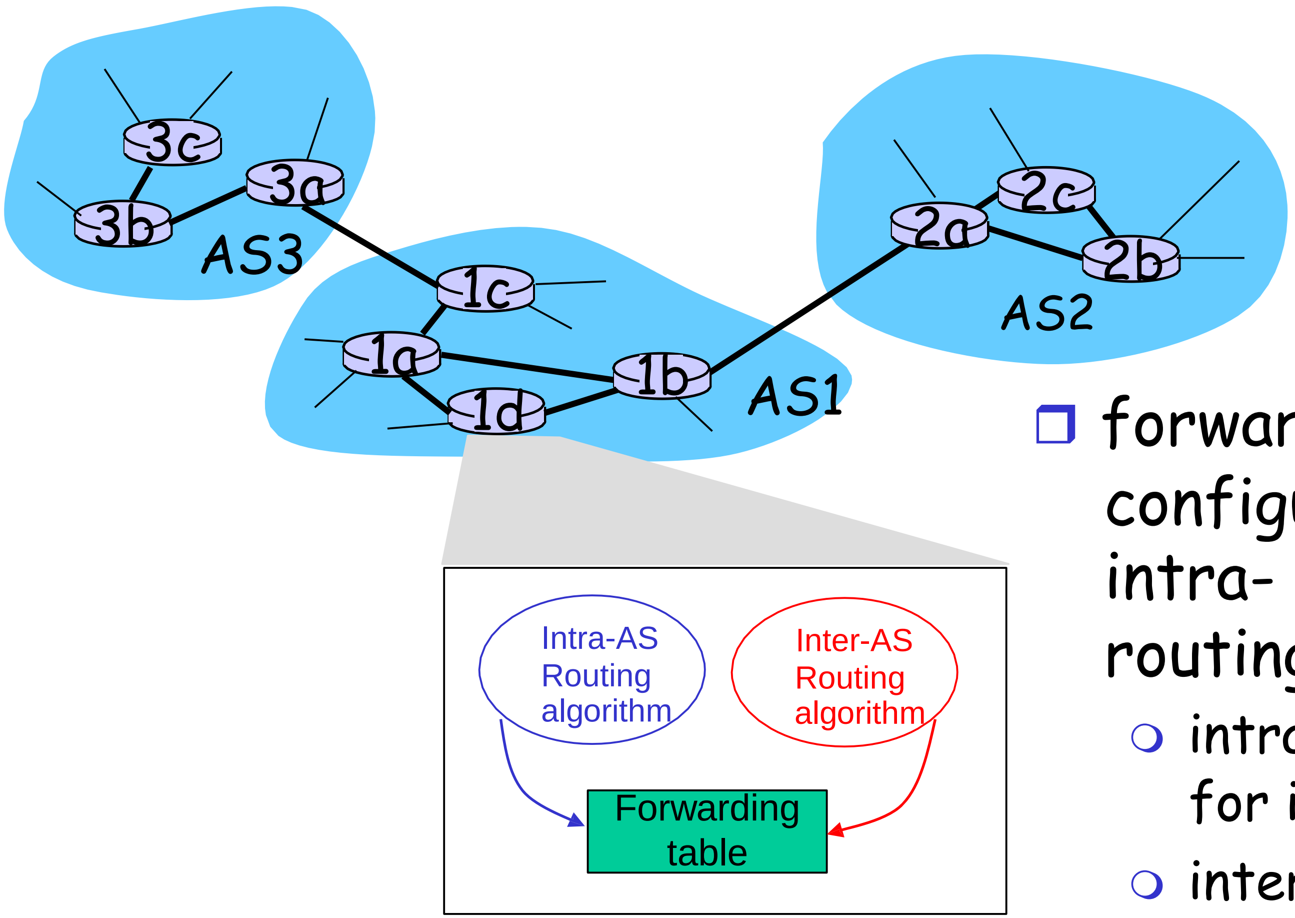
- can't store all dest's in routing tables!
- routing table exchange would swamp links!

administrative autonomy

- internet = network of networks
- each network admin may want to control routing in its own network

- Aggregate routers into regions known as "autonomous systems" (AS) (a.k.a. domains).
- An autonomous system (AS) is a collection of routers under the same administrative and technical control.
- Routers in the same AS run (typically) same routing protocol
  - intra-AS routing protocol
  - Routers in different AS can run different intra-AS routing protocol
- Between AS's: inter-AS routing protocol
- Gateway router: Direct link(s) to router(s) in other AS'es.

Interconnected ASes



- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS sets entries for internal dests
  - inter-AS & intra-As sets entries for external dests

Intra-AS Routing

- also known as Interior Gateway Protocols (IGP)
- most common Intra-AS routing protocols:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First (IS-IS protocol essentially same as OSPF)
  - EIGRP (Cisco): Enhanced Interior Gateway Routing Protocol

Inter-AS Routing

Inter-AS routing protocol in today's Internet:

BGP: Border Gateway Protocol

Packet-switching networks
Outline

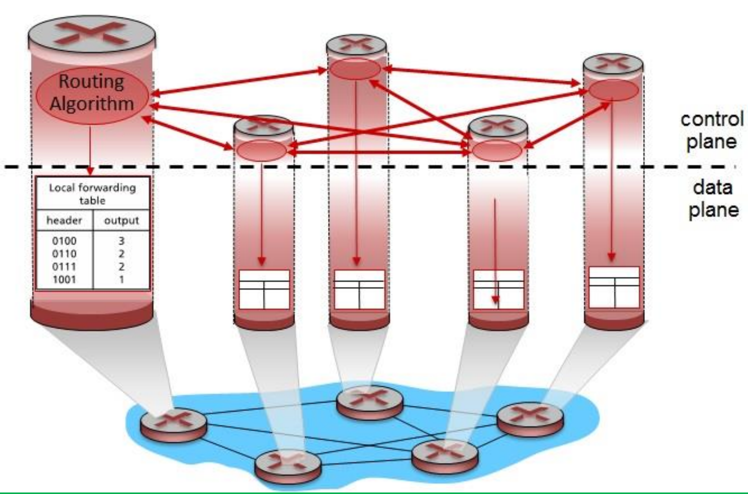
- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What’s inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

Software defined networking (SDN)

- Internet network layer: historically has been implemented via distributed, per-router approach
  - *monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different “middleboxes” for different network layer functions: firewalls, load balancers, NAT boxes, ..
- ~2005: renewed interest in rethinking network control plane

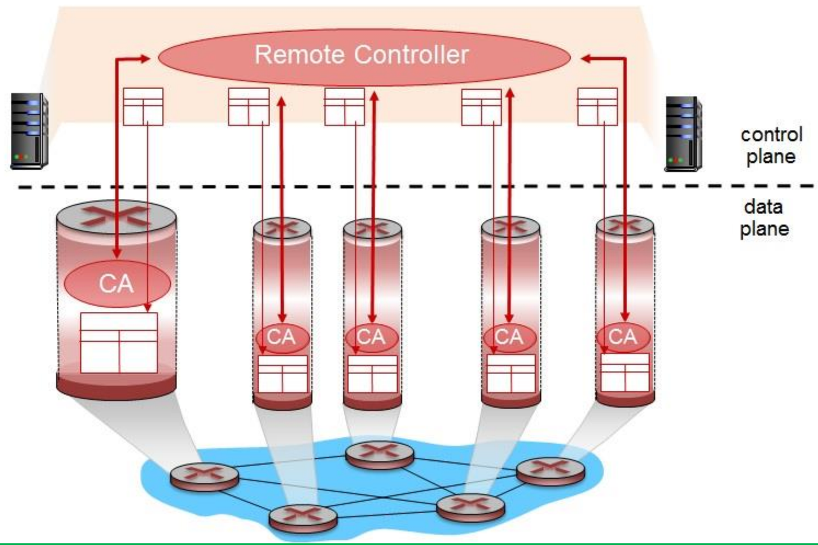
Recall: per-router control plane

Individual routing algorithm components *in each and every router* interact with each other in control plane to compute forwarding tables



Recall: logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables

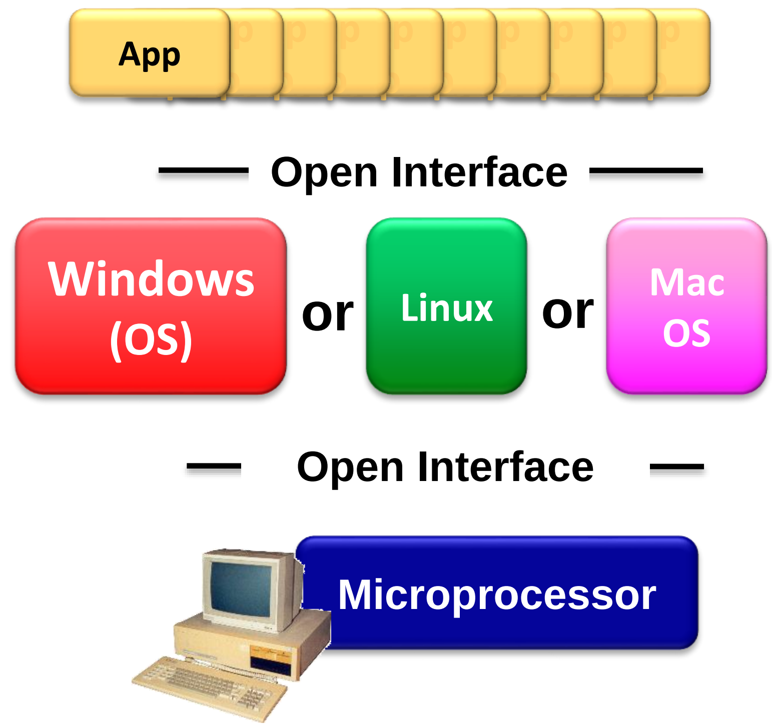


Software defined networking (SDN)

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 (orchestration): 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

Analogy: mainframe to PC evolution

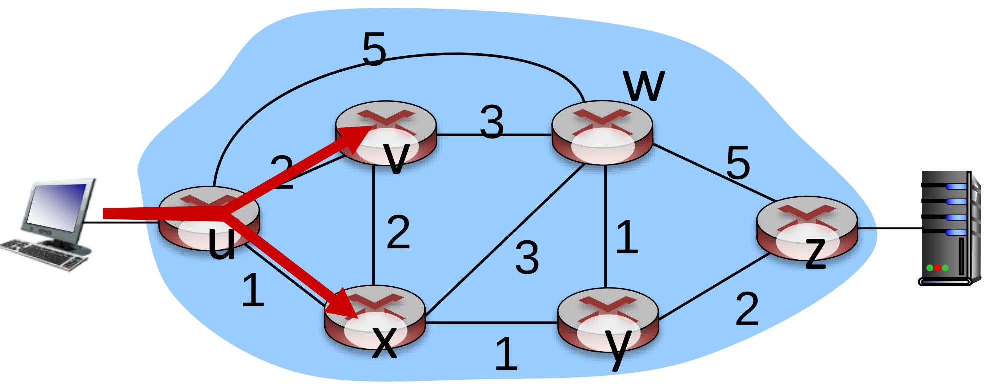


Vertically integrated
Closed, proprietary
Slow innovation
Small industry

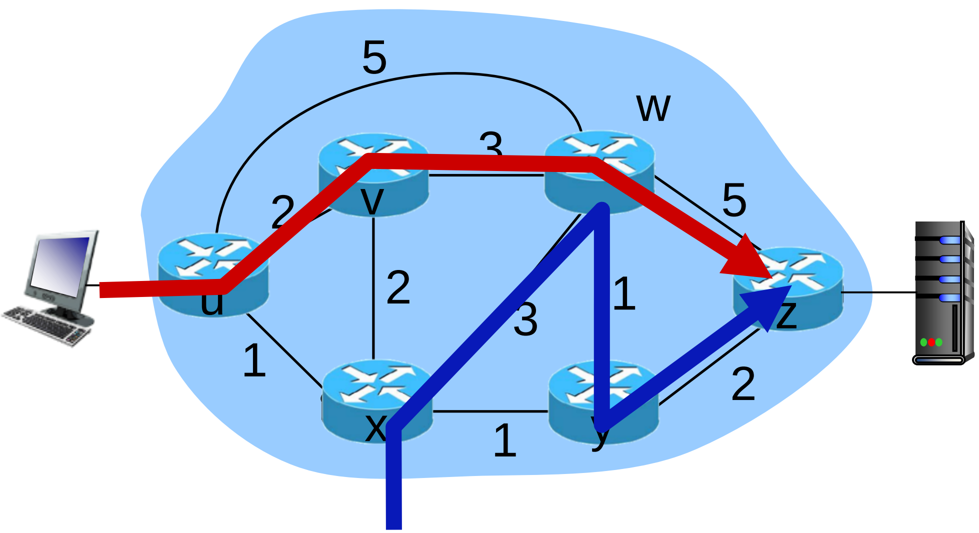
Horizontal
Open interfaces
Rapid innovation
Huge industry

slide courtesy: N. McKeown

Traffic engineering: difficult



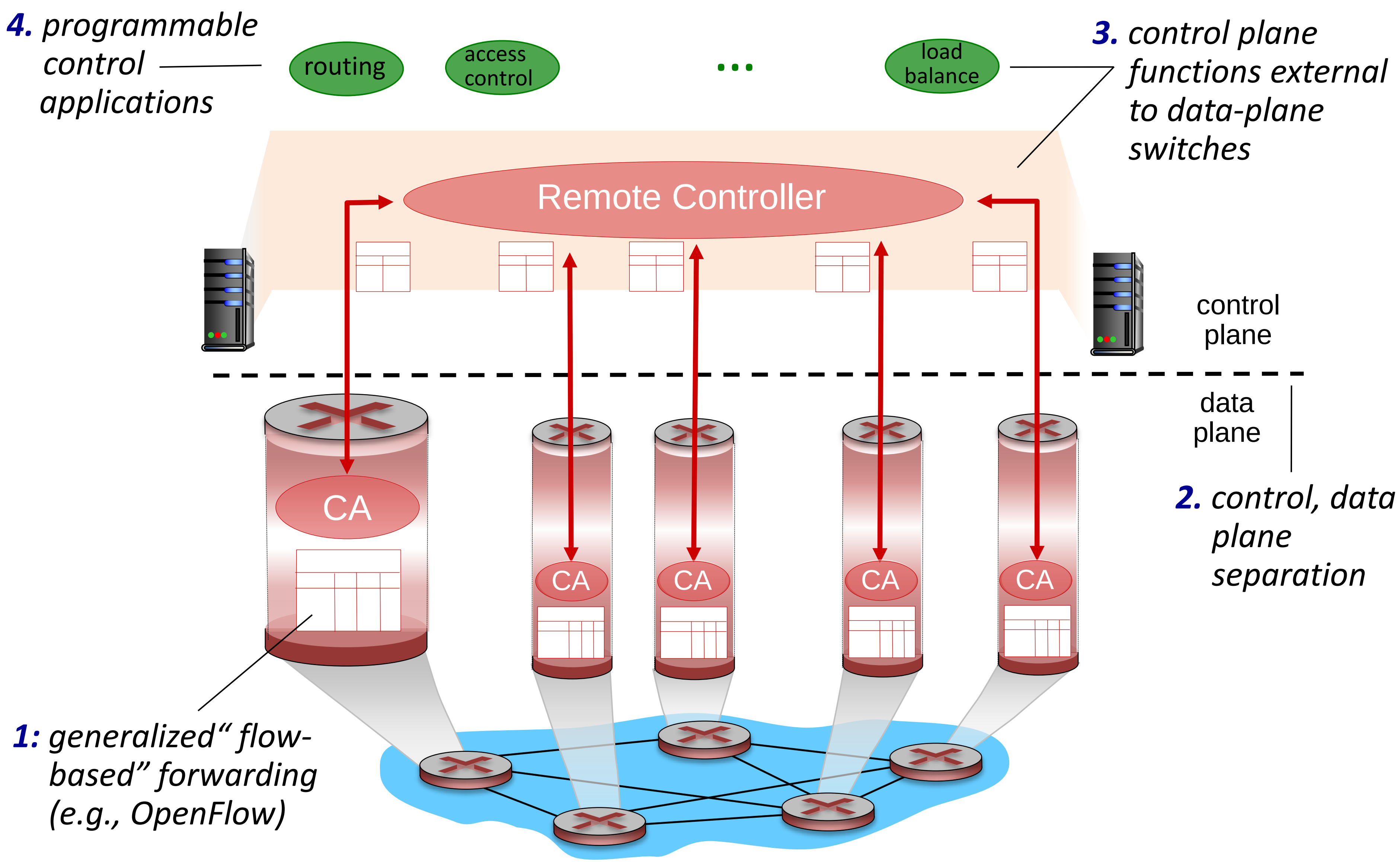
Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?
A: can't do it using current (traditional) protocols based on shortest-path computation (or need a new routing algorithm)



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

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

Software defined networking (SDN)



Layers in SDN architecture

Network-control apps

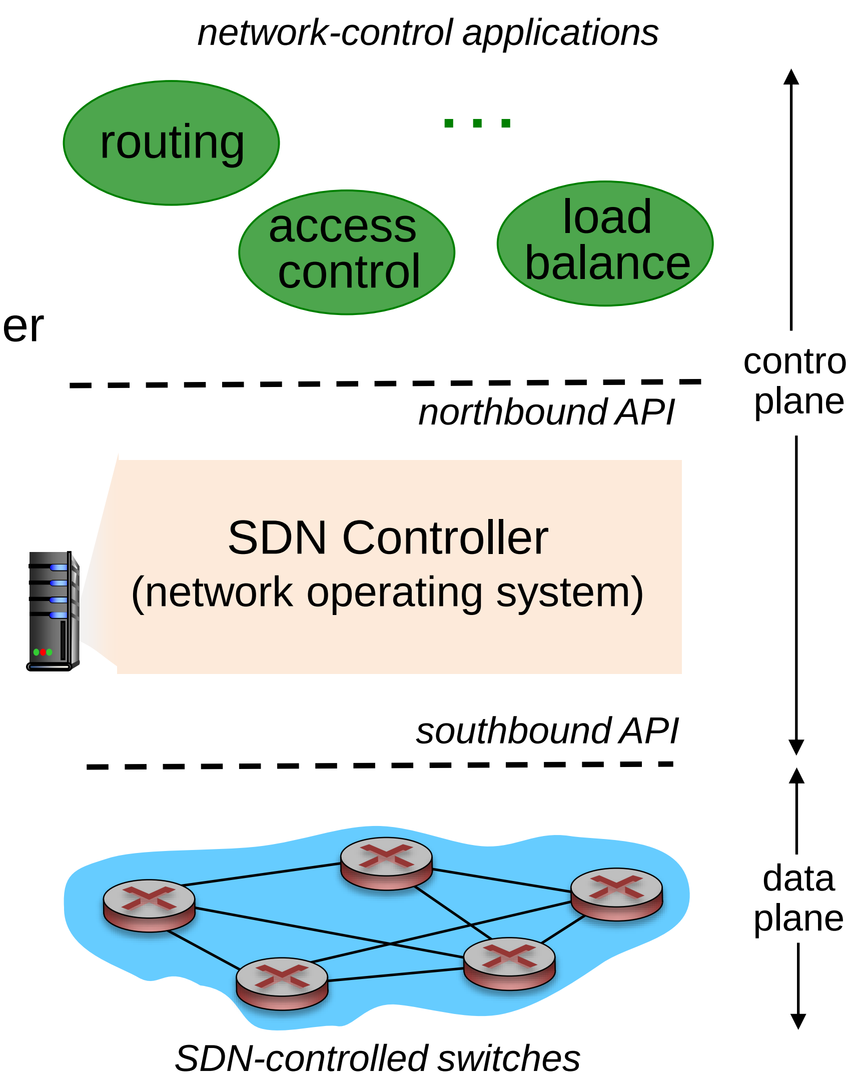
- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- use network state information provided by SDN controller
- unbundled: can be provided by 3rd party: distinct from routing vendor, or SDN controller

SDN controller (network OS)

- maintains network state information, statistics, and flow tables
- (often) implemented as distributed system for performance and fault-tolerance

Data plane switches

- fast, simple, commodity switches
- implementing generalized data-plane forwarding in hardware
- communicate with controller (using protocols such as 'OpenFlow') to provide state information and receive flow tables

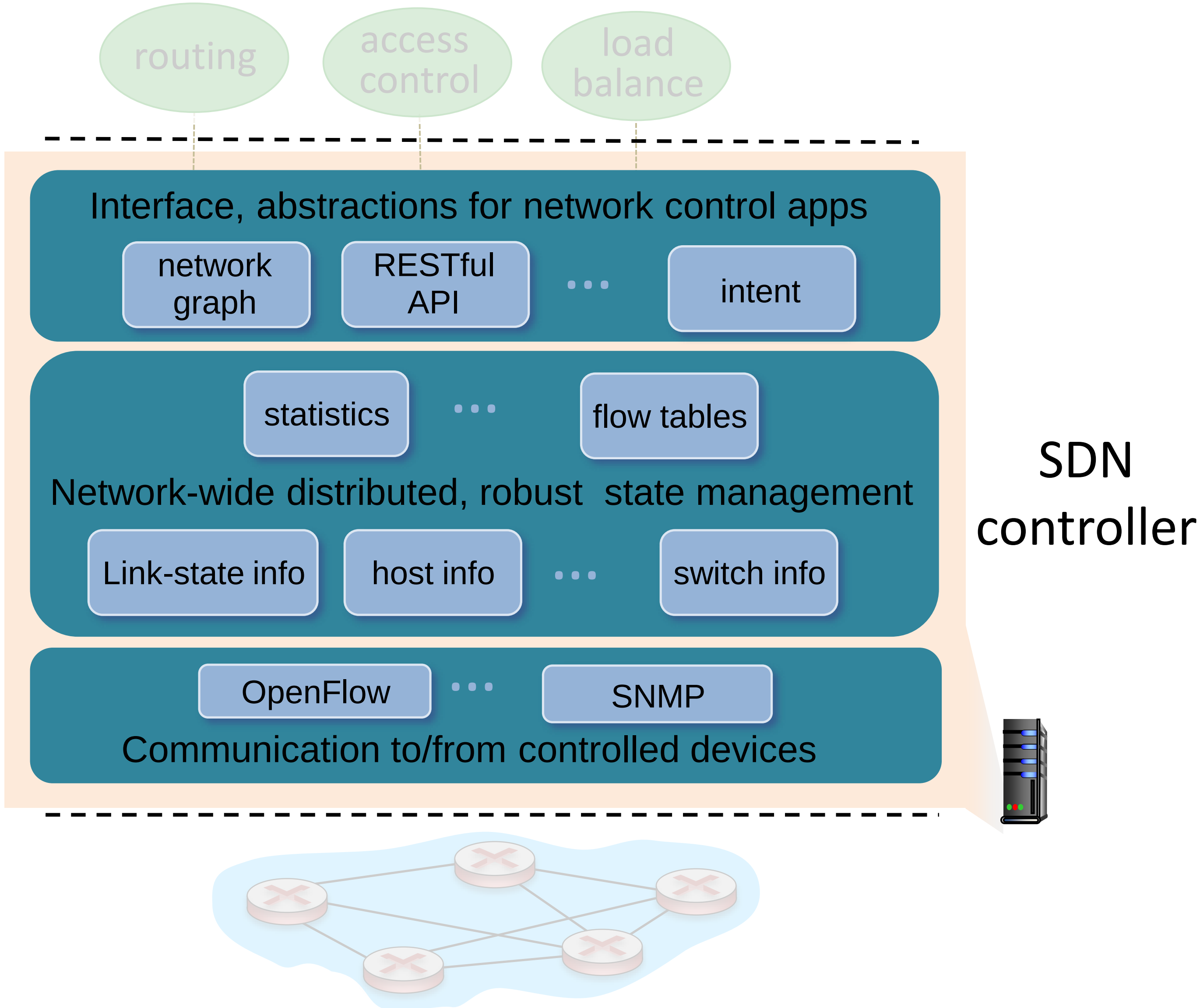


Components of SDN controller

Interface layer to network control apps: abstractions API

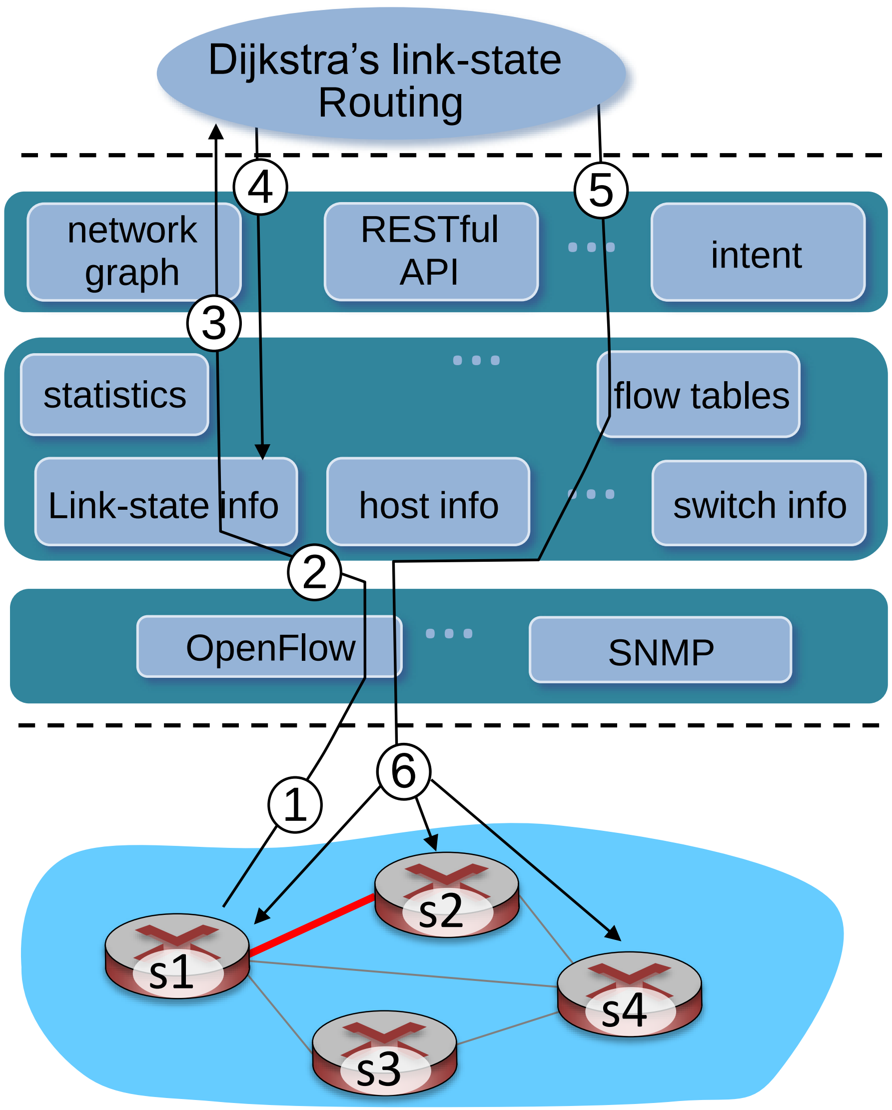
Network-wide state management layer: state of networks links, switches, services: a *distributed database*

communication layer: communicate between SDN controller and controlled switches



Example controllers: OpenDaylight (ODL) controller, ONOS controller, etc.

SDN: control/data plane interaction example



1. S1, experiencing link failure using 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
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

Observation: SDN is primarily for networks under the same administrative control.

Packet-switching networks
Outline

- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What’s inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

ICMP: Internet Control Message Protocol

Ping, Traceroute: use ICMP

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs 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

Used in 'traceroute'

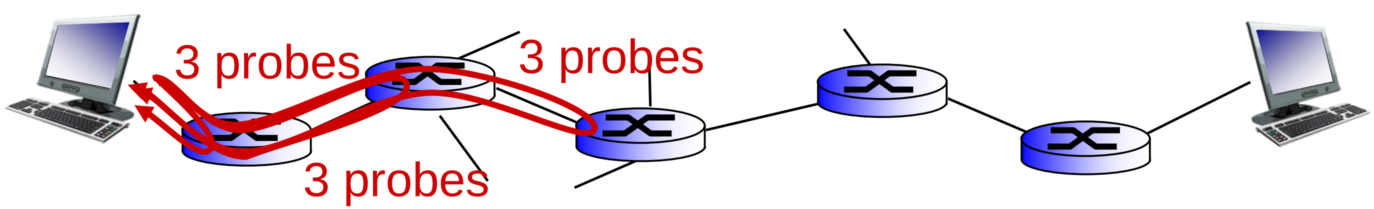
Traceroute and ICMP

Traceroute uses ICMP and UDP

- Source sends series of UDP segments to dest
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram
  - And sends to source an ICMP message (type 11, code 0)
  - Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion

- UDP segment eventually arrives at destination host
- Destination returns ICMP "port unreachable" packet (type 3, code 3)
- When source gets this ICMP, stops.



* Learn about other network tools such as 'netstat', 'tcpdump', 'ifconfig' (or 'ipconfig'), … (Lab1)