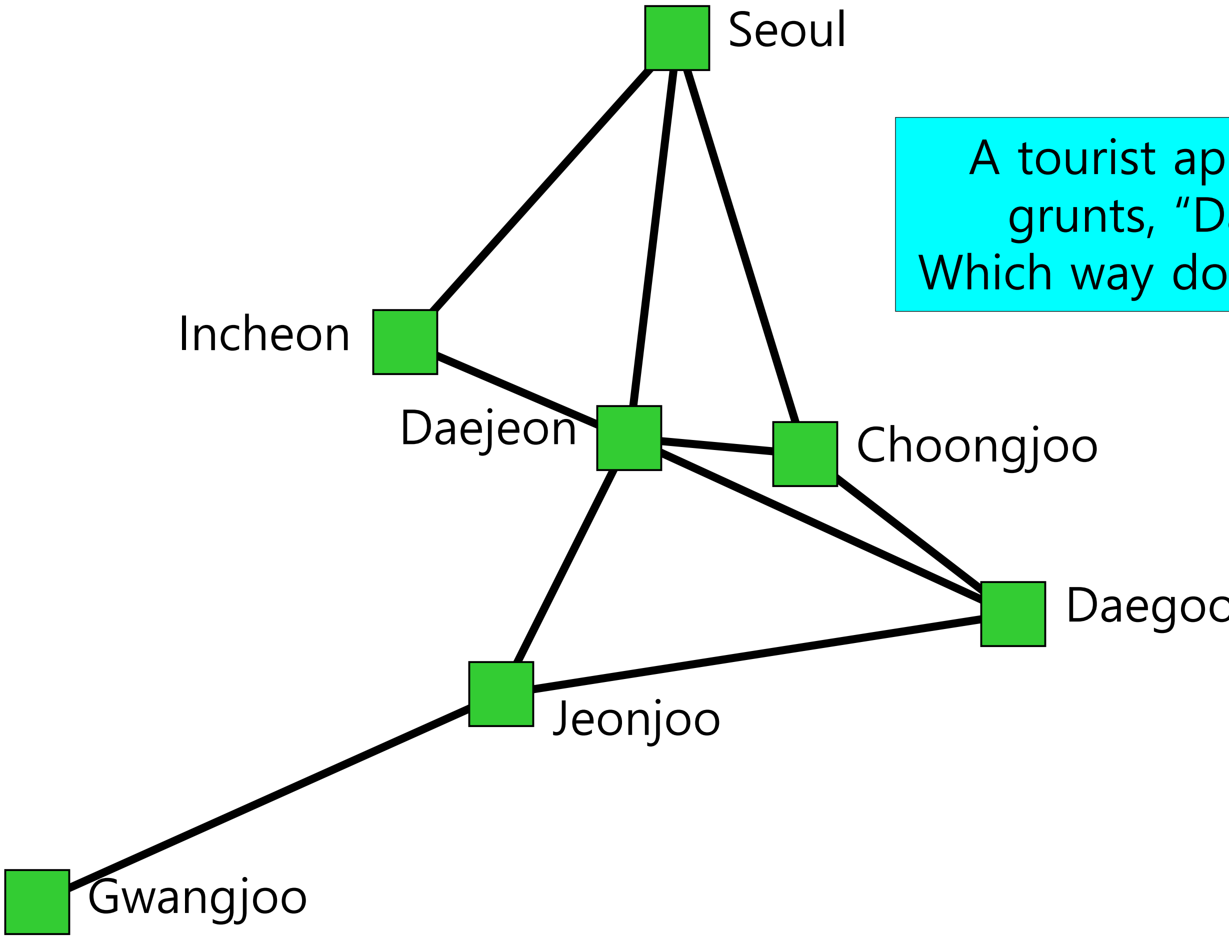
ITH508 컴퓨터망

Network Layer Control Plane

Hwangnam Kim
hnkim@korea.ac.kr
School of Electrical Engineering
Korea University

ROUTING PROTOCOLS

Routing



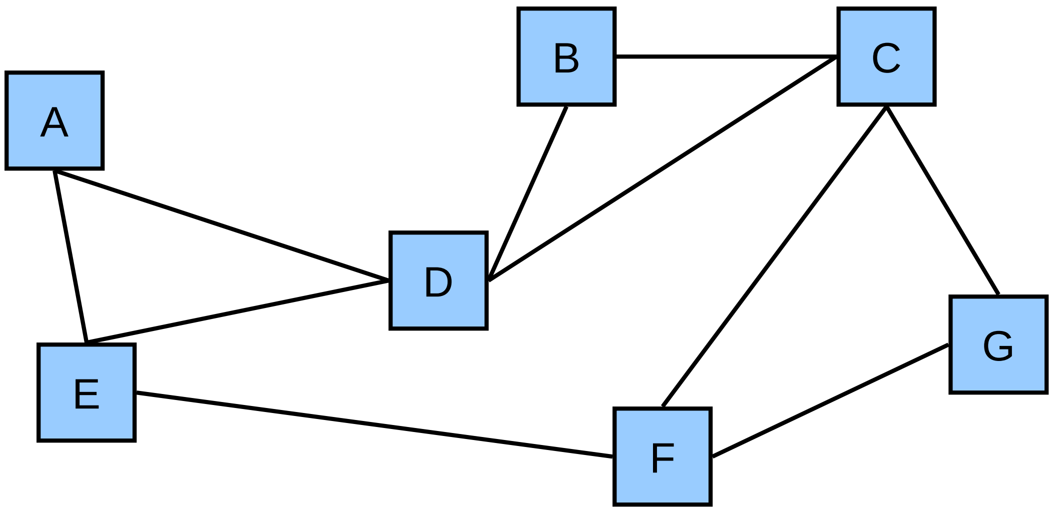
A tourist appears and grunts, "Daegoo?" Which way do you point?

Routing

- **Definition**
  - The task of constructing and maintaining forwarding information (**in hosts or routers**)
- **Goals**
  - Capture the notion of "best" routes
  - Propagate changes effectively
  - Require limited information exchange

Routing: Ideal Approach

- Maintain information about each link
- Calculate fastest path between each directed pair



For each direction, maintain:

- •Bandwidth
- •Latency
- •Queueing delay

Routing: Ideal Approach

- **Problems**
  - Unbounded amount of information
  - Queueing delay can change rapidly
  - Graph connectivity can change rapidly
- **Solution**
  - Dynamic
    - Periodically recalculate routes
  - Distributed
    - No single point of failure
    - Reduced computation per node
  - Abstract Metric
    - "Distance" may combine many factors
    - Use heuristics

Routing Overview

- **Algorithms**
  - **Static** shortest path algorithms
    - **Bellman-Ford**
      - Based on local iterations
    - **Dijkstra's algorithm**
      - Build tree from source
  - **Distributed, dynamic** routing algorithms
    - **Distance vector routing**
      - Distributed Bellman-Ford
    - **Link state routing**
      - Implement Dijkstra's algorithm at each node

SCALABLE ROUTING

Hierarchical Routing

Routing study thus far - idealized
- all routers identical
- network "flat" but *not* true in practice

Scale: with 200 million destinations:
- Cannot store all destinations 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

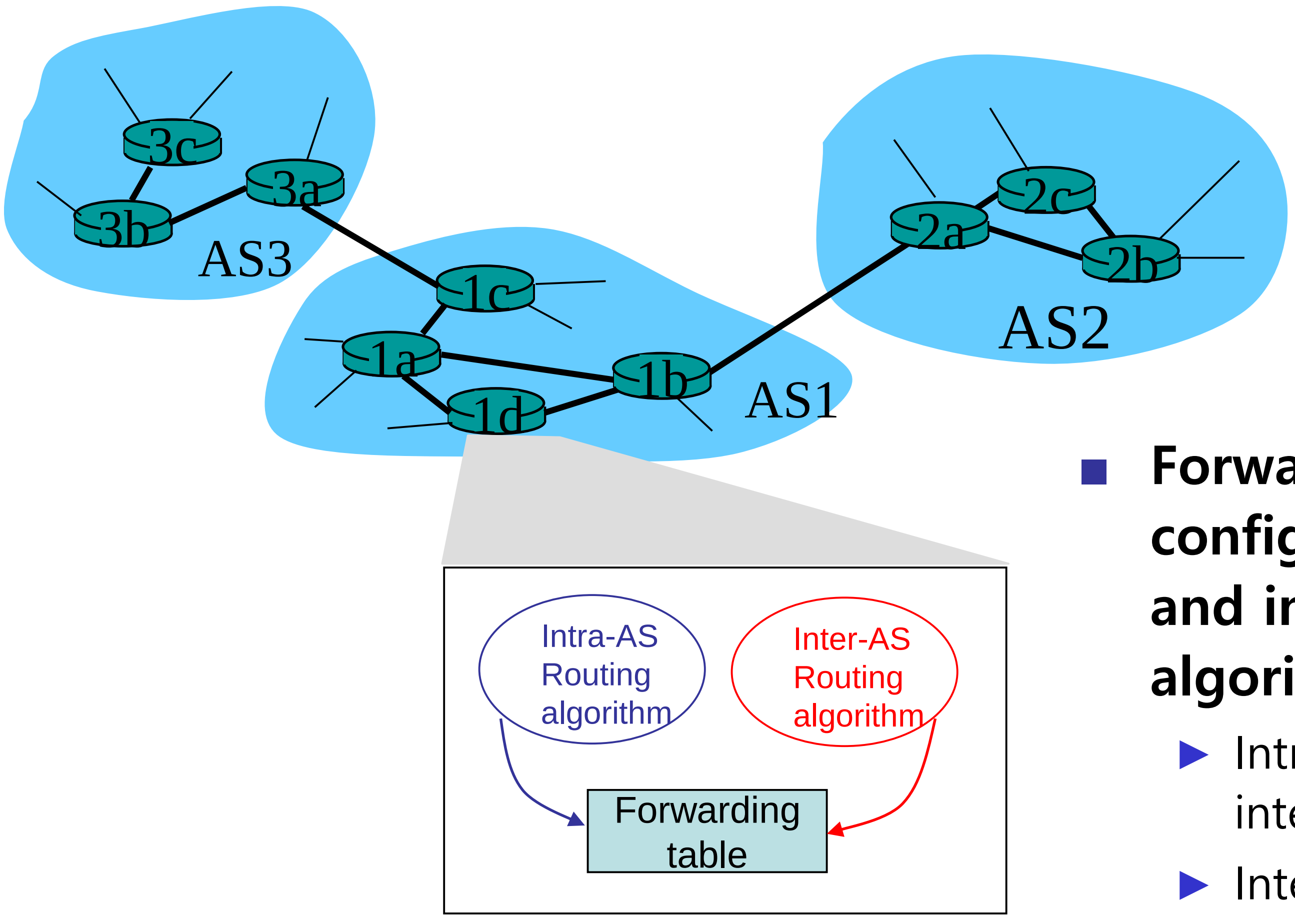
Hierarchical Routing

- Aggregate routers into regions, "autonomous systems" (AS)
- Routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - Routers in different AS can run different intra-AS routing protocol

Gateway router

- Direct link to router in another AS

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)
  - IGRP: Interior Gateway Routing Protocol (Cisco proprietary for decades, until 2016)

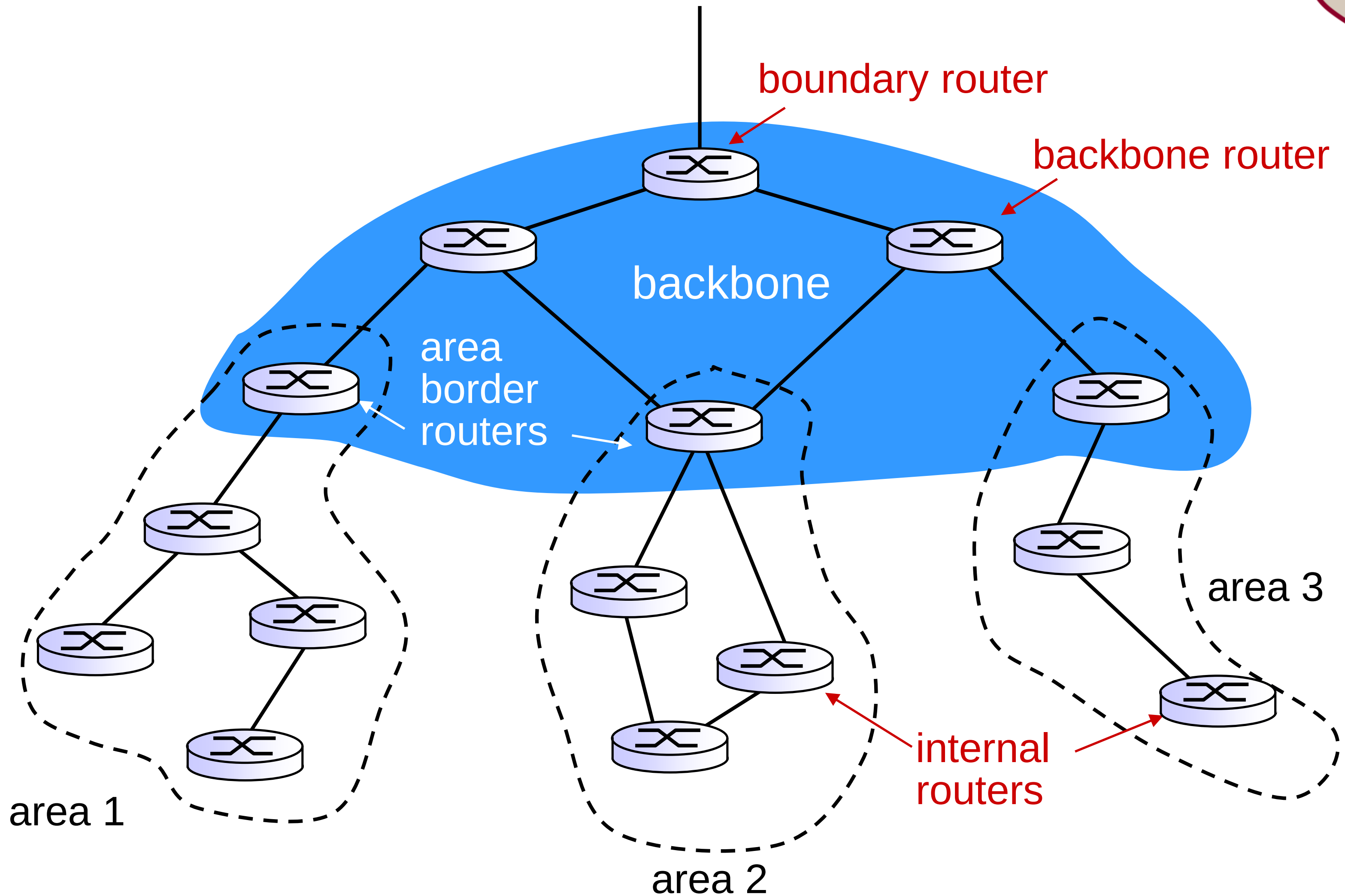
OSPF (Open Shortest Path First)

- **"open": publicly available**
- **Uses link-state algorithm**
  - Link state packet dissemination
  - Topology map at each node
  - Route computation using Dijkstra's algorithm
- **Router floods OSPF link-state advertisements to all other routers in entire AS**
  - Carried in OSPF messages directly over IP (rather than TCP or UDP
  - Link state: for each attached link
- **IS-IS routing protocol: nearly identical to OSPF**

OSPF "advanced" features

- **Security**
  - All OSPF messages authenticated (to prevent malicious intrusion)
- **Multiple same-cost paths allowed**
  - Only one path in RIP
- **For each link, multiple cost metrics for different TOS**
  - e.g., satellite link cost set low for best effort ToS; high for real-time ToS
- **Integrated uni- and multi-cast support:**
  - Multicast OSPF (MOSPF) uses same topology data base as OSPF
- **Hierarchical OSPF in large domains.**

Hierarchical OSPF



Hierarchical OSPF

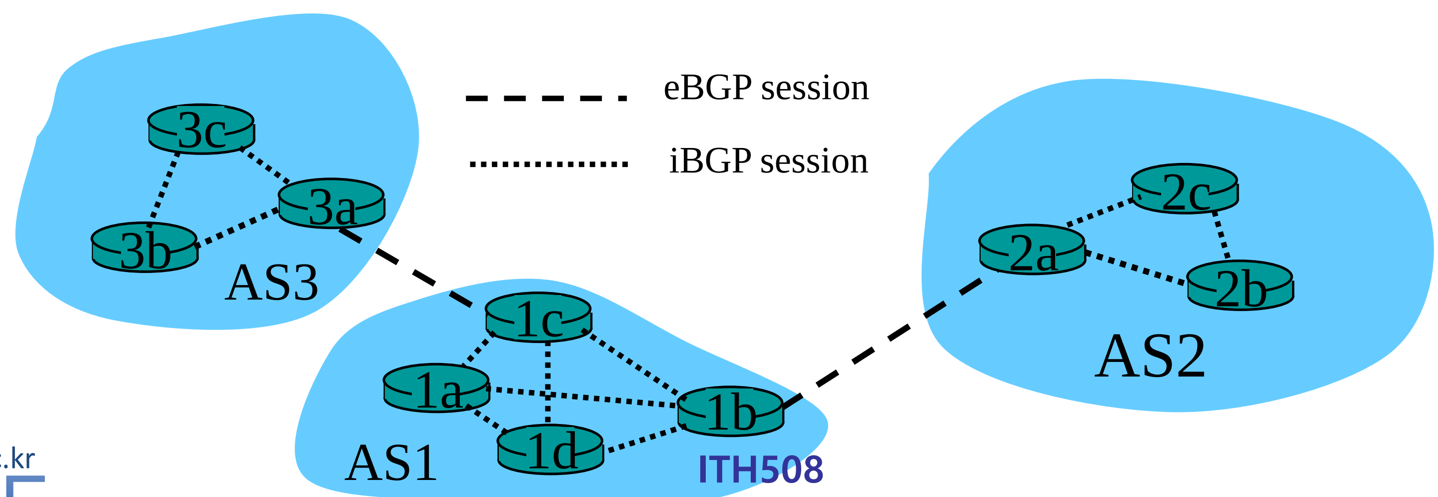
- **Two-level hierarchy: local area, backbone.**
  - Link-state advertisements only in area
  - Each nodes has detailed area topology; only known direction (shortest path) to nets in other areas.
- **Area border routers: "summarize" distances to nets in own area, advertise to other Area Border routers.**
- **Backbone routers: run OSPF routing limited to backbone.**
- **Boundary routers: connect to other AS'es.**

Internet inter-AS routing: BGP

- **BGP (Border Gateway Protocol):** ***the*** **de facto inter-domain routing protocol**
  - "glue that holds the Internet together"
- **BGP provides each AS a means to:**
  1. Obtain subnet **reachability information** from neighboring ASs.
  2. Propagate **reachability information** to all AS-internal routers.
  3. Determine "good" routes to **subnets** based on **reachability information and policy**.

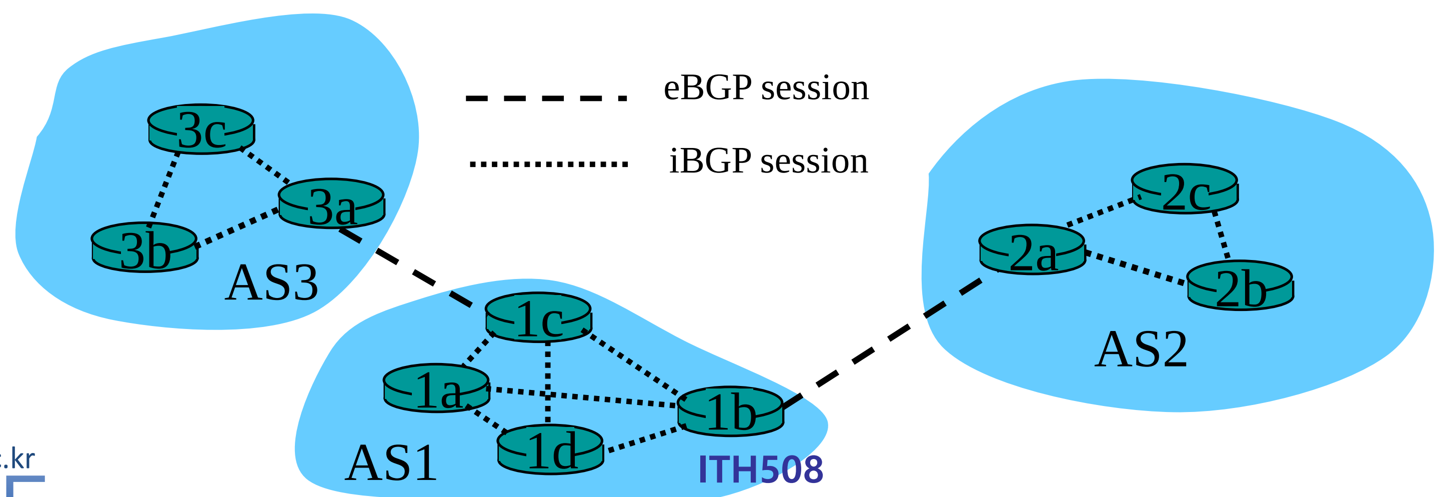
BGP Basics

- **Pairs of routers (BGP peers) exchange routing info over semi-permanent TCP connections: BGP sessions**
  - ▶ **BGP sessions need not correspond to physical links.**
- **When AS2 advertises a prefix (subnet) to AS1:**
  - ▶ AS2 *announce* it will forward datagrams towards that **prefix**.
  - ▶ AS2 can aggregate **prefixes** in its advertisement



Distributing Reachability

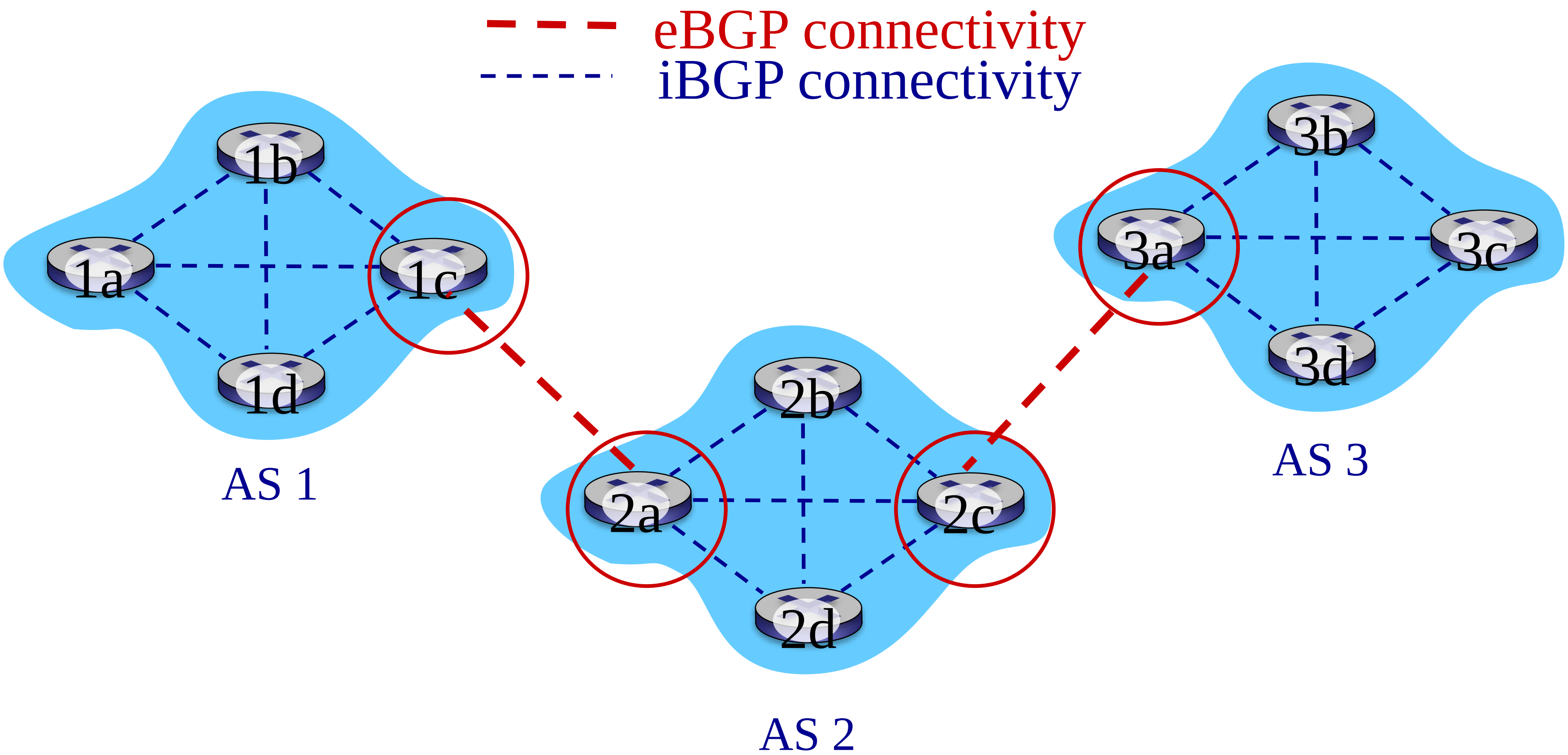
- **Using eBGP session between 3a and 1c, AS3 sends prefix reachability info to AS1.**
  - 1c can then use iBGP to distribute new prefix info to all routers in AS1
  - 1b can then re-advertise new reachability info to AS2 over 1b-to-2a eBGP session
- **When router learns of new prefix, it creates entry for prefix in its forwarding table.**

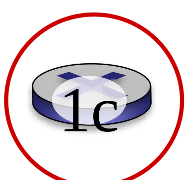


Path Attributes & BGP routes

- **Advertised prefix includes BGP attributes.**
  - **prefix + attributes** = "route"
- **Two important attributes:**
  - **AS-PATH: contains ASs** through which **prefix advertisement has passed**: e.g, AS 67, AS 17
  - **NEXT-HOP:** indicates a **specific AS router** to next-hop AS.
    - **The IP address of the router interface that begins the AS-PATH**
    - **The IP address of the border router that announced the route**
- ***Policy-based routing:***
  - Gateway receiving route advertisement uses *import policy* to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to *advertise* path to other other neighboring ASes

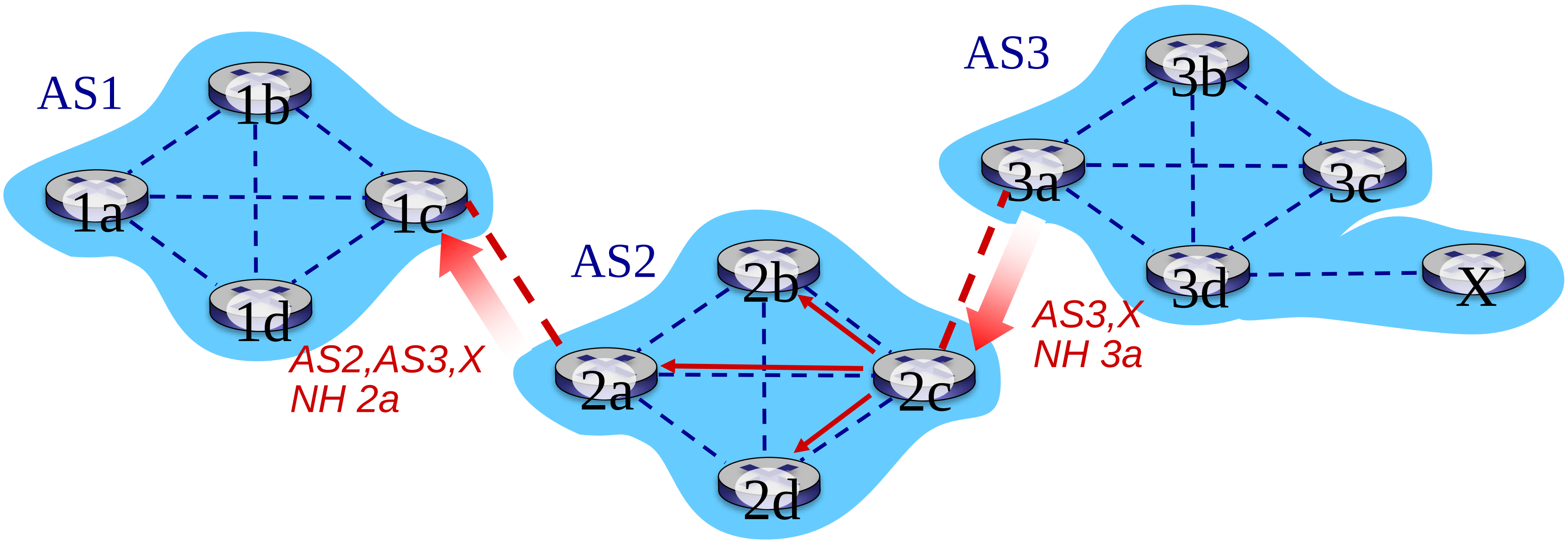
Example: eBGP, iBGP connections





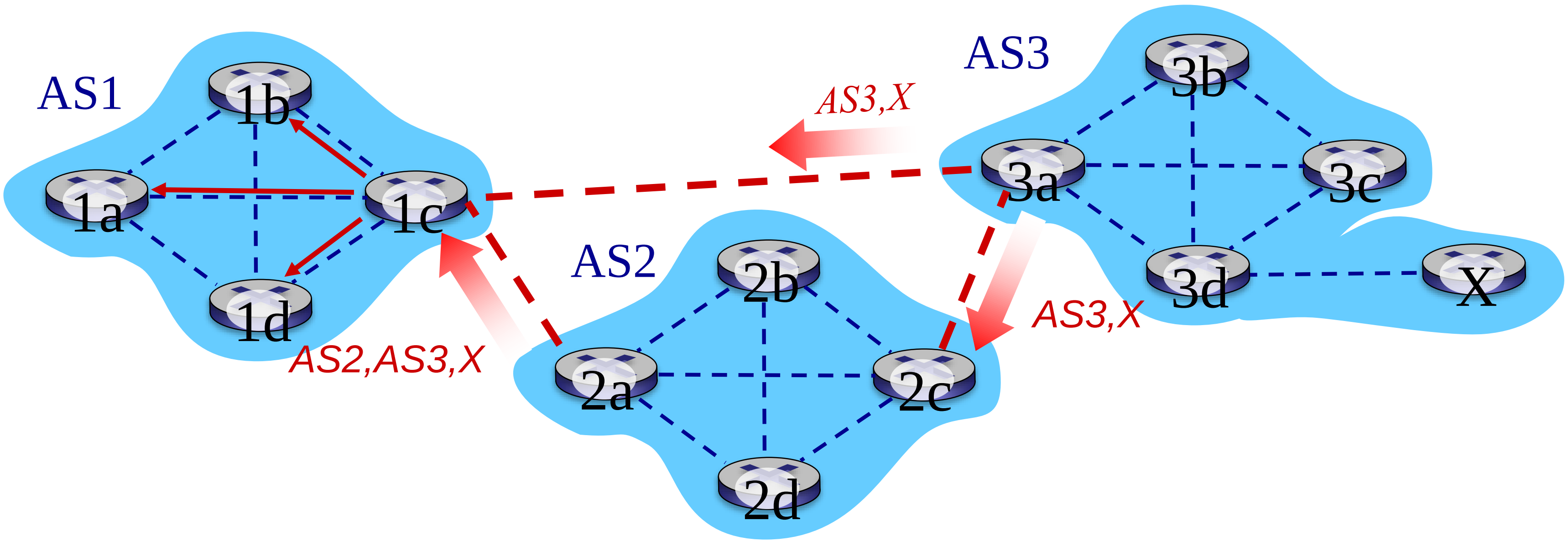
gateway routers run both eBGP and iBGP protools

Example: BGP Path Advertisement



- AS2 router 2c receives path advertisement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 policy, AS2 router 2a advertises (via eBGP) path AS2, AS3, X to AS1 router 1c

Example: BGP Path Advertisement



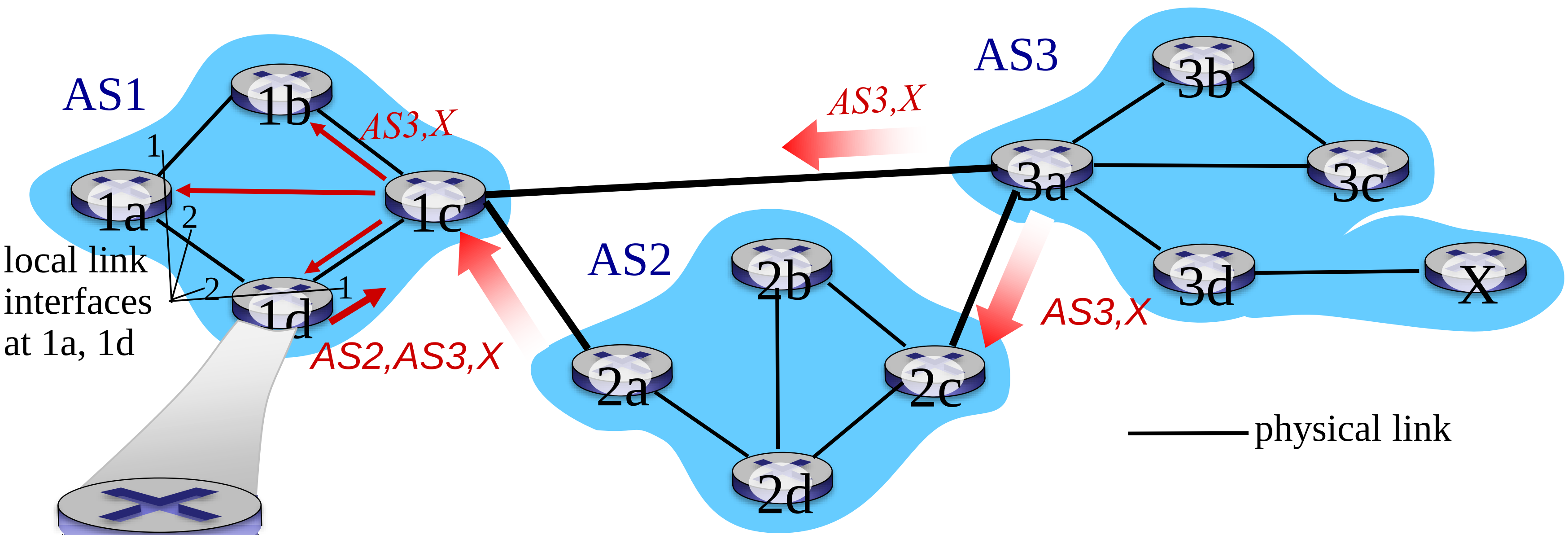
- **Gateway router may learn about multiple paths to destination:**
  - AS1 gateway router 1c learns path *AS2,AS3,X* from 2a
  - AS1 gateway router 1c learns path *AS3,X* from 3a
  - Based on policy, AS1 gateway router 1c chooses path *AS3,X, and advertises path within AS1 via iBGP*

BGP Messages

- **BGP messages exchanged using TCP.**
- **BGP messages:**
  - **OPEN:** opens TCP connection to peer and authenticates sender
  - **UPDATE:** advertises new path (or withdraws old)
  - **KEEPALIVE** keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - **NOTIFICATION:** reports errors in previous msg; also used to close connection

BGP, OSPF, Forwarding Table Entries

Q: how does router set forwarding table entry to distant prefix?



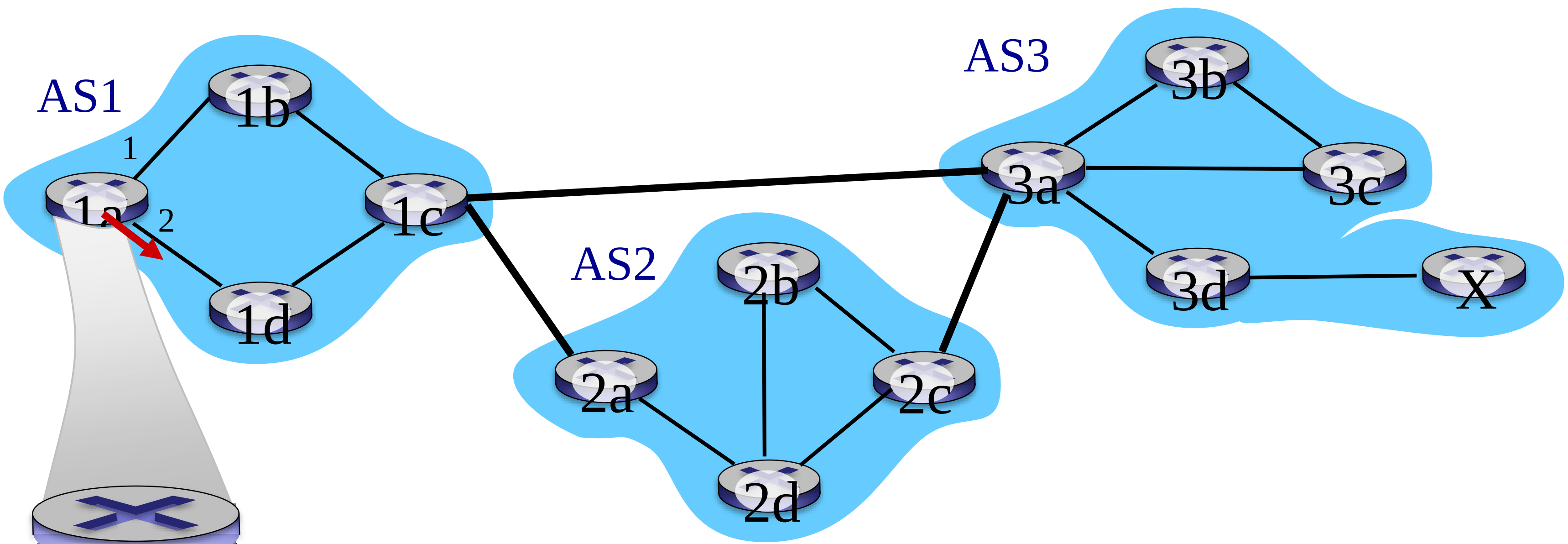
dest	interface
…	…
X	1
	…

…

- **Recall: 1a, 1b, 1d learn about dest X via iBGP from 1c: "path to X goes through 1c"**
- **1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1**

BGP, OSPF, Forwarding Table Entries

Q: how does router set forwarding table entry to distant prefix?



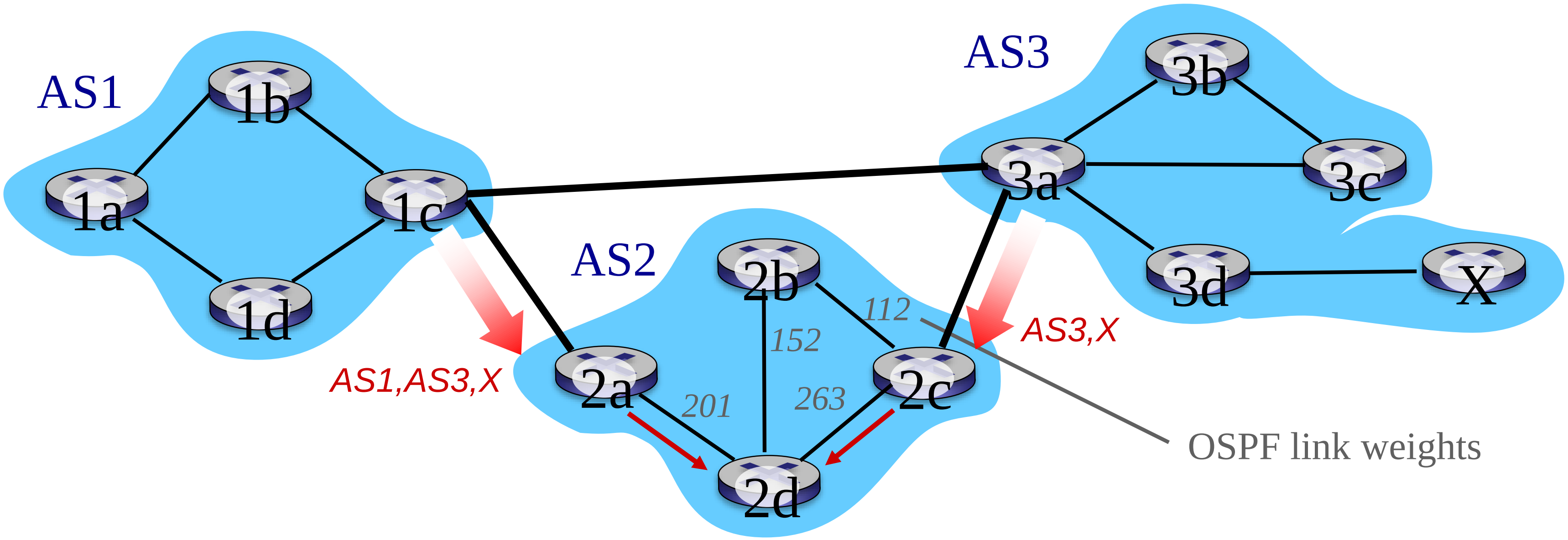
dest	interface
…	…
X	2
…	…

- Recall: 1a, 1b, 1c learn about dest X via iBGP from 1c: "path to X goes through 1c"
- 1d: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 1
- **1a: OSPF intra-domain routing: to get to 1c, forward over outgoing local interface 2**

BGP Route Selection

- **Router may learn about more than 1 route to some prefix. Router must select route.**
- **Elimination rules:**
  1. Local preference value attribute: **policy** decision
  2. Shortest AS-PATH
  3. Closest NEXT-HOP router
  4. Additional criteria

Hot Potato Routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- hot potato routing: choose local gateway that has **least intra-domain cost** (e.g., 2d chooses 2a, even though more AS hops to X)

Why Intra-AS routing and Inter-AS routing?

- **Policy:**
  - Inter-AS: **admin wants control over how** its traffic routed, who routes through its net.
  - Intra-AS: single admin, so no policy decisions needed
- **Scale:**
  - Hierarchical routing saves table size, reduced update traffic
- **performance:**
  - Intra-AS: can focus on performance
  - Inter-AS: policy may dominate over performance

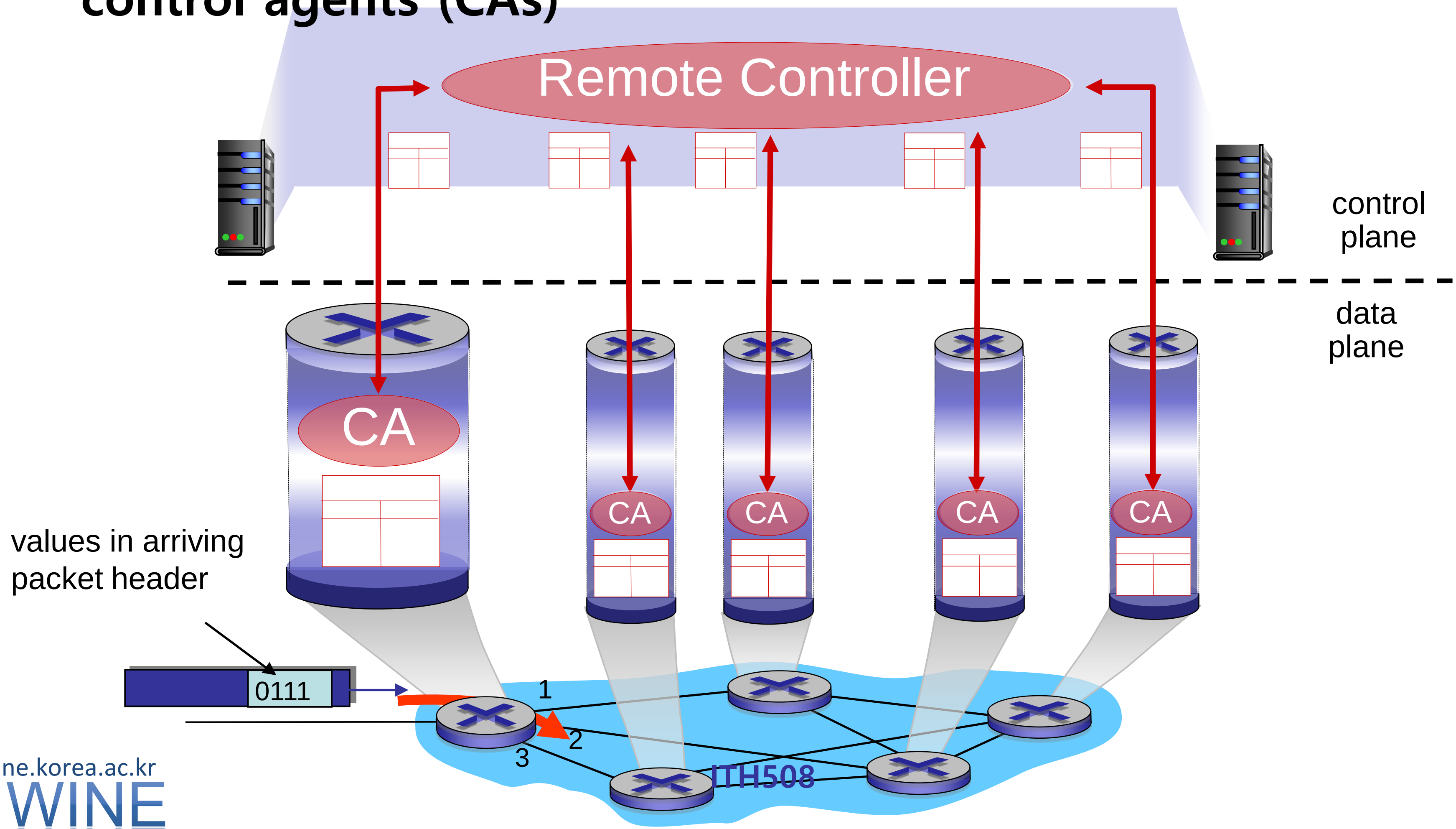
Software Defined Networking (SDN)

- **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: Logically Centralized Control Plane

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

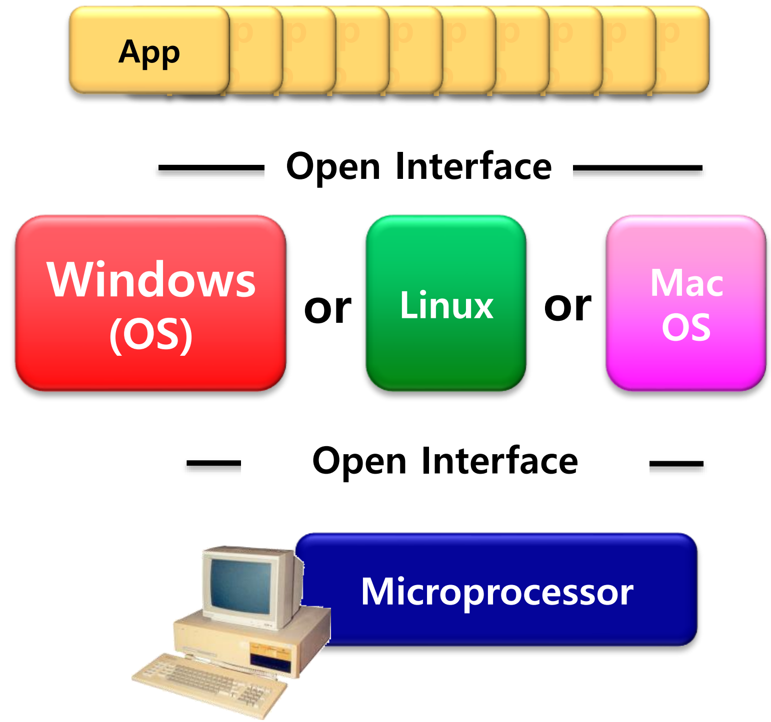


Software defined networking (SDN)

Why **a** ***logically centralized*** **control plane?**

- **Easier network management: avoid router misconfigurations, greater flexibility of traffic flows**
- **Table-based forwarding (OpenFlow API) allows "programming" routers**
  - Centralized "programming" easier: compute tables centrally and distribute them
  - 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*

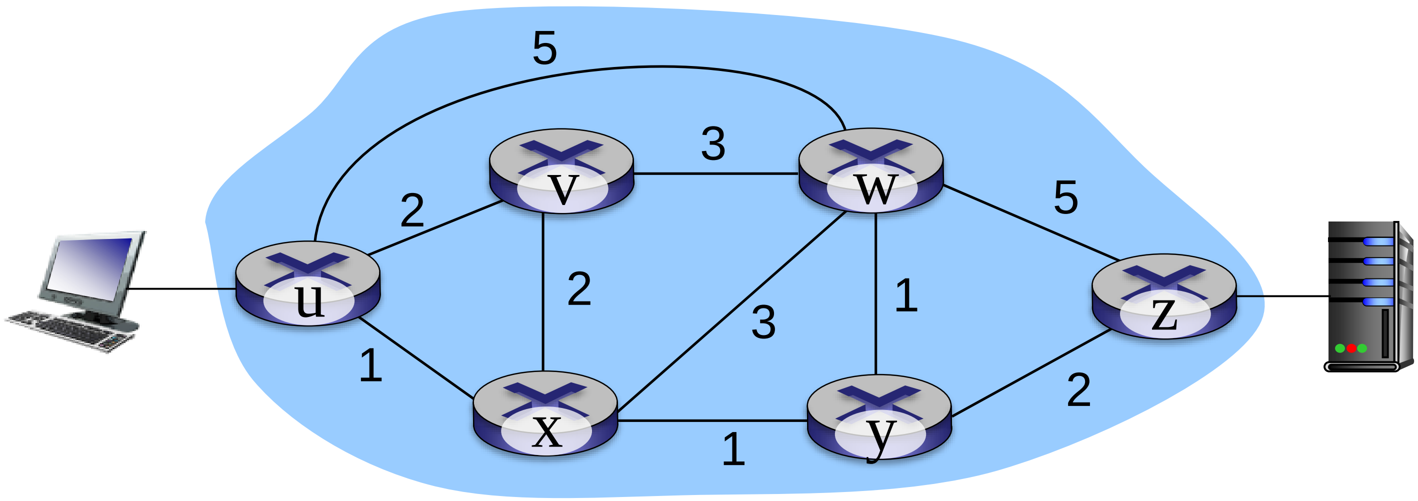


Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

* Slide courtesy: N. McKeown

Traffic Engineering: Difficult in Traditional Routing

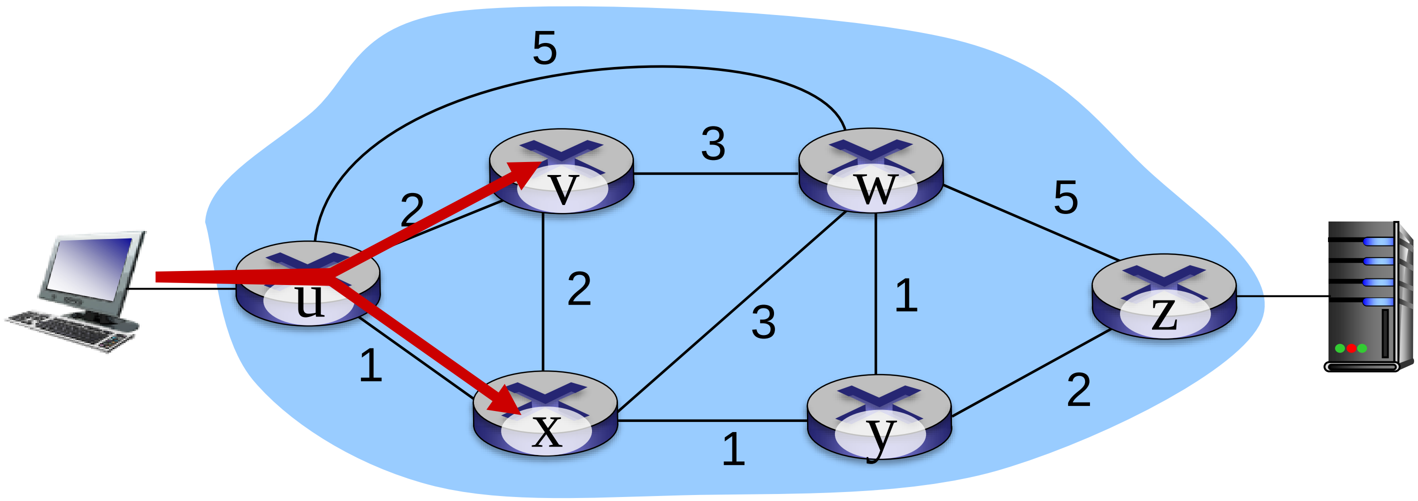


Q: what if network operator wants **u-to-z traffic to flow along *uvwz***, **x-to-z traffic to flow *xwyz***?

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

Link weights are only control "knobs": wrong!

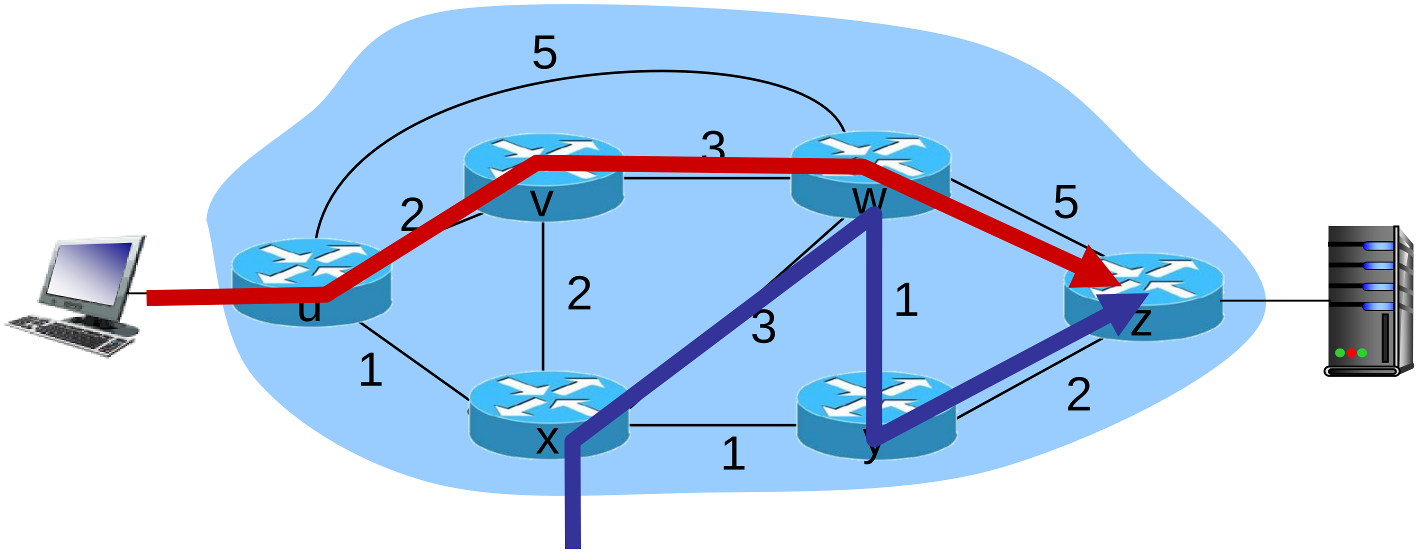
Traffic Engineering: Difficult in Traditional Routing



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)

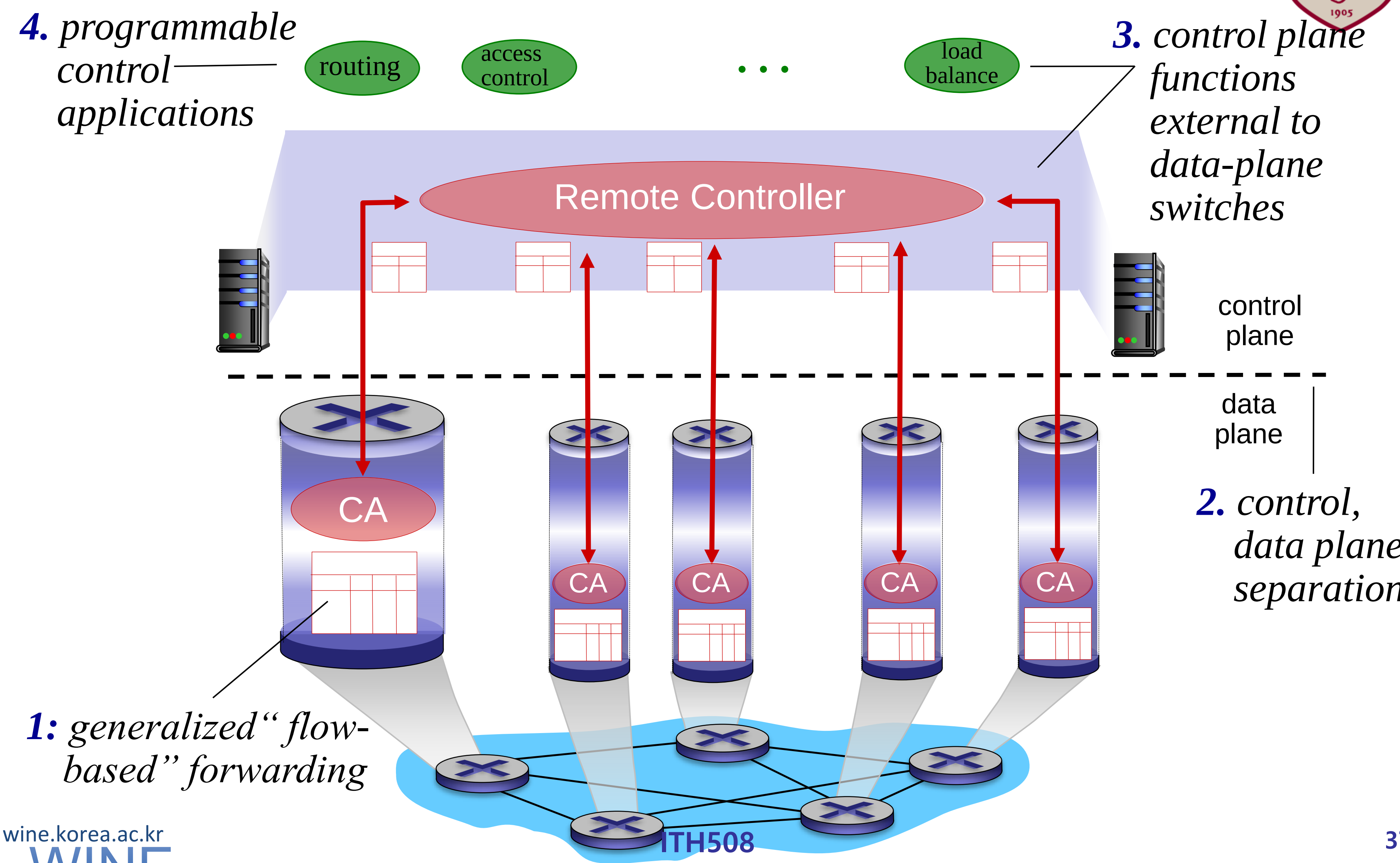
Traffic Engineering: Difficult in Traditional Routing



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)

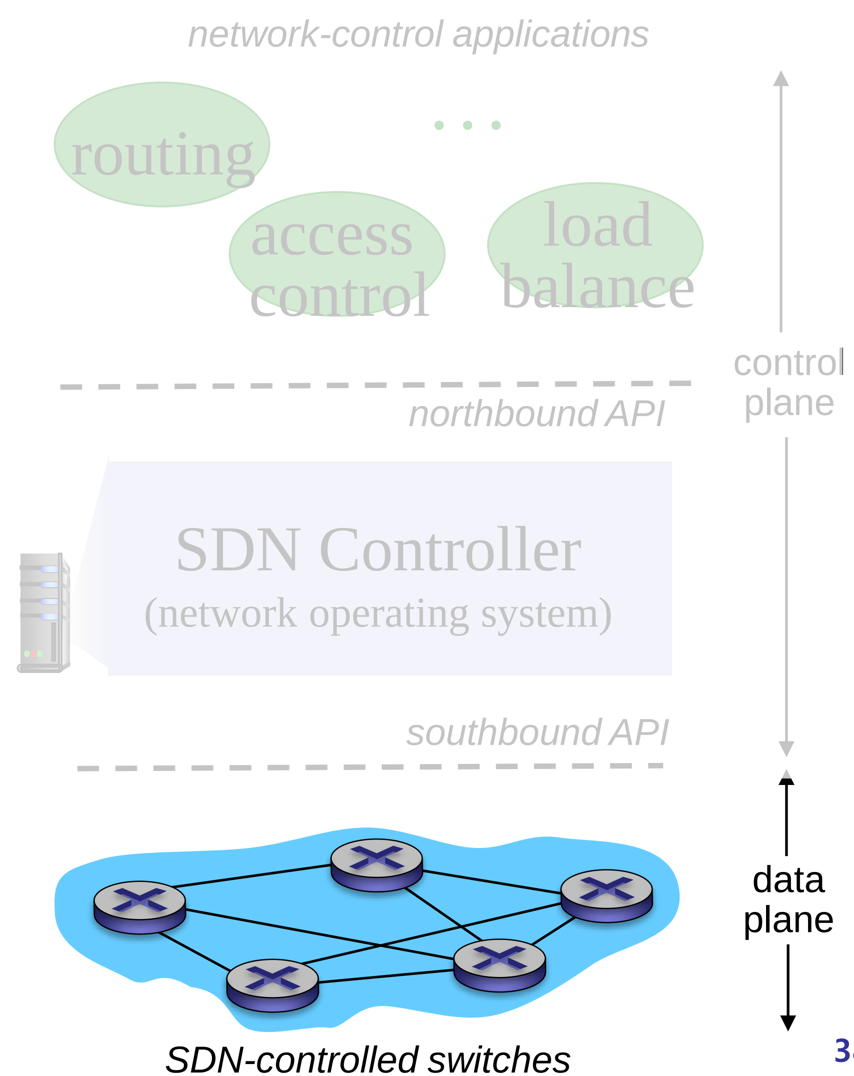
Software Defined Networking (SDN)



SDN Perspective: Data Plane Switches

Data plane switches

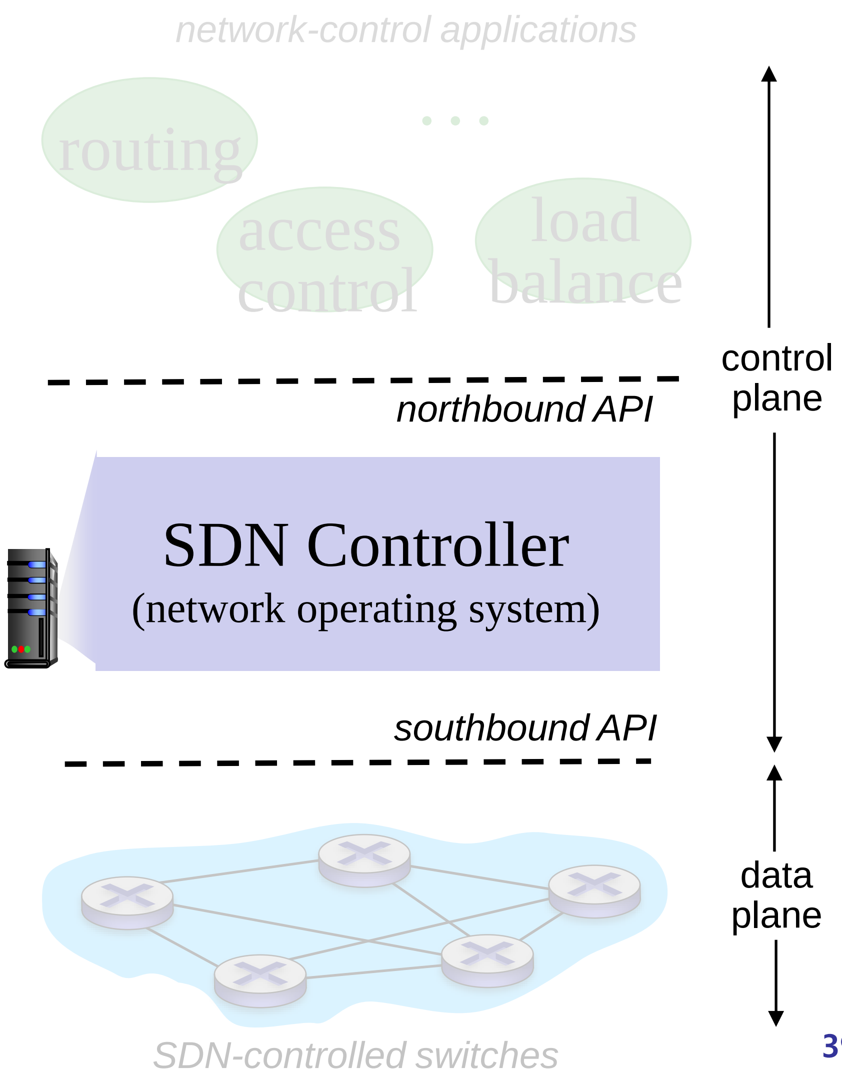
- **Fast, simple, commodity switches implementing generalized data-plane forwarding in hardware**
- **Switch table computed, installed by controller**
- **API for table-based switch control (e.g., OpenFlow)**
  - Defines what is controllable and what is not
- **Protocol for communicating with controller (e.g., OpenFlow)**



SDN Perspective: SDN Controller

SDN controller (network OS):

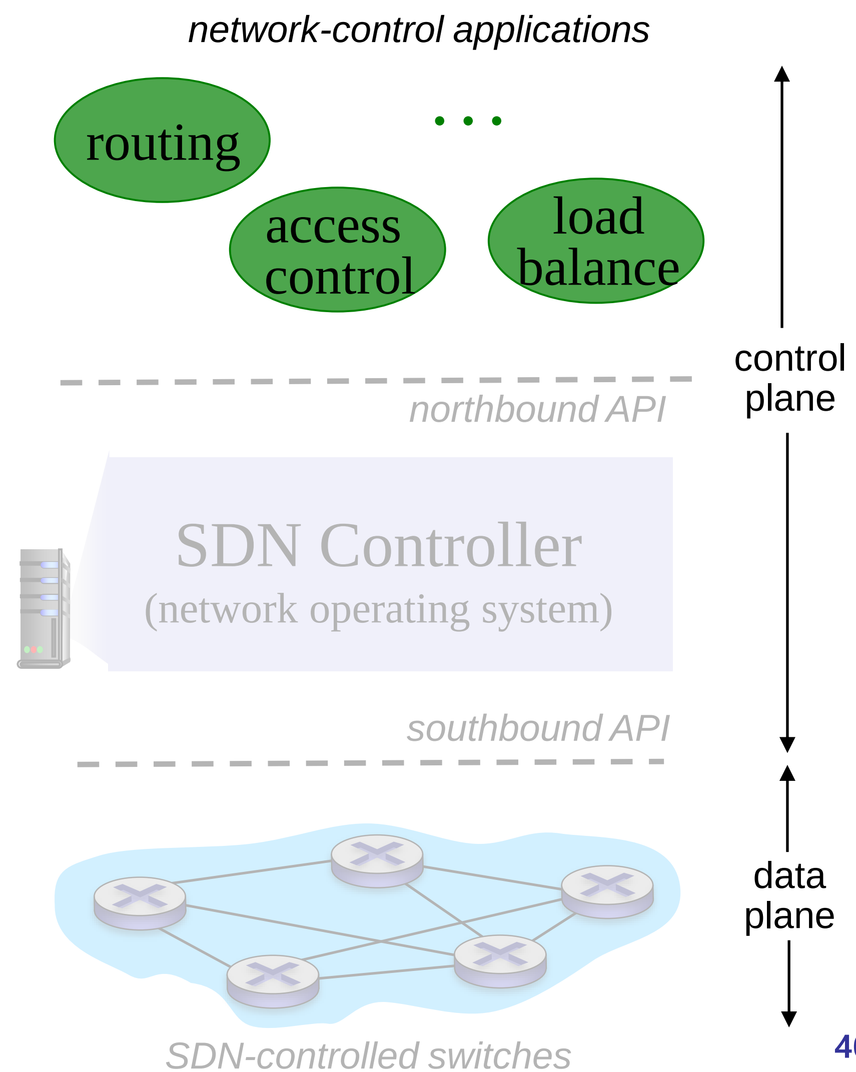
- Maintain network state information
- Interacts with network control applications "above" via northbound API
- Interacts with network switches "below" via southbound API
- Implemented as distributed system for performance, scalability, fault-tolerance, robustness



SDN Perspective: Control Applications

Network-control apps:

- **"brains" of control**
  - Implement control functions using lower-level services, API provided by SND controller
- ***Unbundled***
  - Can be provided by 3rd party: distinct from routing vendor, or SDN controller

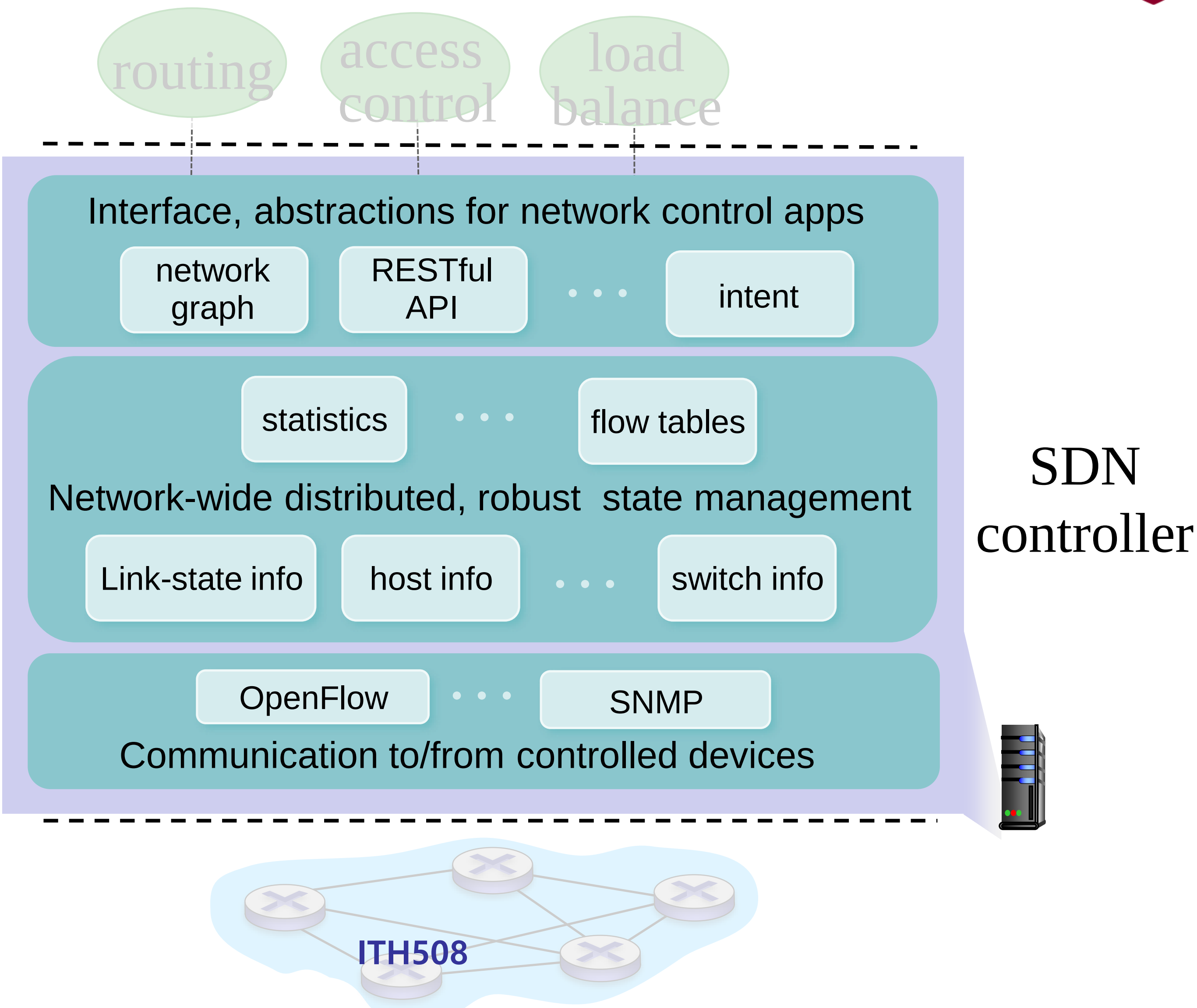


Components of SDN Controller

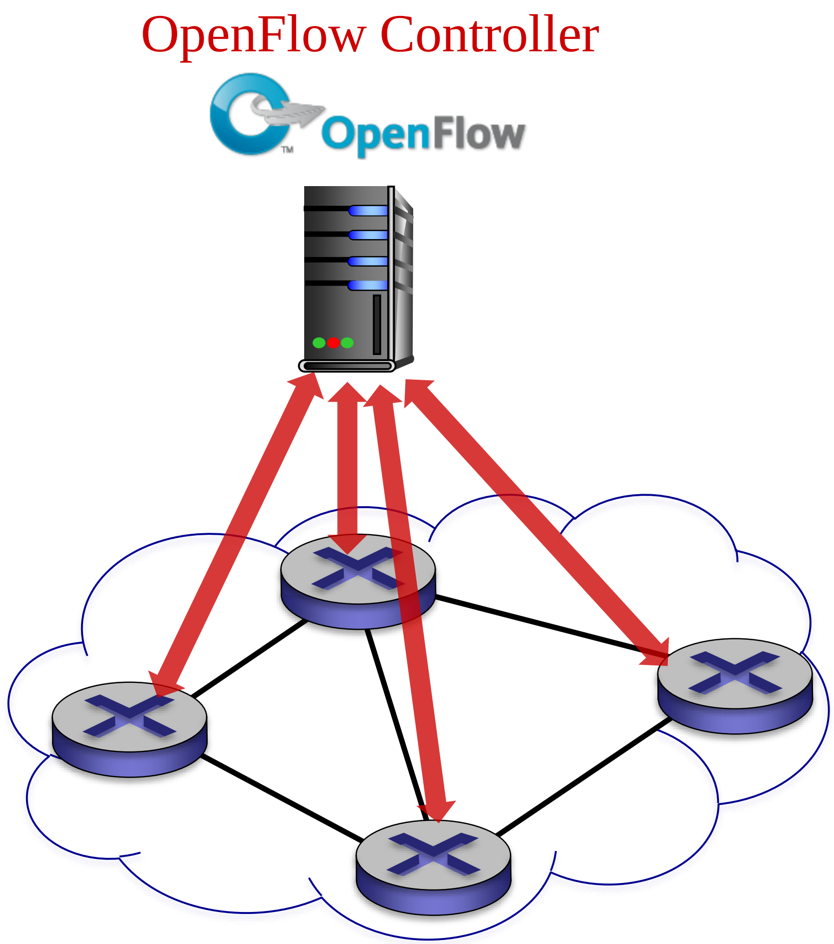
Interface layer to network control apps: abstractions API

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

***Communication layer*:** communicate between SDN controller and controlled switches



OpenFlow Protocol

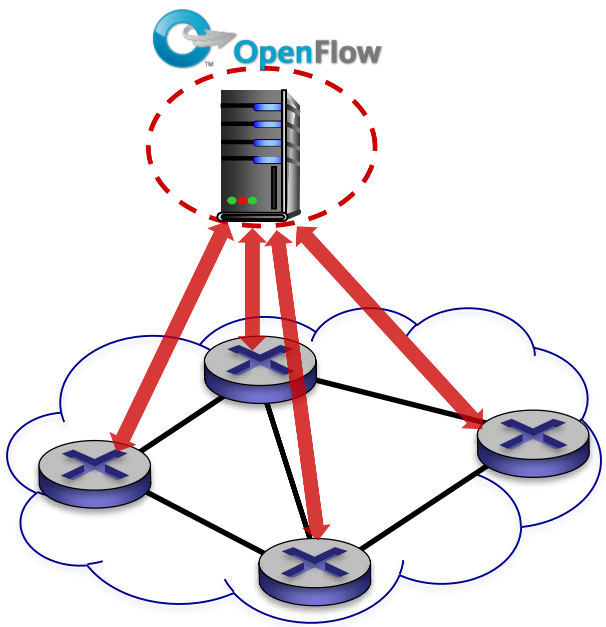


- **Operates between controller, switch**
- **TCP used to exchange messages**
  - Optional encryption
- **Three classes of OpenFlow messages:**
  - Controller-to-switch
  - Asynchronous (switch to controller)
  - Symmetric (misc)

OpenFlow: Controller-to-Switch Messages

Key controller-to-switch messages

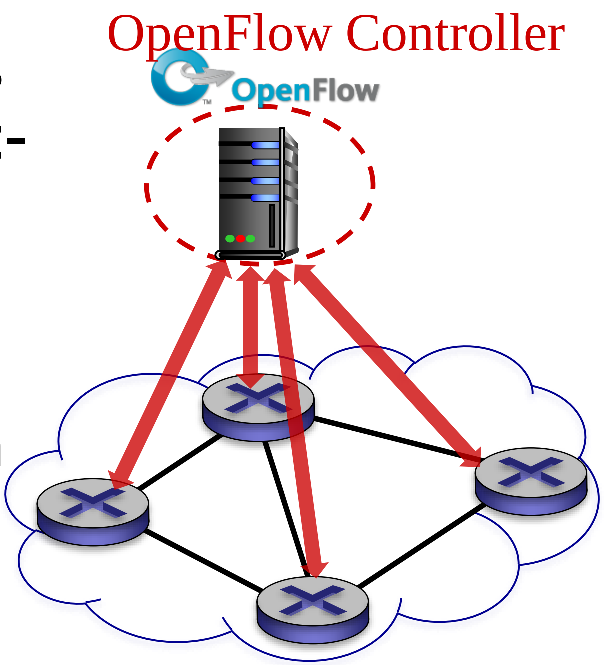
- *Features:* controller queries switch features, switch replies
- *Configure:* controller queries/sets switch configuration parameters
- *Modify-state:* add, delete, modify flow entries in the OpenFlow tables
- *Packet-out:* controller can send this packet out of specific switch port



OpenFlow: Controller-to-Switch Messages

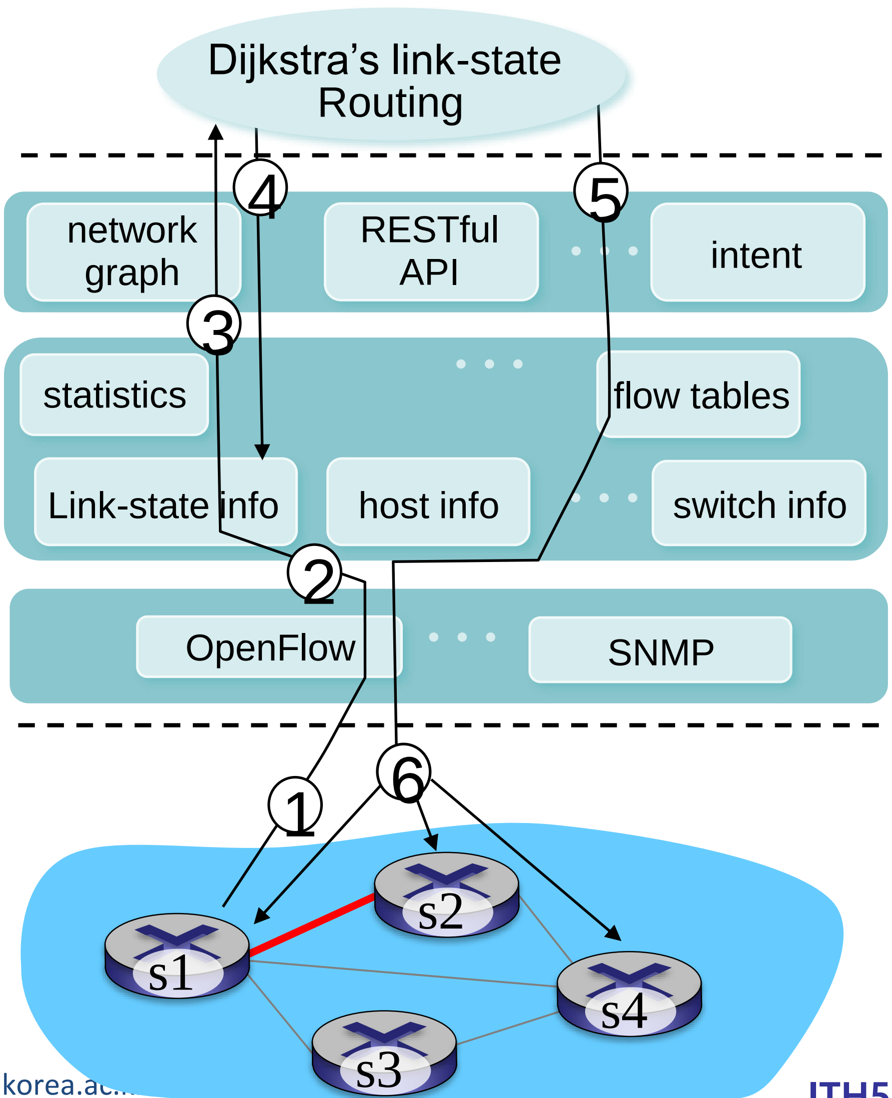
Key switch-to-controller messages

- ***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.**



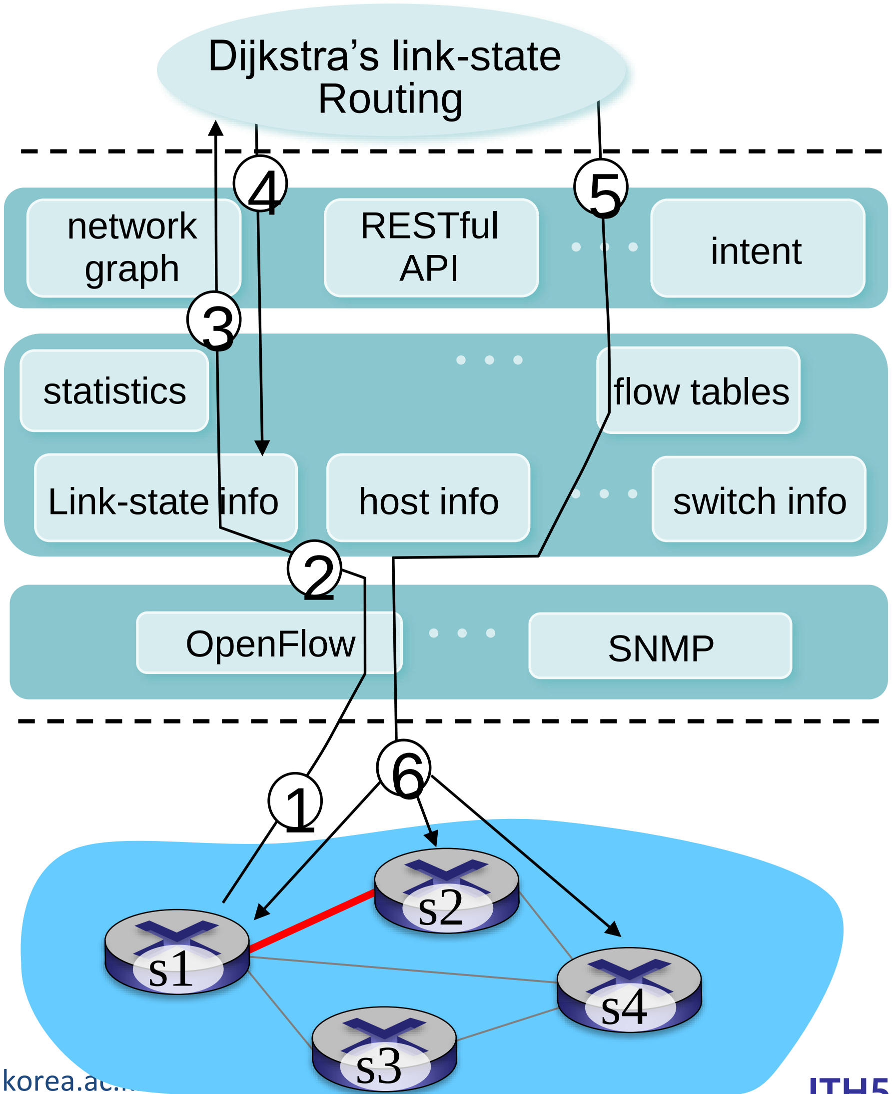
Fortunately, network operators don't "program" switches by creating/sending OpenFlow messages directly. Instead use higher-level abstraction at controller

SDN: Control/Data Plane Interaction Example



1. S1, experiencing link failure 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

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**