Network Layer Control Plane: BGP and SDN

November 12, 2024

Min Suk Kang
Associate Professor
School of Computing/Graduate School of Information Security

NetS&P
Research Lab @ KAIST

KAIST

Network layer: "control plane" roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF → open shortest path first
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol

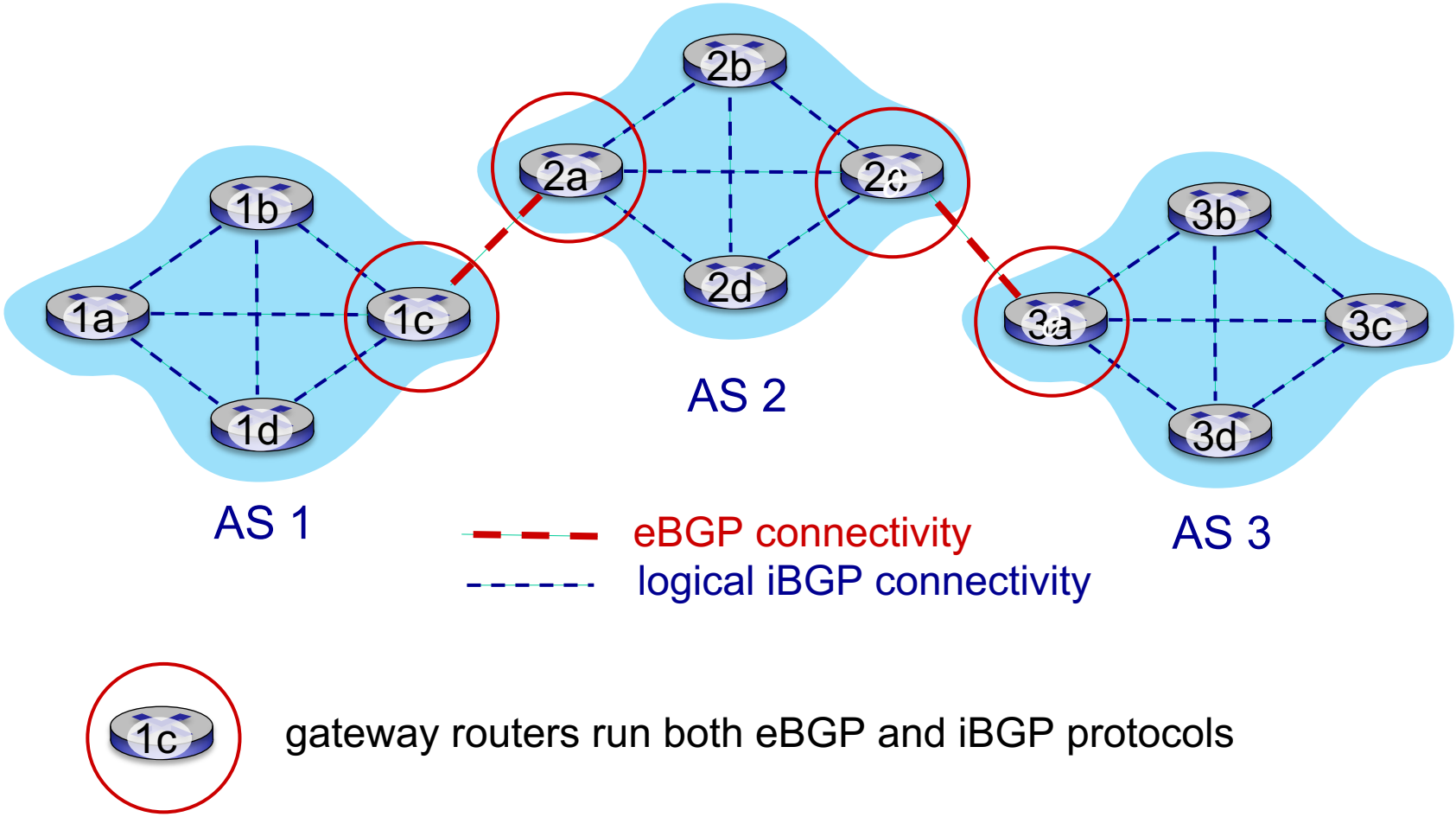
- network management, configuration
  - SNMP
  - NETCONF/YANG

Internet inter-AS routing: BGP

- BGP (Border Gateway Protocol): *the* de facto inter-domain routing protocol
  - "glue that holds the Internet together"
- allows subnet to advertise its existence, and the destinations it can reach, to rest of Internet: *"I am here, here is who I can reach, and how"*
- 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*

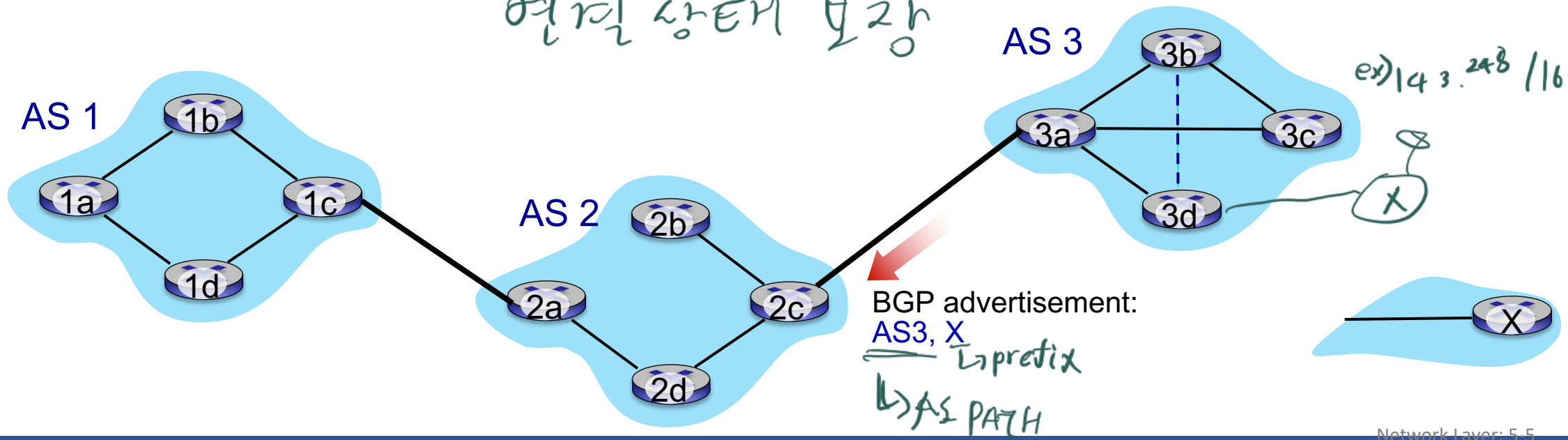
BGP는 일반적으로 eBGP를 의미한다.

eBGP, iBGP connections



BGP basics

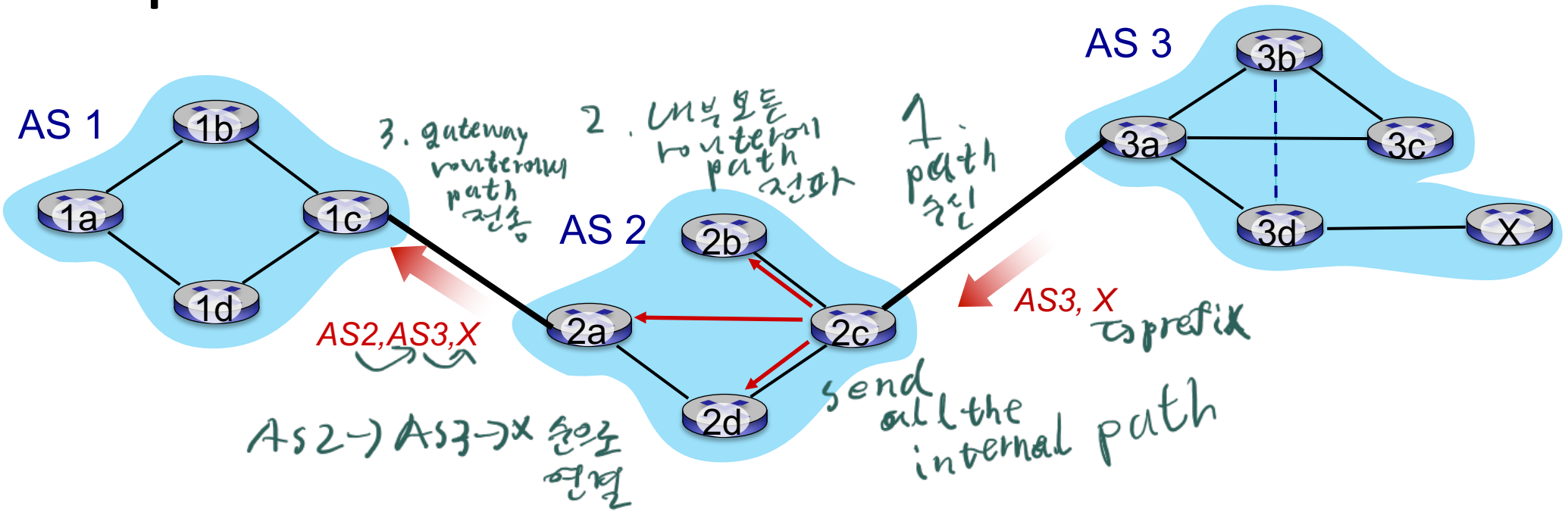
- BGP session: two BGP routers ("peers") exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a "path vector" protocol)
- when AS3 gateway 3a advertises path AS3,X to AS2 gateway 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



Path attributes and BGP routes

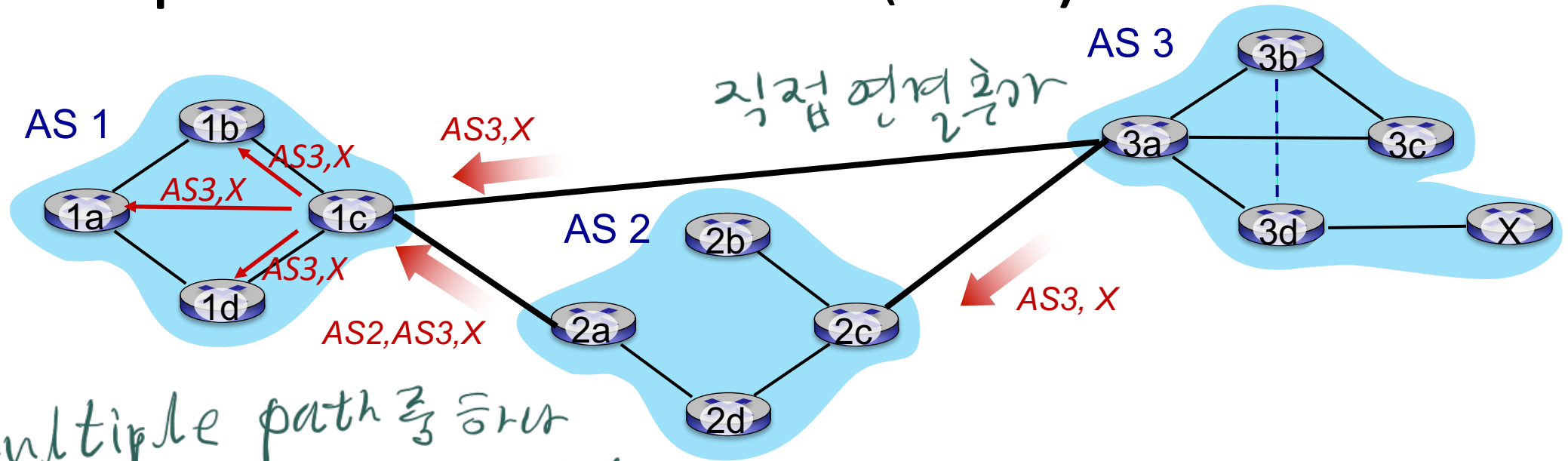
- BGP advertised route: prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
    - AS-PATH: list of ASes through which prefix advertisement has passed
    - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- policy-based routing:
  - gateway receiving 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 other neighboring ASes

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

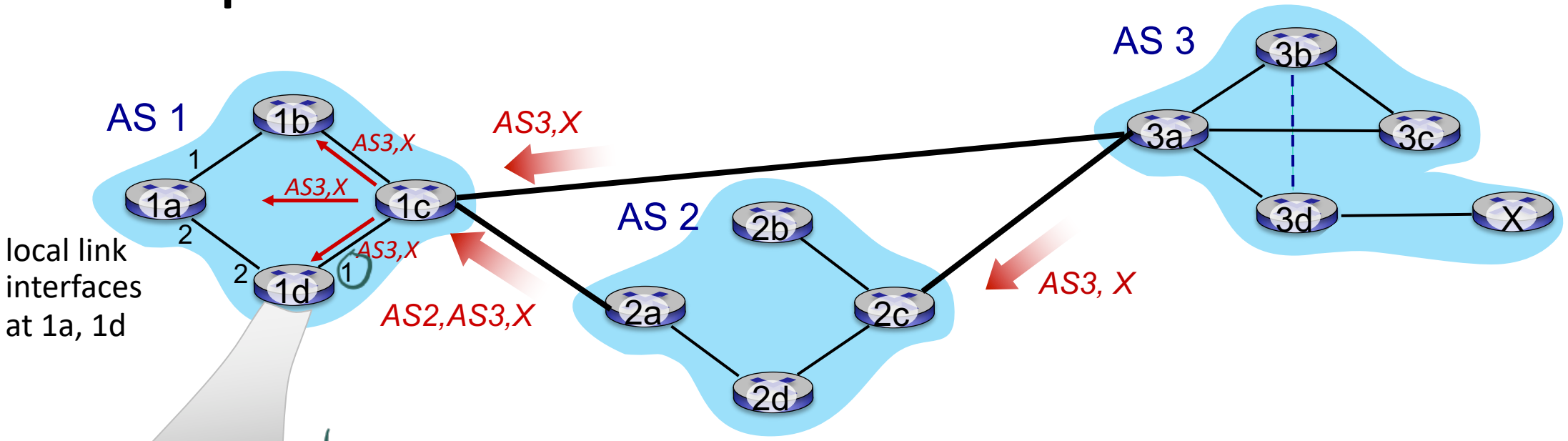
BGP path advertisement (more)

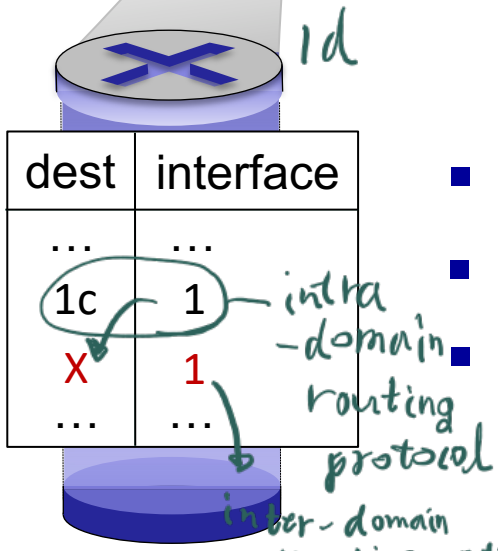


gateway router may learn about multiple paths to destination:

- 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

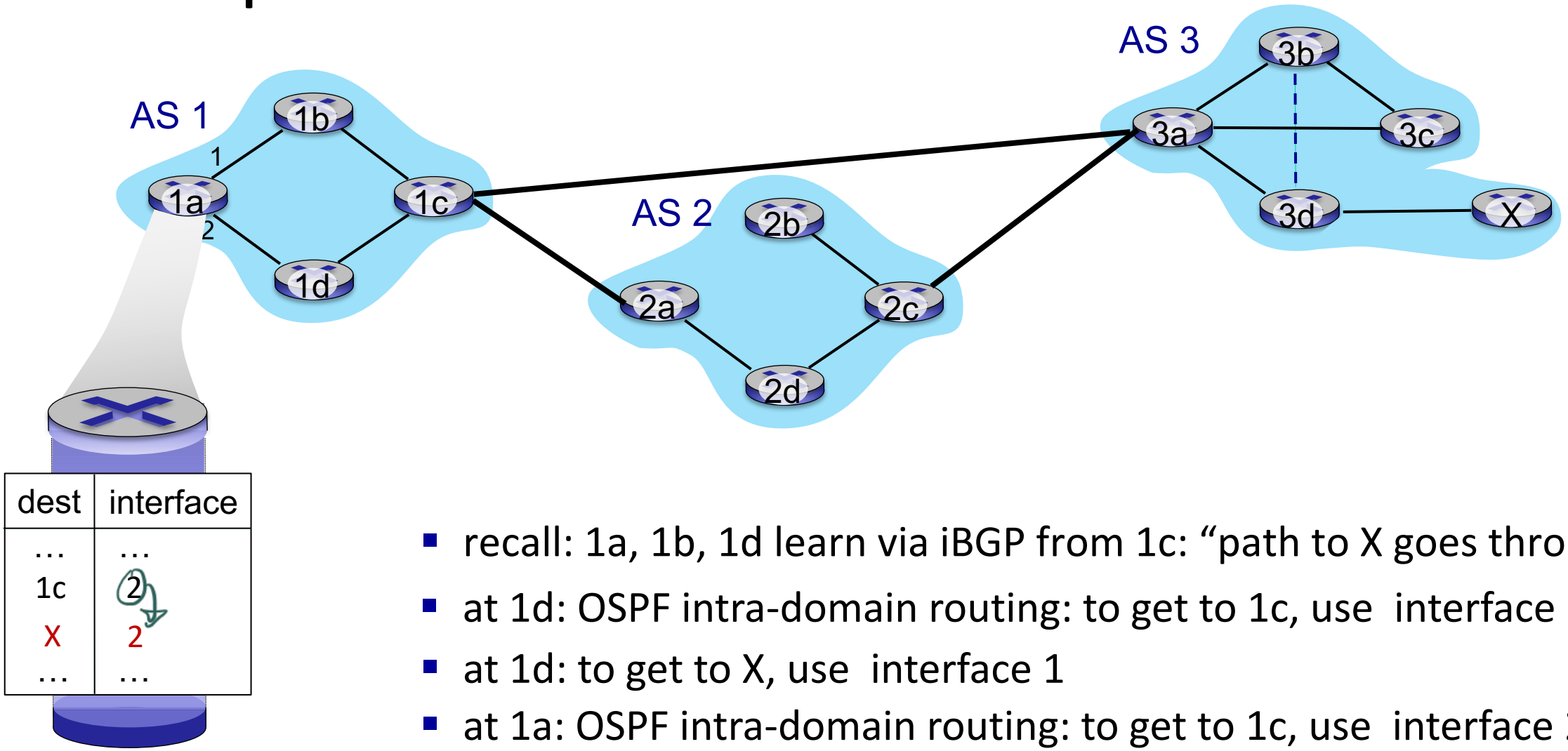
BGP path advertisement





- recall: 1a, 1b, 1d learn via iBGP from 1c: "path to X goes through 1c"
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1

BGP path advertisement



- recall: 1a, 1b, 1d learn via iBGP from 1c: “path to X goes through 1c”
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1
- at 1a: OSPF intra-domain routing: to get to 1c, use interface 2
- at 1a: to get to X, use interface 2

Why different Intra-, Inter-AS routing ?

policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its network
- intra-AS: single admin, so policy less of an issue

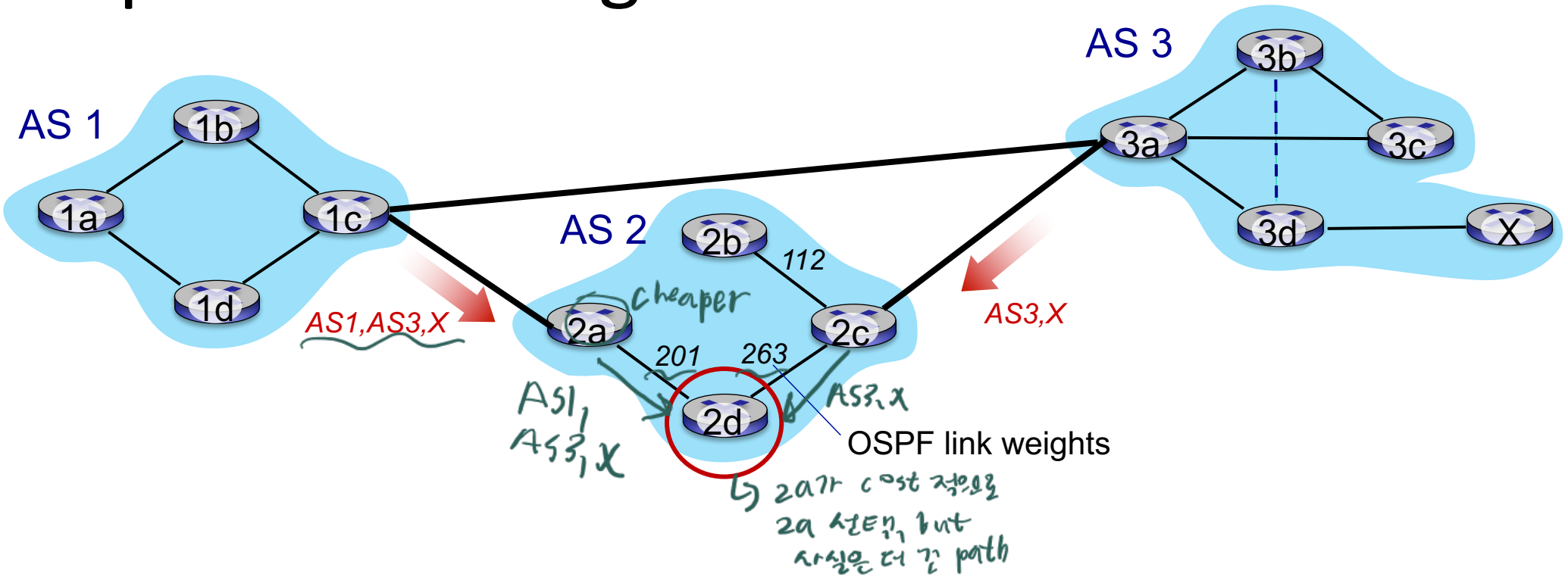
scale:

- hierarchical routing saves table size, reduced update traffic

performance:

- intra-AS: can focus on performance
- inter-AS: policy dominates over performance

Hot potato routing

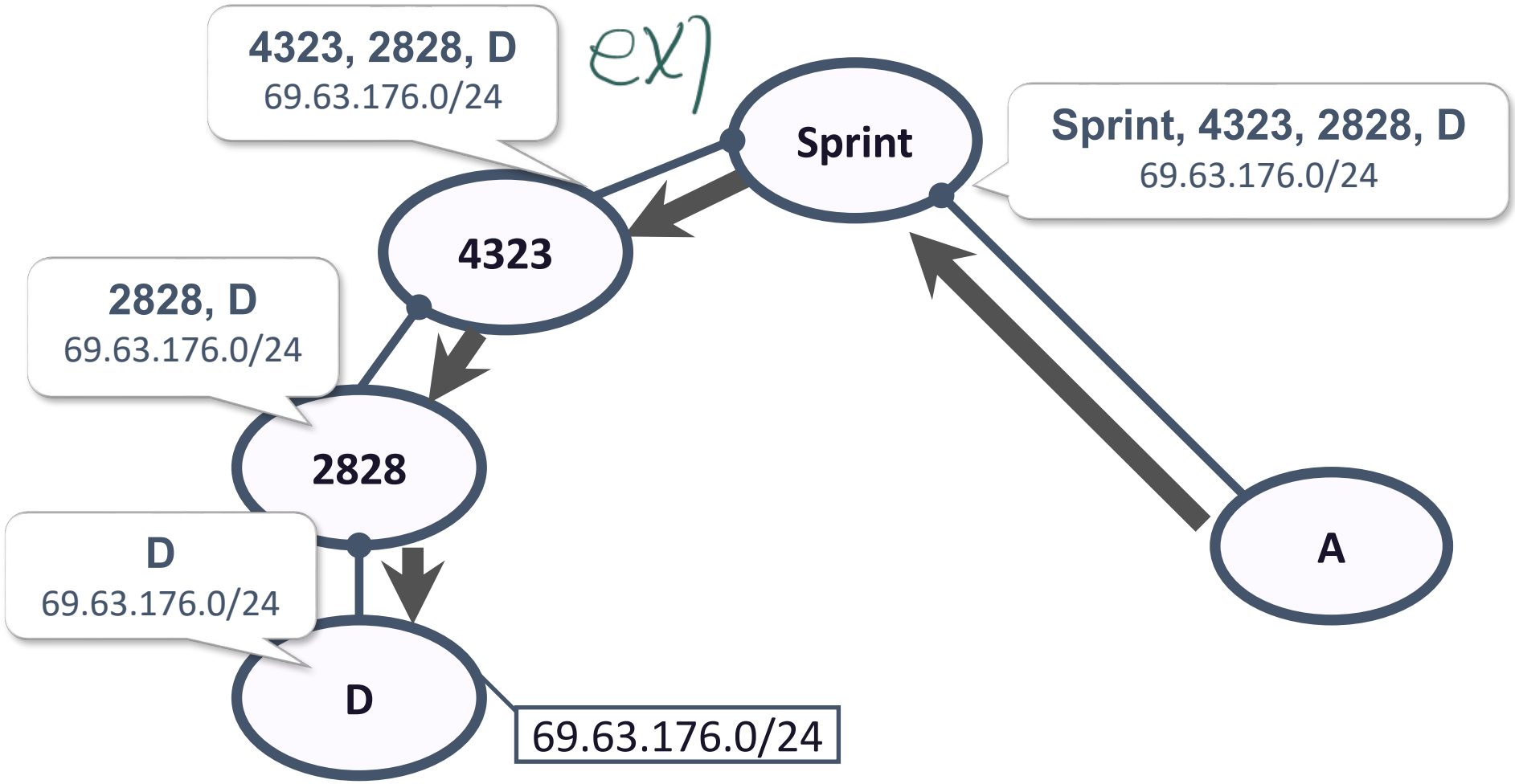


- 2d learns (via iBGP) it can route to X via 2a or 2c
- hot potato routing: choose local gateway that has least *intra-domain* cost (e.g., 2d chooses 2a, even though more AS hops to *X*): don't worry about inter-domain cost!

Border Gateway Protocol

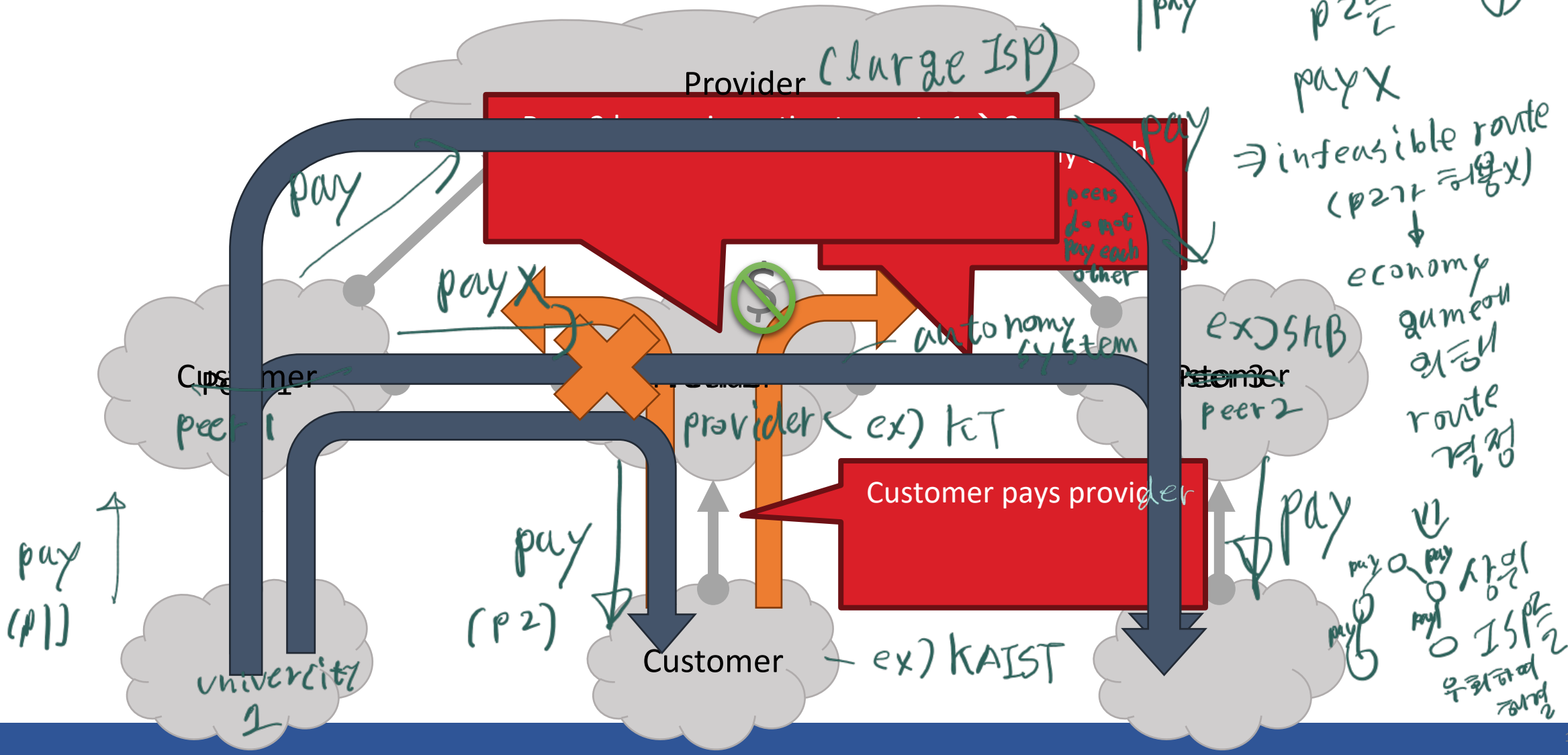
- Border Gateway Protocol (BGP)
  - De facto inter-domain protocol of the Internet
  - Uses a **path vector** routing
    - =/= link state routing
      - flood the routing table to entire network
    - =/= distance vector routing
      - advertise the cost but not actual paths
  - Policy based routing protocol
- Relatively simple protocol, but…
  - Complex, manual configuration
  - Entire world sees advertisements
    - Errors can screw up traffic globally
  - Policies driven by economics
    - Not by performance (e.g. shortest paths)

BGP

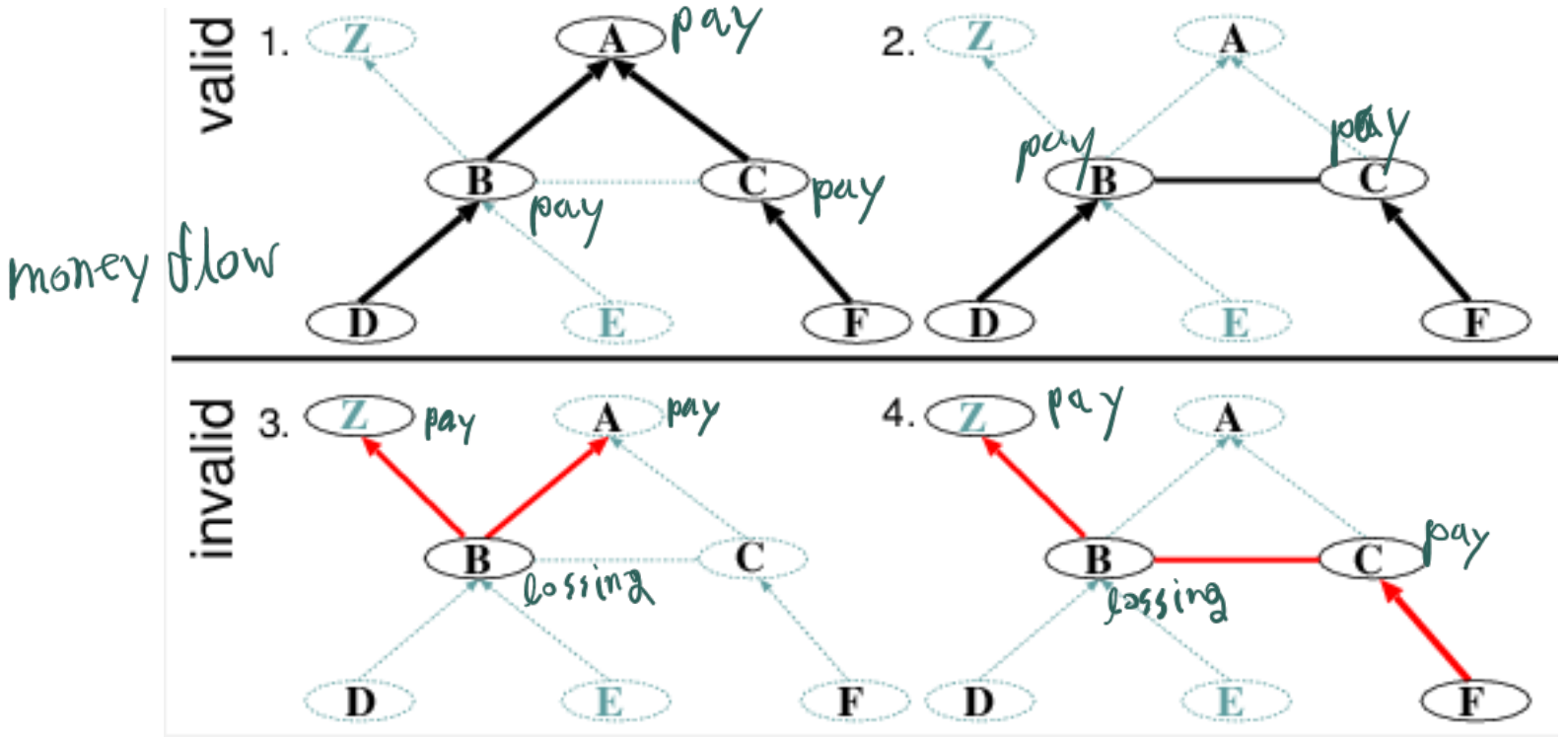

4323, 2828, D
69.63.176.0/24
ex)
Sprint
Sprint, 4323, 2828, D
69.63.176.0/24
4323
2828, D
69.63.176.0/24
2828
D
69.63.176.0/24
A
D
69.63.176.0/24

15

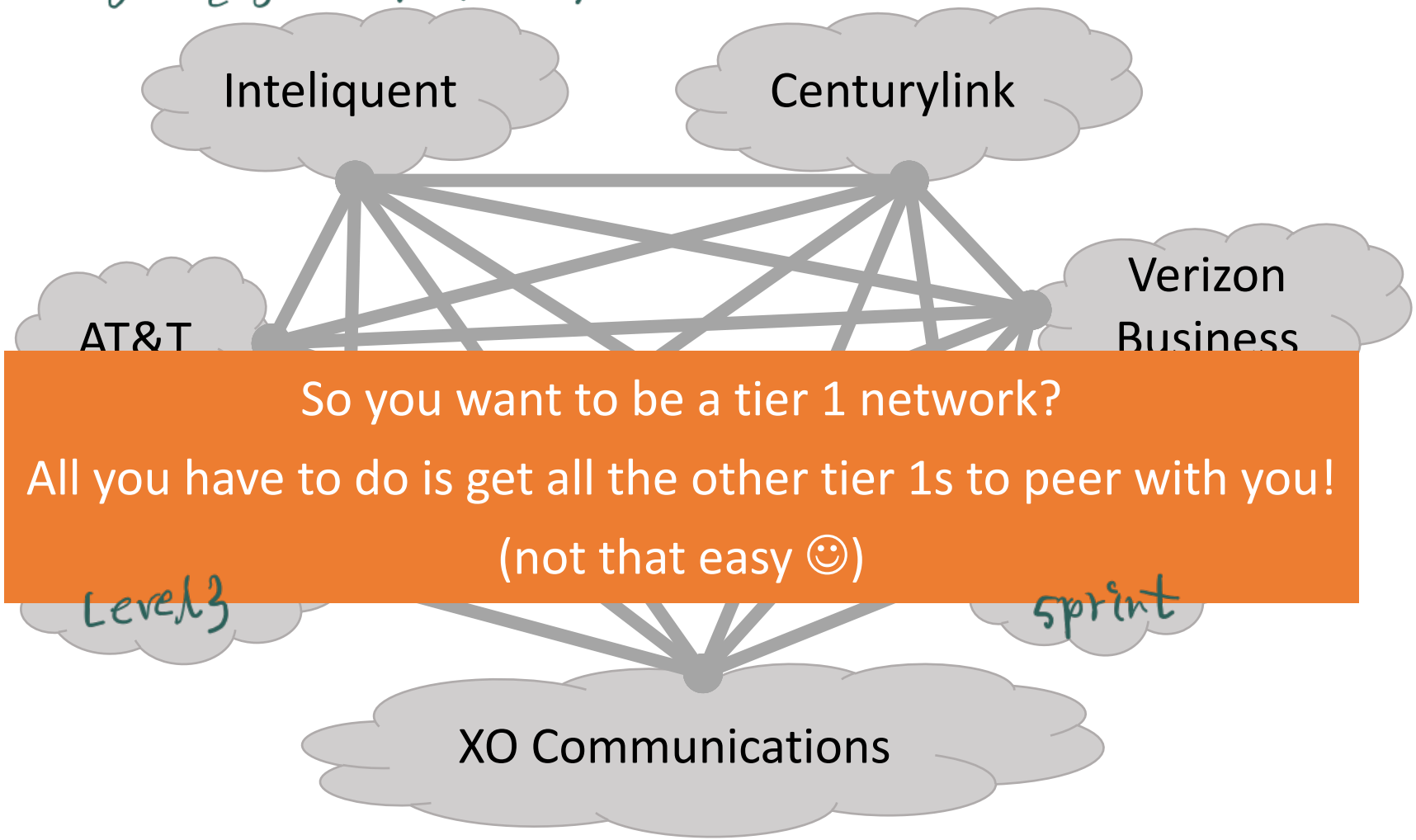
BGP's Business Relationships



BGP paths: valids and invalids



Tier-1 ISP Peering

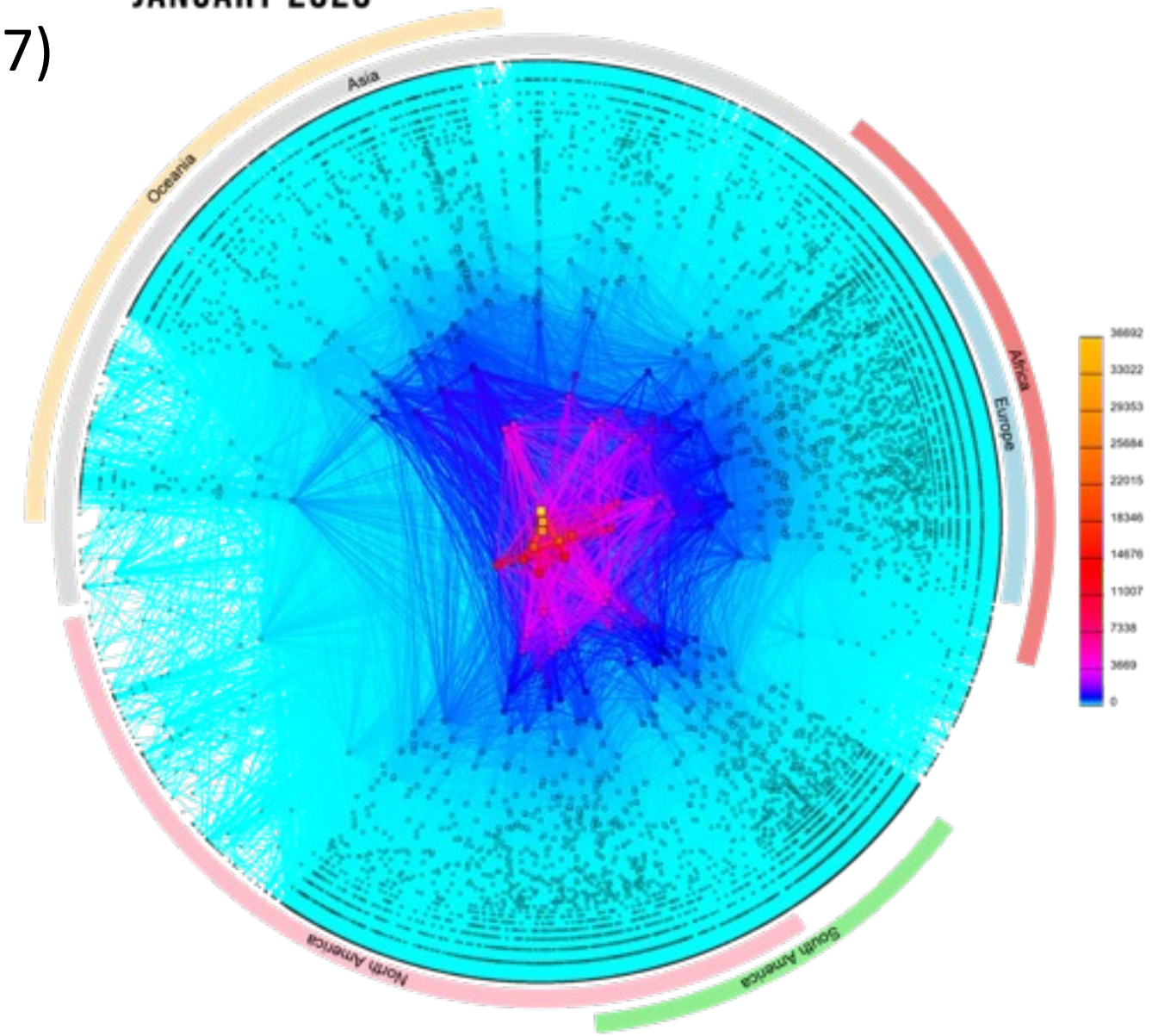


So you want to be a tier 1 network?

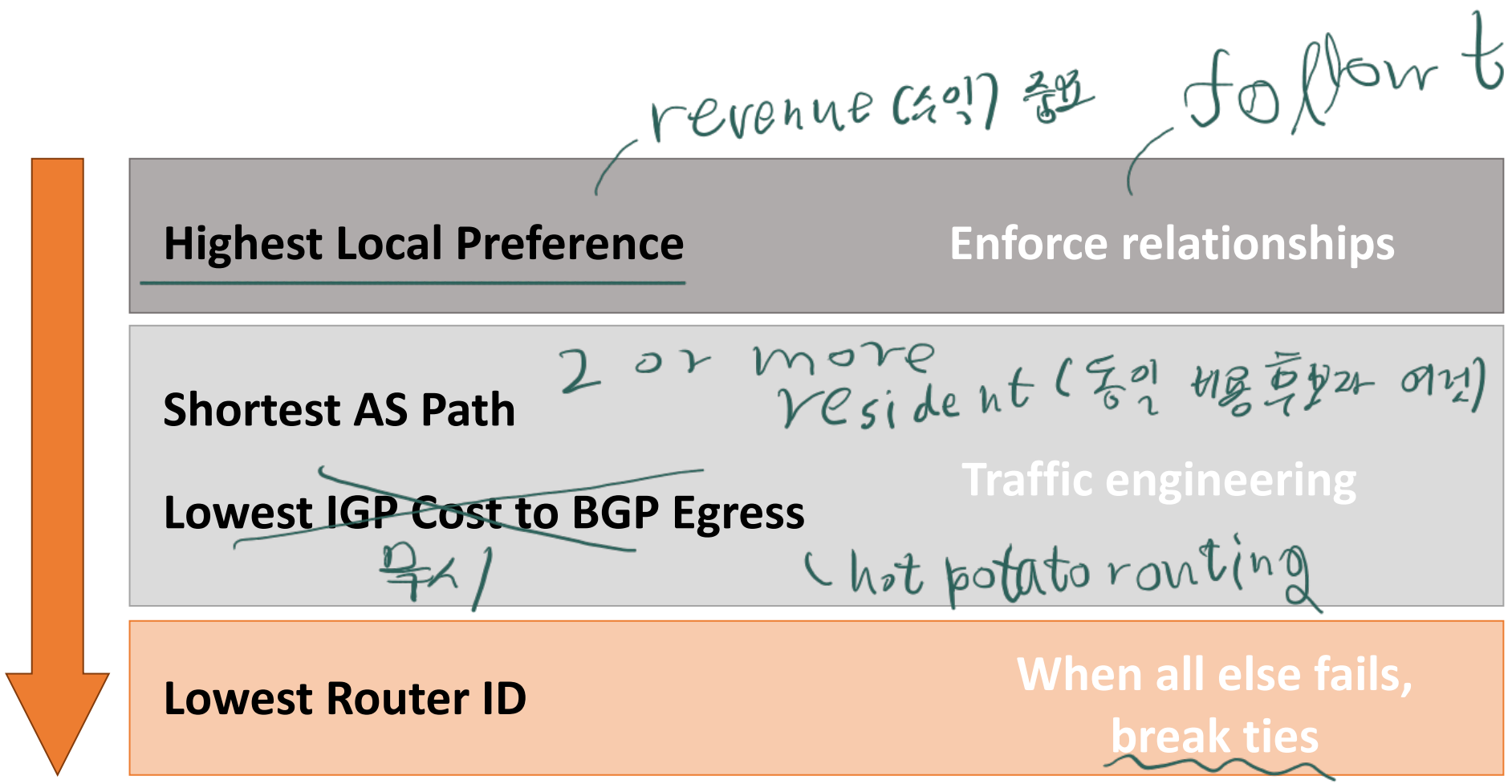
All you have to do is get all the other tier 1s to peer with you!

(not that easy ☺)

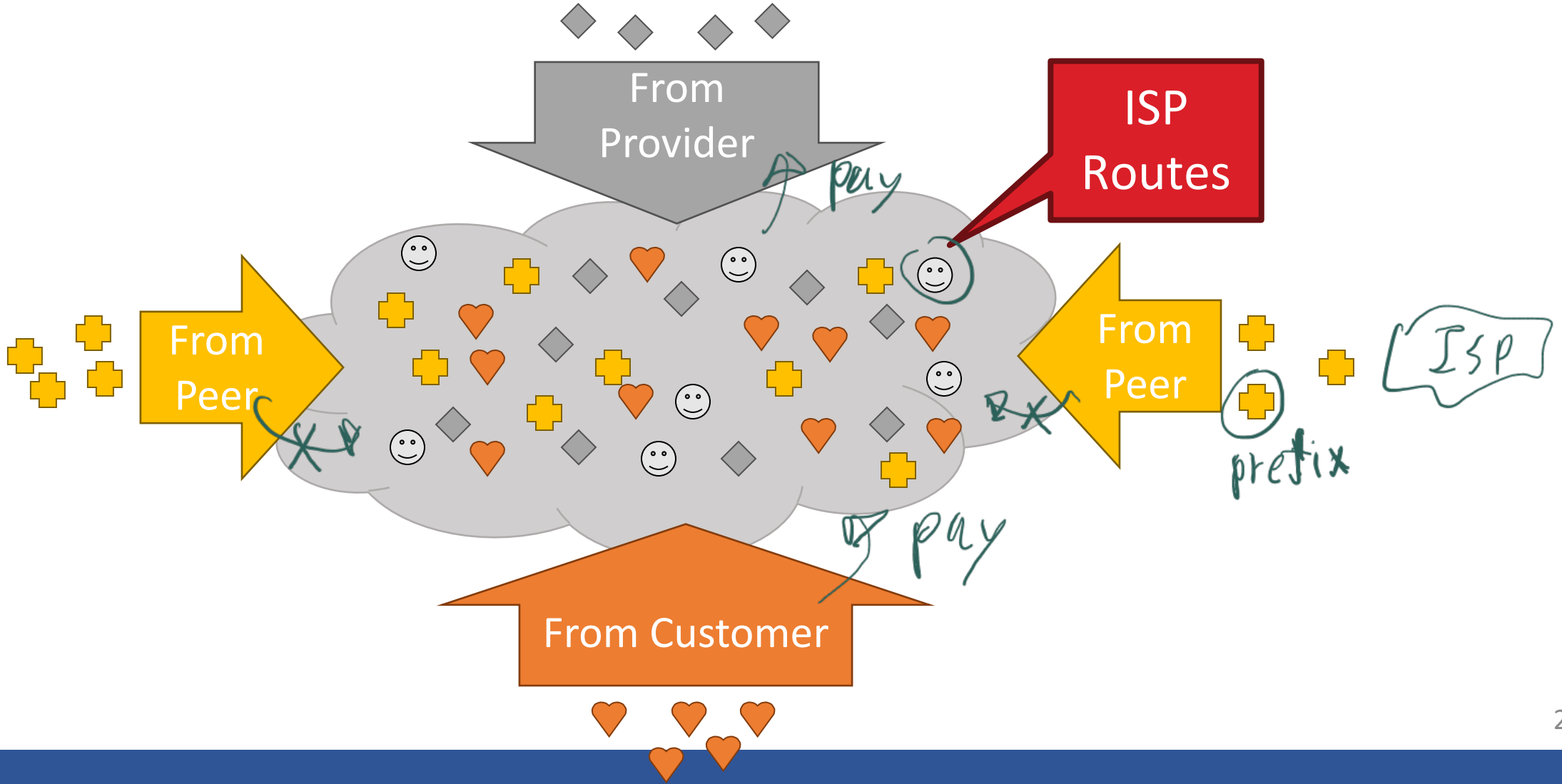
AS level topology by CAIDA (2017)



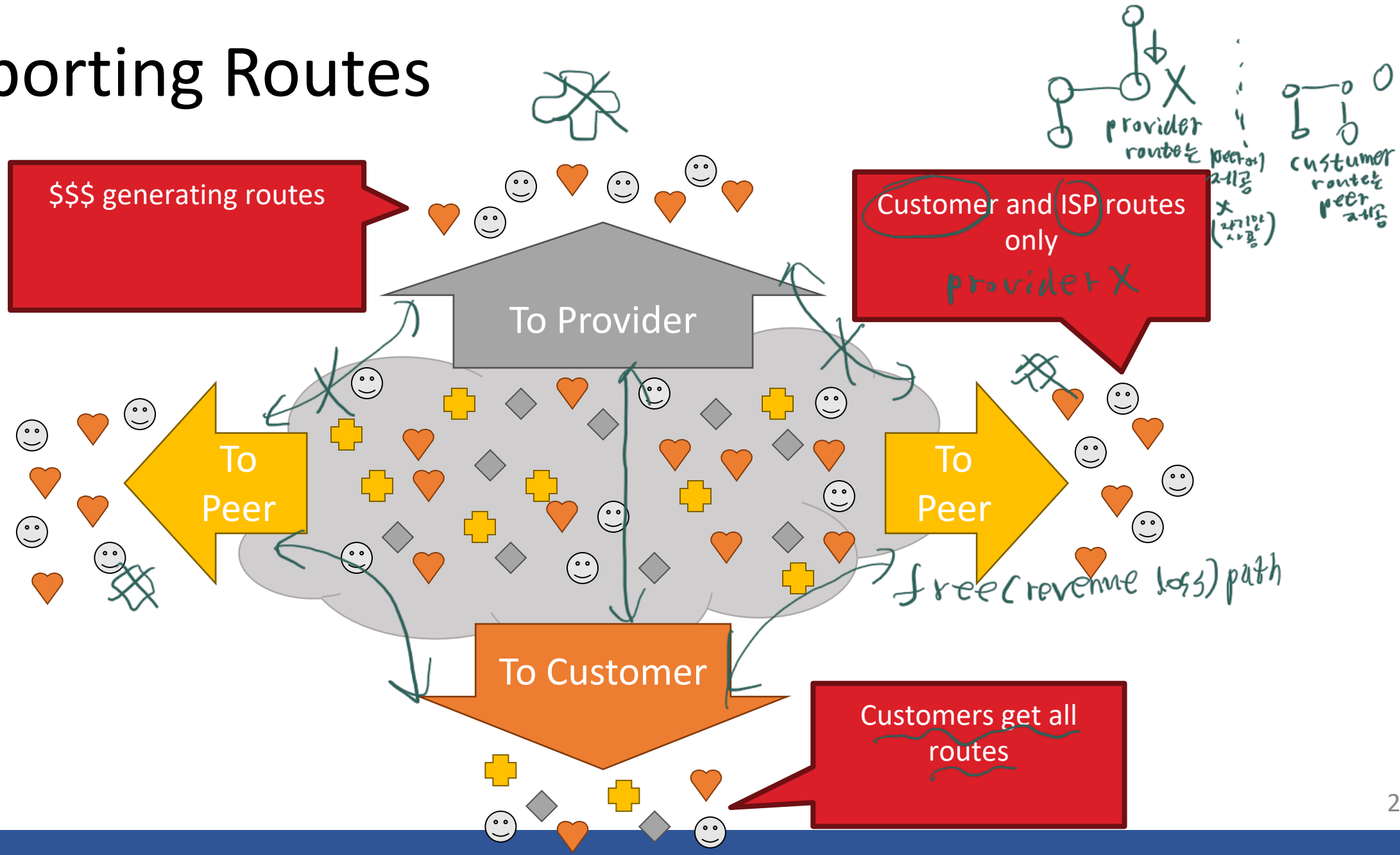
Route Selection Summary



Importing Routes



Exporting Routes



BGP: Robustness?

simple but errorable

- Robustness: Is BGP robust enough to prevent/avoid mistakes?

Business All Industry Technology Tr

Ministry finds KT work incorrect command as outage

Ministry to review introducing measures fo

By Kim Da-sol

Published : Oct 29, 2021 - 17:22 Updated : Oct 29, 2021 - 17:23

An incorrect command input -- in fact, forge routing setting command to the router dev outage in less than a minute.

Authorities found that the network path inf sent to border gateway protocol (BGP), was which is used for internal path setting at KT

The ministry's investigation team also foun features in place to prevent the spread of w

SOC PRIME

What Is BGP and How Its Failure Took Facebook Down?

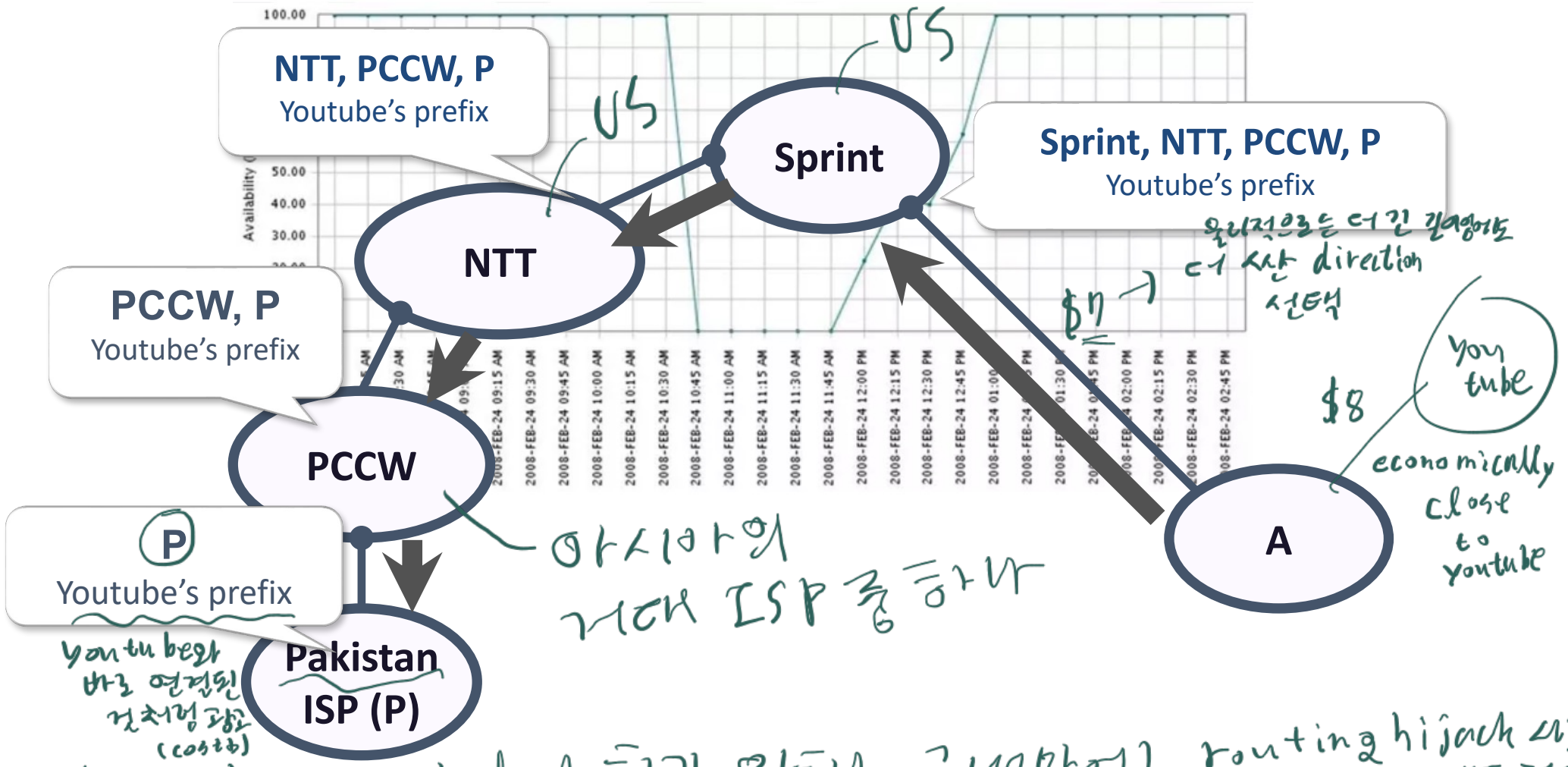
WRITTEN BY
Eugene Tkachenko

October 08, 2021 · 4 min read

On October 4, 2021, Facebook – and all the major services Facebook

ex)

Pakistan Youtube Outage Event (2008)



Interception in real world

Cost가 싼 것으로 속여 오는 신호가 자신을 거치도록 변경

route hijacking attack

S2W

Feb 17 · 18 min read · Listen

Post Mortem of KlaySwap Incident through BGP Hijacking | EN

Author: S2W TALON with eyez (Lead by Sojun Ryu)

> *Last Modified : 2022.02.16.*

TIMELINE

Preliminary Preparation

The first transaction in the attacker's Account
2021.06.29 08:31

Test Factory Contract created
2022.01.05 08:49

Malicious API server domain created
2022.01.06 07:58

Certificate of malicious API server domain enabled
2022.01.06 08:02

Factory Contract for theft created
2022.01.07 02:49

BGP Hijacking

BGP Hijacking started (211.249.216.0/21)
2022.02.03 10:04

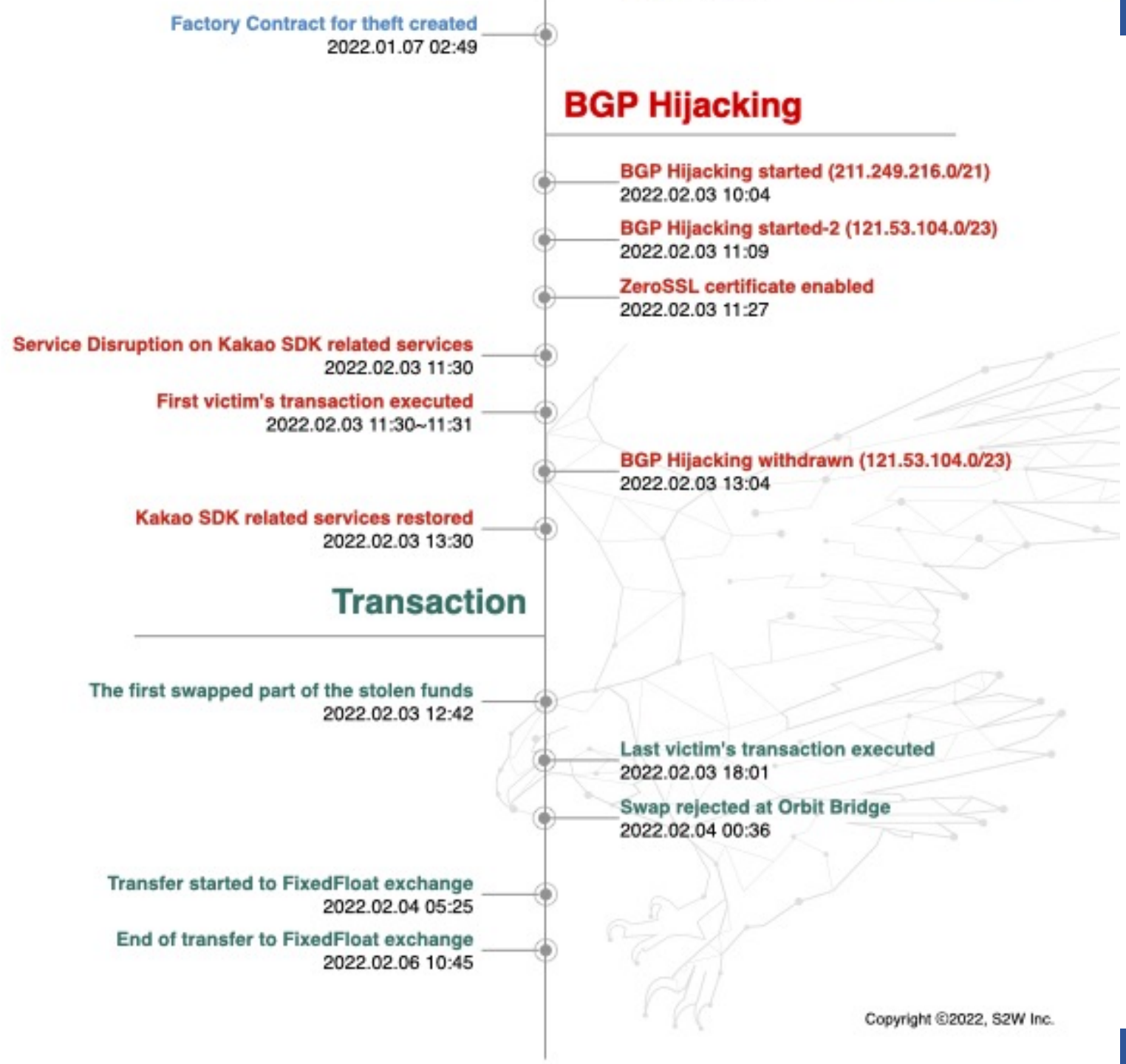
BGP Hijacking started-2 (121.53.104.0/23)
2022.02.03 11:09

ZeroSSL certificate enabled
2022.02.03 11:27

Service Disruption on Kakao SDK related services
2022.02.03 11:30

First victim's transaction executed
2022.02.03 11:30~11:31

BGP Hijacking withdrawn (121.53.104.0/23)
2022.02.03 13:04


Factory Contract for theft created
2022.01.07 02:49
BGP Hijacking
BGP Hijacking started (211.249.216.0/21)
2022.02.03 10:04
BGP Hijacking started-2 (121.53.104.0/23)
2022.02.03 11:09
ZeroSSL certificate enabled
2022.02.03 11:27
Service Disruption on Kakao SDK related services
2022.02.03 11:30
First victim's transaction executed
2022.02.03 11:30~11:31
BGP Hijacking withdrawn (121.53.104.0/23)
2022.02.03 13:04
Kakao SDK related services restored
2022.02.03 13:30
Transaction
The first swapped part of the stolen funds
2022.02.03 12:42
Last victim's transaction executed
2022.02.03 18:01
Swap rejected at Orbit Bridge
2022.02.04 00:36
Transfer started to FixedFloat exchange
2022.02.04 05:25
End of transfer to FixedFloat exchange
2022.02.06 10:45
Copyright ©2022, S2W Inc.

Search

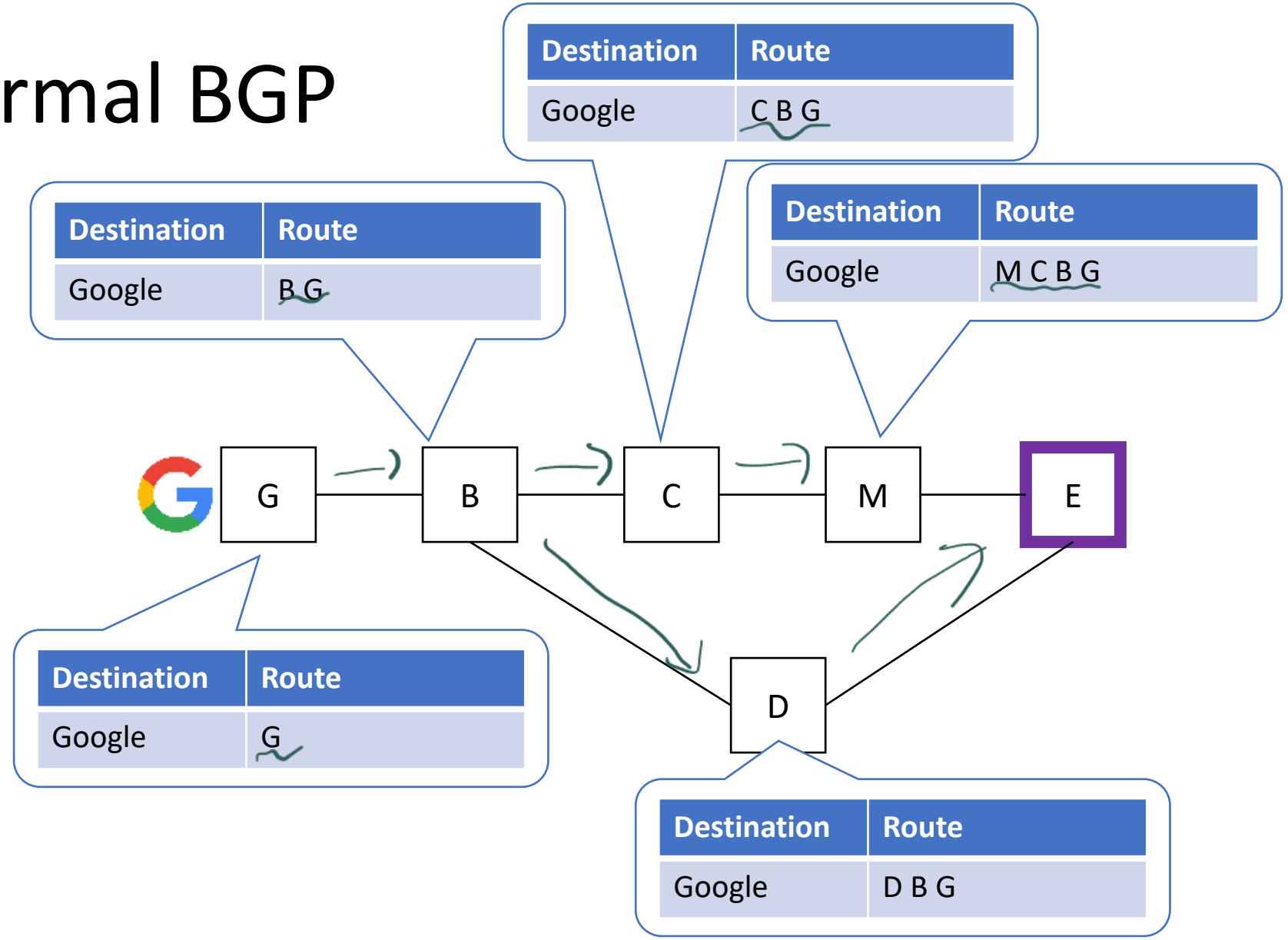
Write

Truth Behind the Celer Network cBridge cross-chain bridge incident: BGP hijacking

SlowMist ·
Published in Coinmonks · 8 min read · Aug 20, 2022

59

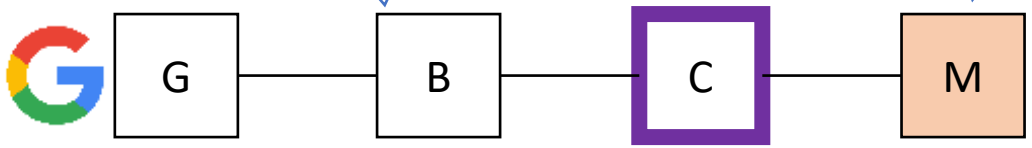
Normal BGP


Destination
Route
Google
C B G
Destination
Route
Google
B G
Destination
Route
Google
M C B G
G
B
C
M
E
D
Destination
Route
Google
G
Destination
Route
Google
D B G

1) Prefix hijacking → 자신이 prefix인 것으로 속이기

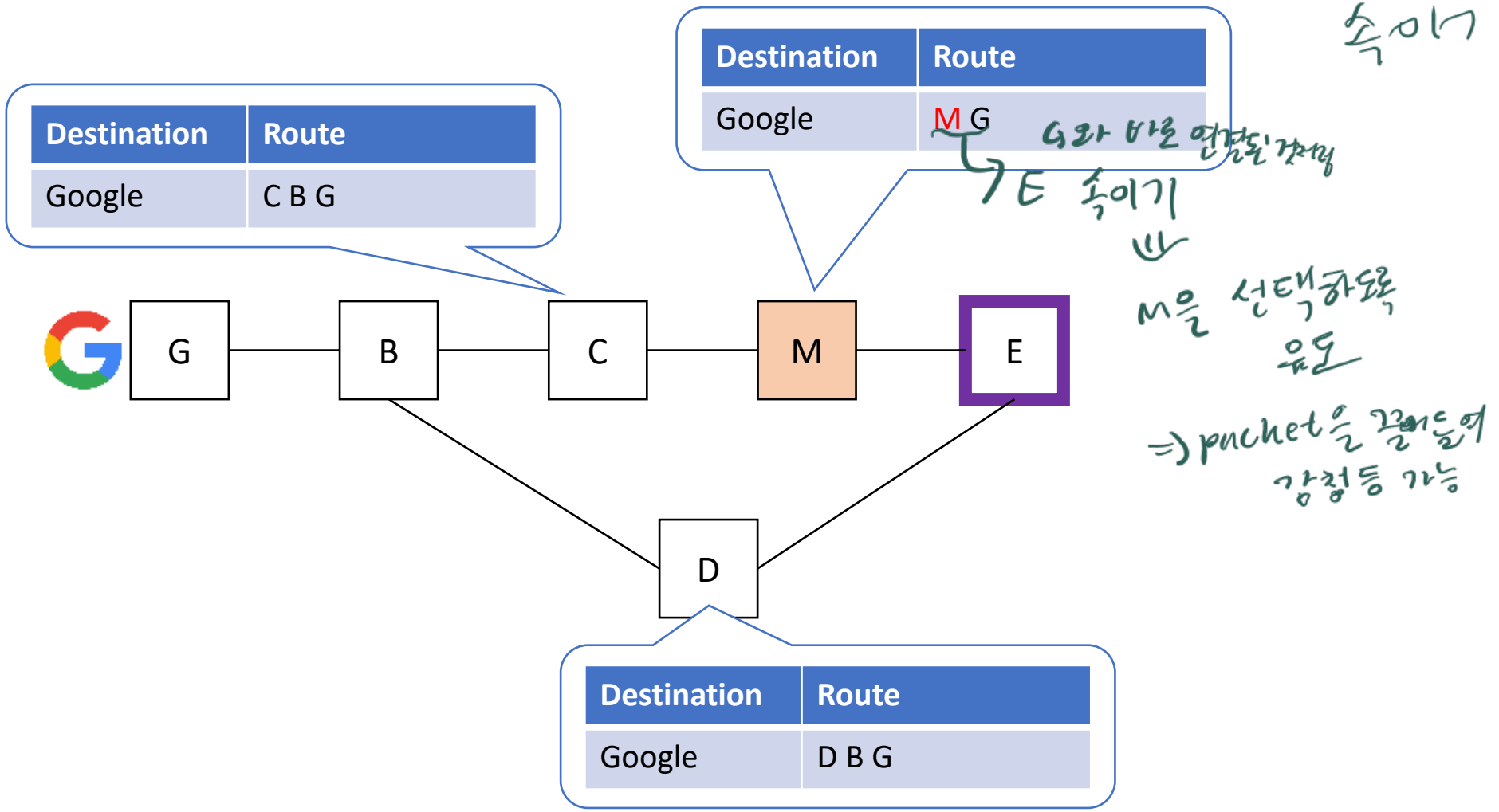
Destination	Route
Google	B G

Destination	Route
Google	M



2) One-hop prefix hijacking

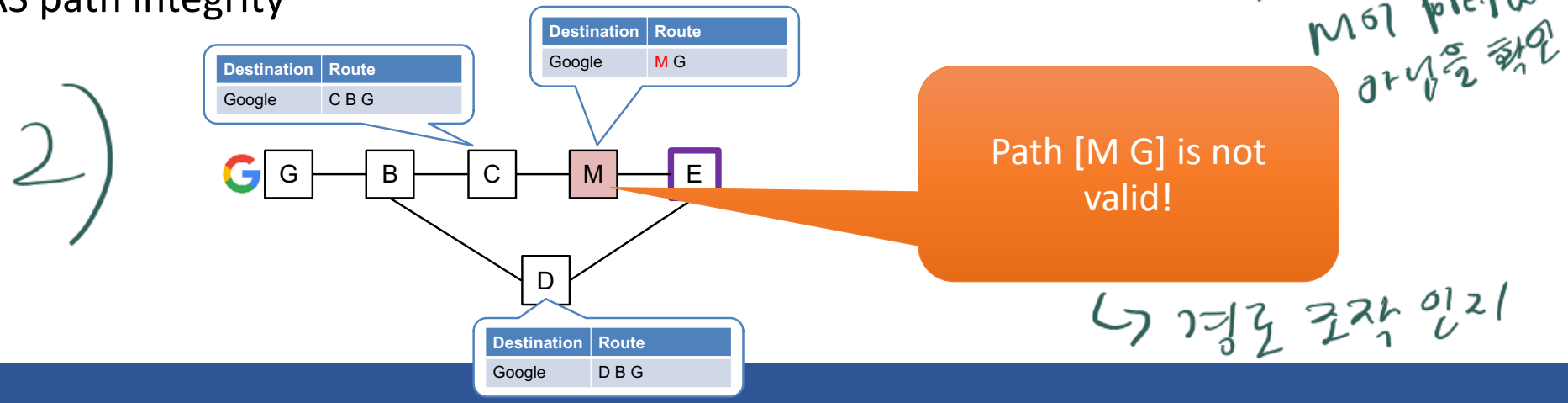
→ 자신이 prefix와 직접 연결된 것으로 속이기



RPKI and BGPSEC

- Resource Public Key Infrastructure (RPKI)
  - Ensures particular AS owns particular address blocks

- BGPSEC
  - Provides AS path integrity



Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane New! (강의 끝날)
- Internet Control Message Protocol

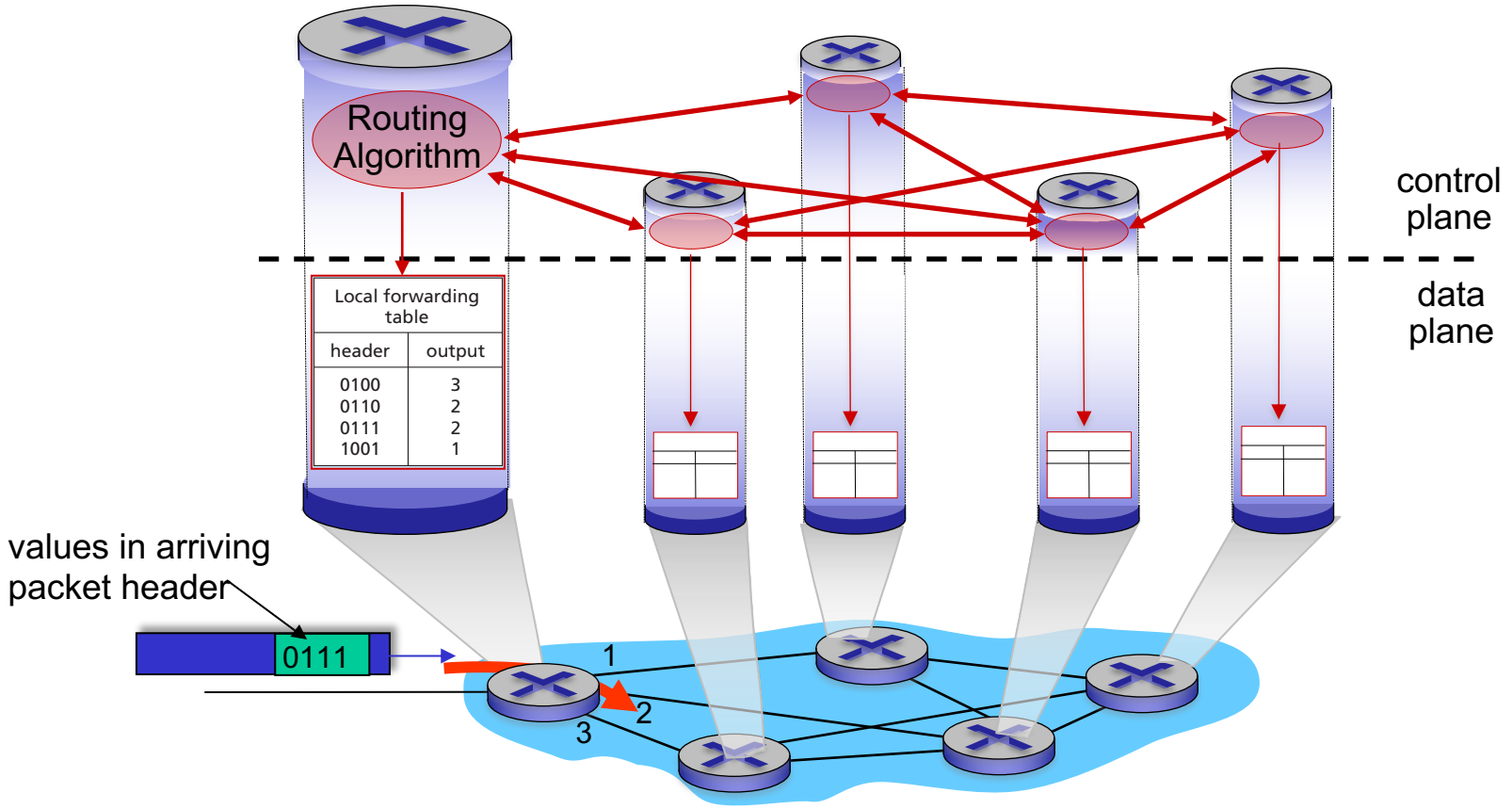
- network management, configuration
  - SNMP
  - NETCONF/YANG

Software defined networking (SDN)

- Internet network layer: historically implemented via distributed, per-router control 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

Per-router control plane

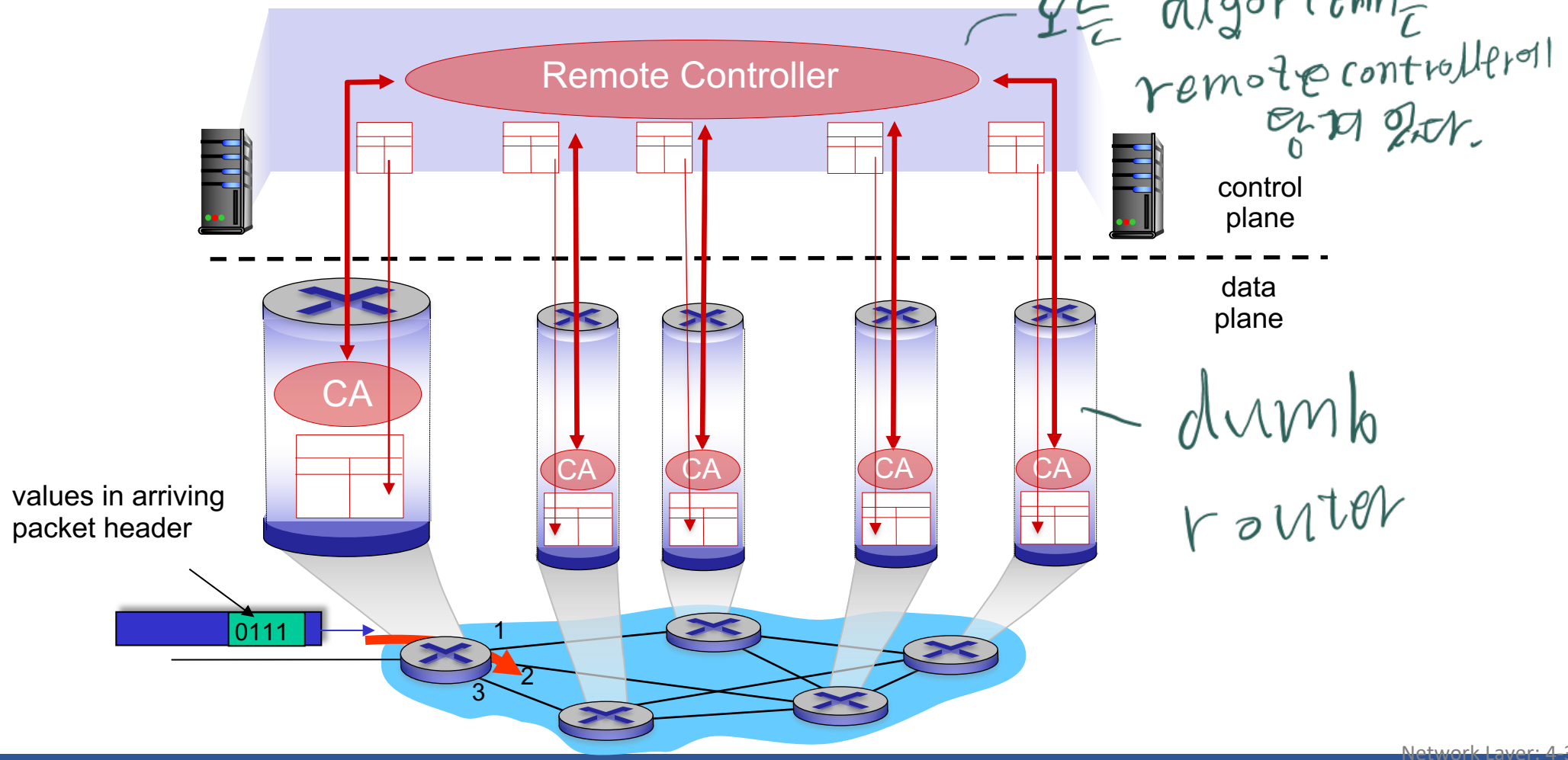
Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



Software-Defined Networking (SDN) control plane

↳ lab 3 : own SDN

Remote controller computes, installs forwarding tables in routers

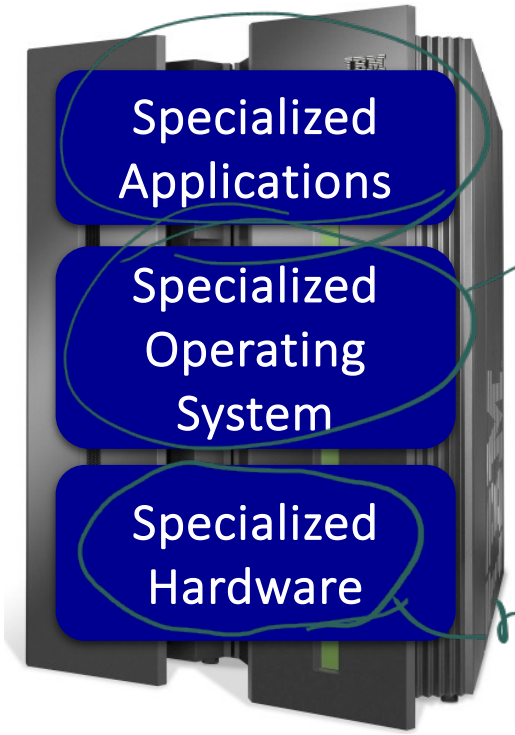


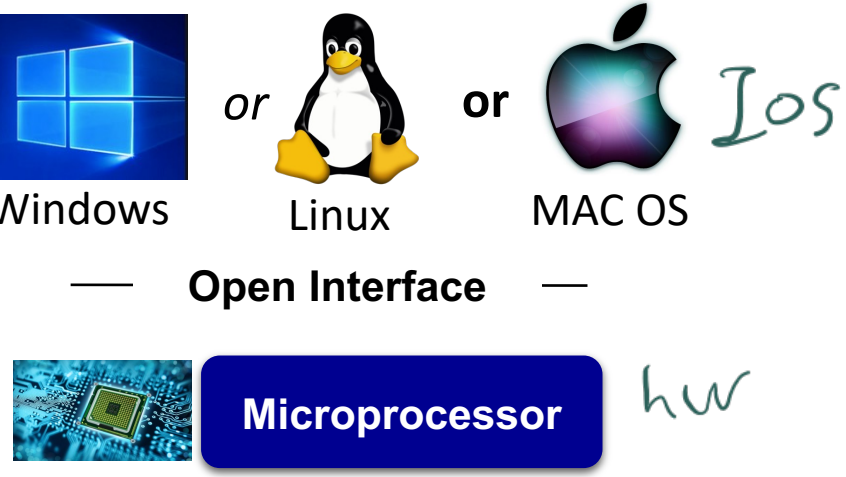
Software defined networking (SDN)

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



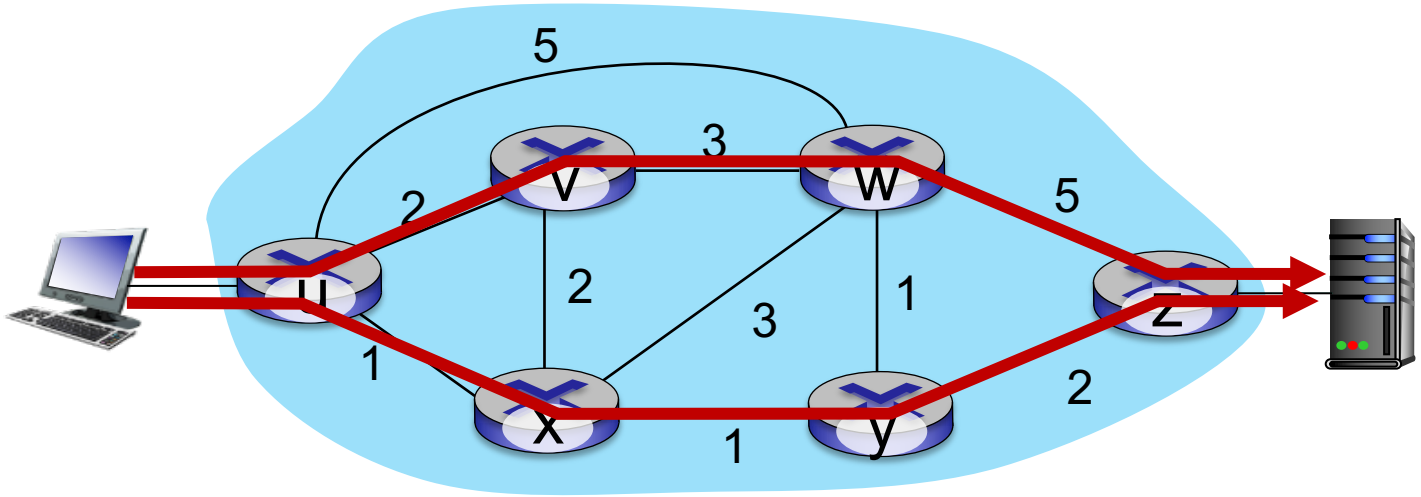


Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

* Slide courtesy: N. McKeown

Traffic engineering: difficult with traditional routing

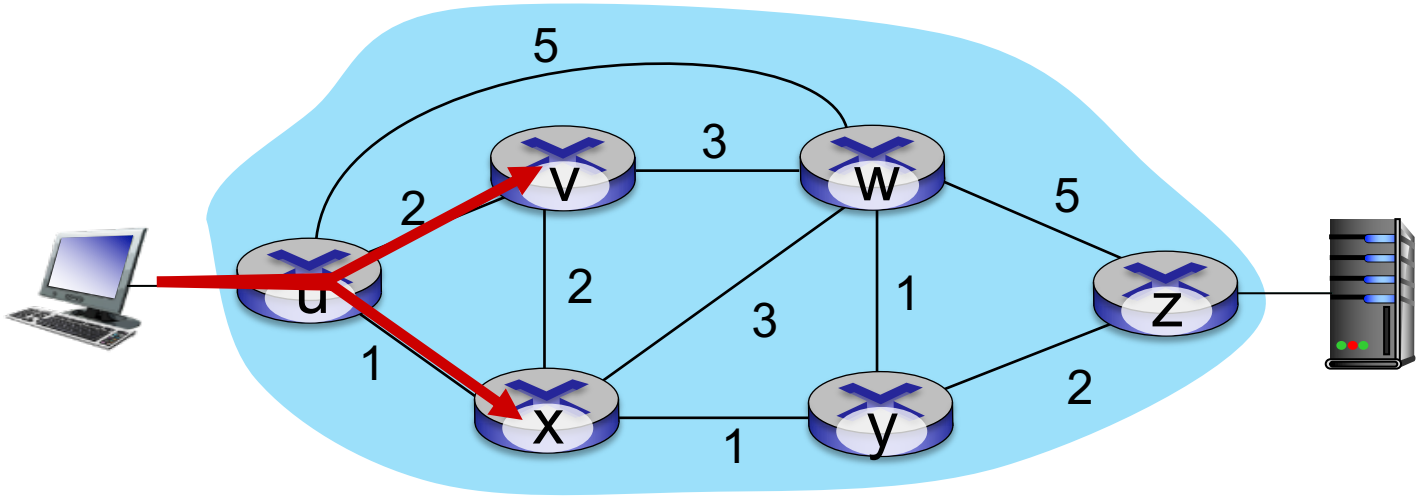


Q: what if network operator wants u-to-z traffic to flow along *uvwz*, rather than *uxyz*?

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!

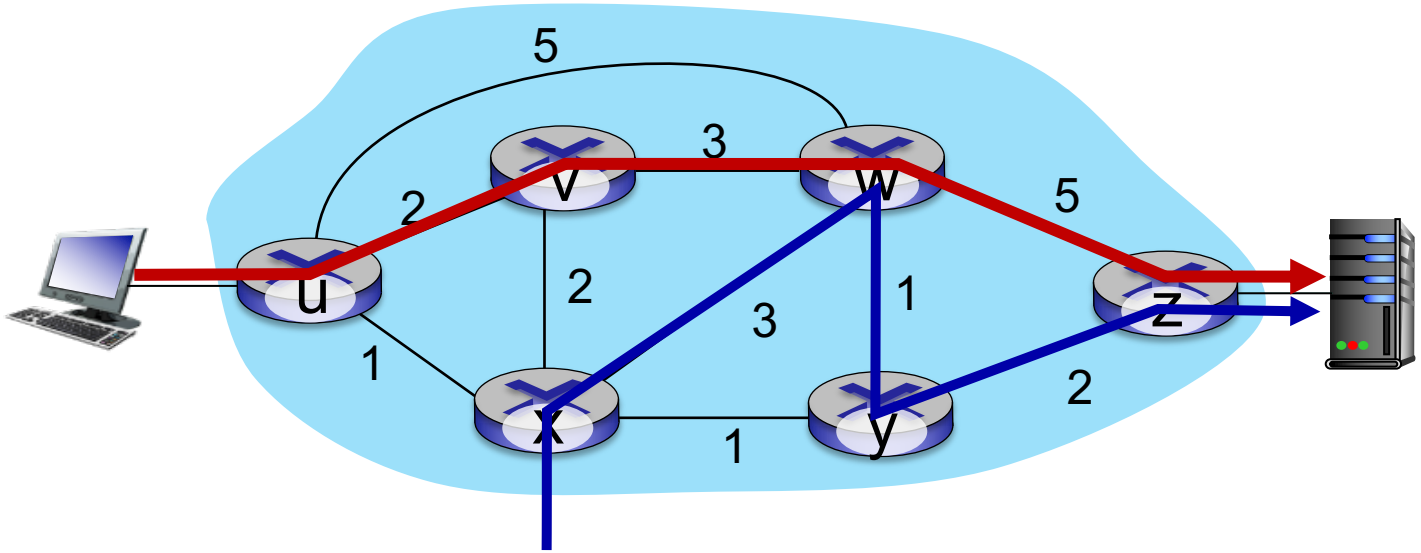
Traffic engineering: difficult with traditional routing



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

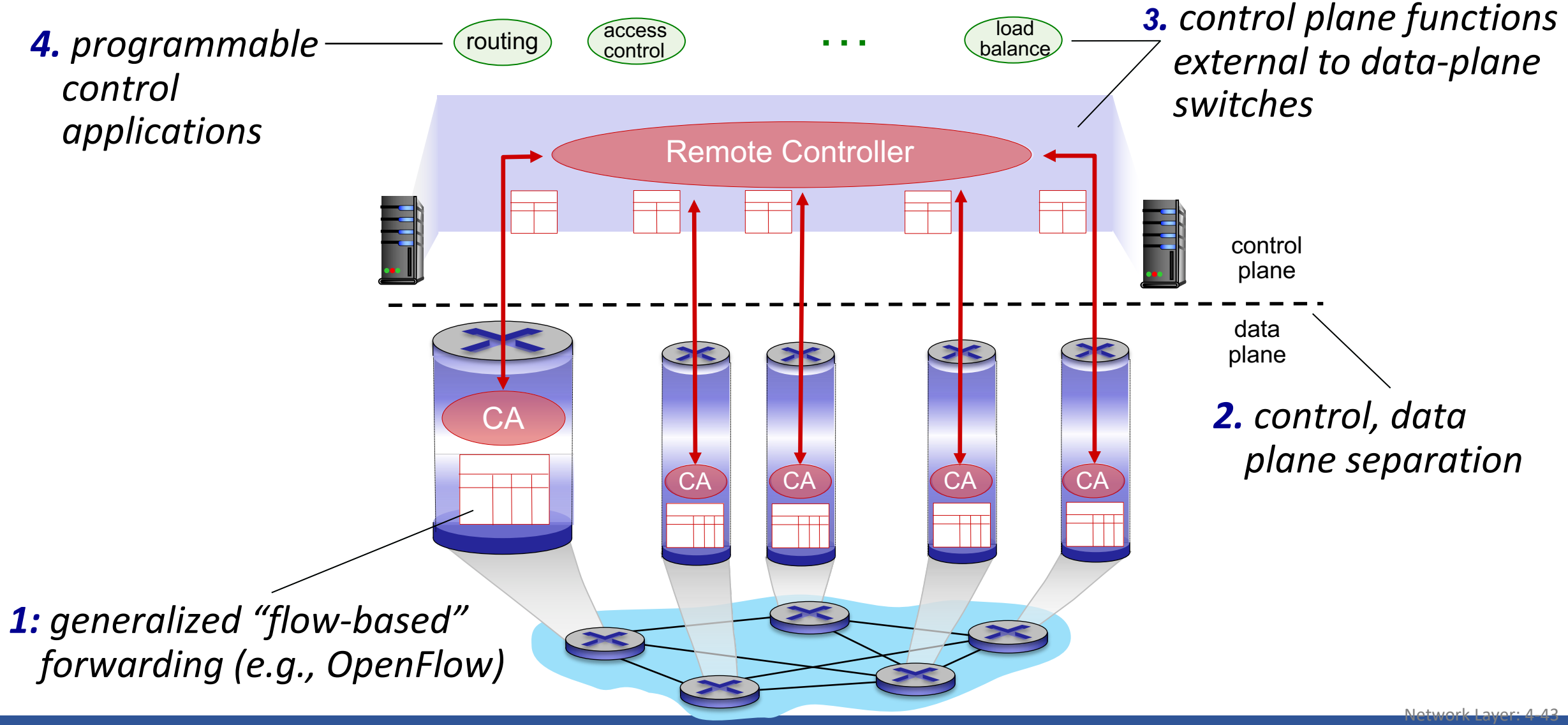


Q: what if w wants to route blue and red traffic differently from w to z?

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

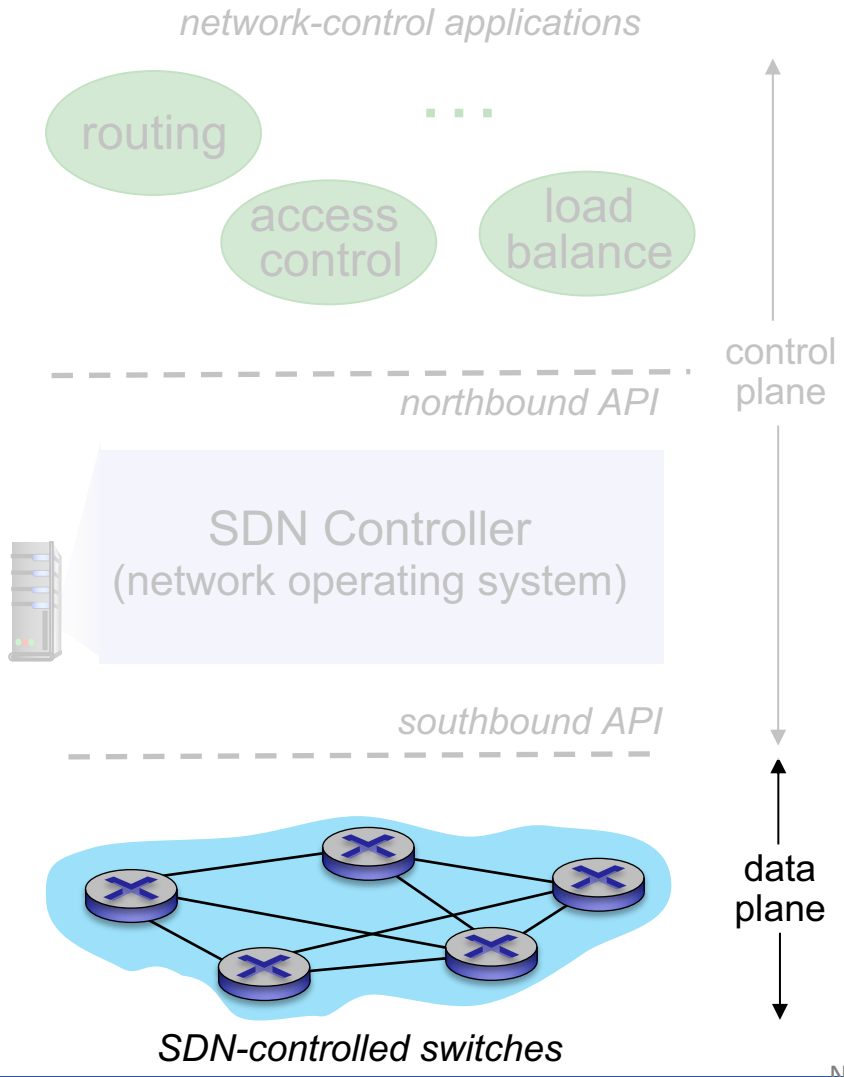
Software defined networking (SDN)



Software defined networking (SDN)

Data-plane switches:

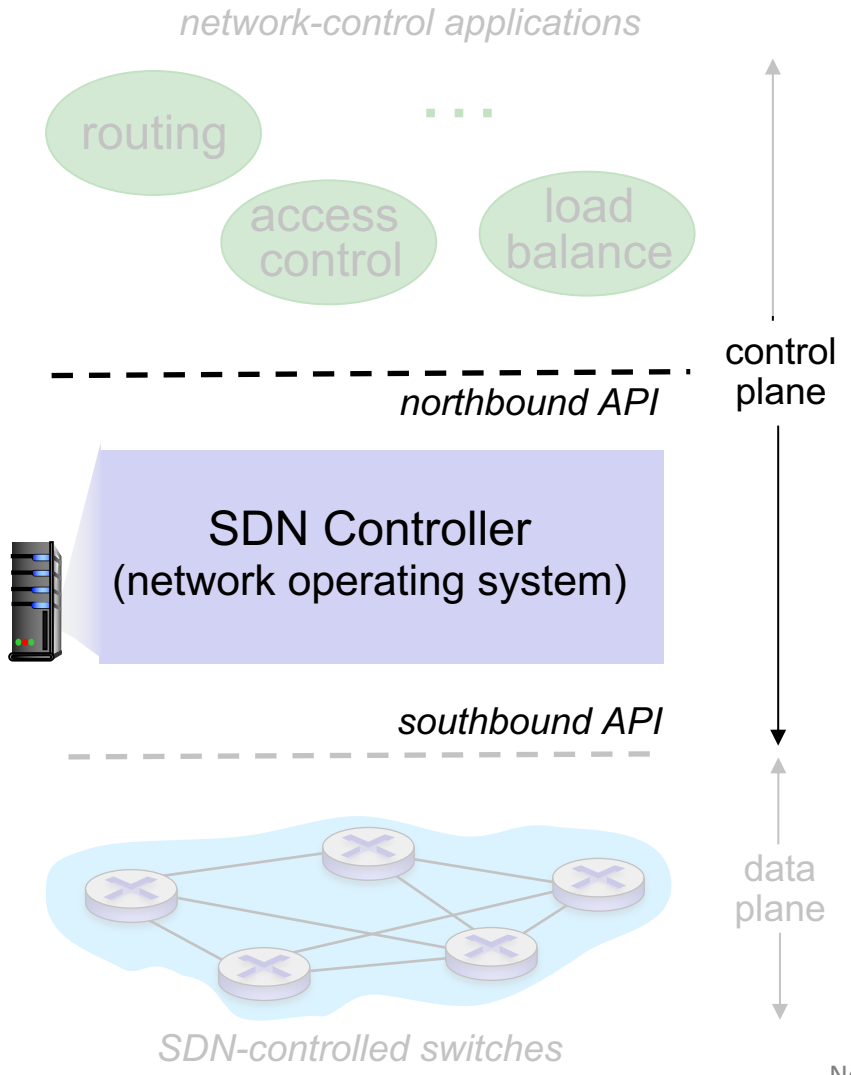
- fast, simple, commodity switches implementing generalized data-plane forwarding (Section 4.4) in hardware
- flow (forwarding) table computed, installed under controller supervision
- API for table-based switch control (e.g., OpenFlow)
  - defines what is controllable, what is not
- protocol for communicating with controller (e.g., OpenFlow)



Software defined networking (SDN)

SDN controller (network OS):

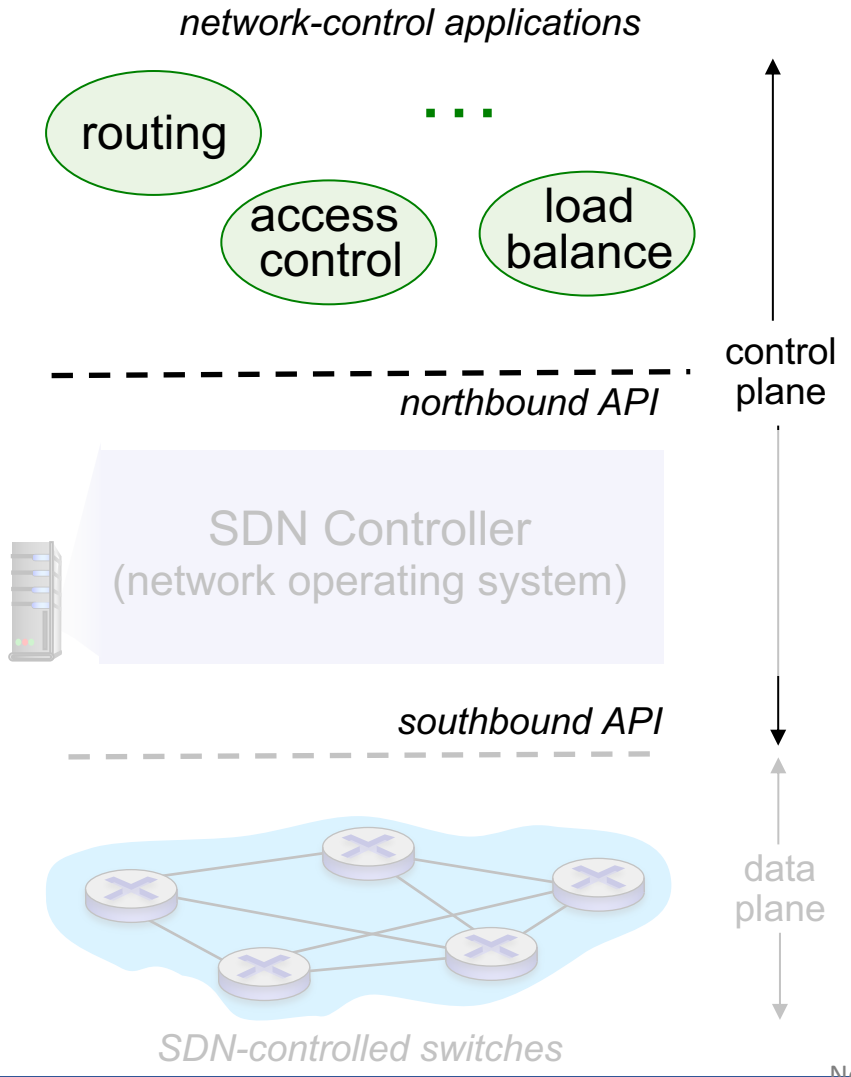
- maintain network state information
- interacts with network control applications “above” via northbound API
- interacts with network switches “below” via southbound API
- implemented as distributed system for performance, scalability, fault-tolerance, robustness



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

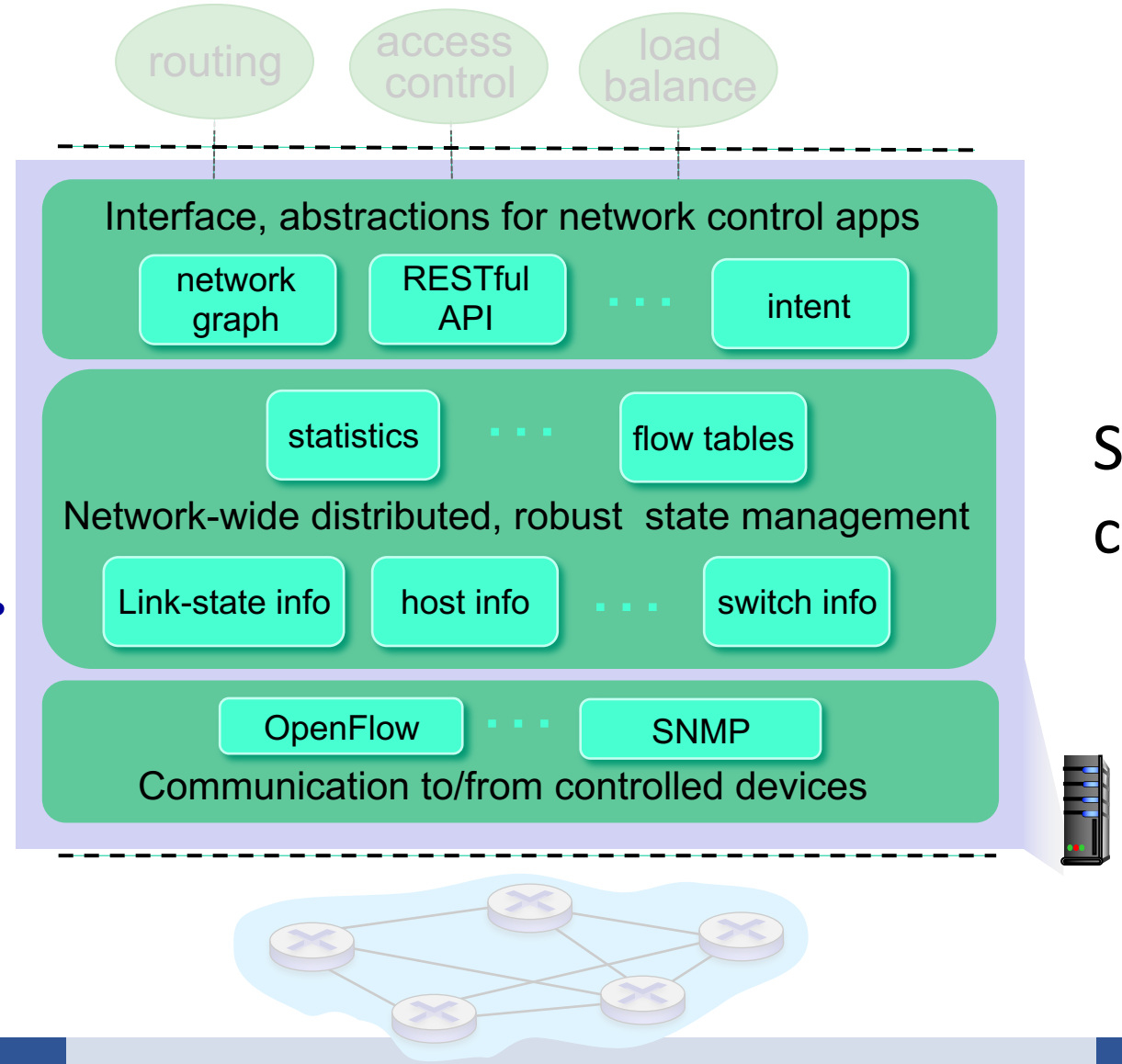


Components of SDN controller

interface layer to network control apps: abstractions API

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

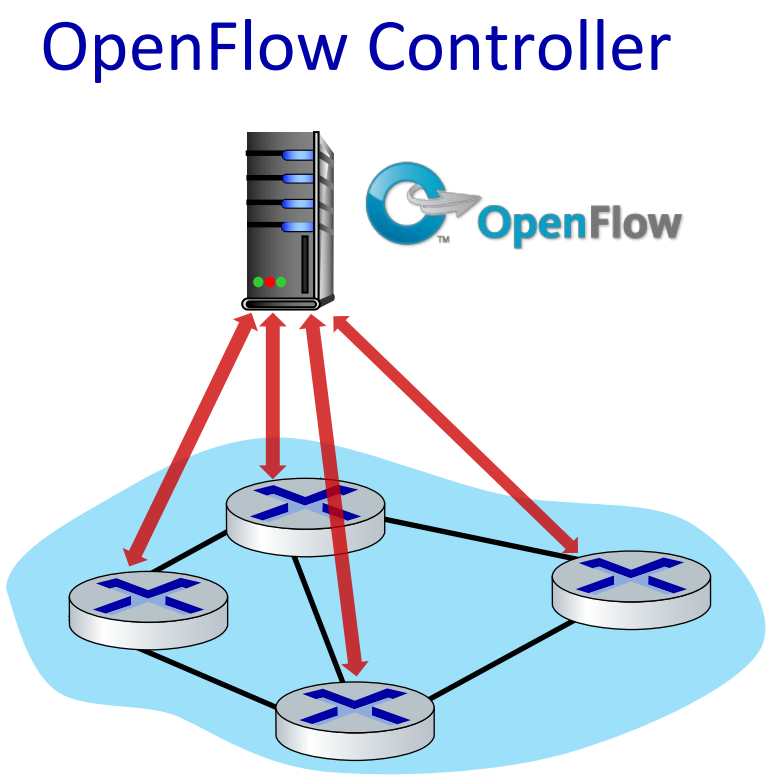
communication: communicate between SDN controller and controlled switches



SDN controller

OpenFlow protocol

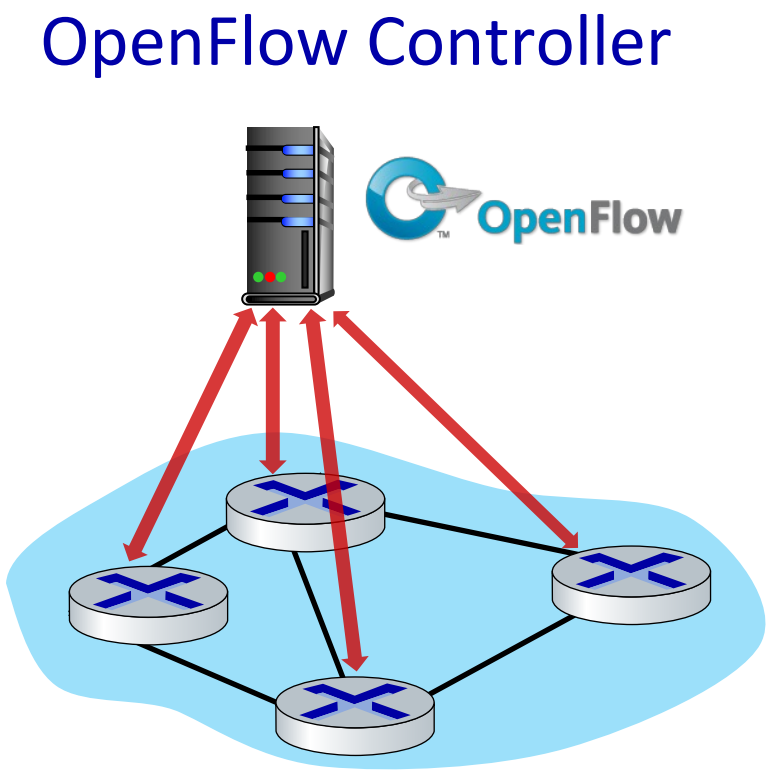
- operates between controller, switch
- TCP used to exchange messages
  - optional encryption
- three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc.)
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions



OpenFlow: controller-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

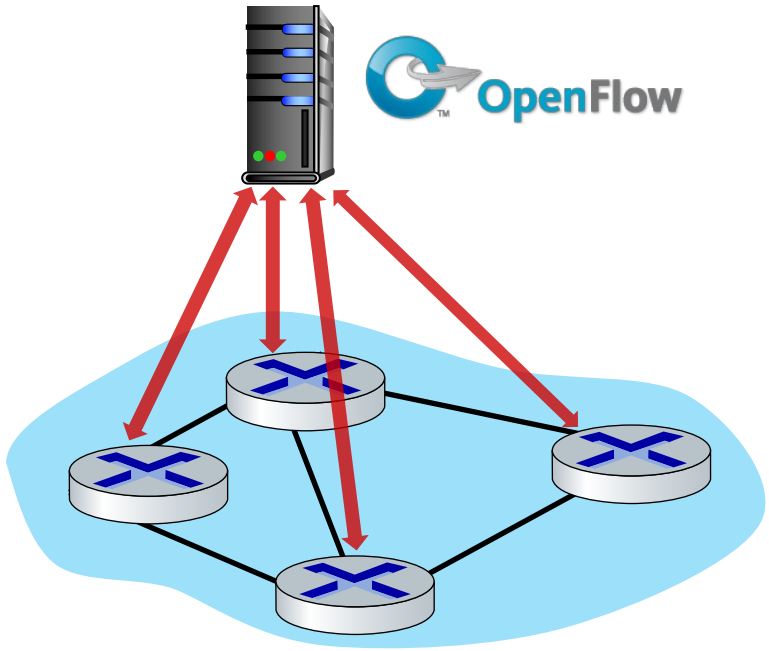


OpenFlow: switch-to-controller messages

Key switch-to-controller messages

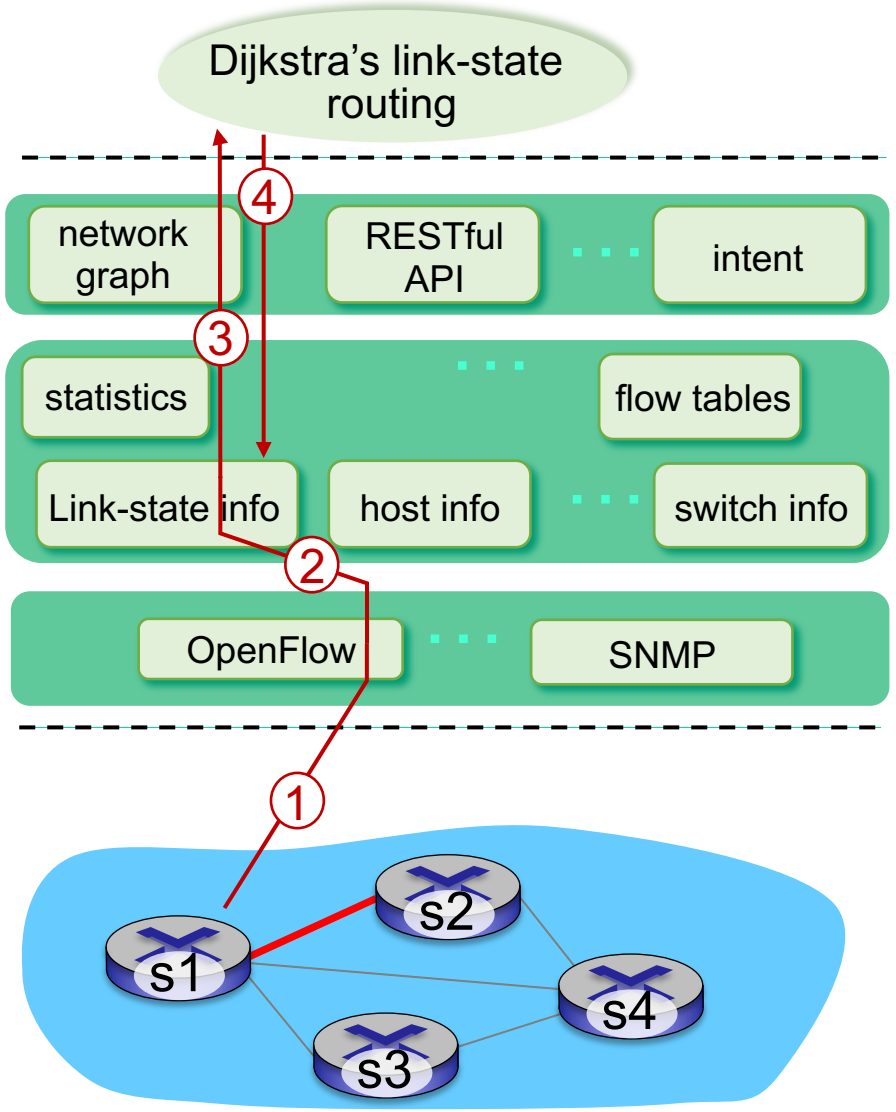
- *packet-in:* transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed:* flow table entry deleted at switch
- *port status:* inform controller of a change on a port.

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



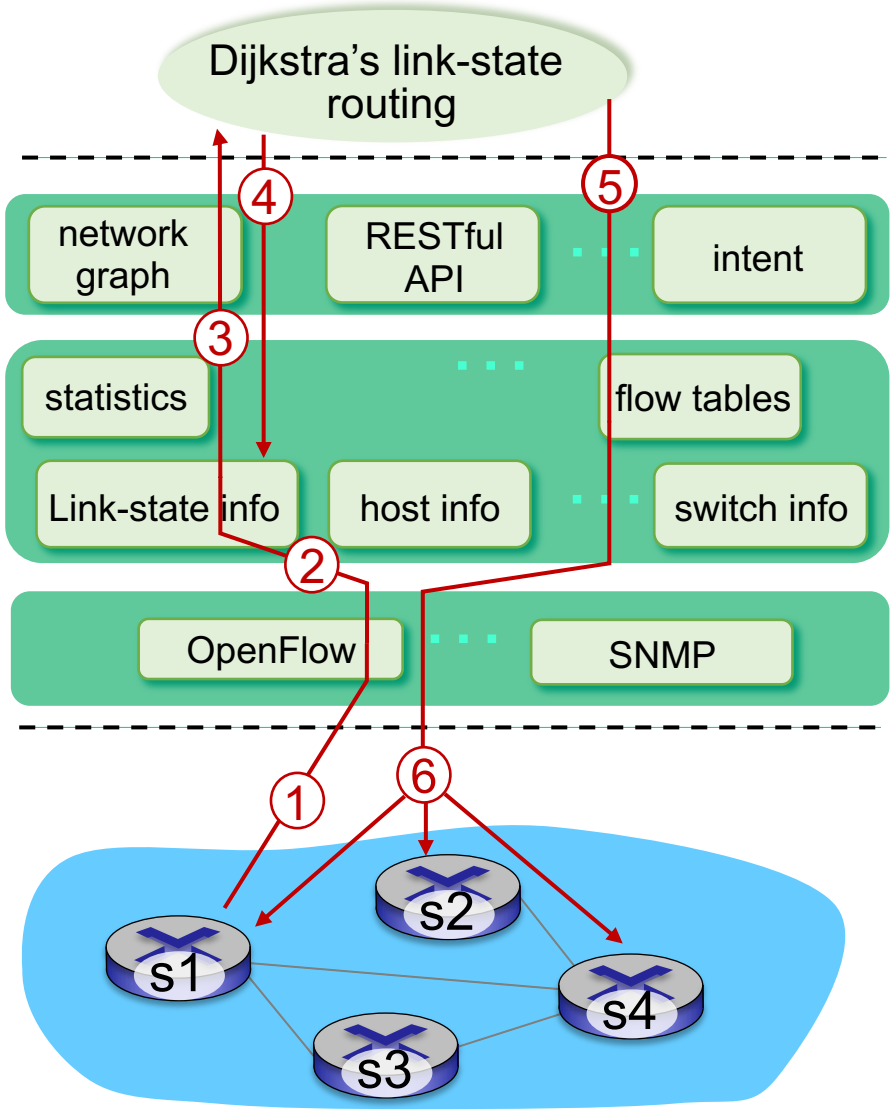
① S1, experiencing link failure uses OpenFlow port status message to notify controller

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

Next…

- *Chapter 6.3-6.4 Multiple Access Links and Protocols*