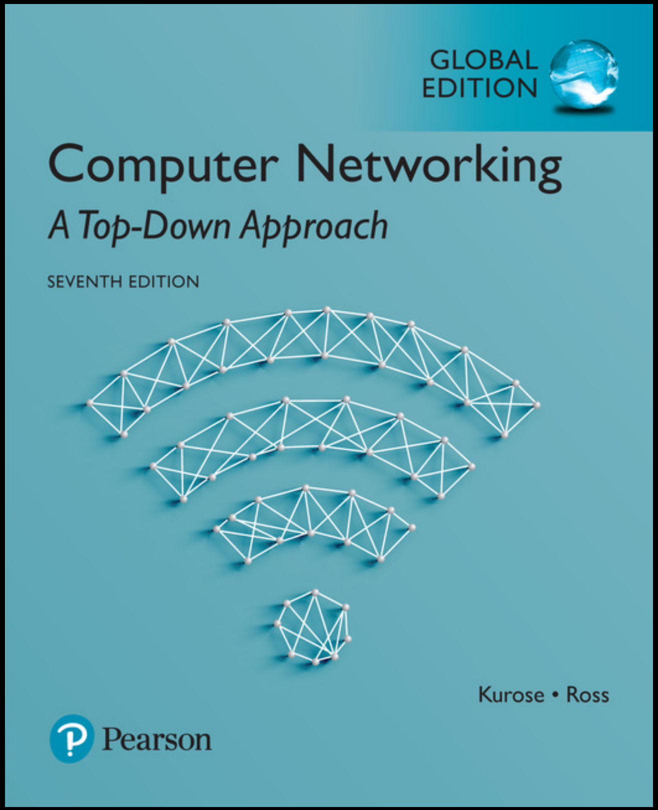
Computer Network

Network Layer II

School of Electric and Computer Engineering
Pusan National University, KOREA
Younghwan Yoo



Computer Networking

A Top–Down Approach

7th edition

Jim Kurose, Keith Ross

Pearson

April 2016

Contents

Computer Network introduction

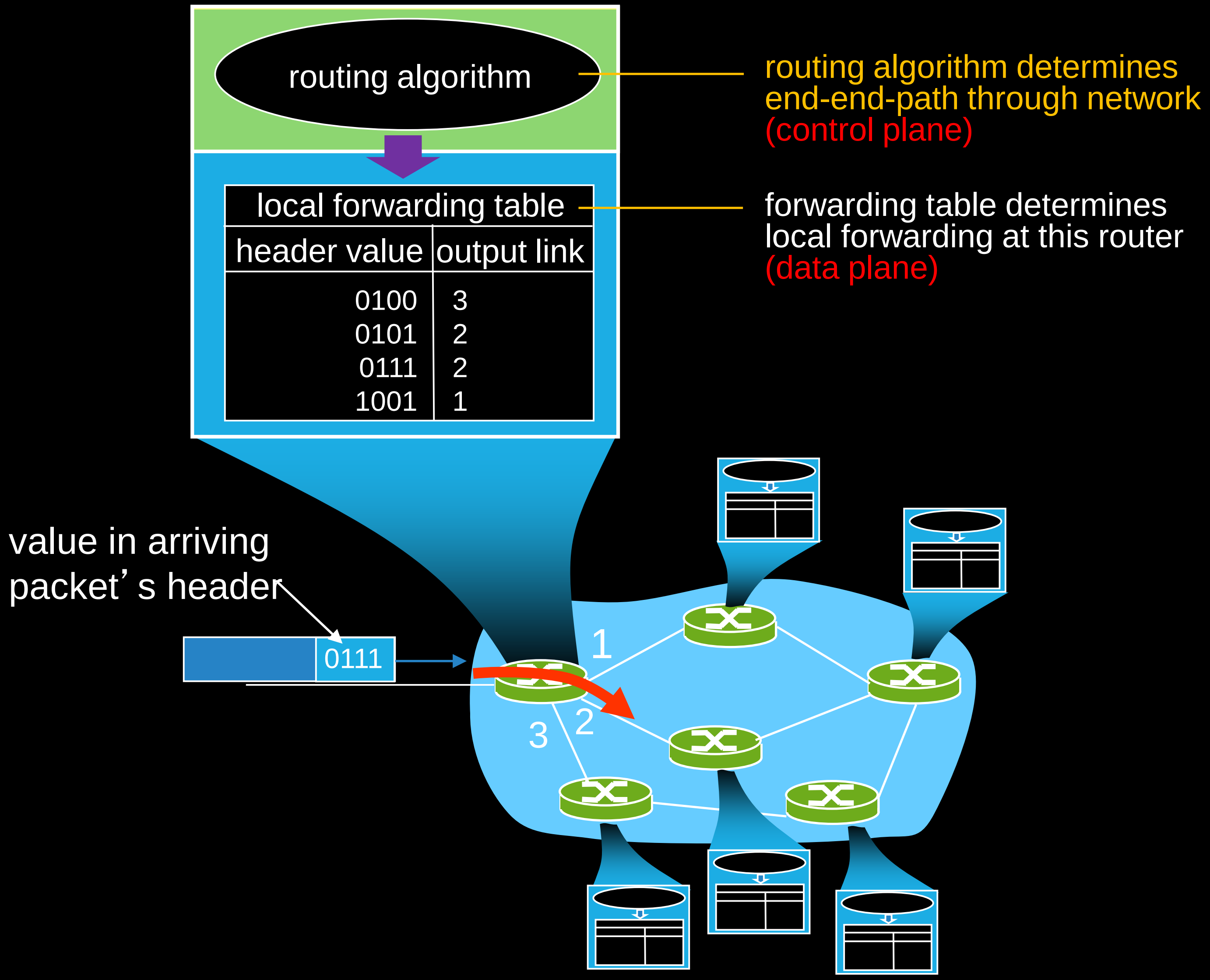
Contents

Computer Network introduction

01. Introduction to Routing

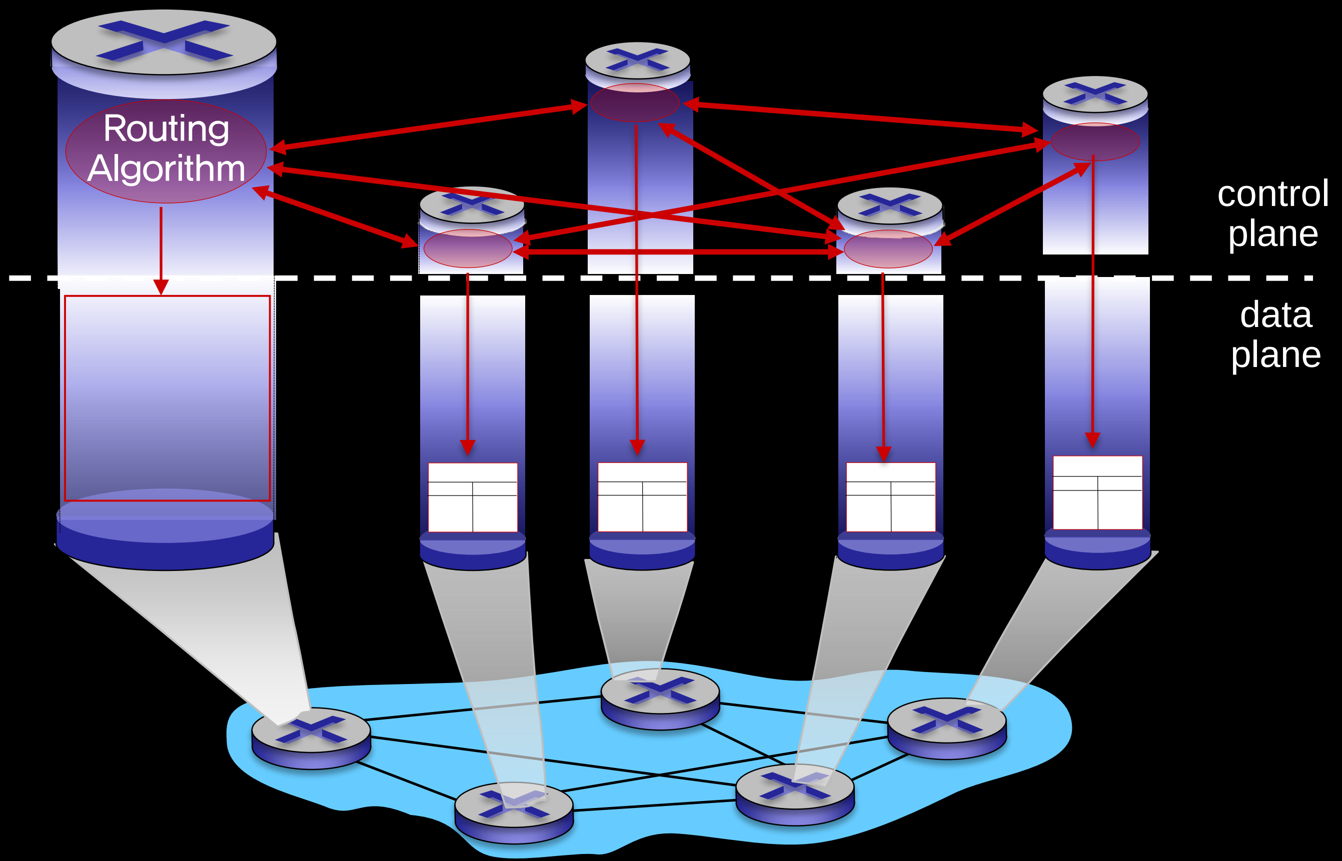
Two Key Functions in Network Layer

- **Routing**: determine route taken by packets from source to destination
- **Forwarding**: move packets from router's input to appropriate router output
- **Two approaches for control pl.**
  - per-router control (traditional)
  - logically centralized control (software defined network)



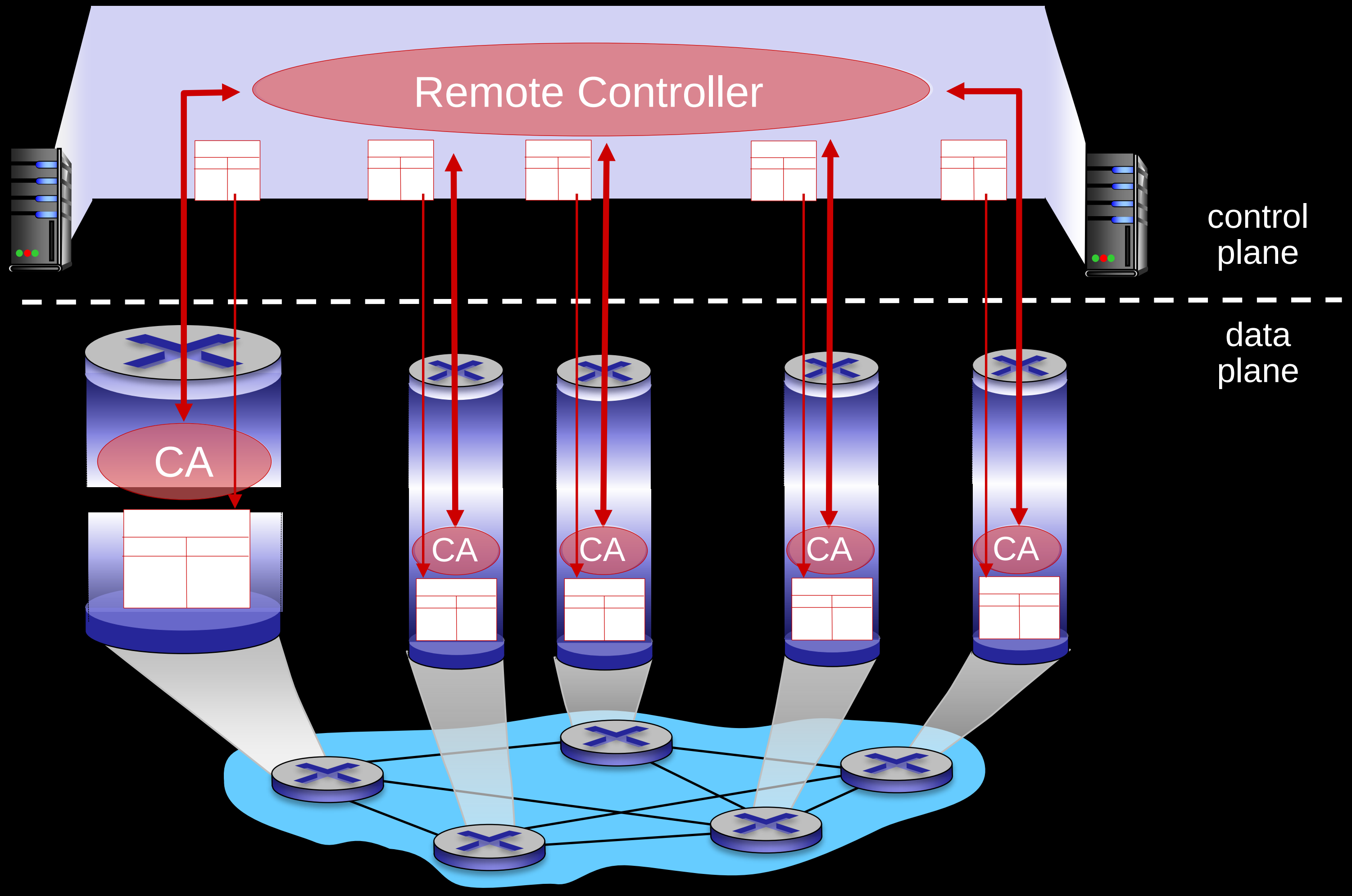
Per-router Control Plane

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

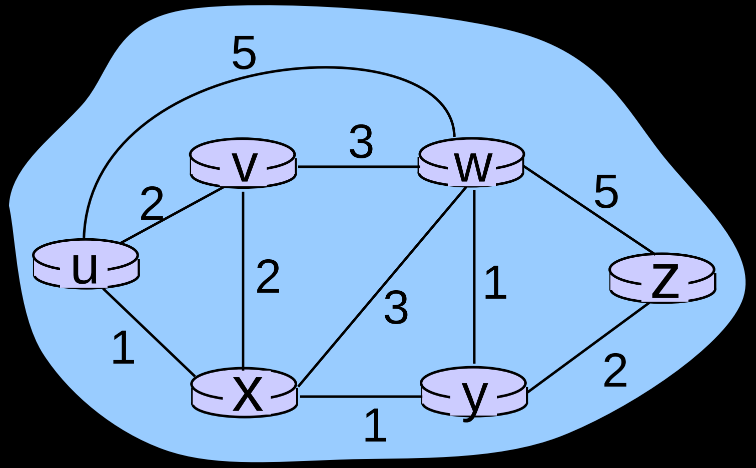


Logically Centralized Control Plane

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

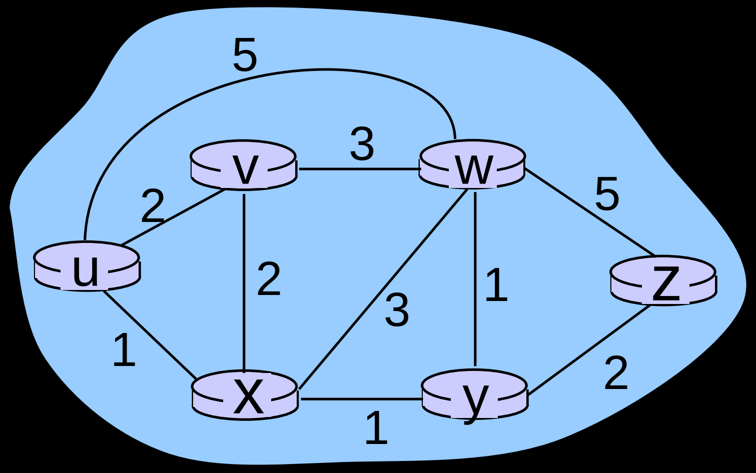


- Goal of routing: determine “good” paths (equivalently, routes), from sending hosts to receiving host, through network of routers
  - path: path: sequence of routers packets will traverse in going from given initial source host to given final destination host
  - “good”?: “least cost”, “fastest”, “least congested”



- Graph abstraction of the network
  - graph: G = (N,E)
  - N = set of routers = { u, v, w, x, y, z }
  - E = set of links ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

Path Cost



$c(x,x')$ = cost of link (x,x')
e.g., $c(w,z) = 5$

cost could always be 1, or inversely related to bandwidth, or inversely related to congestion

cost of path $(x_1, x_2, x_3, \ldots, x_p) = c(x_1,x_2) + c(x_2,x_3) + \ldots + c(x_{p-1},x_p)$

key question: what is the least-cost path between u and z ?
routing algorithm: algorithm that finds that least cost path

Routing Algorithm Classification

Static vs. Dynamic

- Static
  - routes are changed manually by administrator
- Dynamic
  - routes are changed more quickly
  - periodic update in response to link cost changes

Global vs. Decentralized

- Global
  - all routers have complete topology and all link cost information
  - "link state" algorithms
- Decentralized
  - router knows
    - directly connected neighbors
    - path costs to others and the next router to reach them
  - "distance vector" algorithms

02. Link-State Routing

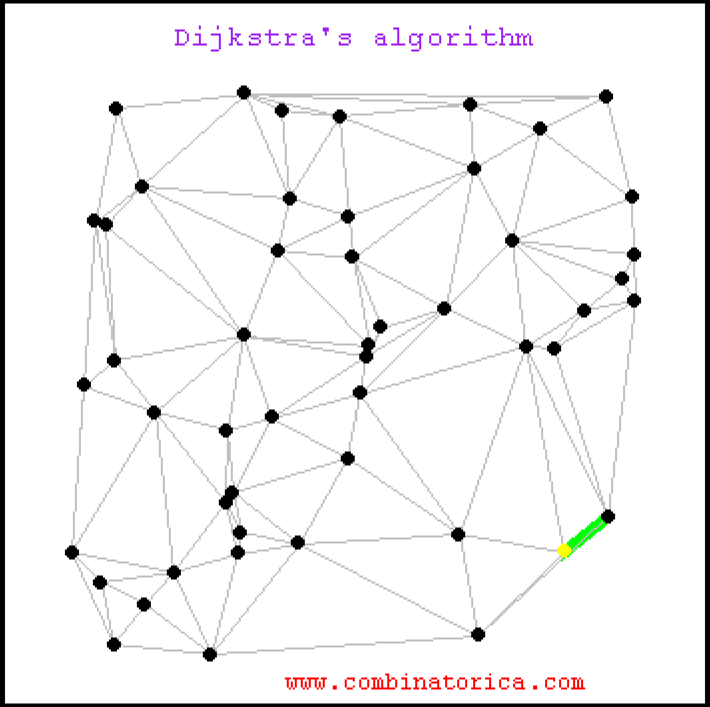
- Link–state advertisement (LSA) message
  - neighbor node information
  - link information to neighbors

Operation

1) A router constructs LSA when
   - neighbor changes
   - link goes up/down
   - also, periodically
2) The LSA is broadcast to all routers in the network
3) Based on LSA from all other nodes, each router constructs the complete topology
4) Least cost paths between two nodes are computed by Dijkstra's algorithm

Dijkstra's Algorithm

- Computes least cost paths from one node ("source") to all other nodes
  - gives forwarding table for that node
- Iterative algorithm
  - after *k* iterations, least cost path to *k* destinations are known



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiBgrGE4KDcAhVCkZQKHYdKA5EQjRx6BAgBEAU&url=https%3A%2F%2Fbrilliant.org%2Fwiki%2Fdijkstras-short-path-finder%2F&psig=AOvVaw1l0acGxCXie7YqMjH2VD-O&ust=1531731855165992

Dijkstra’s Algorithm Operation

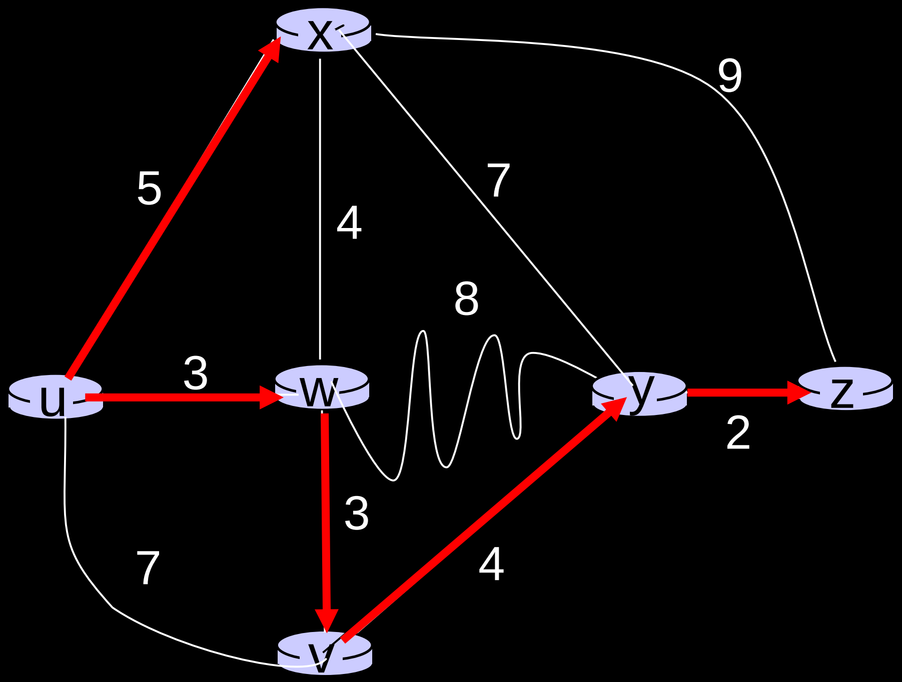
```
Initialization:
  N' = {u}
  for all nodes v
    if v adjacent to u
        then D(v) = c(u,v)
    else D(v) = ∞

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for all v adjacent to w and not in N' :
      D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   shortest path cost to w plus cost from w to v */
until all nodes in N'
```

notation：

- c(x,y)：link cost from node x to y； = ∞ if not direct neighbors
- D(v)：current value of cost of path from source to dest. v
- p(v)：predecessor node along path from source to v
- N'：set of nodes whose least cost path definitively known

Dijkstra's Algorithm: Example #1

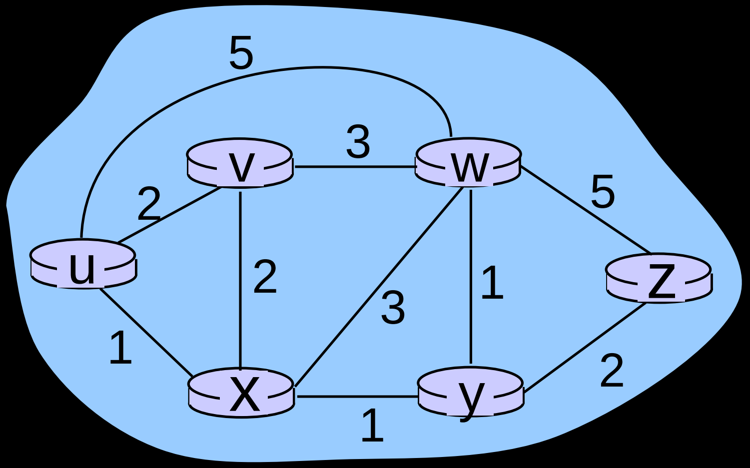
Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					



notes:

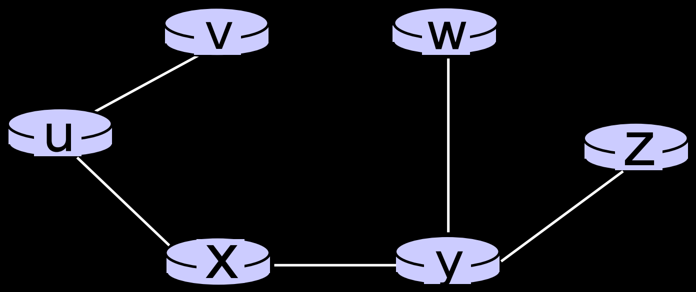
- construct shortest path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)

Dijkstra's Algorithm: Example #2



Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

resulting shortest-path tree from u:

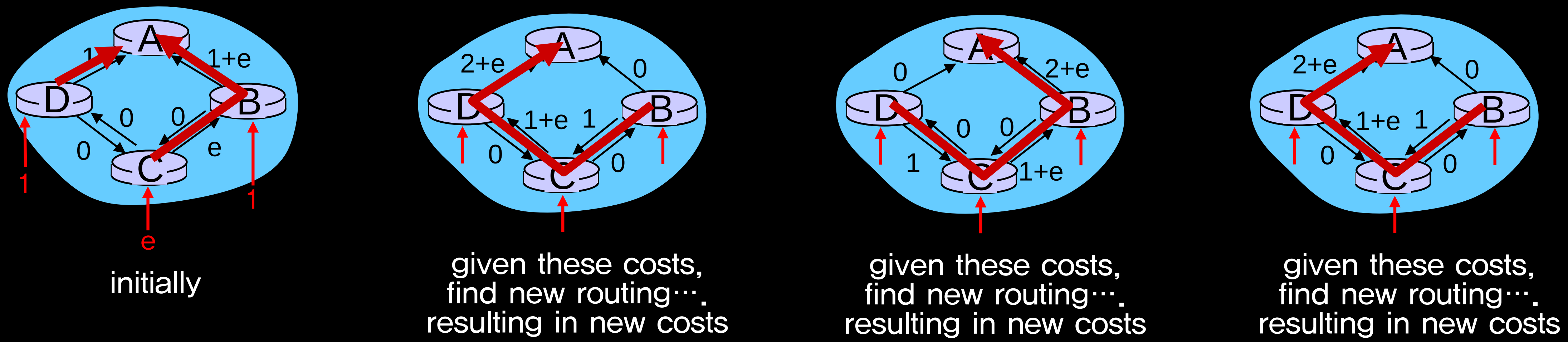


resulting forwarding table in u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Note: Oscillation Problem

- Can happen when link cost equals the amount of carried traffic



03. Distance Vector Routing

let

$D_x(y)$:= cost of least-cost path from x to y

then

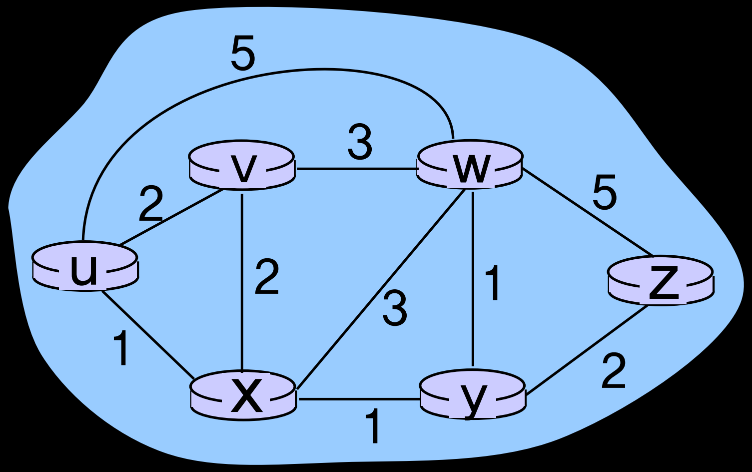
$$D_x(y) = \min_v \{c(x,v) + D_v(y)\}$$

cost from neighbor v to destination y

cost to neighbor v

min taken over all neighbors v of x

Bellman-Ford Example



clearly, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

B-F equation says:

$$
\begin{aligned}
d_u(z) &= \min \{\, c(u,v) + d_v(z), \\
&\qquad c(u,x) + d_x(z), \\
&\qquad c(u,w) + d_w(z) \,\} \\
&= \min \{2 + 5, \\
&\qquad 1 + 3, \\
&\qquad 5 + 3\} = 4
\end{aligned}
$$

Distance Vector Algorithm

- Node x initially:
  - knows cost to each neighbor v: $c(x,v)$
  - maintains its neighbors' distance vectors. For each neighbor v, x maintains
    $D_v = [D_v(y): y \in N]$
- Operation
  - from time-to-time, each node sends its own distance vector estimate to neighbors
  - when x receives new DV estimate from neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

each node:

wait for (change in local link cost or msg from neighbor)

↓

recompute estimates

↓

if DV to any dest has changed, *noti*
fy neighbors

Distance Vector Algorithm: Example

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

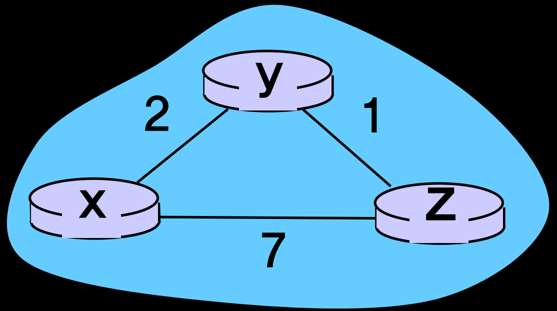
node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

$$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\} = \min\{2+0\ ,\ 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\} = \min\{2+1\ ,\ 7+0\} = 3$$



time

Distance Vector Algorithm: Example

$D_x(y) = \min\{c(x,y) + D_y(y),\ c(x,z) + D_z(y)\}$
$= \min\{2+0\ ,\ 7+1\} = 2$

$D_x(z) = \min\{c(x,y) + D_y(z),\ c(x,z) + D_z(z)\}$
$= \min\{2+1\ ,\ 7+0\} = 3$

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

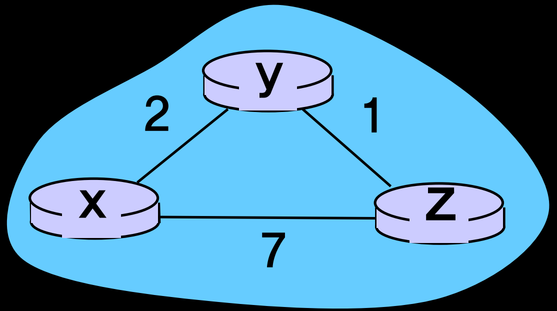
from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

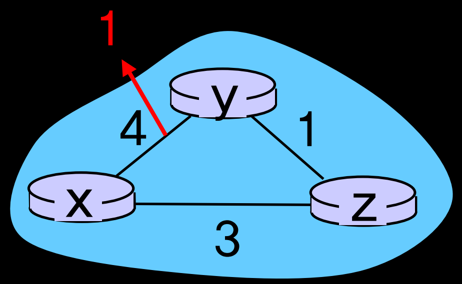
time



Note: “Good News Travels Fast”

link cost changes:

- Node detects local link cost change
- Updates routing info, recalculates distance vector
- If DV changes, notify neighbors



“Good news travels fast”

t_0 : *y* detects link-cost change, updates its DV, informs its neighbors.

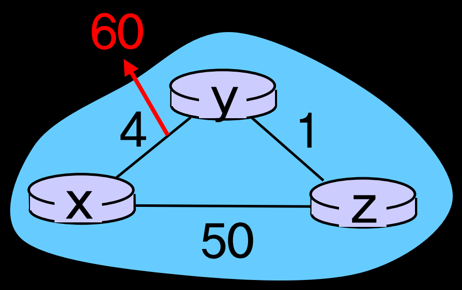
t_1 : *z* receives update from *y*, updates its table, computes new least cost path to *x* , sends its neighbors its DV.

t_2 : *y* receives *z*’s update, updates its distance table. *y*’s least costs do *not* change, so *y* does *not* send a message to *z*.

Note: “Bad News Travels Slow”

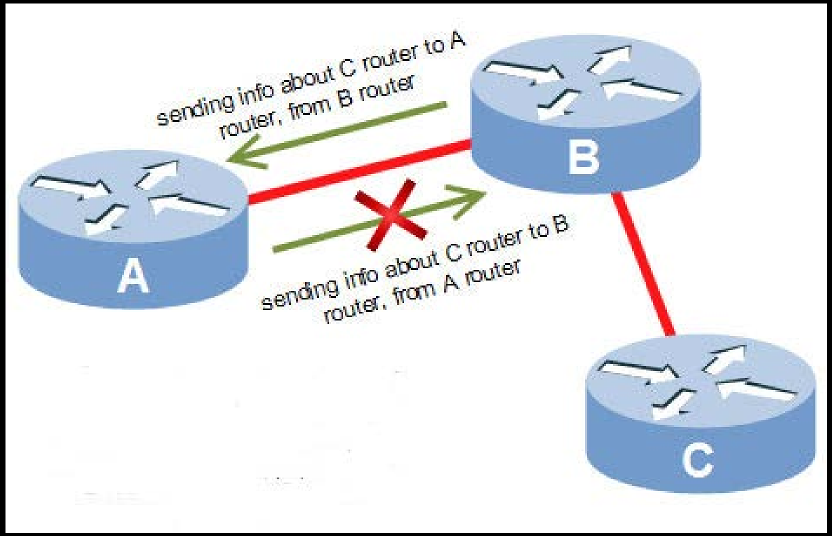
link cost changes:

- Node detects local link cost change
- Bad news travels slow – “count to infinity” problem!
- 44 iterations before algorithm stabilizes



Poisoned reverse:

- If Z routes through Y to get to X :
  - Z tells Y its (Z’s) distance to X is infinite (so Y won’t route to X via Z)



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiLkJKC9qLcAhXCjZQKHREXBBUQjRx6BAgBEAU&url=https%3A%2F%2Falistairkey.wordpress.com%2Ftag%2Fdistance-vector%2F&psig=AOvVaw1_4JDVSMuf7pCzXYdSMtU3&ust=1531806491343471

Distance Vector vs. Link State

Distance Vector	Link State
Entire routing table is sent as an update	Incremental updates
Slow convergence	Fast convergence
Updates are sent to directly connected neighbor only	Updates are broadcast to entire network
Routers don't know the entire network topology	Routers know the entire network topology of that area
Count–to–infinity problem	No routing loops
Simple configuration	Complex configuration (for large network)
Example: RIP (Routing Information Protocol)	Example: OSPF (Open Shortest Path First)

04. Intra-AS Routing: OSPF

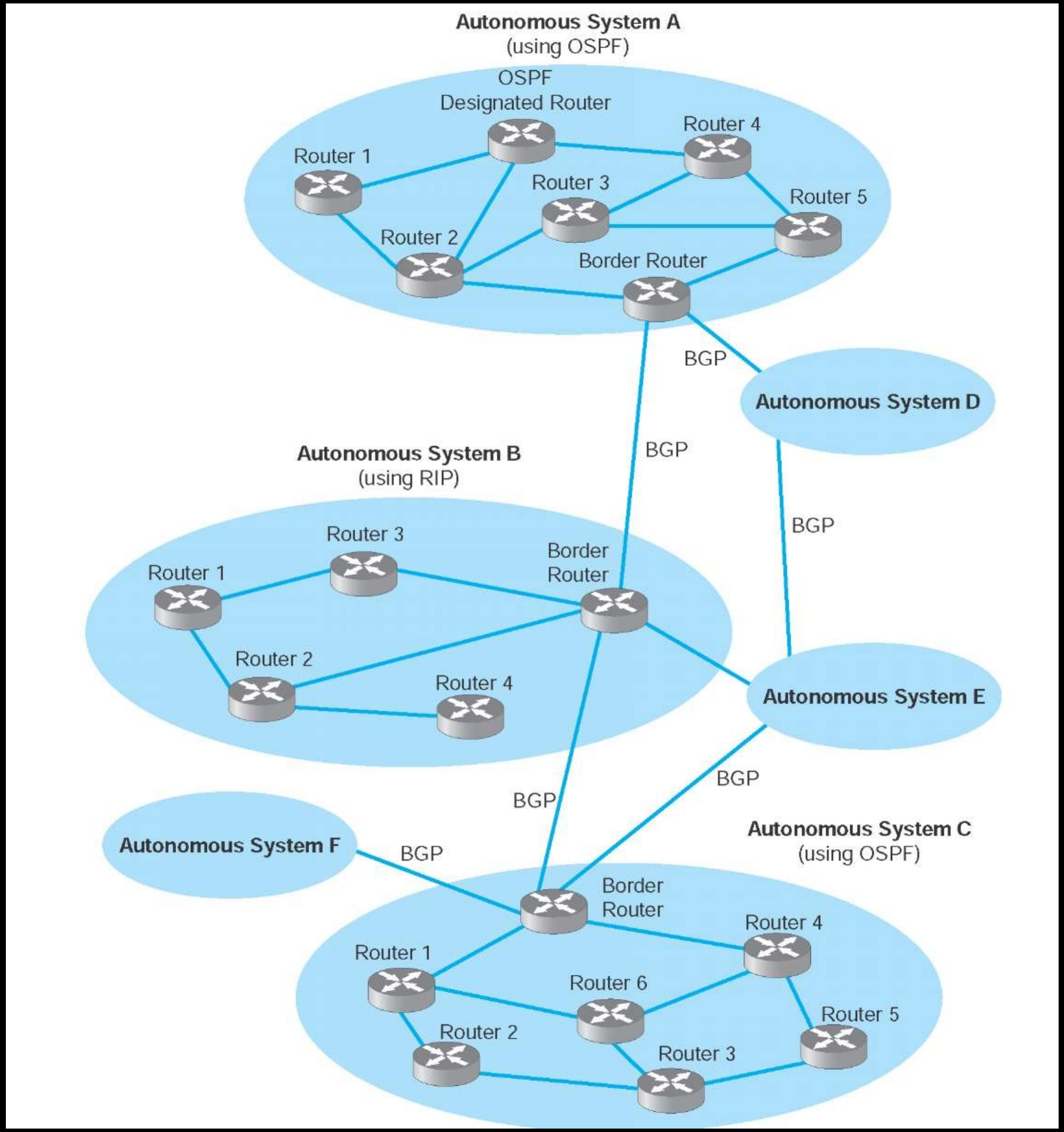
Scalability of Routing

- So far we have assumed
  - all routers are identical
  - network is "flat"

  but not true in practice
- In actual, with billions of destinations
  - can't store all destinations in routing tables! ➡ Scalability issue
  - routing table exchange would swamp links!
- Administrative autonomy
  - internet = network of networks
  - each network admin may want to control routing in its own network

Autonomous System

- Aggregate routers into regions known as "autonomous systems" (AS)
- Autonomous system (a.k.a. domain)
  - a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators on behalf of a single administrative entity (from Wikipedia)



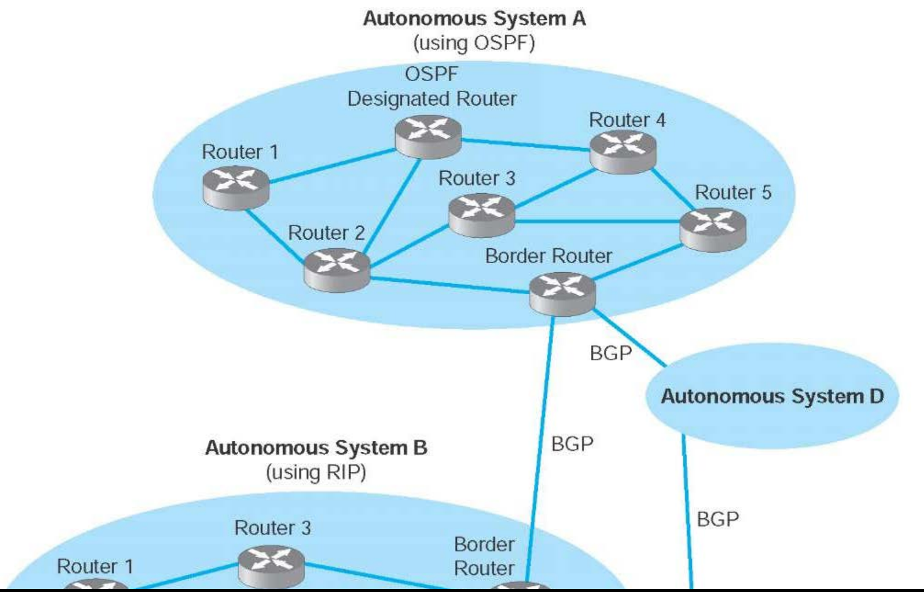
출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiGhrnsm6PcAhUHj5QKHQzuCclQjRx6BAgBEAU&url=http%3A%2F%2Fwhat-when-how.com%2Fdata-communications-and-networking%2Frouting-data-communications-and-networking%2F&psig=AOvVaw3EoL_Etg6pl2m8FcN-F-FE&ust=1531816626795883

Intra-AS routing

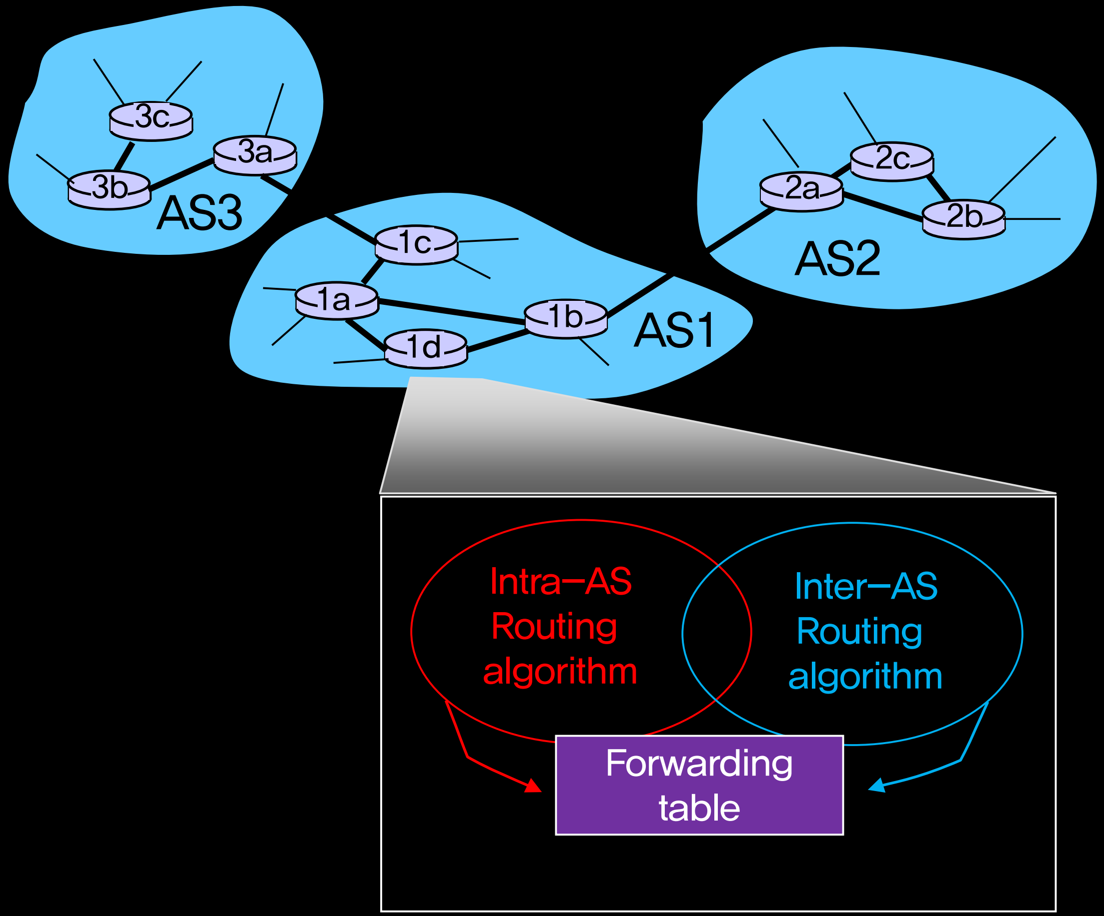
- Routing among hosts, routers in same AS ("network")
- All routers in AS must run same intra-domain protocol
- Routers in different AS can run different intra-domain routing protocol
- Gateway router (a.k.a. border router): at "edge" of its own AS, has link(s) to router(s) in other AS'es

Inter-AS routing

- Routing among AS'es
- Gateway routers perform inter-domain routing (as well as intra-domain routing)

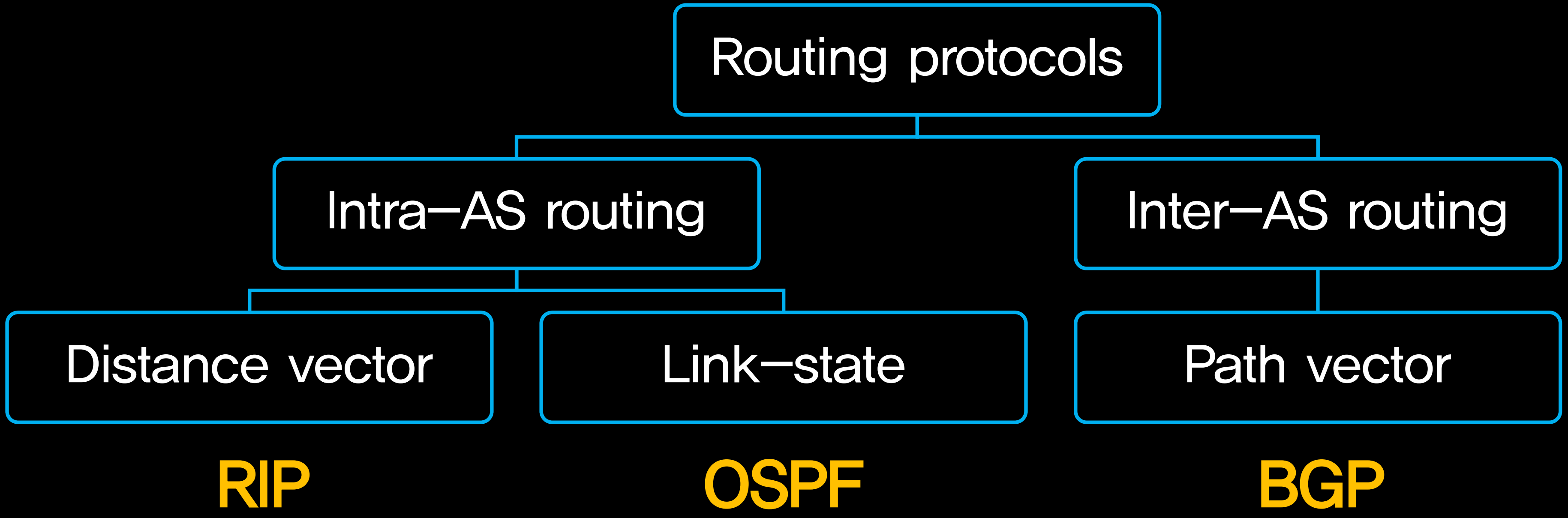


Interconnected ASes



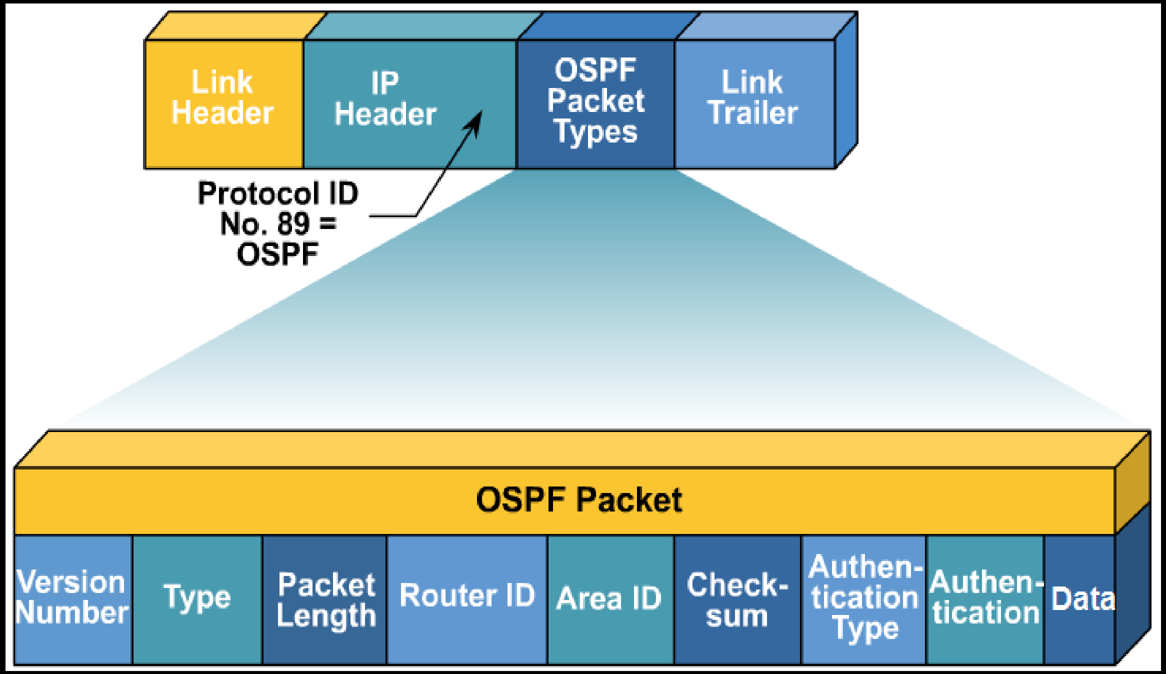
- Forwarding table configured by both intra– and inter–AS routing algorithm
  - intra–AS routing determine entries for destinations within AS
  - inter–AS & intra–AS determine entries for external destinations

Routing Protocols



OSPF (Open Shortest Path First)

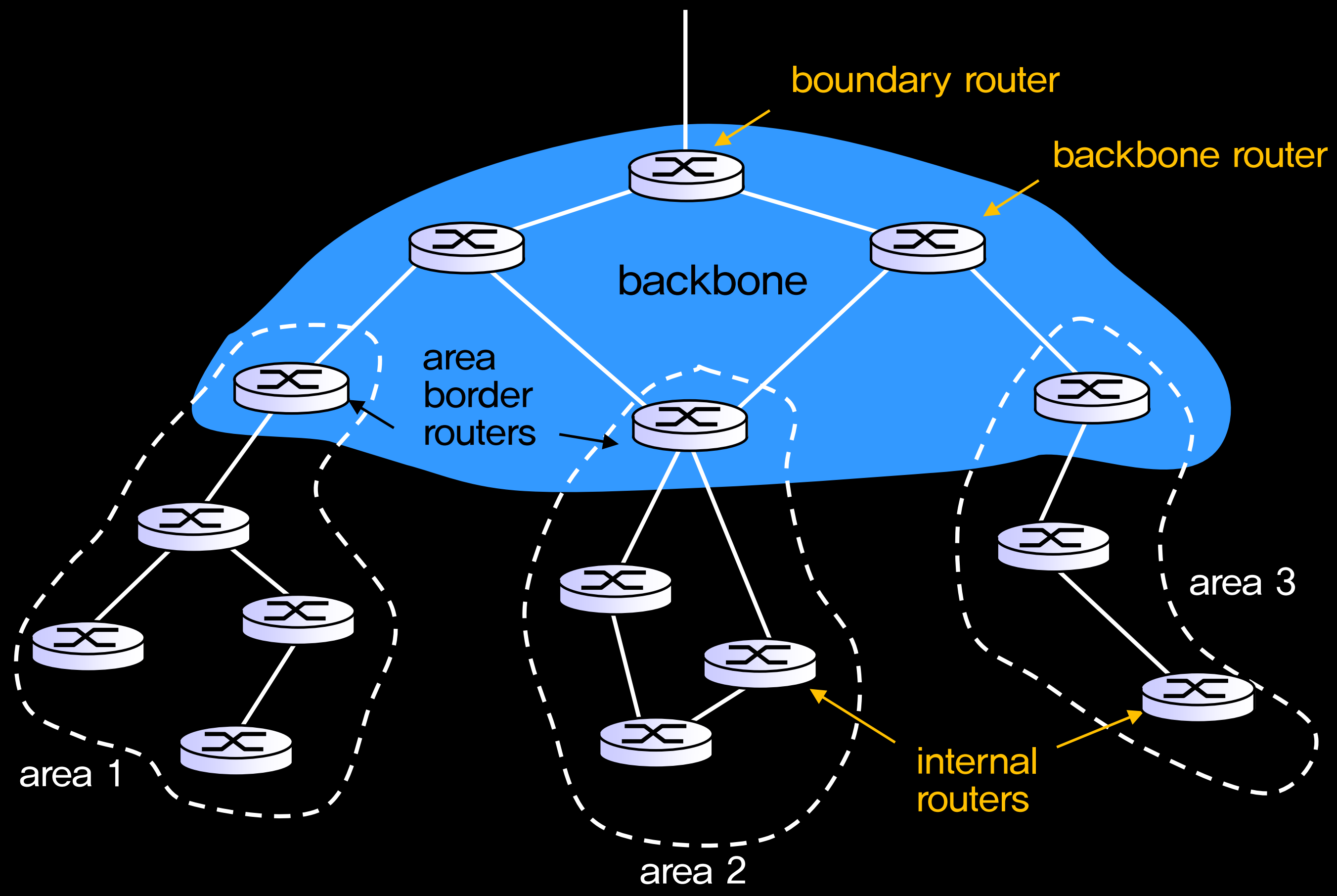
- Most common intra-AS routing protocol
- "open": publicly available
- Uses link-state algorithm
  - LSAs to all other routers in entire AS
    - carried directly over IP
  - topology map at each node
  - route computation using Dijkstra's algorithm
- Security: all OSPF messages authenticated (to prevent malicious intrusion)



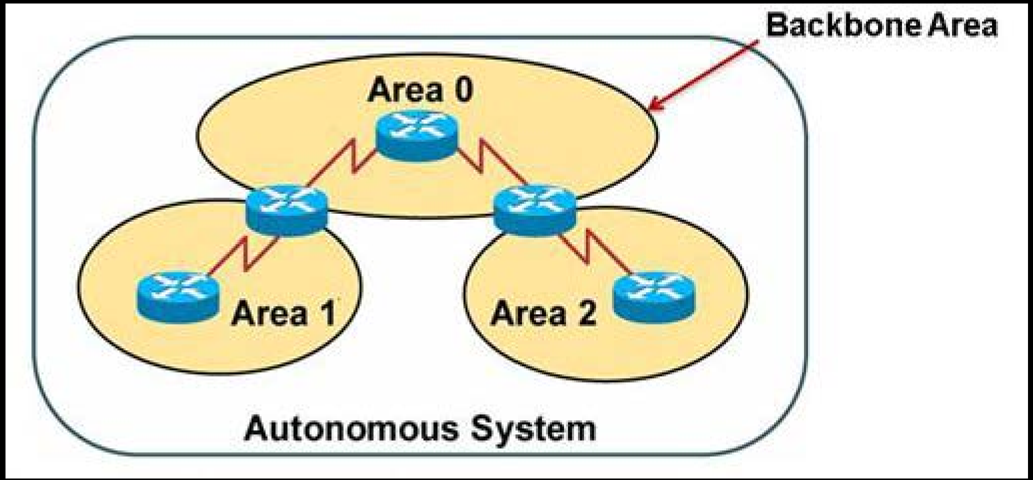
출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj5ic3_i6jcAhVMv5QKHdVTA9cQjRx6BAgBEAU&url=https%3A%2F%2Flearningnetwork.cisco.com%2Fthread%2F97672&psig=AOvVaw2f0bkwlGbRf21voeka86K8&ust=1531984118257984

Hierarchical OSPF

- Hierarchical OSPF in large domains
  - minimizes routing update traffic
  - localizes the impact of topology changes to an area



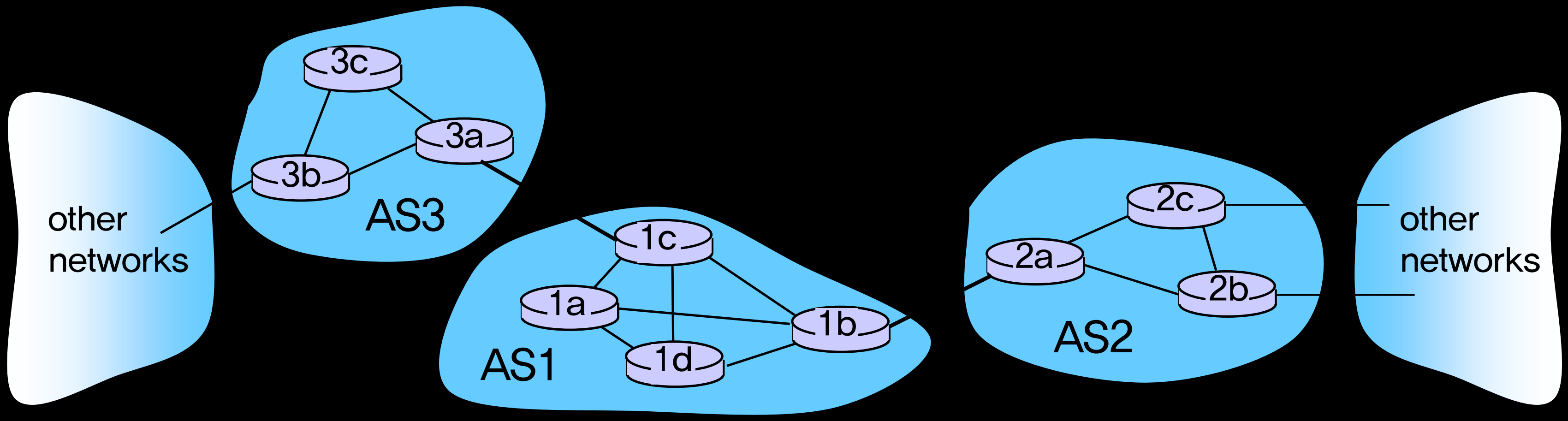
- Two-level hierarchy: local area, backbone
  - LSAs only in area
  - each nodes has detailed area topology; only know direction (shortest path) to nets in other areas
- Area border routers
  - "summarize" distances to nets in own area, advertise to other area border routers
- Backbone routers
  - run OSPF routing limited to backbone
  - connect to other AS'es



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&ved=2ahUKEwiLk6ugzqDcAhXLmZQKHWJtBu8QjRx6BAgBEAQ&url=http%3A%2F%2Fwww.srmuniv.ac.in%2Fsites%2Fdefault%2Ffiles%2Ffiles%2FUNIT-III(3).pdf&psig=AOvVaw12MwLt2_fOhEKh8oY2sbQf&ust=1531726885064814

05. Inter-AS Routing: BGP

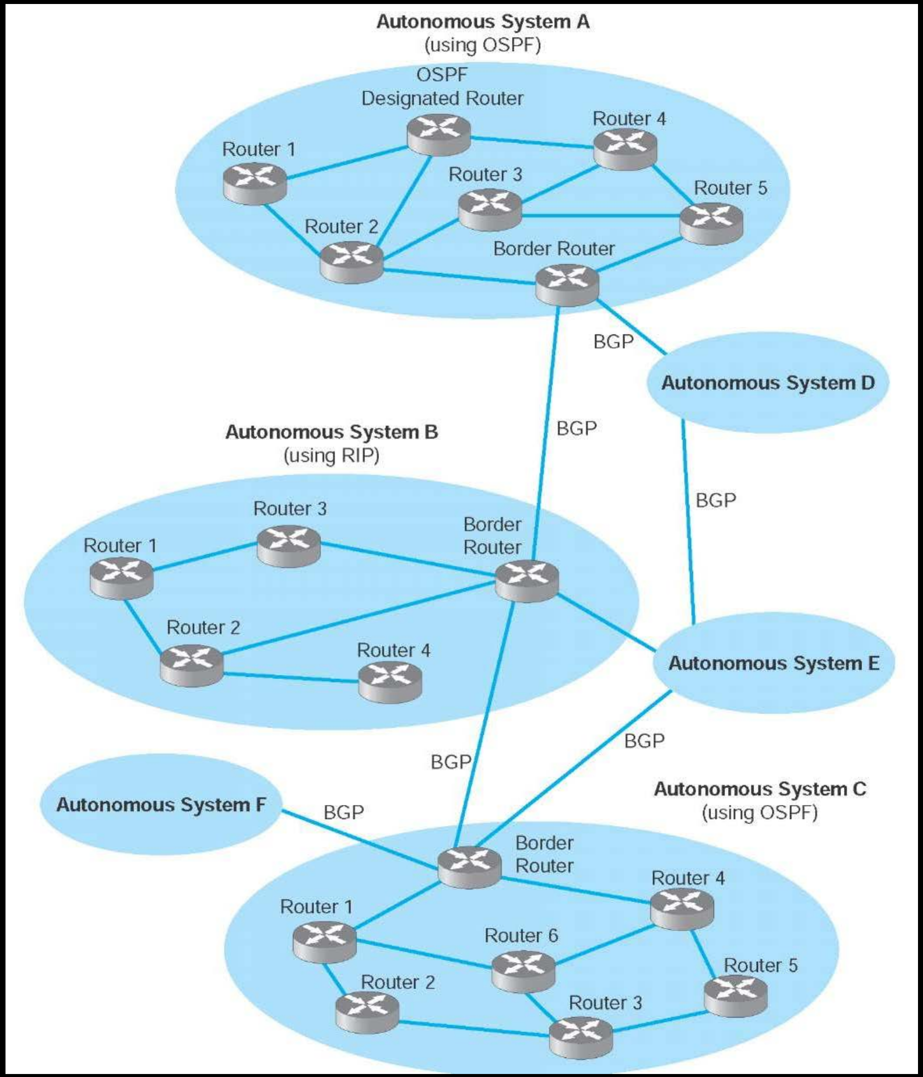
Inter-AS Tasks



- Suppose router in AS1 receives datagram destined outside of AS1:
  - router should forward packet to gateway router, but which one?
- AS1 must:
  - learn which destinations are reachable through AS2, which through AS3
  - propagate this reachability info to all routers in AS1
- Job of inter-AS routing!

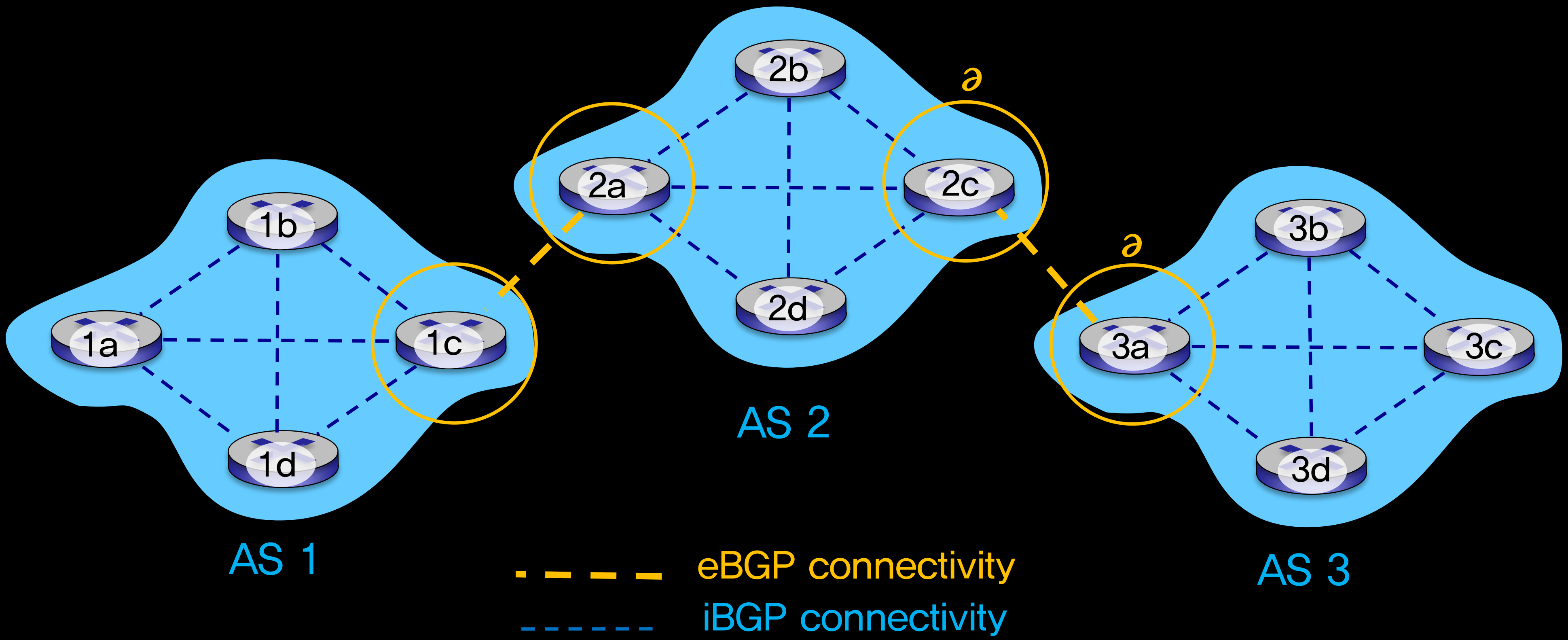
BGP (Border Gateway Protocol)

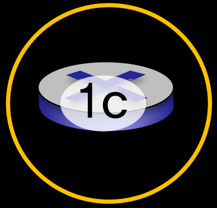
- de facto inter-domain routing protocol
  - "glue that holds the Internet together"
  - allows subnet to advertise its existence to rest of Internet: "I am here"
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine "good" routes to other networks based on reachability information and policy



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiGhrnsm6PcAhUHj5QKHQzuCclQjRx6BAgBEAU&url=http%3A%2F%2Fwhat-when-how.com%2Fdata-communications-and-networking%2Frouting-data-communications-and-networking%2F&psig=AOvVaw3EoL_Etg6pl2m8FcN-F-FE&ust=1531816626795883

eBGP, iBGP Connections

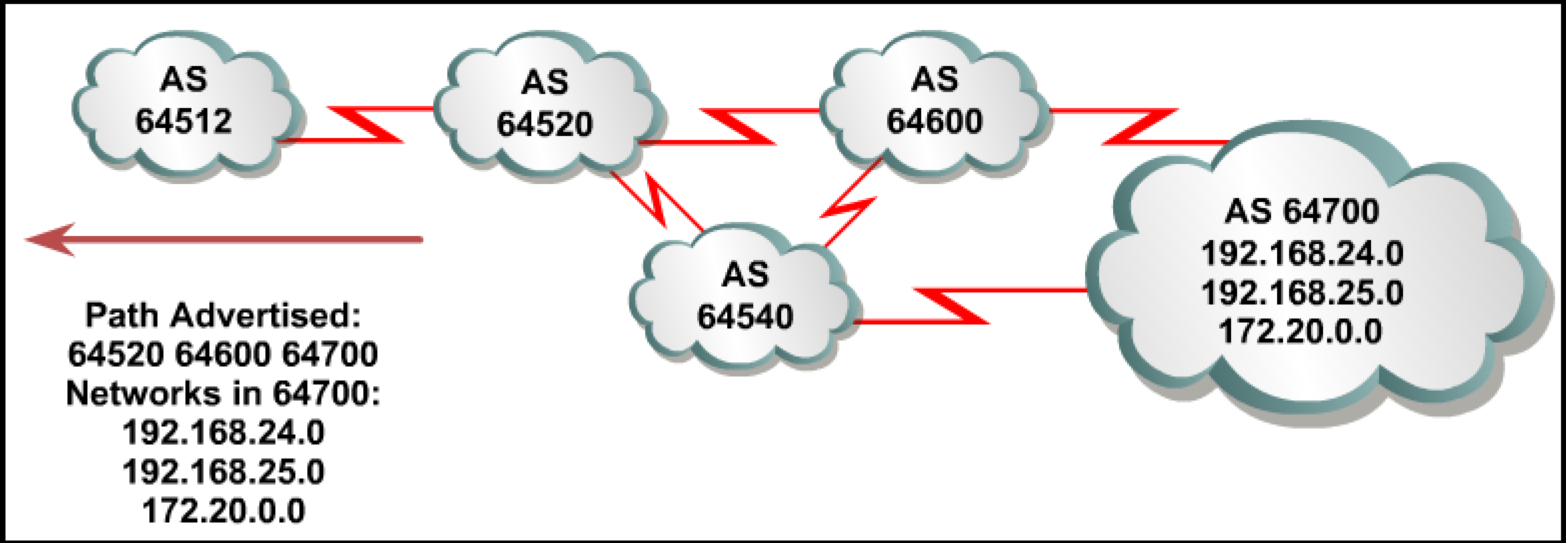




gateway routers run both eBGP and iBGP protocols

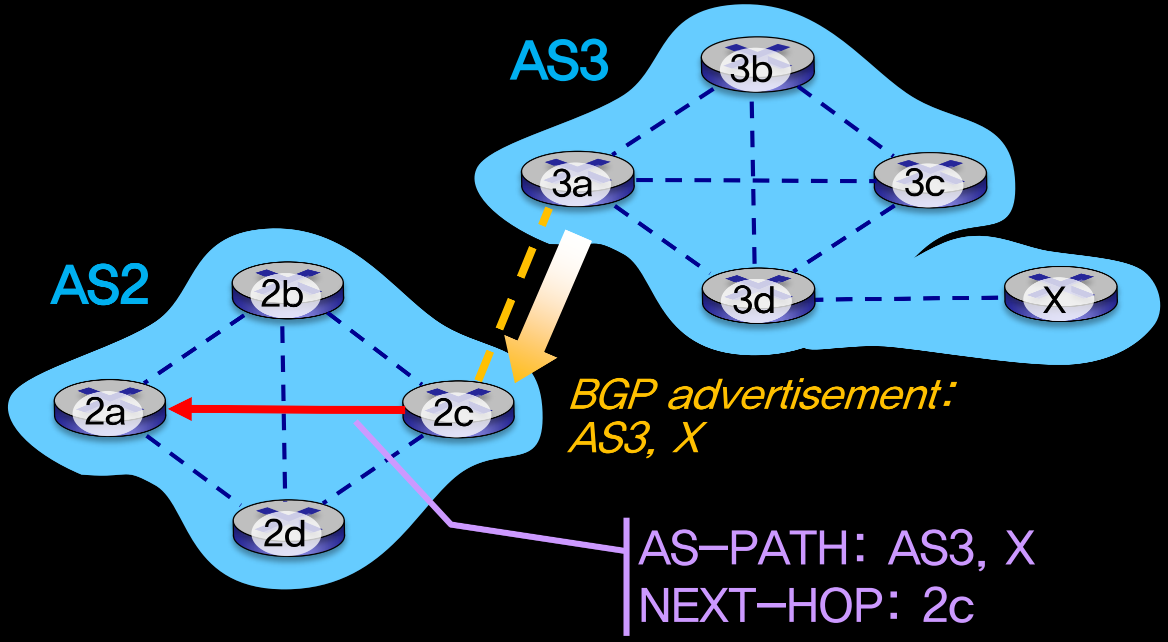
BGP Basics

- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising paths to different destination network prefixes ("path vector" protocol)
- When AS64520 advertises a path toward AS64700 to AS64512:
  - AS64520 promises to AS64512 it will forward datagrams towards 192.168.24/21



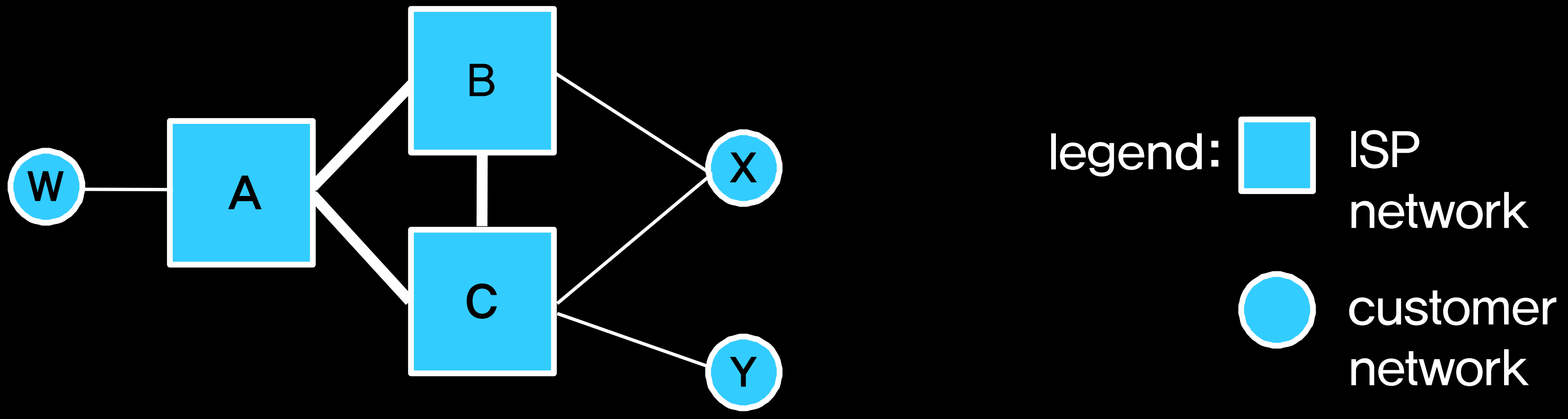
출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj6qKy2mbHcAhXPNpQKHYdmDpsQjRx6BAgBEAU&url=http%3A%2F%2Fmars.tekkom.dk%2Fw%2Findex.php%2FPederrs_CCNP_1&psig=AOvVaw2WQD2hzrcgVialmwXdj4Rd&ust=1532296885093008

- Advertised prefix includes BGP attributes
  - network prefix + attributes = “route”
- Two important attributes:
  - AS-PATH: list of ASes through which prefix advertisement has passed
  - NEXT-HOP: indicates specific internal-AS router to next-hop AS



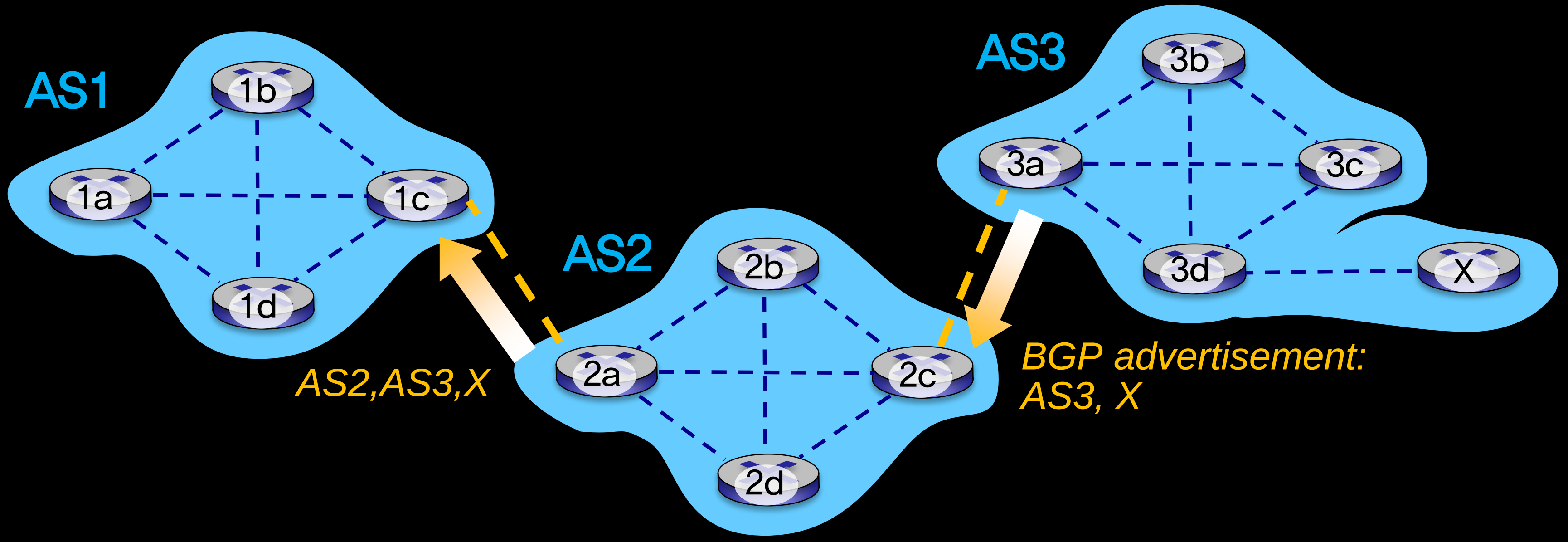
- Policy-based routing:
  - gateway receiving route advertisement uses import policy to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to advertise path to other neighboring ASes

BGP Policy Achievement via Advertisements



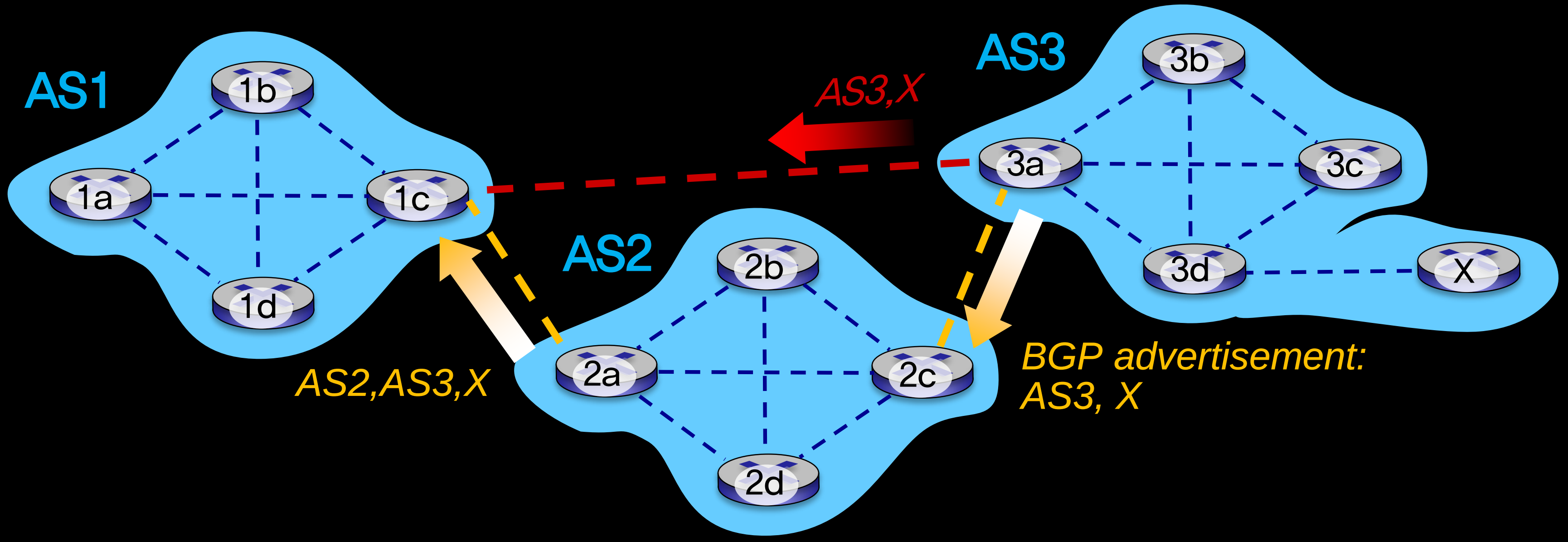
- A advertises path Aw to B and to C
- B chooses not to advertise BAw to C:
  - B gets no “revenue” for routing CBAw, since none of C, A, w are B’s customers
  - C does not learn about CBAw path
- C will route CAw (not using B) to get to w

BGP Path Advertisement: Example



- 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

BGP Path Advertisement: Example (cont'd)



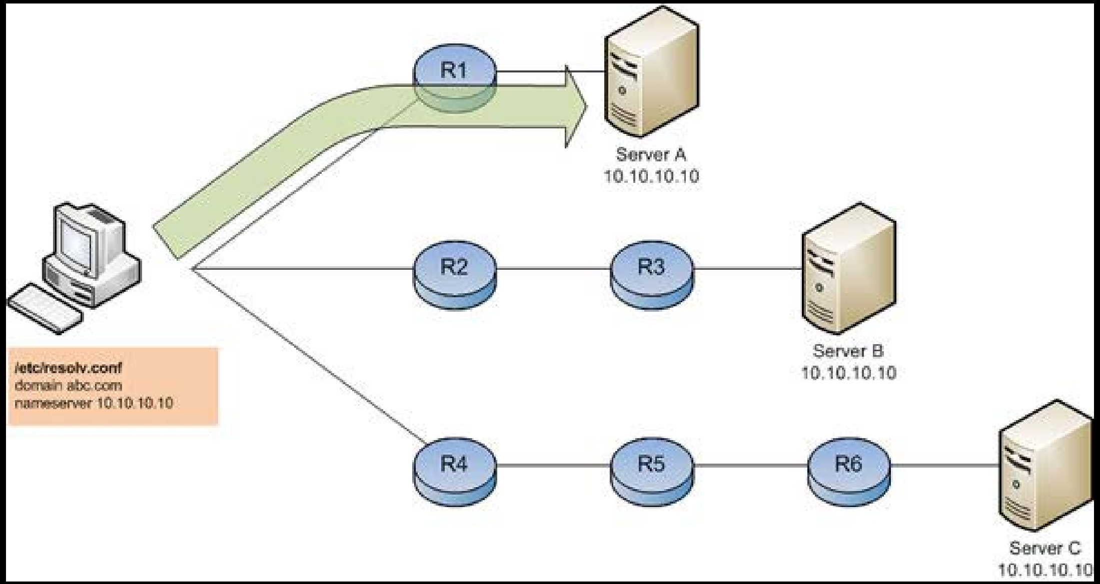
Gateway router may learn about multiple paths to destination:

- 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

- IP address class (additional)

Class D	1	1	1	0	multicast group id (28 bits)	Multicast
Class E	1	1	1	1	undefined	Anycast(unused)

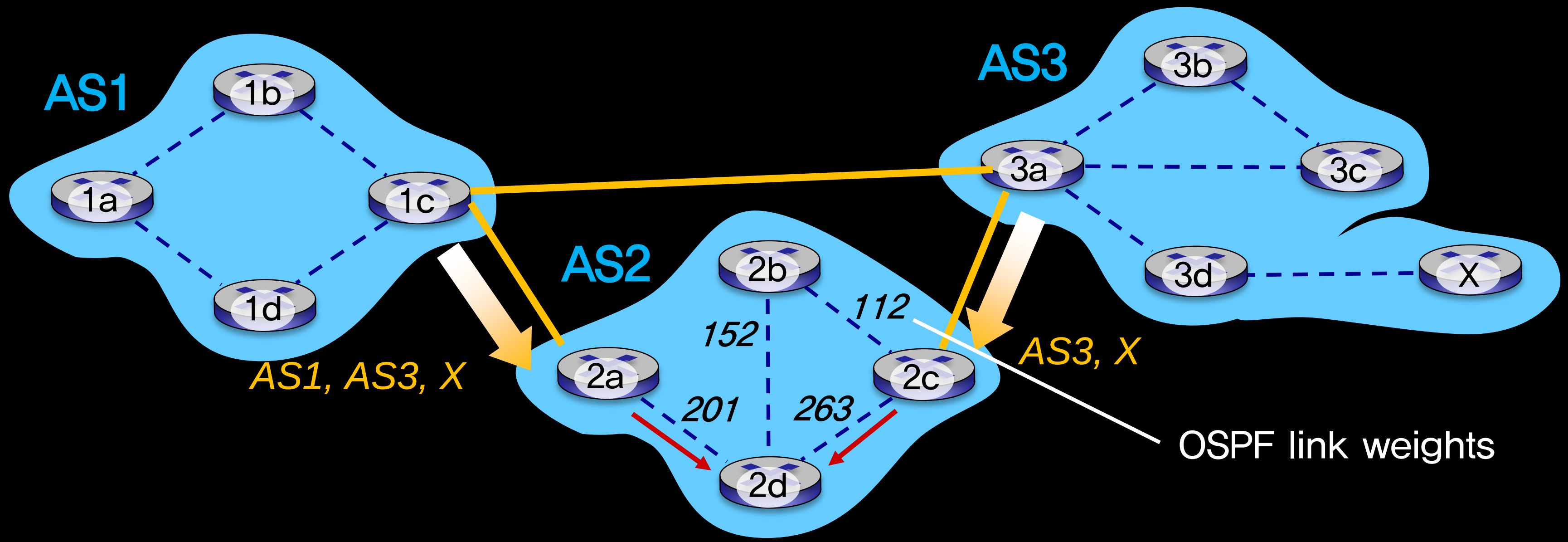
- Anycast DNS using BGP
  - same IP address to each DNS server
  - when a BGP router receives multiple route advertisements for this IP addr., it thinks they are different paths to the same physical location
  - each router locally pick the “best” route to that IP address



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj74bmgprHcAhUEnZQKHe5VCroQjRx6BAgBEAU&url=http%3A%2F%2Fpingbin.com%2F2011%2F03%2Fwhat-is-any-cast-dns%2F&psig=AOvVaw27aez2QiygUFpV5s_Aja50&ust=1532300473992824

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): don't worry about inter-domain cost!
  - the idea behind is to get packets out of its AS as quickly as possible without worrying about the cost of the remaining portions of the path outside its AS to destination



BGP Route Selection

- Router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. Closest NEXT-HOP router: hot potato routing
  4. additional criteria

Difference between Intra- and Inter-AS Routing

Why different?

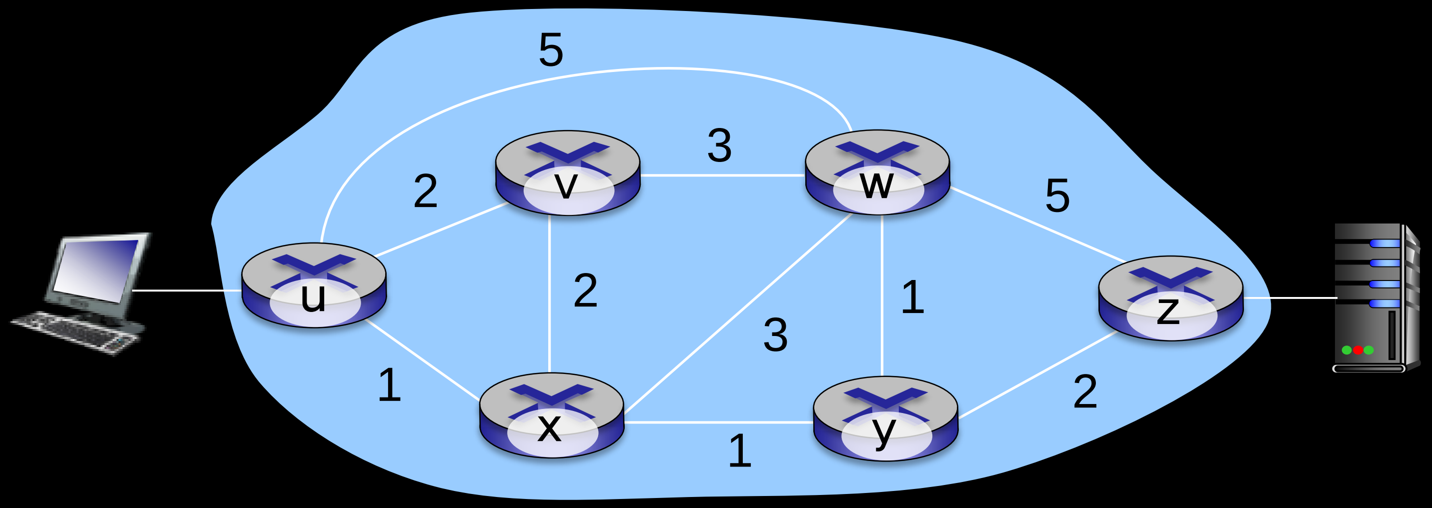
- Scalability: hierarchical routing saves table size, reduced update traffic

What different?

	Inter-AS	Intra-AS
policy	admin wants control over how its traffic routed, who routes through its network	single admin, so no policy decisions needed
performance	can focus on performance	policy may dominate over performance

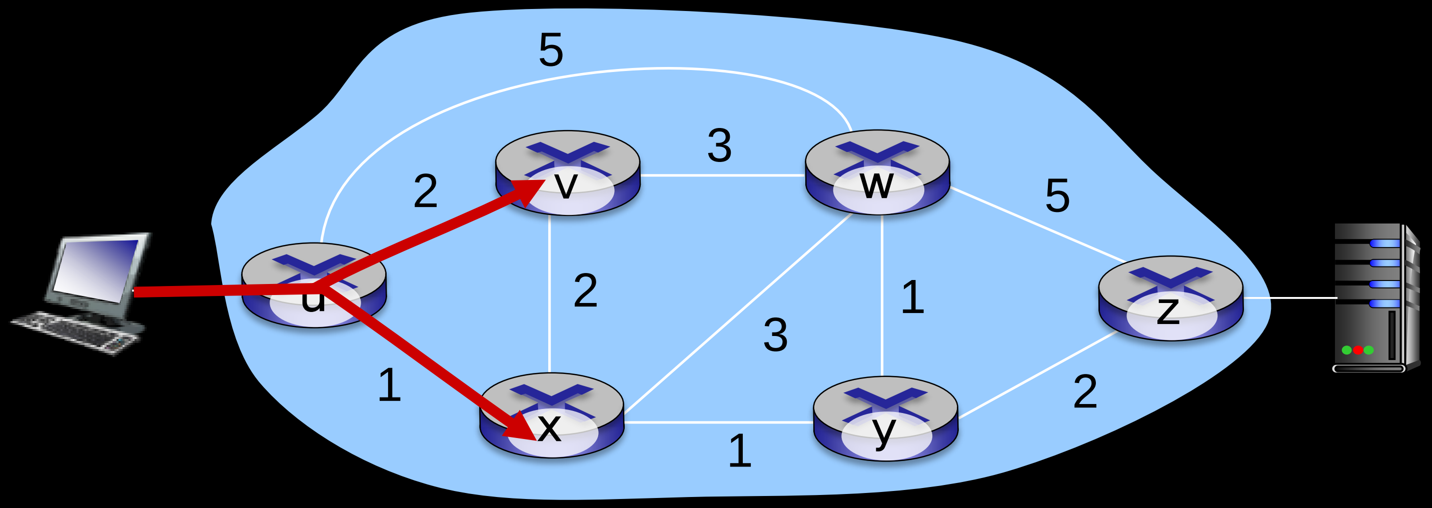
06. Software Defined Networking

Traffic Engineering with 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

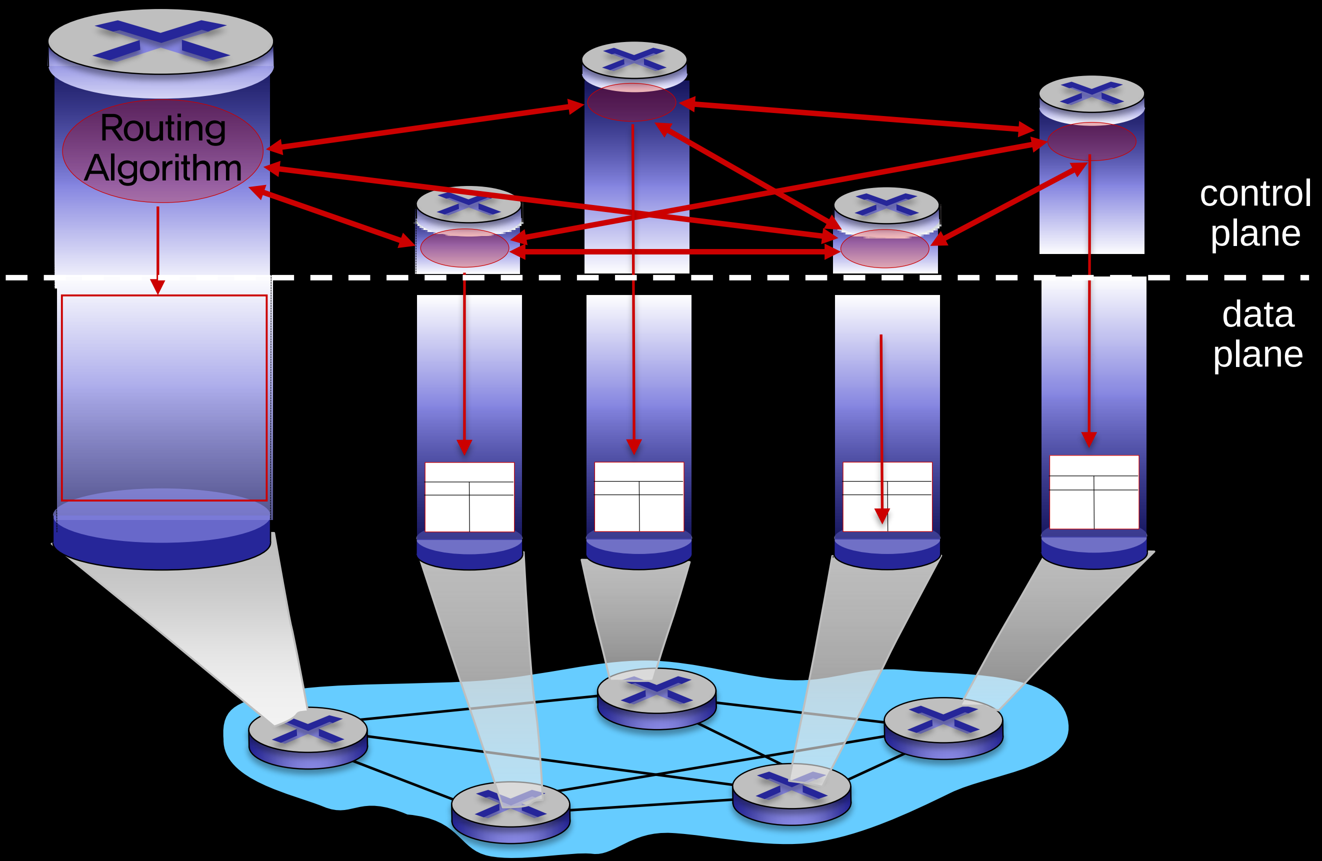
Traffic Engineering with Traditional Routing



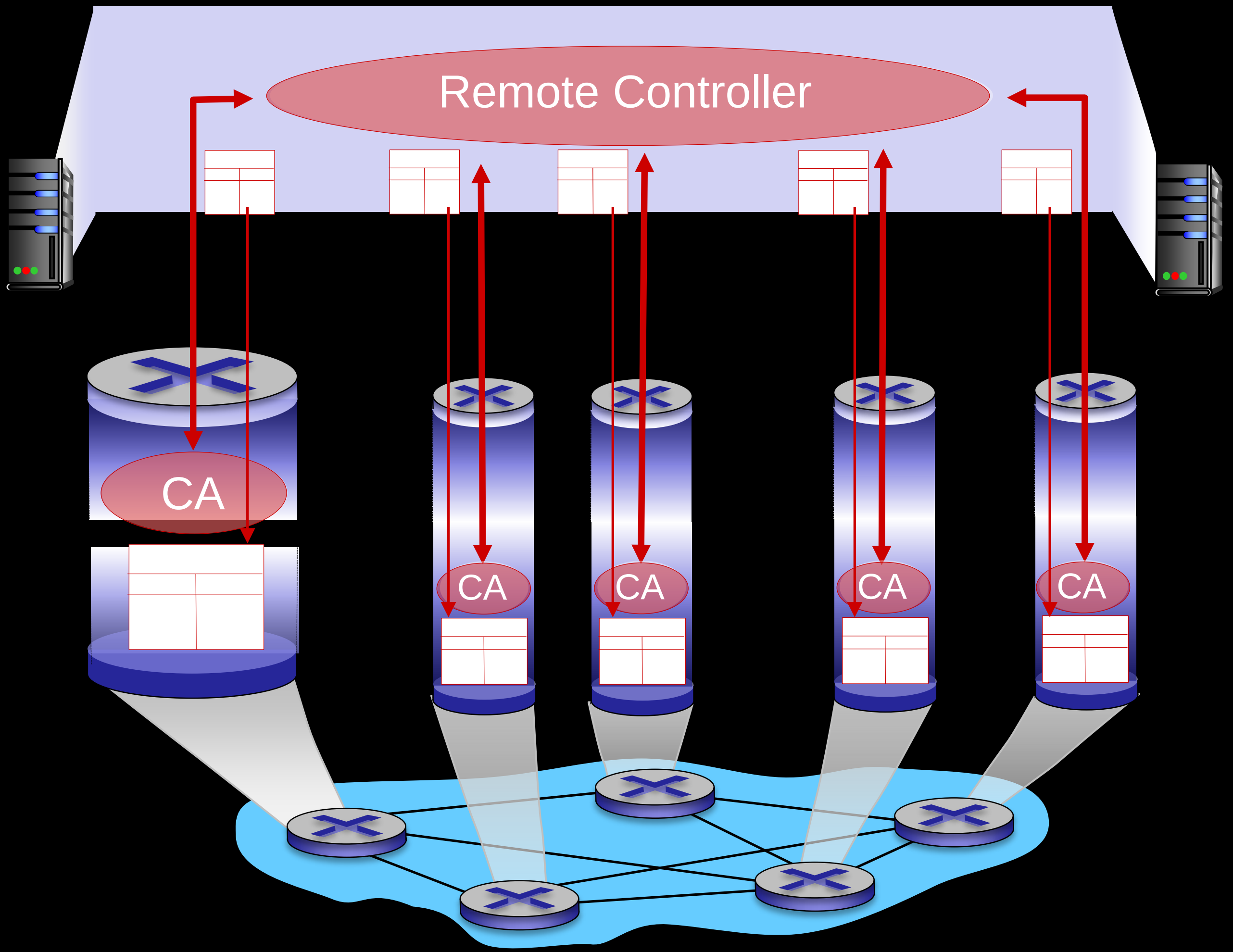
- Q: What if network operator wants to split u-to-z traffic along uvwz and uxyz (load balancing)?
- A: Can't do it (or need a new routing algorithm)

Traditional Network Layer

- 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, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different “middle boxes” for different network layer functions: firewalls, load balancers, NAT boxes, ..

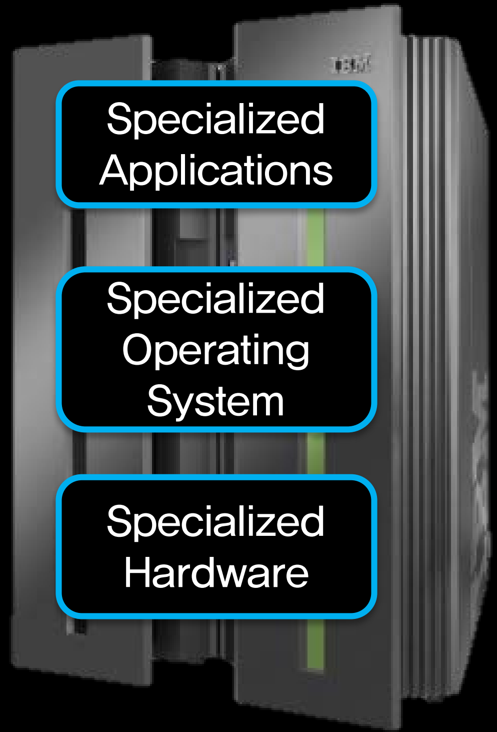


Logically Centralized Control Plane



- A distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables
- Programmable routers
  - centralized "programming" easier: compute tables centrally and distribute
  - distributed "programming" more difficult: compute tables as result of distributed algorithm (protocol) in each and every router
- Open (non-proprietary) control plane

Analogy: Mainframe to PC Evolution



App

Open Interface

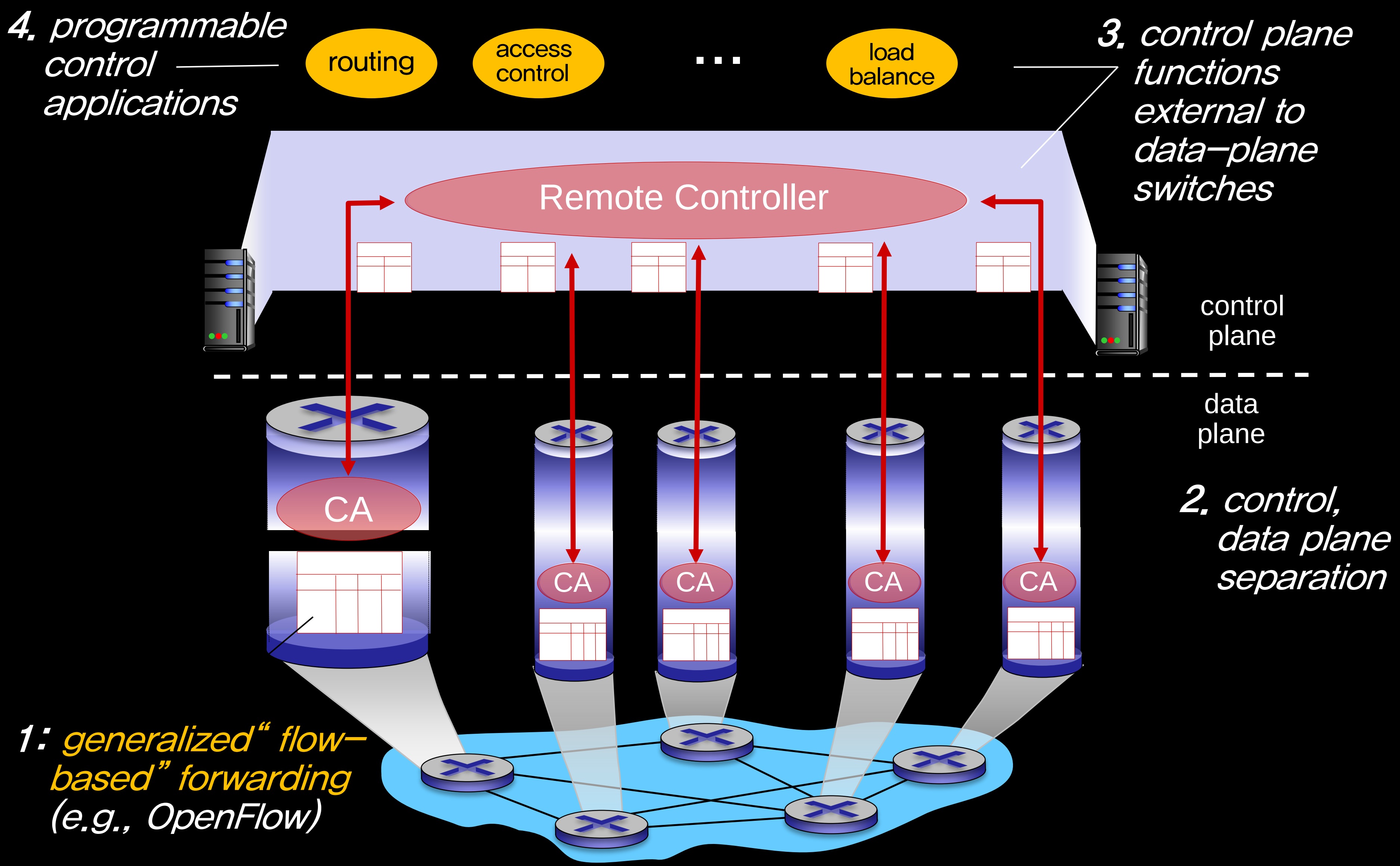
Windows (OS) or Linux or Mac OS

Open Interface

Vertically integrated
Closed, proprietary
Slow innovation
Small industry

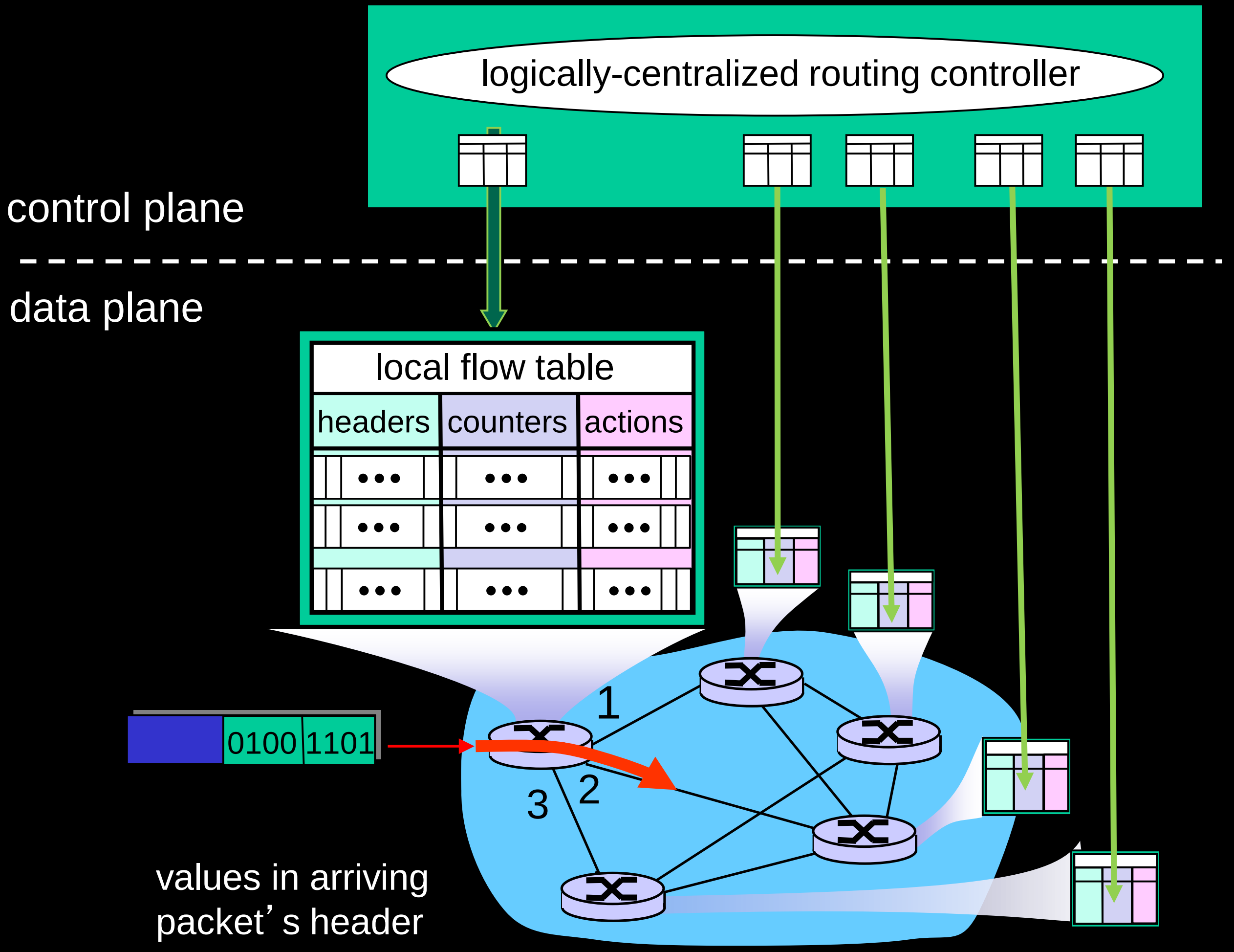
Horizontal
Open interfaces
Rapid innovation
Huge industry

Software Defined Networking (SDN)



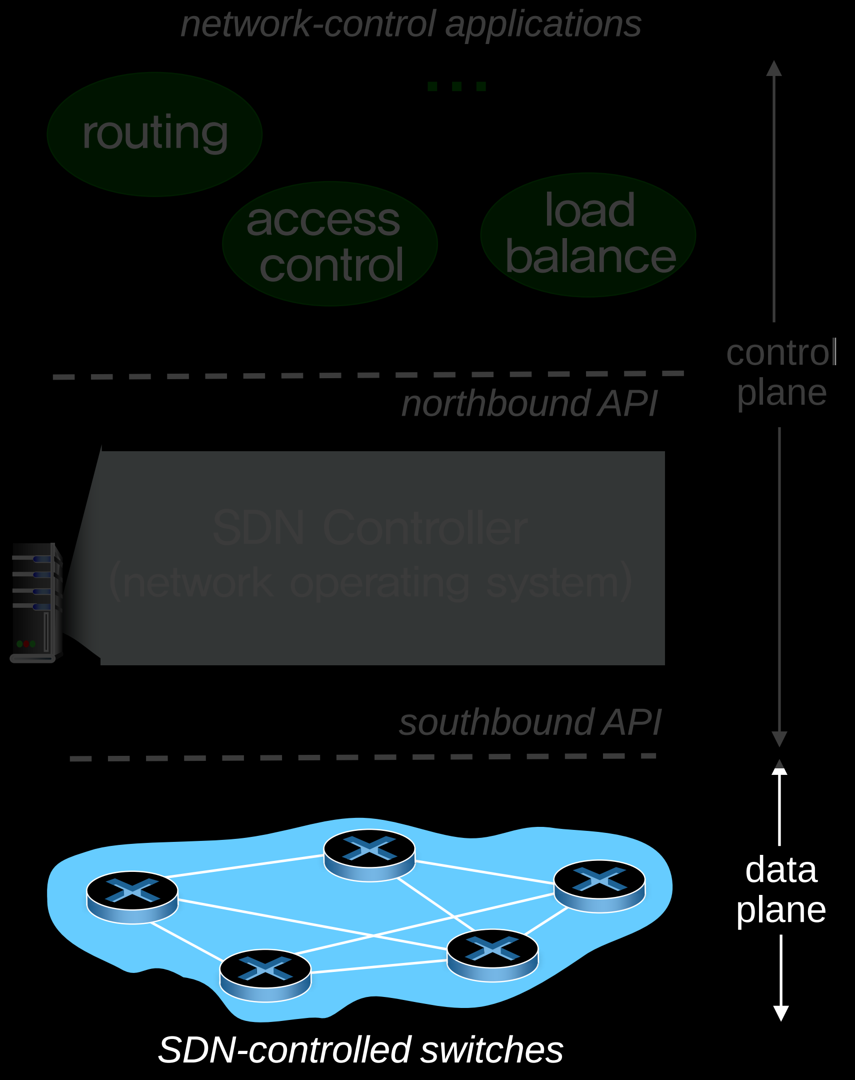
Generalized Forwarding

- Each router contains a flow table that is computed and distributed by a logically centralized routing controller
- Generalized forwarding: forwarding based on any set of header field values (cf. destination-based forwarding)



Data Plane Switches

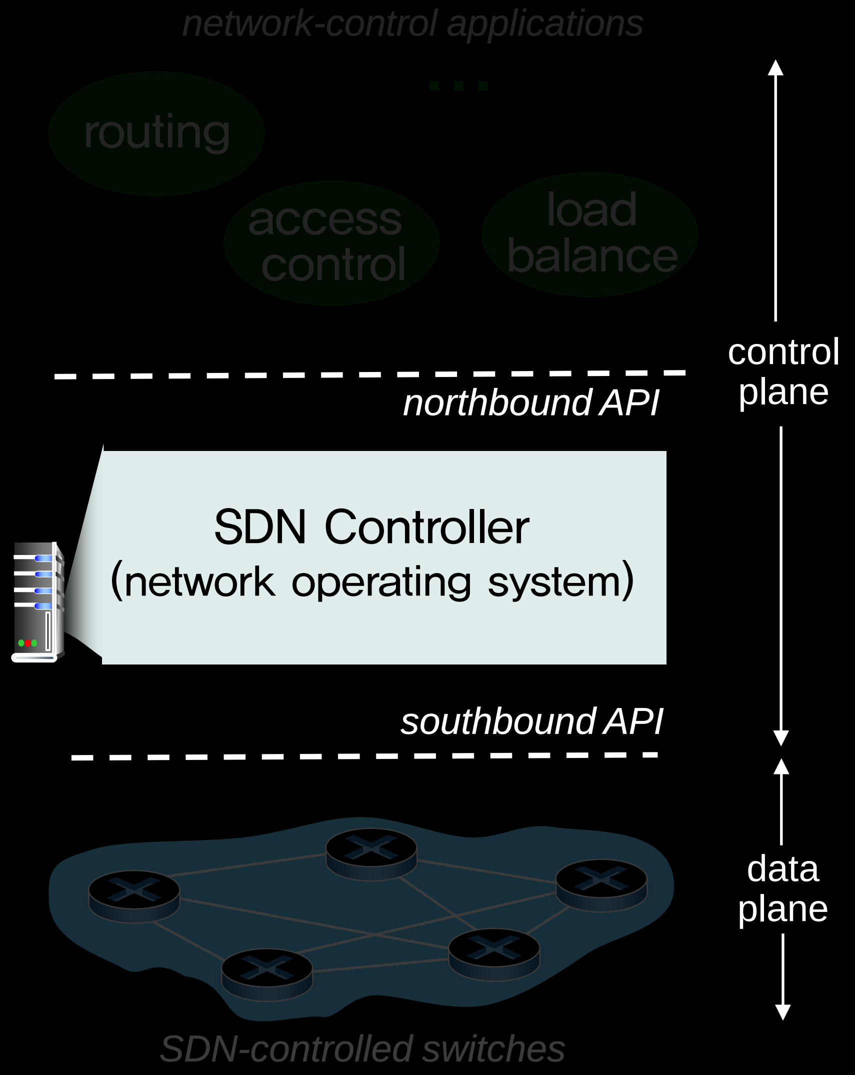
- Fast, simple, commodity switches implementing generalized data-plane forwarding (Section 4.4) in hardware
- Switch flow table computed, installed by controller
- API for table-based switch control (e.g., OpenFlow)
- Protocol for communicating with controller (e.g., OpenFlow)



SDN Controller (Network OS)

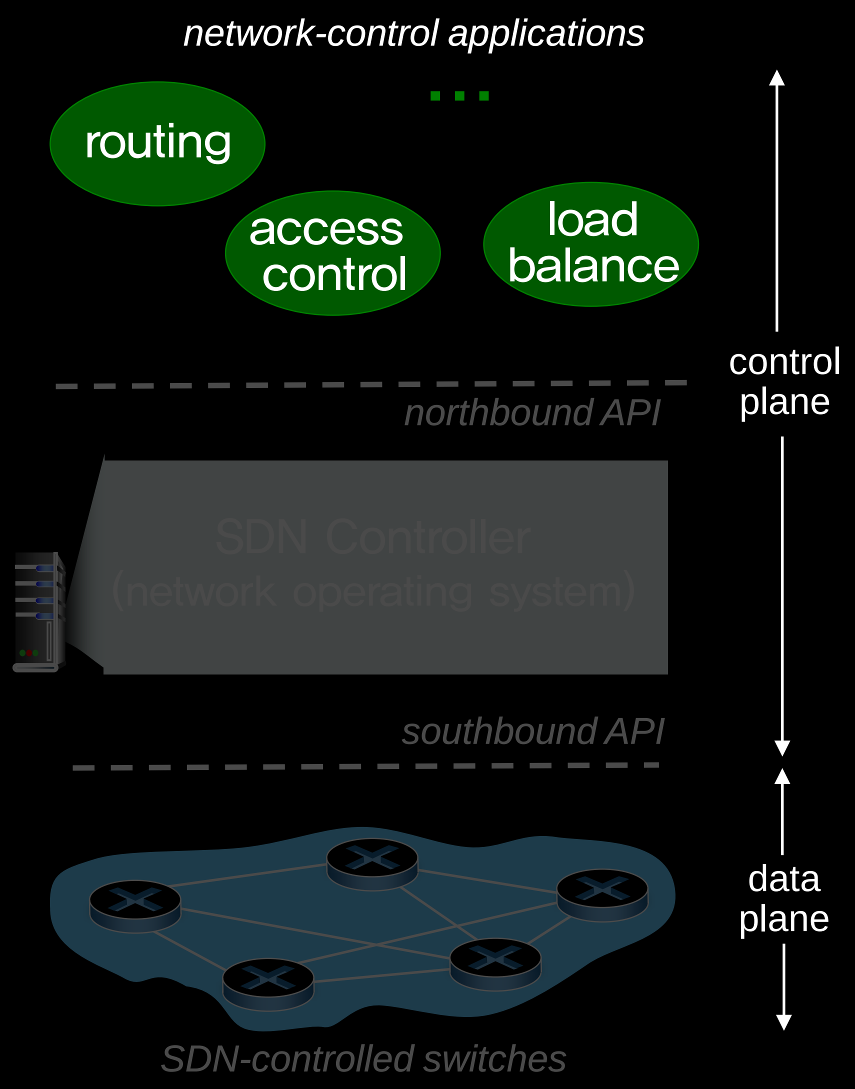
- Maintain network state information
- Interacts with network control applications "above" via northbound API
- Interacts with network switches "below" via southbound API
- Implemented as distributed system for performance, scalability, fault-tolerance, robustness

  => "logically" centralized



Network Control Applications

- "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



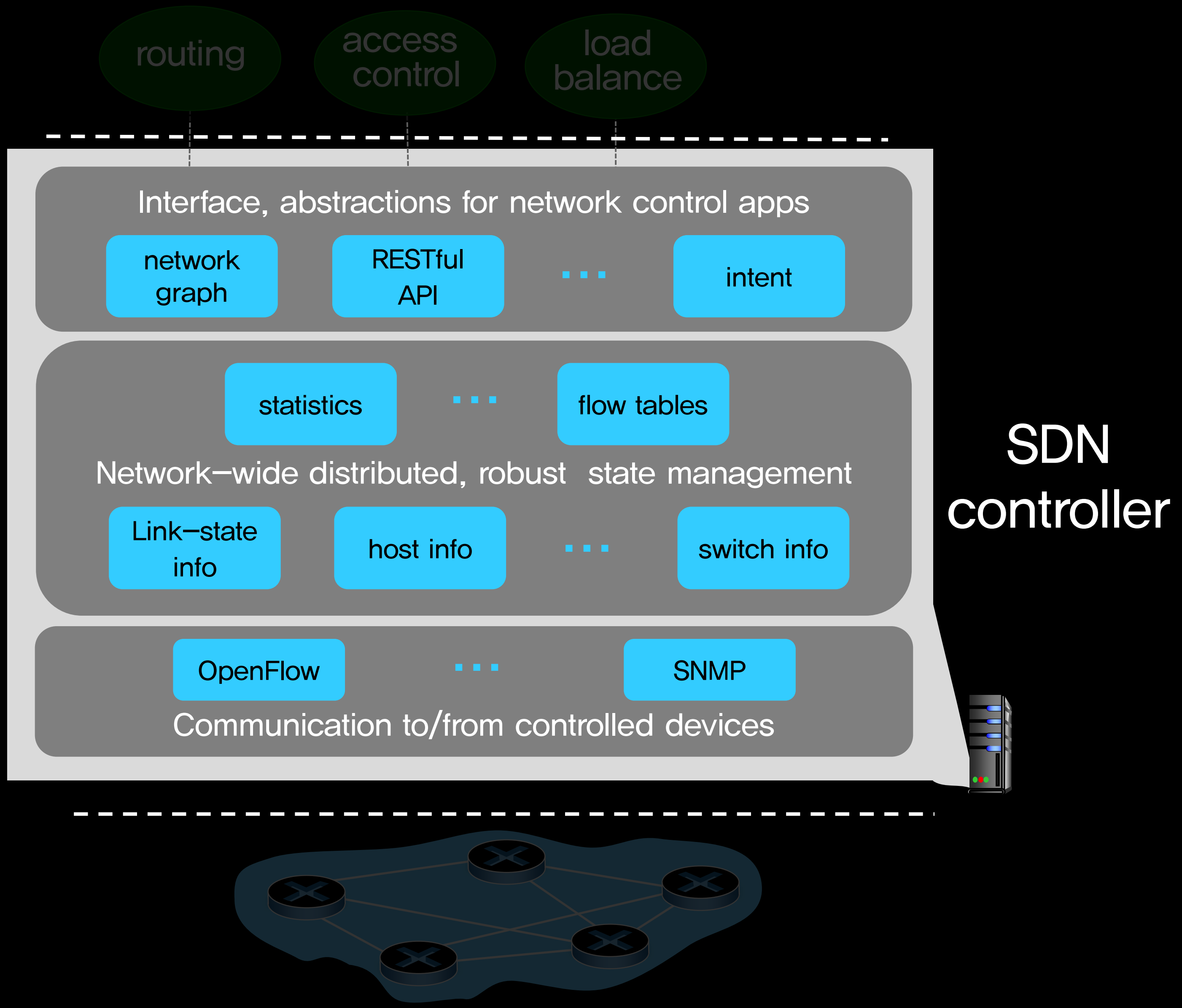
07. OpenFlow

Components of SDN Controller

Interface layer to network control apps: abstractions API

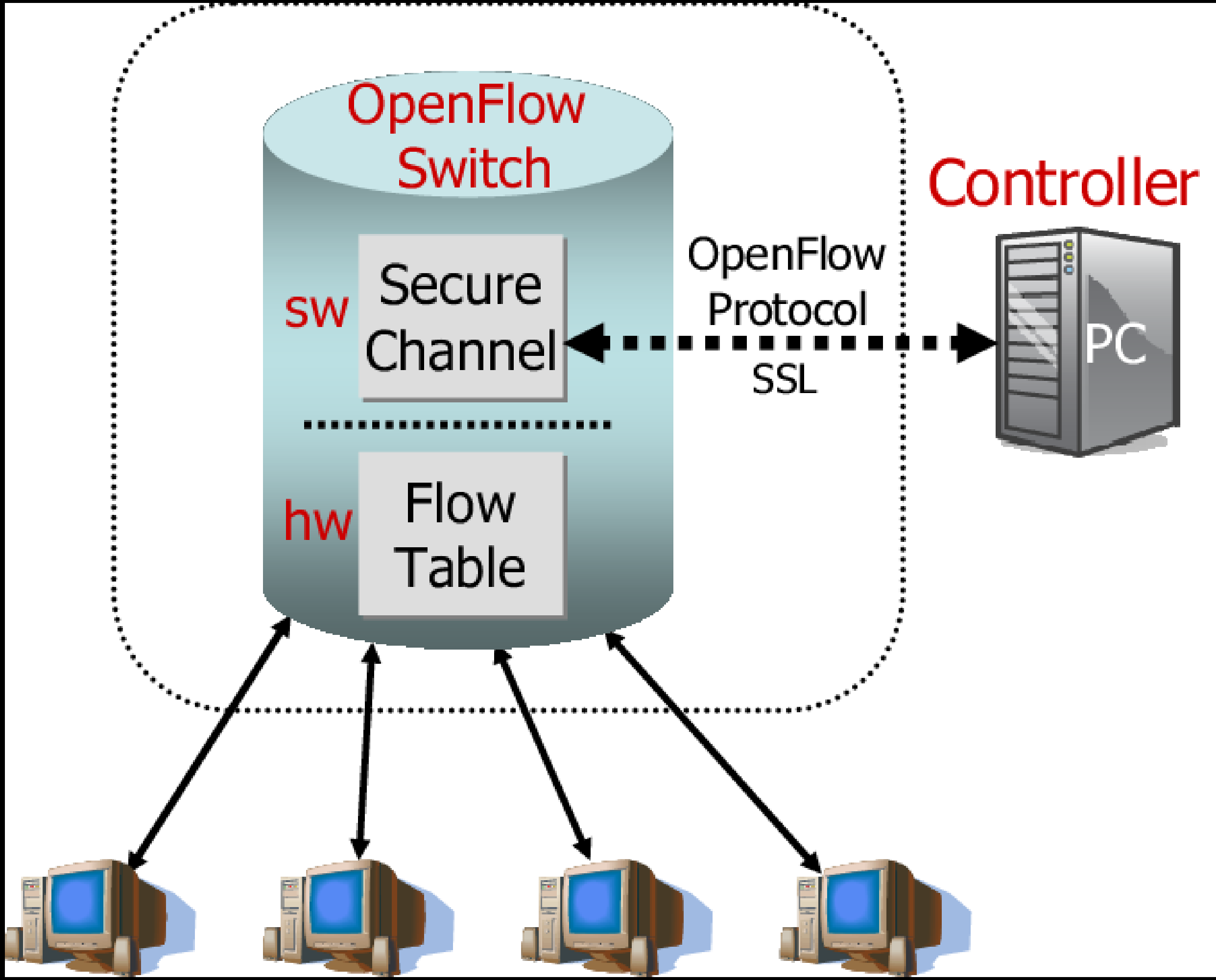
Network-wide state management layer: state of networks links, switches, services: a *distributed database*

communication layer: communicate between SDN controller and controlled switches



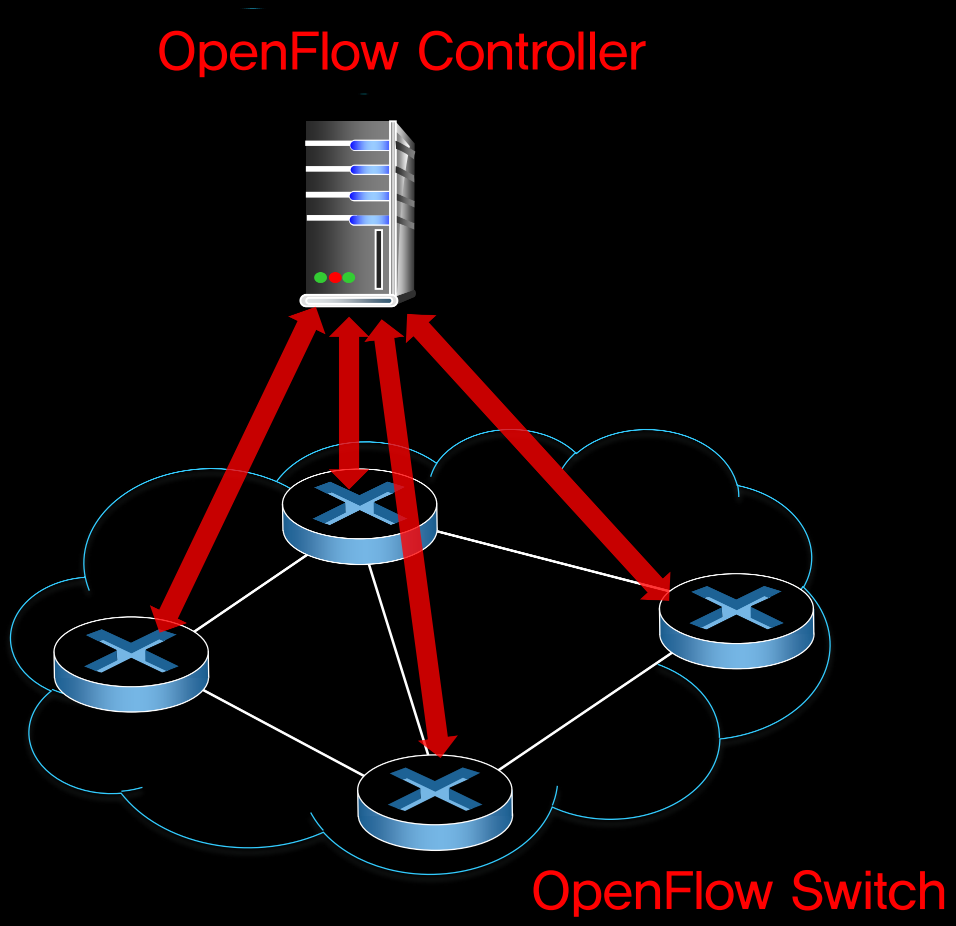
OpenFlow

- Open Networking Foundation (ONF)
  - OpenFlow spec. Ver. 1.5.1 in 2015
    - the components and the basic functions of the switch
    - based on an Ethernet switch, and firstly deployed at Stanford University
- OpenFlow switch consists of
  - flow table
  - secure channel
  - OpenFlow protocol



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiDj_Pf9rTcAhUlIpQKHVXKAZAQjRx6BAgBEAU&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FIdealized-OpenFlow-Switch-The-Flow-Table-is-controlled-by-a-remote-controller-via-the_fig1_220195143&psig=AOvVaw1k5V_b095lYnlfH80meDr-&ust=1532425166601373

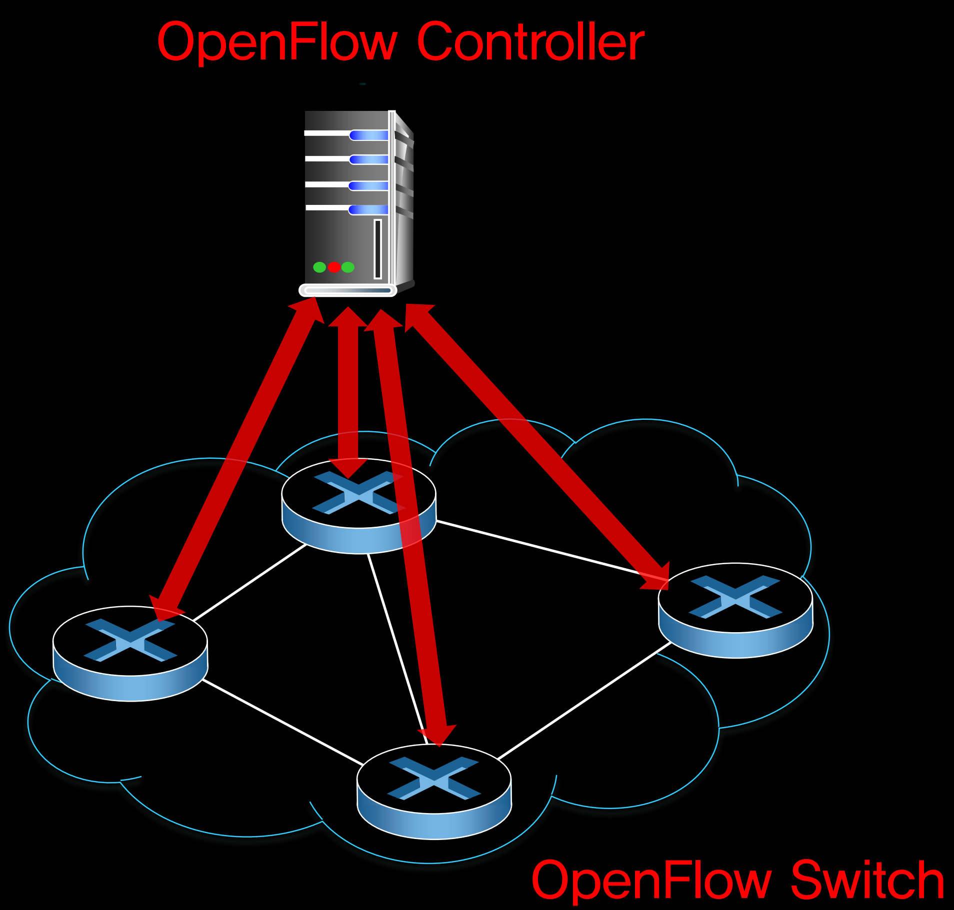
OpenFlow Protocol



- Operates between controller and switch
- TCP used to exchange messages
  - optional encryption
  - Transport Layer Security (TLS) or plain TCP
- Three classes of OpenFlow messages:
  - controller–to–switch (synchronous)
  - switch–to–controller (asynchronous)
  - symmetric (misc)

OpenFlow Protocol: Controller-to-Switch Messages

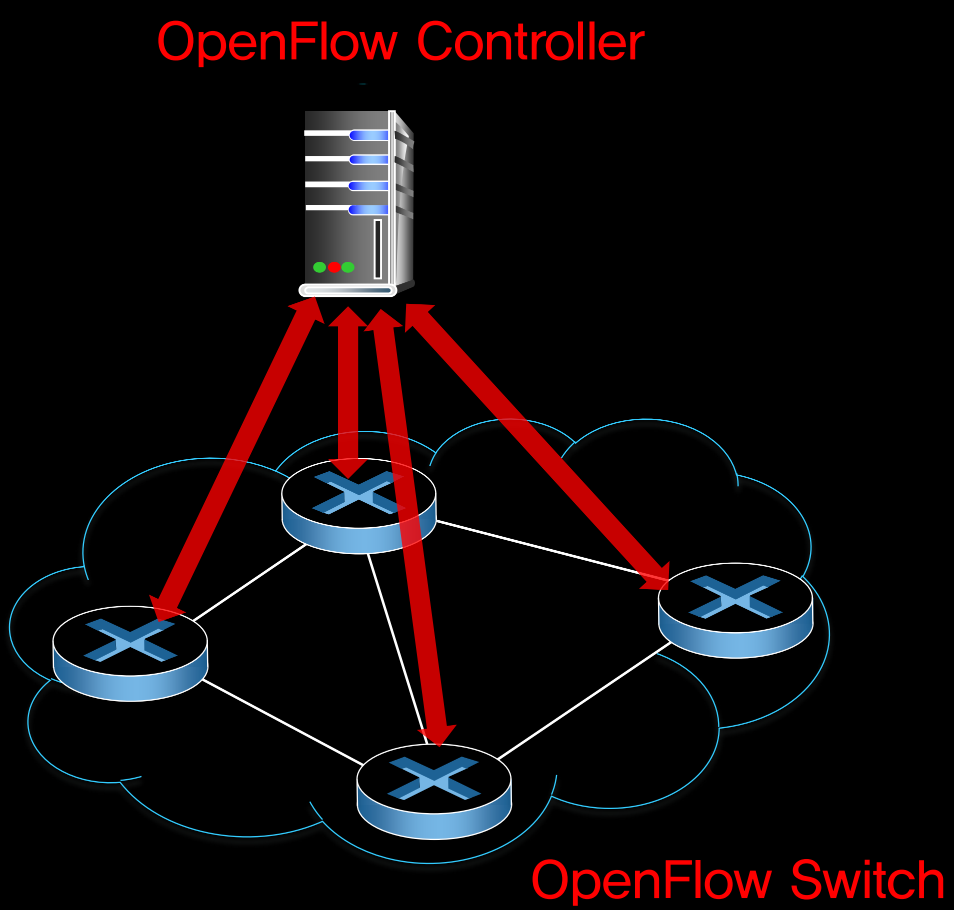
- Configuration: controller queries/sets switch configuration parameters
- Read-state: controller queries statistics and features from switch's flow table and ports
- Modify-state: add, delete, modify flow entries in the OpenFlow tables
- Send-Packet: controller can send a specific packet out of a specified switch port



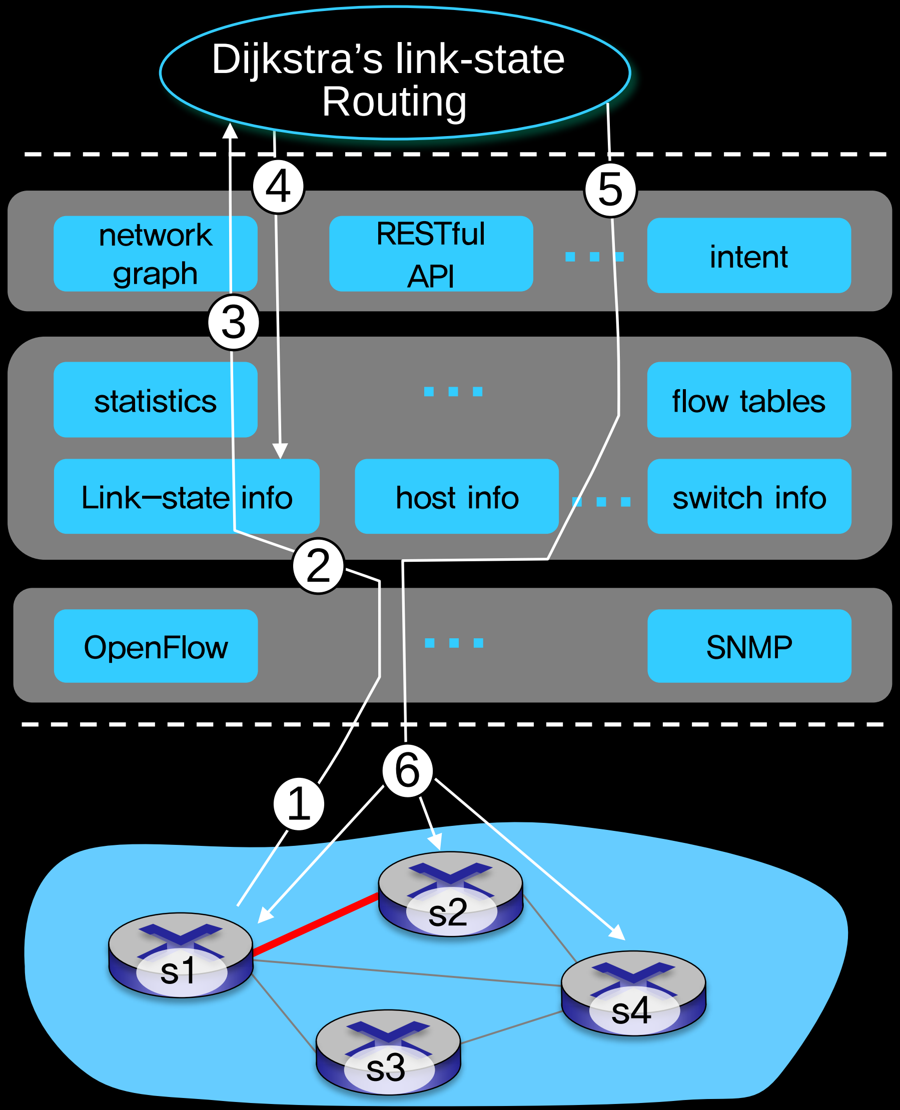
OpenFlow Protocol: Switch-to-Controller Messages

- Flow-removed: flow table entry deleted at switch
- Port status: inform controller of a change on a port
- Packet-in: transfer packet (and its control) to controller

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



Control/Data Plane Interaction thru OpenFlow

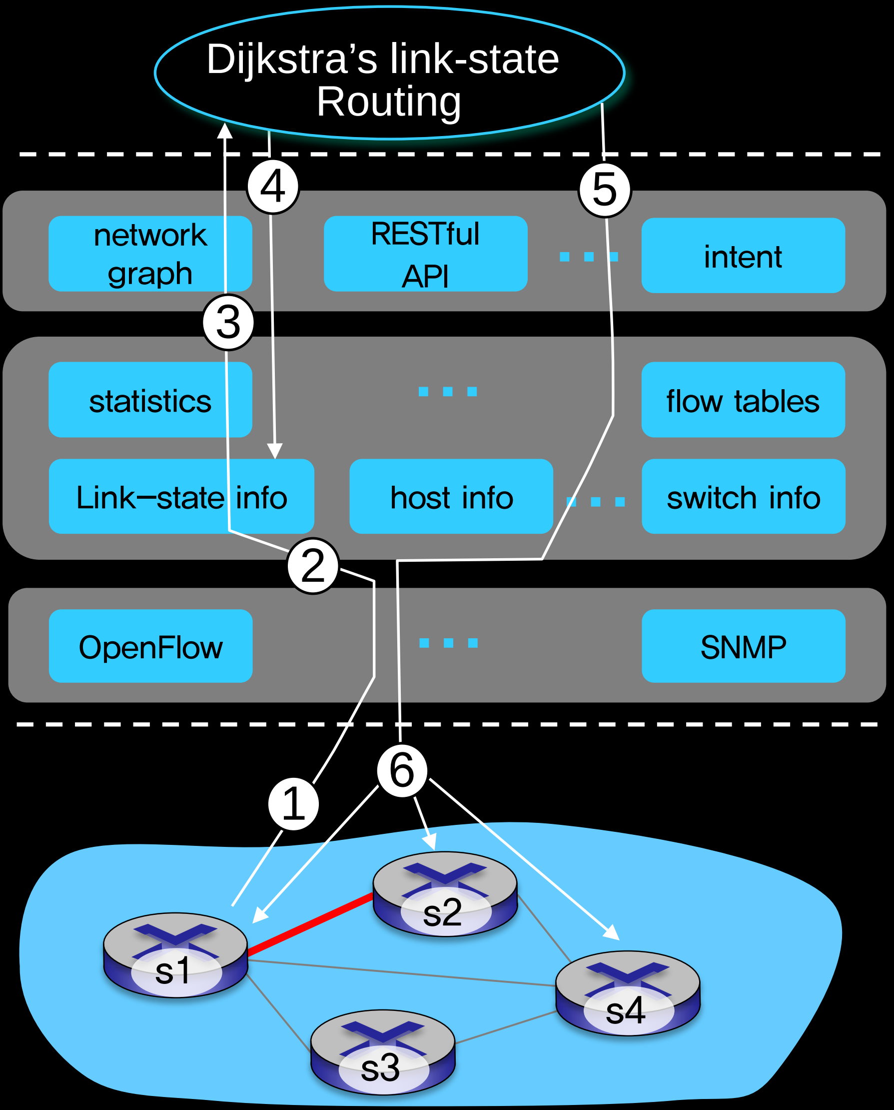


① S1, experiencing link failure using OpenFlow port status message to notify controller

② SDN controller receives OpenFlow message, updates link status info

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

Control/Data Plane Interaction thru OpenFlow (cont'd)



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

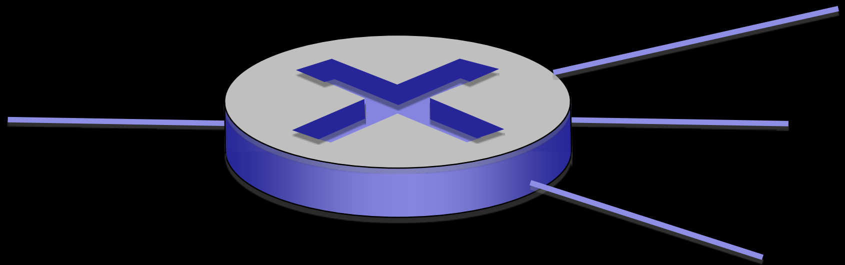
OpenFlow Flow Table Entries

Rule	Action	Statistics

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

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



Switch Port	MAC Src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	TCP Sport	TCP dport

Link layer (MAC Src – VLAN Pri) · Network layer (IP Src – IP Prot) · Transport layer (TCP Sport – TCP dport)

Flow Table Entry Examples (1/2)

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	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

Destination-based layer 2 (switch) forwarding:

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

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

Flow Table Entry Examples (2/2)

Firewall:

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

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

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

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

Match-plus-Action

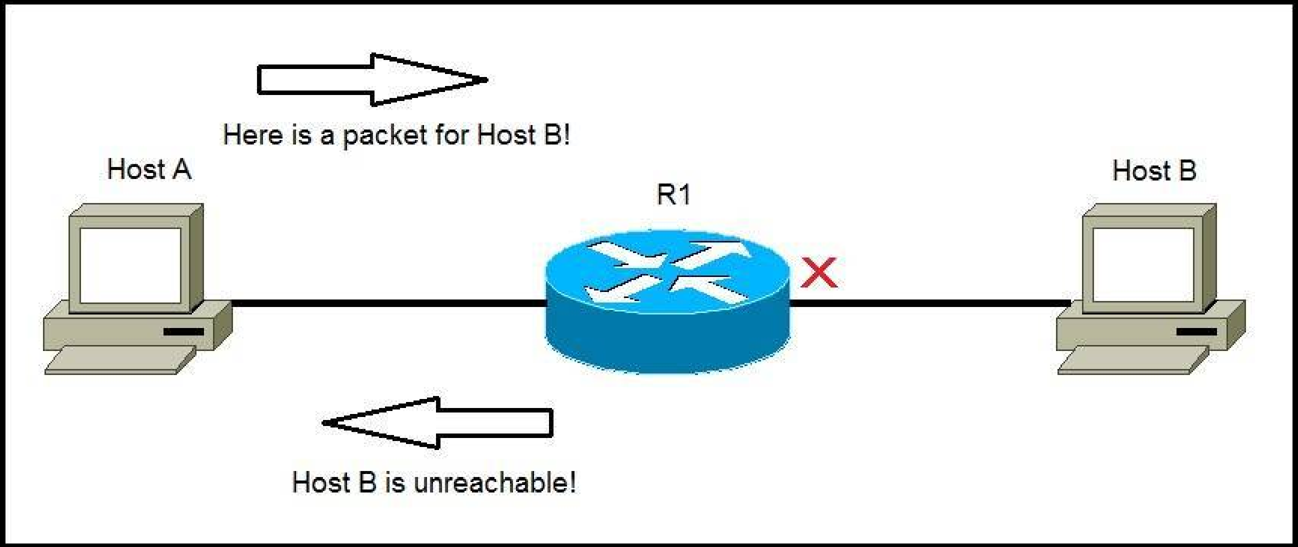
- Match and action are unified for different kinds of devices

Device	Match	Action
Router	longest matching with destination IP prefix	forward packets out of a link
Switch	destination MAC address	forward or flood
Firewall	IP addresses and port numbers	permit or deny
NAT	IP addresses and port numbers	rewrite address and port

08. Internet Control Message Prot.

ICMP (Internet Control Message Protocol)

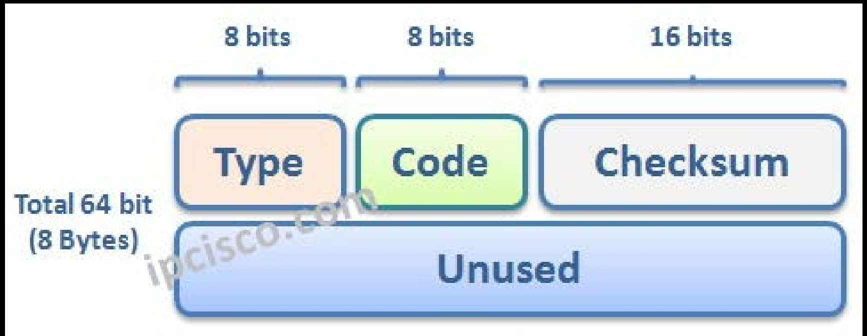
- Used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
- ICMP messages carried in IP datagrams



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj9qOa_zbTcAhXGFZQKHTUsDaoQjRx6BAgBEAU&url=http%3A%2F%2Fgeek-university.com%2Fccna%2Finternet-control-message-protocol-icmp%2F&psig=AOvVaw2lxzsHOVPsH2cgsKC2thm5&ust=1532413959697084

ICMP (Internet Control Message Protocol)

- ICMP message: type, code plus first 8 bytes of IP datagram causing error



출처 – https://www.google.co.kr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjjzcz5zbTcAhUFKpQKHWyWBGMQjRx6BAgBEAU&url=https%3A%2F%2Fipcisco.com%2Finternet-control-message-protocol%2F&psig=AOvVaw2lxzsHOVPsH2cgsKC2thm5&ust=1532413959697084

- ICMPv6: new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

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

Summary

Summary (1/2)

01

Introduction to Routing

- goal of routing: find "good" paths from src to dst
- routing algorithm classification

02

Link-State Routing

- routing based on complete network topology
- Dijkstra's algorithm: Computes least cost paths from one node ("source") to all other nodes

03

Distance Vector Routing

- knows path cost and next hop toward all other nodes
- Bellman-Ford equation

04

Intra-AS Routing: OSPF

- Autonomous System (AS): a collection of connected IP prefixes under the control of a single administrative entity

Summary (2/2)

05

Inter–AS Routing: BGP

- "glue that holds the Internet together"
- policy–based routing

06

Software Defined Networking

- separation control plane from data plane
- logically centralized remote controller

07

OpenFlow

- programmable Ethernet switch
- OpenFlow protocol for controller and switch communication

08

Internet Control Message Protocol

- used by hosts & routers for error reporting
- type and code