Introduction to Computer Networks

CS640

IP More Discussion

https://pages.cs.wisc.edu/~mgliu/CS640/F22/

Ming Liu
mgliu@cs.wisc.edu

Today

Last lecture

- How to decide the forwarding path among routers at scale?

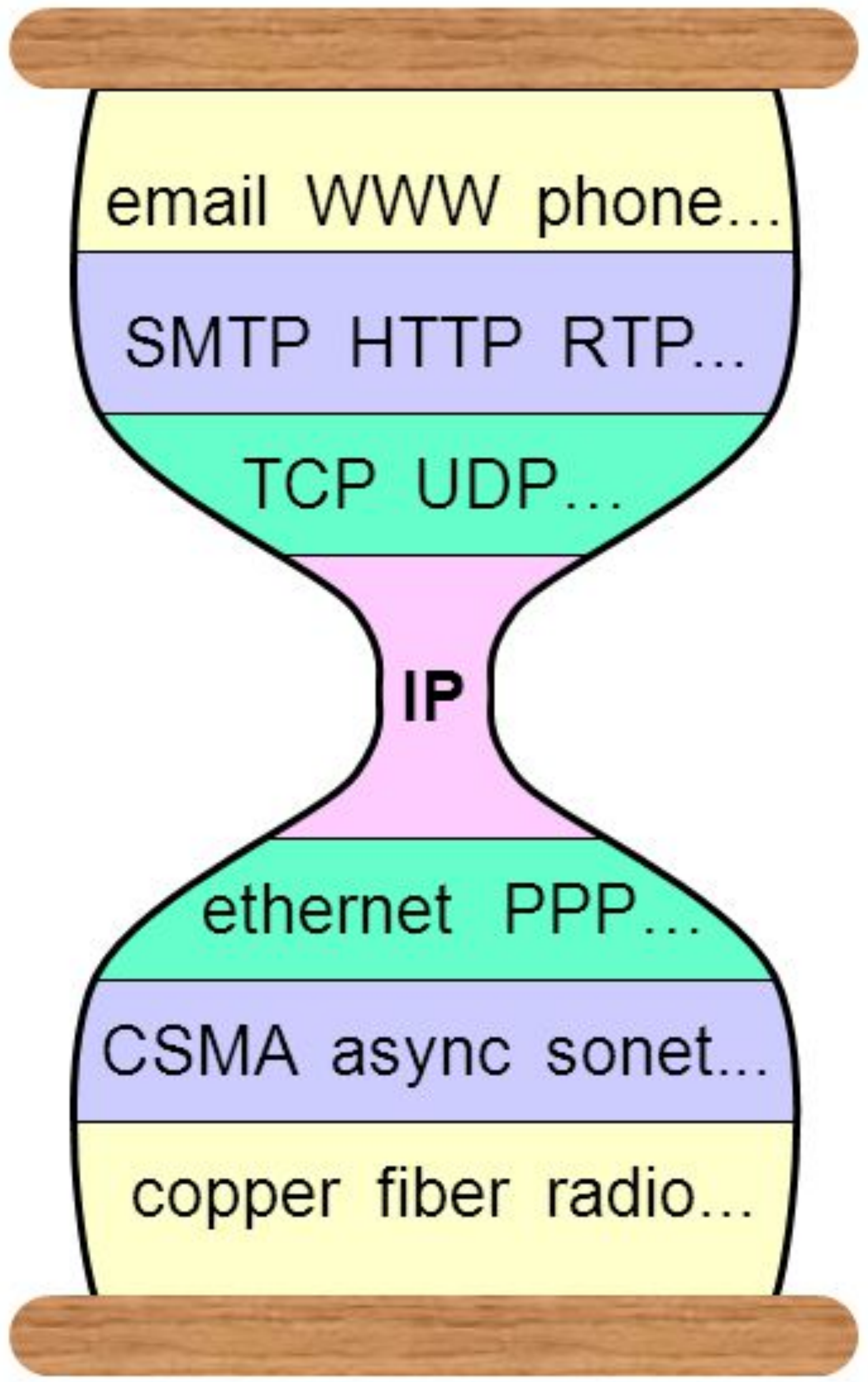
Today

- How to address some limitations in the IP layer?

Announcements

- Lab3 is due 11/04/2022, 11:59 PM
- Quiz3 next Tuesday

IP is powerful, but…



Q1: How to handle a rising number of hosts?

Q1: How to handle a rising number of hosts?

A: Two techniques:

- #1: Private IP address

Private IPv4 addresses

Three ranges are reserved for private use

- Class A: 10.0.0.0 to 10.255.255.255
- Class B: 172.16.0.0 to 172.31.255.255
- Class C: 192.168.0.0 to 192.168.255.255

Private network: a computer network uses the private IPv4 addresses

- Not allocated to any specific organization
- IP packets whose source or destination address is a private IP cannot be routed through the public Internet

Network Address Translation (NAT)

NAT is a process in which one or more local IP address is translated into one or more global IP address, providing Internet access to the local hosts

- A common functionality at the border router

NAT type

- #1: Static NAT
- #2: Dynamic NAT
- #3: Port Address Translation (PAT)

Q1: How to handle a rising number of hosts?

A: Two techniques:

- #1: Private IP address
- #2: IPv6

IPv6 Background

IETF started to design a new version of IP in 1991

Solicitation of suggestions from the community

- First version was completed in 1994

IPv6 Planned Features

#1: 128-bit address space

#2: Support diverse Quality of Service (QoS) apps

#3: Support security and authentication

#4: Auto-configuration

- Hosts are auto-config with an IP address and a domain name
- Try to make systems more plug-n-play

IPv6 Planned Features

#5: Enhanced routing functionality (e.g., Mobile hosts)

#6: Support efficient Multicast

#7: Rely on simple protocol extensions

#8: Enable a smooth transition path from IPv4

IPv6 Address Space

Allocation is classless

Prefixes specify the unicast, multicast, anycast cases

Prefixes can be used to map between v4 and v6

Lots of addresses with 128 bits!

- ~1500 address per square foot of the earth's surface

IPv6 Address Notation

Set of eight 16-bit values separated by colons

- E.g., 47CD:1234:3200:0000:0000:4325:B792:0428

Large number of zeros omitted with series of colons

- E.g., 47CD:1234:3200::4325:B792:0428

Address prefixes (slash notation) are the same as IPv4

- E.g., FEDC:BA98:7600::/40 describes a 40 bit prefix

IPv6 Address Prefix Assignments

0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP (non-IP addresses used by ISO)
0000 010	Reserved for IPX (non-IP addresses used by IPX)
0000 011	Unassigned
0000 1	Unassigned
1	Unassigned
1	Unicast Address Space
10	Unassigned
11	Unassigned
100	Unassigned
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link Local Use addresses
1111 1110 11	Site Local Use addresses
1111 1111	Multicast addresses

IPv6 Unicast Assignment

Unicast address assignment is similar to CIDR

- Unicast addresses are started with 001
- Host interfaces belong to subnets
- Addresses are composed of a subnet prefix and a host identifier
- Prefix aggregation is also possible

IPv6 Unicast Assignment

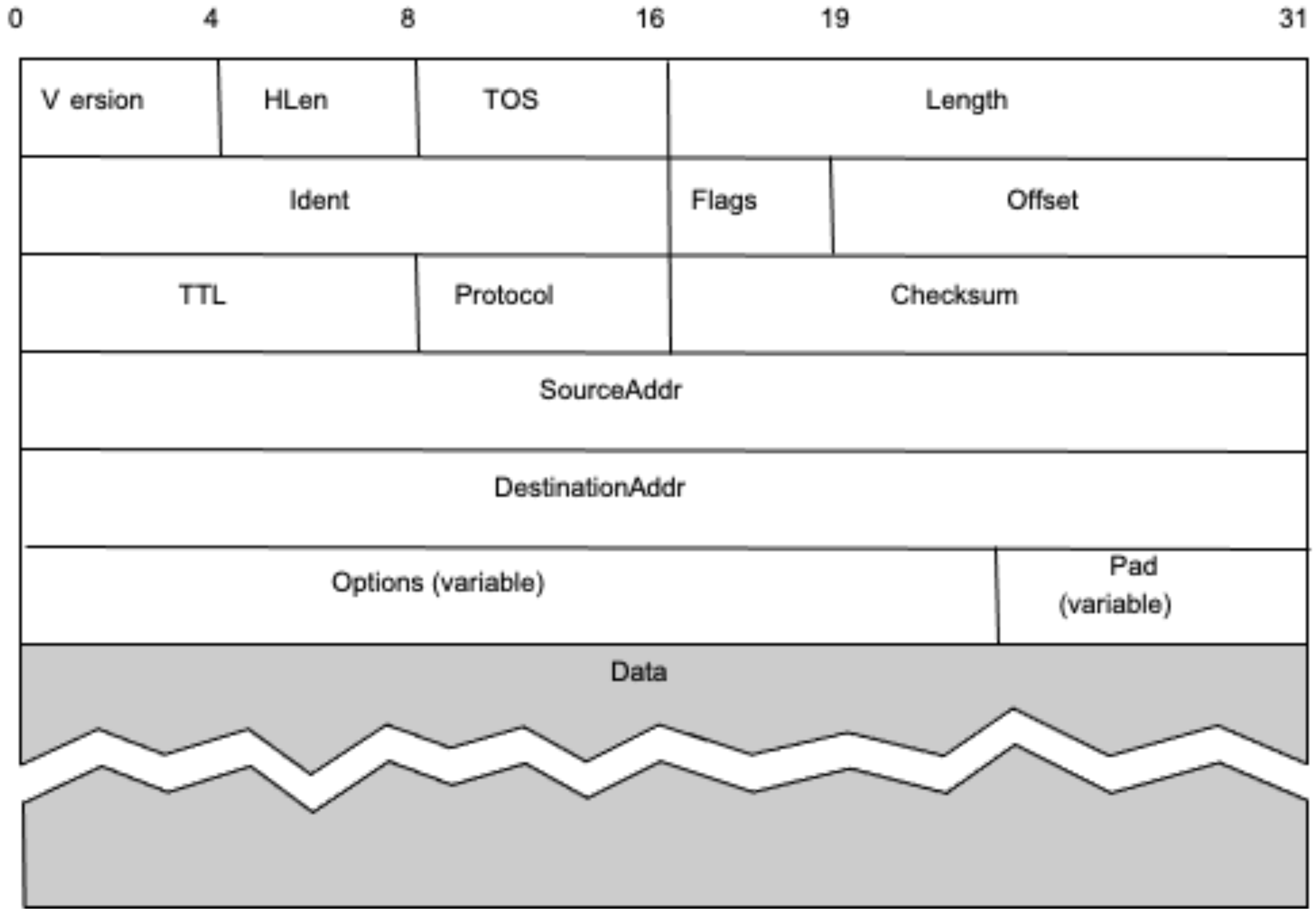
Provider-based plan

- The Internet is organized into a hierarchy of networks
- 3 levels – region, provider, subscriber

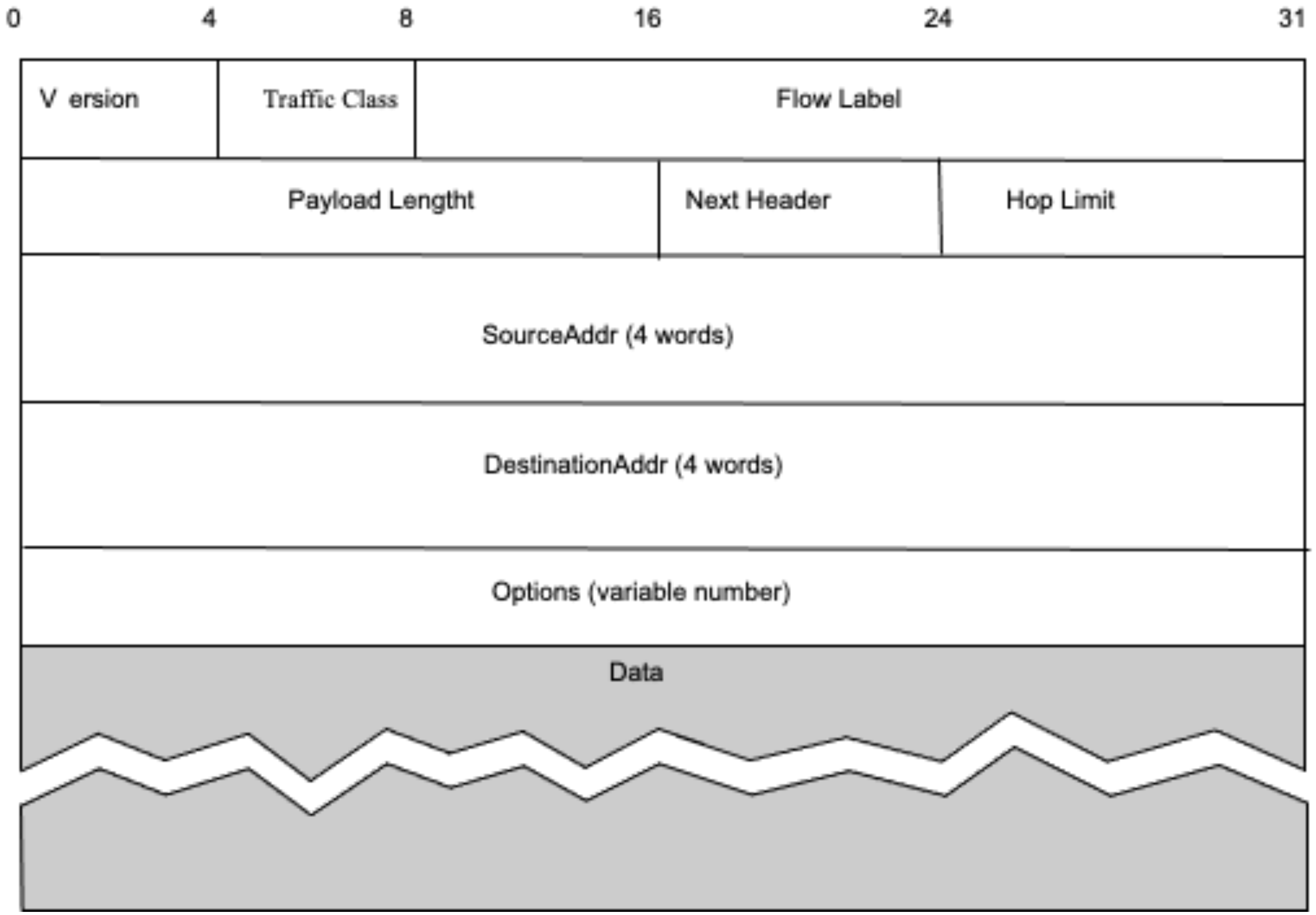
Goal: aggregate routes to reduce BGP overhead

- The provider can advertise a single prefix for all of its subscribers
- Region = 13 bits, Provider = 24 bits, Subscriber = 16 bits, Host = 80 bits
  - E.g., 001, region ID, provider ID, subscriber ID, subnet ID, interface ID

IPv4 Packet Format



IPv6 Packet Format



IPv6 Packet Format Details

Simpler format than IPv4

Version = 6

Traffic class = IPv4 ToS

Treat all packets with the same Flow Label equally

- Support QoS and fair bandwidth allocation

IPv6 Packet Format Details

Payload length does not include header

Hop limit = IPv4 TTL field

Next header combines options and protocol

- If there are no options, then NextHeader is the protocol field

Options are “extension header”

- E.g., routing, fragmentation, authentication, encryption, …

Key Differences in Header

No checksum

- Bit level errors are checked for all over the place

No length variability in the header

- Fixed format speeds processing

No more fragmentation and reassembly in the header

- Incorrectly sized packets are dropped
- Hosts should do path MTU discovery

Transition from v4 to v6

Flag day is not feasible

Dual stack operation — IPv6 nodes run in both v4 and v6 modes and use the version field to decide which stack to use

Tunneling is used to deal with networks where v4 router(s) sit between two v6 routers

- Encapsulate v6 packets in v4 packets until hit the next v6 router

Q2: How to minimize traffic load under a large-scale network?

Q2: How to minimize traffic load under a large-scale network?

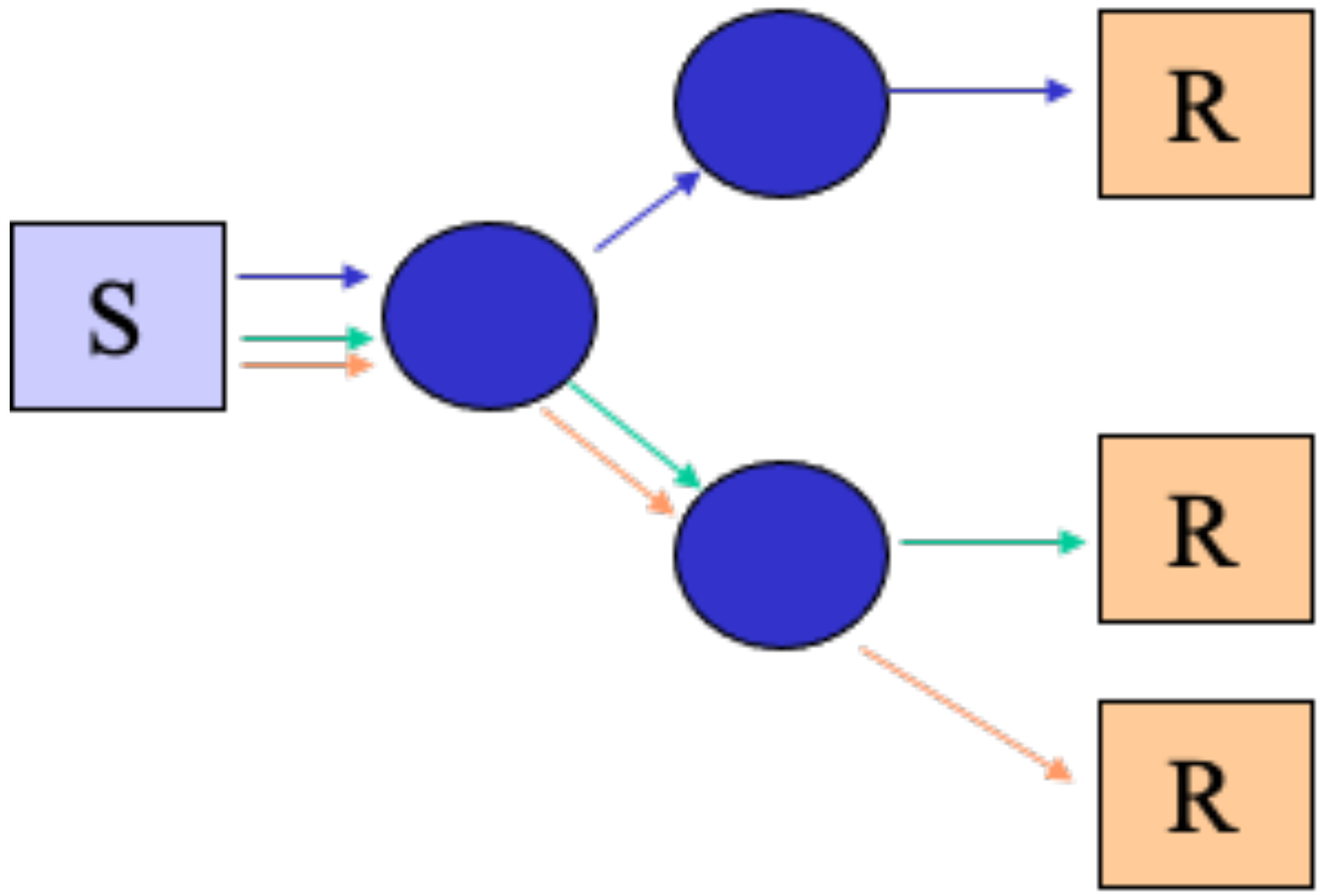
A: Multicast is one solution

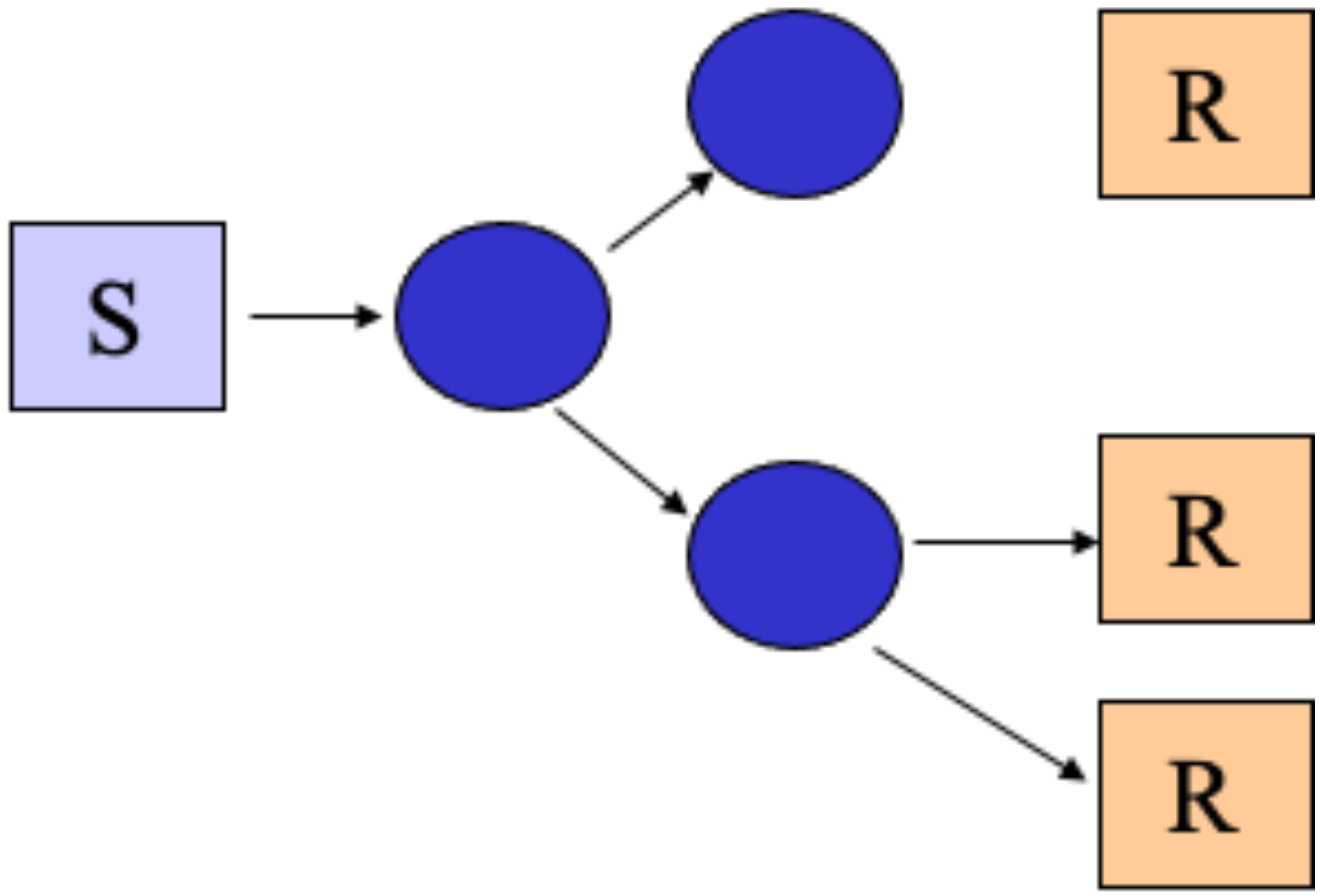
One to Many Communication

Application level one-to-many communication

- Multiple Unicasts

IP multicast





Why Multicast

When sending the same data to multiple receivers

- Better bandwidth utilization
- Less host/router processing
- Fast participation

Benefit applications

- Video/Audio broadcast (One sender)
- Video conferencing (Many senders)
- Real-time news distribution
- Interactive gaming

IP Multicast Service Model

Invented by Steve Deering (Ph.D. 1991)

- It's a different way of routing datagrams

RFC 1112: Host Extensions for IP Multicasting — 1989

IP Multicast Workflow

#1: Configuration

- Members join and leave the group and indicate this to the routers
- The «host group» is identified by a class D IP address

#2: Execution

- Senders transmit IP datagrams to a «host group»
- Routers listen to all multicast addresses, managed by the multicast routing protocol
- Members of the host group could be present anywhere on the Internet

IP Multicast Group Address

Class D address space

- High-order three-bits are set
- 224.0.0.0 ~ 239.255.255.255

Allocation is essentially random — any class D can be used

- Nothing prevents an application from sending to any multicast address

Multicast Packets —> End Hosts

Packets from remote sources will only be forwarded by IP routers onto a local network only if they know there is at least one recipient for that group on that network

Internet Group Management Protocol (IGMP, RFC 2236)

- Used by end hosts to signal that they want to join a specific multicast group
- Used by routers to discover the mapping between end hosts and multicast groups
- Implemented directly over IP

IGMP — Joining a Group

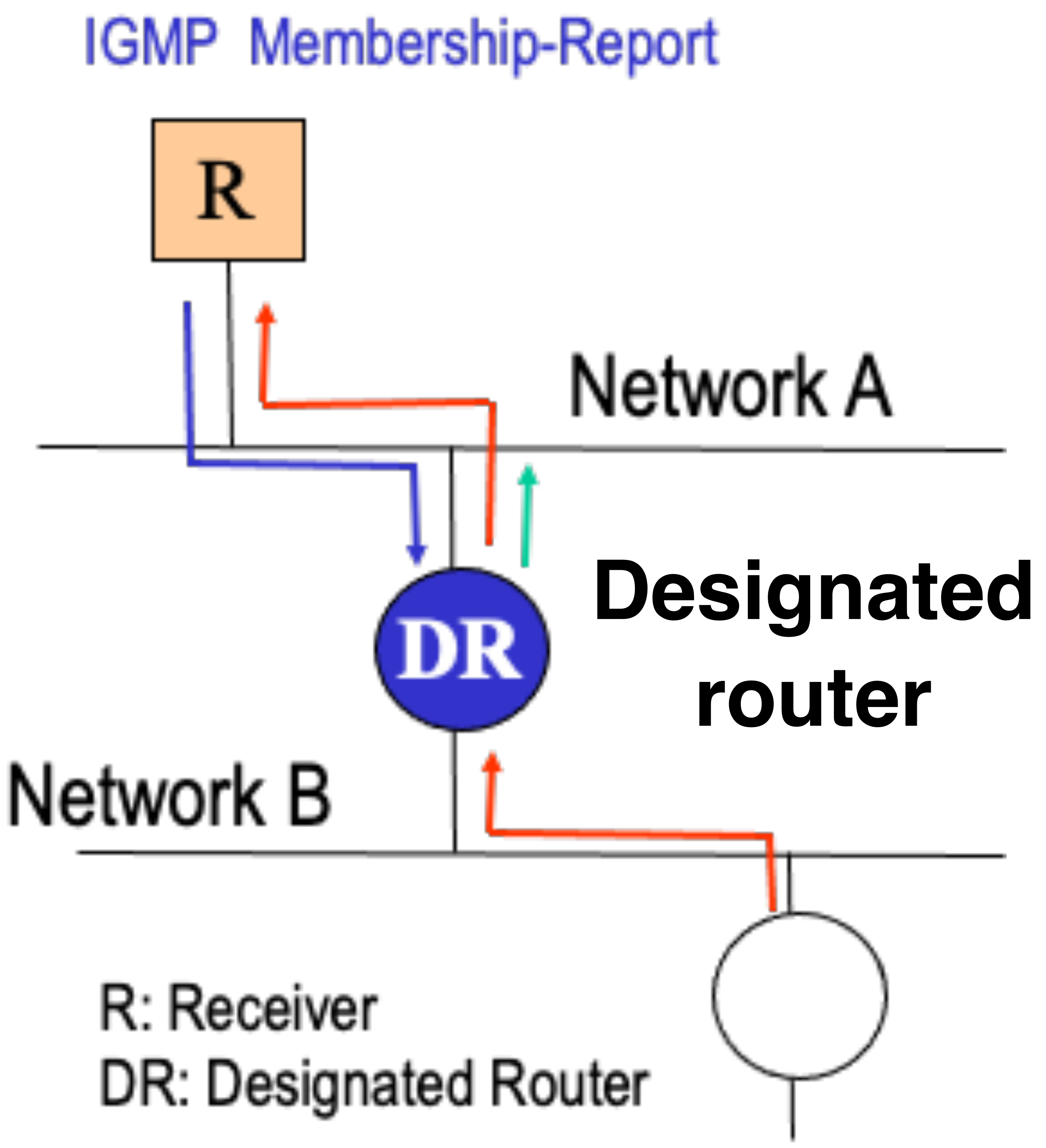
Example: R joins to Group 224.2.0.1

#1: R sends an **IGMP Membership-Report to 224.2.0.1**

#2: DR receives it. DR will start forwarding **packets for 224.2.0.1** to Network A

#3: DR periodically sends **IGMP Membership-Query to 224.0.0.1** (All-SYSTEM.MCAST.NET)

#4: R answers **IGMP Membership-Report to 224.2.0.1**



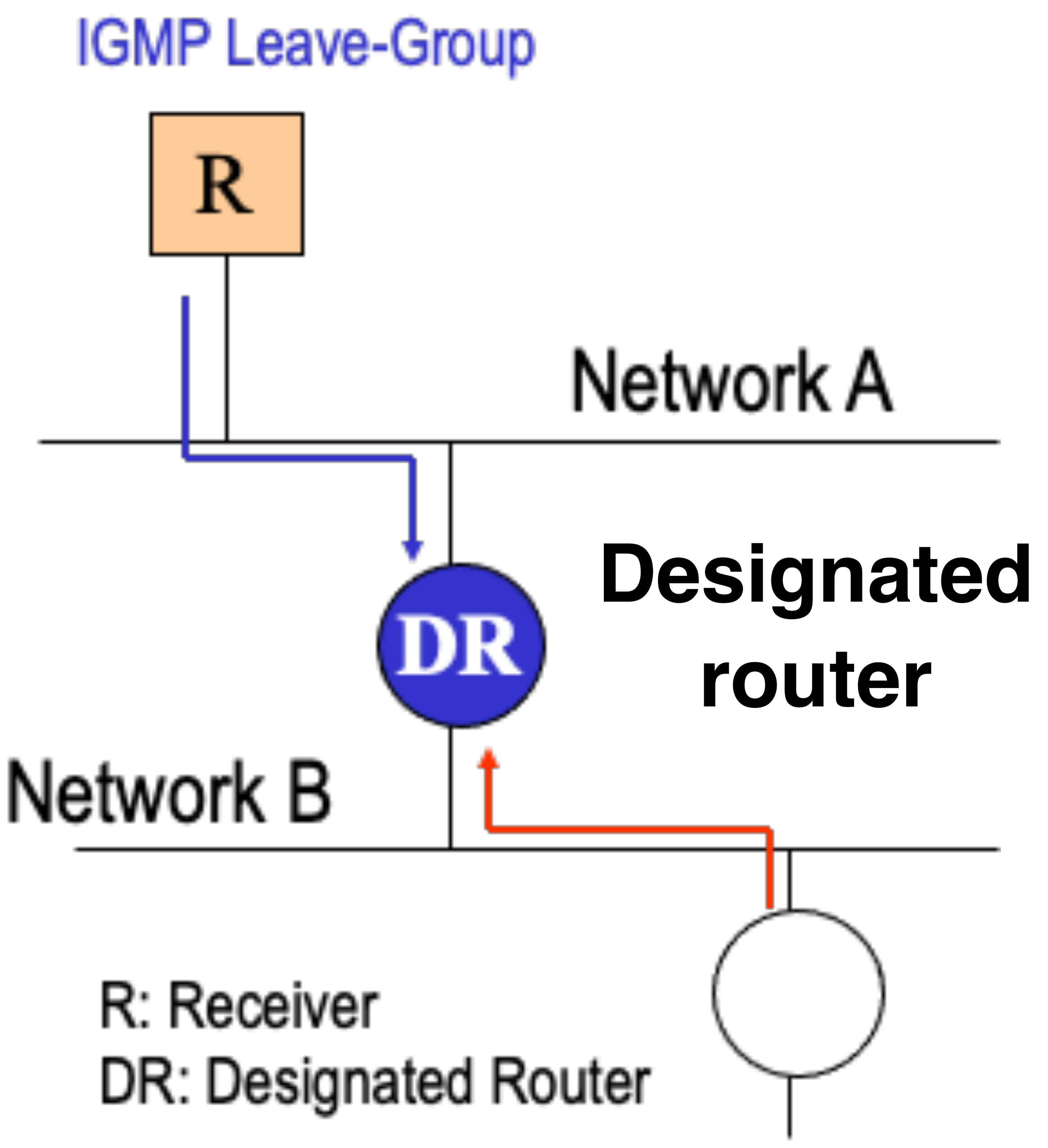
IGMP — Leaving a Group

Example: R leaves from a Group 224.2.0.1

#1: R sends **IGMP Leave-Group to 224.0.0.2** (ALL-ROUTES.MCAST.NET)

#2: DR receives it

#3: DR stops forwarding **packets for 224.2.0.1** if no more 224.2.0.1 group members on Network A



Challenges in the Multicast Model

How can a sender restrict who can receive a packet?

- Need authentication and authorization
- Encryption of data
- Key distribution
- Still an active area of research

Multicast Requires Router Support

Purpose: share the group information among routers to implement better routing for data distribution

Distribution tree structure

Routing protocols are used in conjunction with IGMP

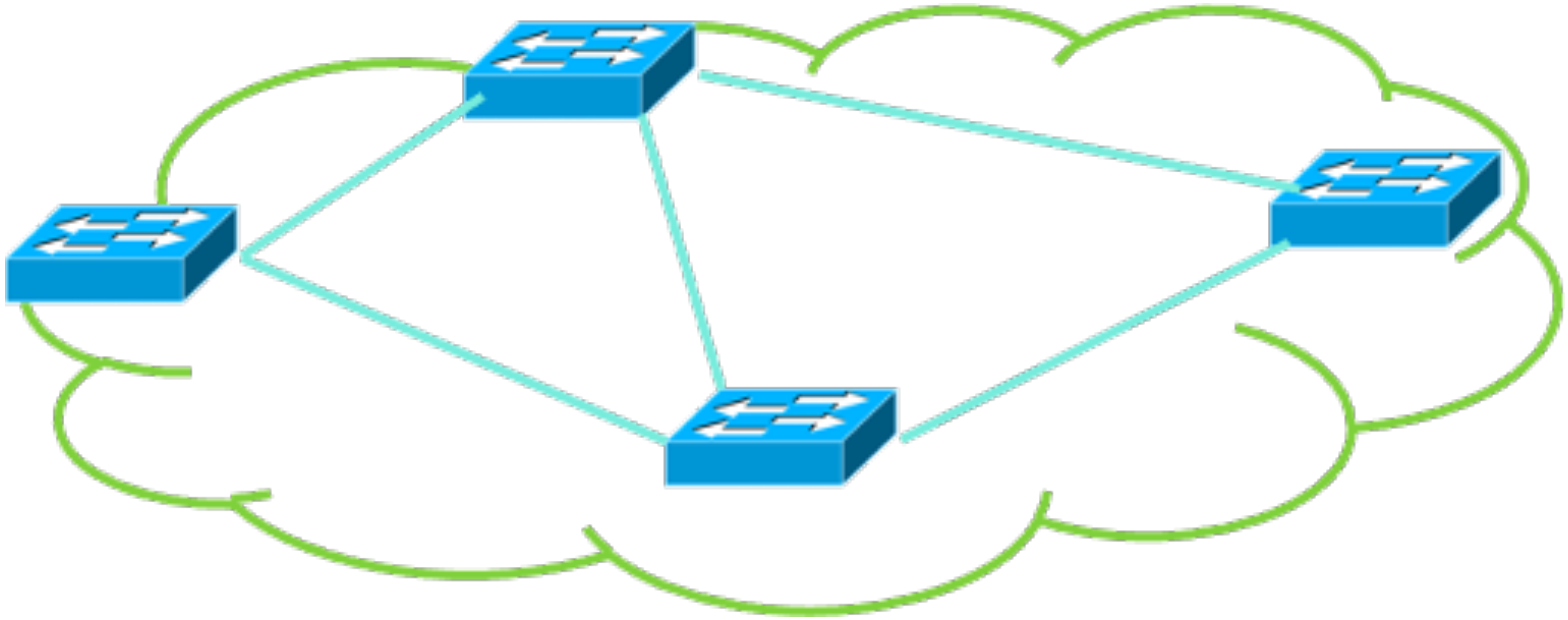
Q3: How to integrate application awareness in the IP layer?

Q3: How to integrate application awareness in the IP layer?

A: Software-defined networking

Traditional Computer Networks

Forward, filter, buffer, mark, rate-limit, and measure packets

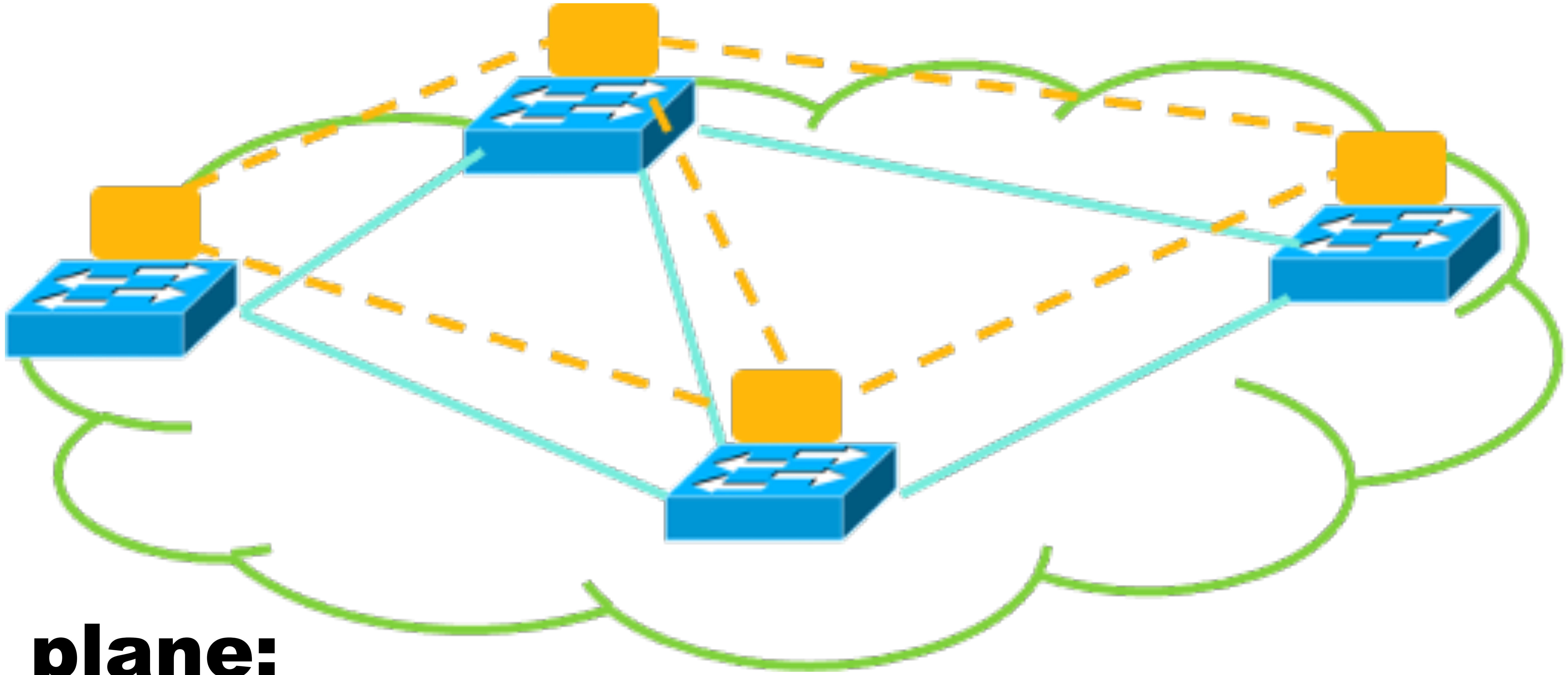


Data plane:

- Packet streaming

Traditional Computer Networks

Track topology changes, compute routes, install forwarding/filtering rules

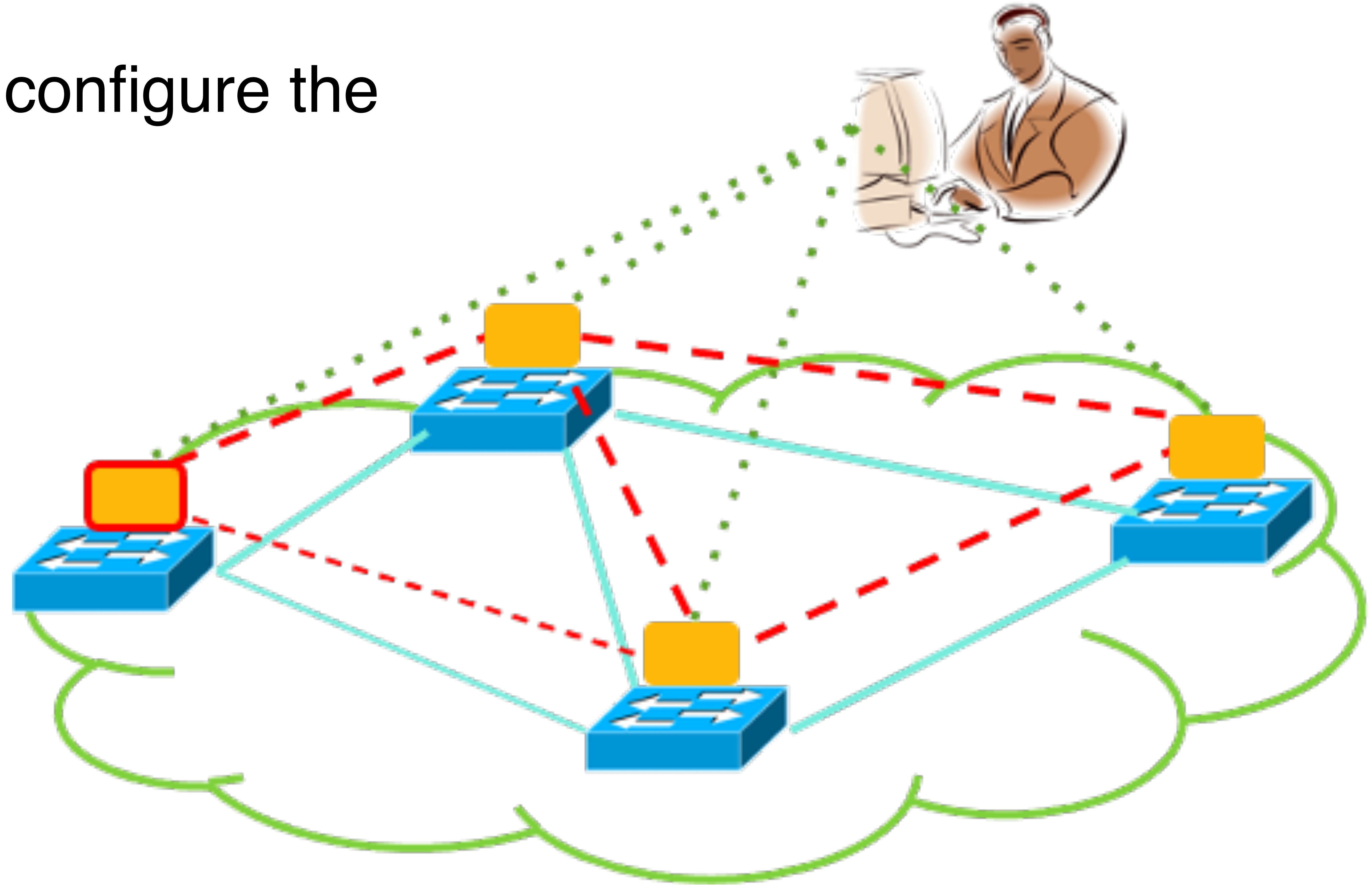


Control plane:

- Distributed algorithms

Traditional Computer Networks

Collect measurements and configure the equipment



Management plane:

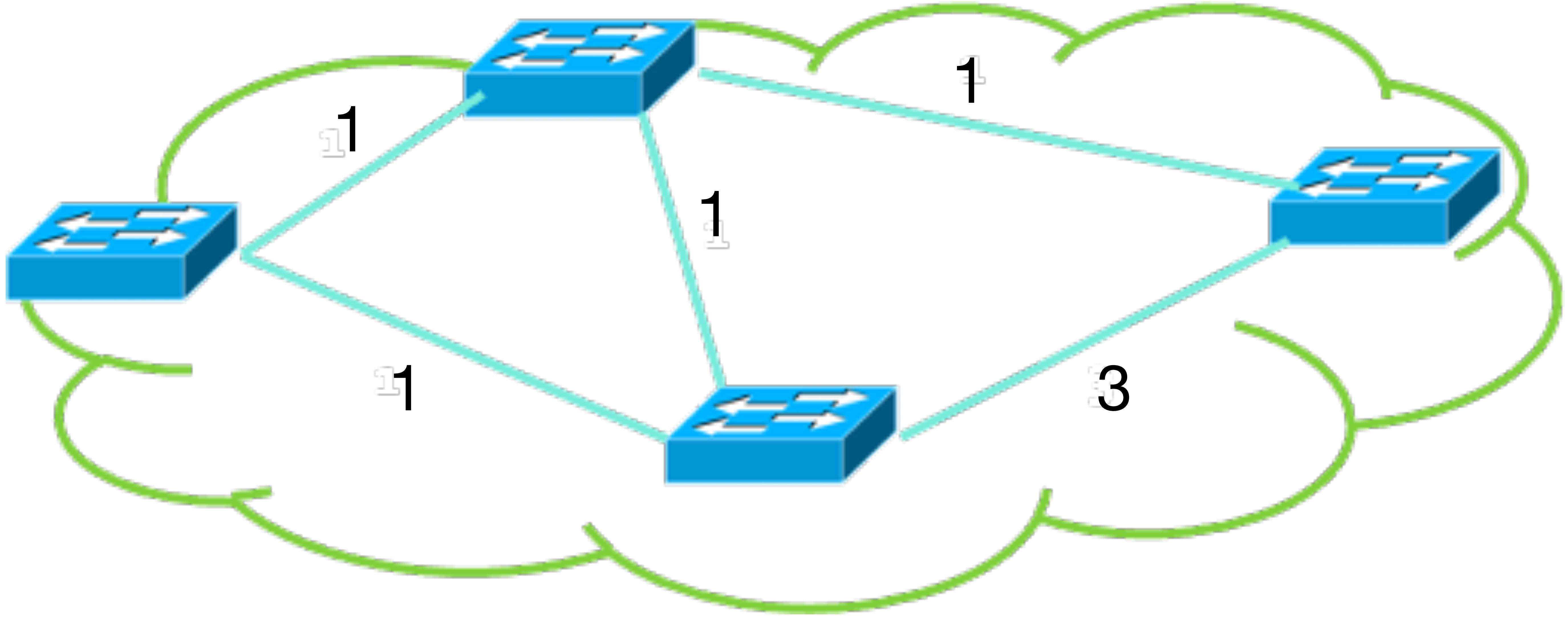
- Human time scale

Shortest-Path Routing

Management: set the link weights

Control: compute shortest paths

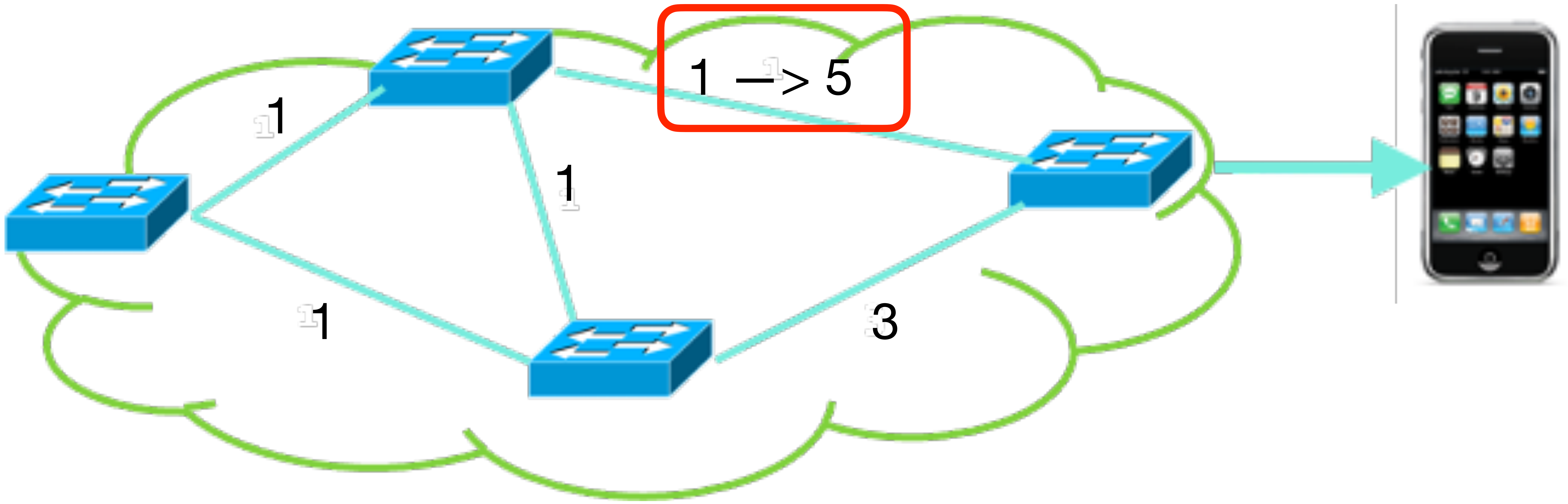
Data: forward packets to next hop



Case #1: Inverting the Control Plane

Traffic engineering

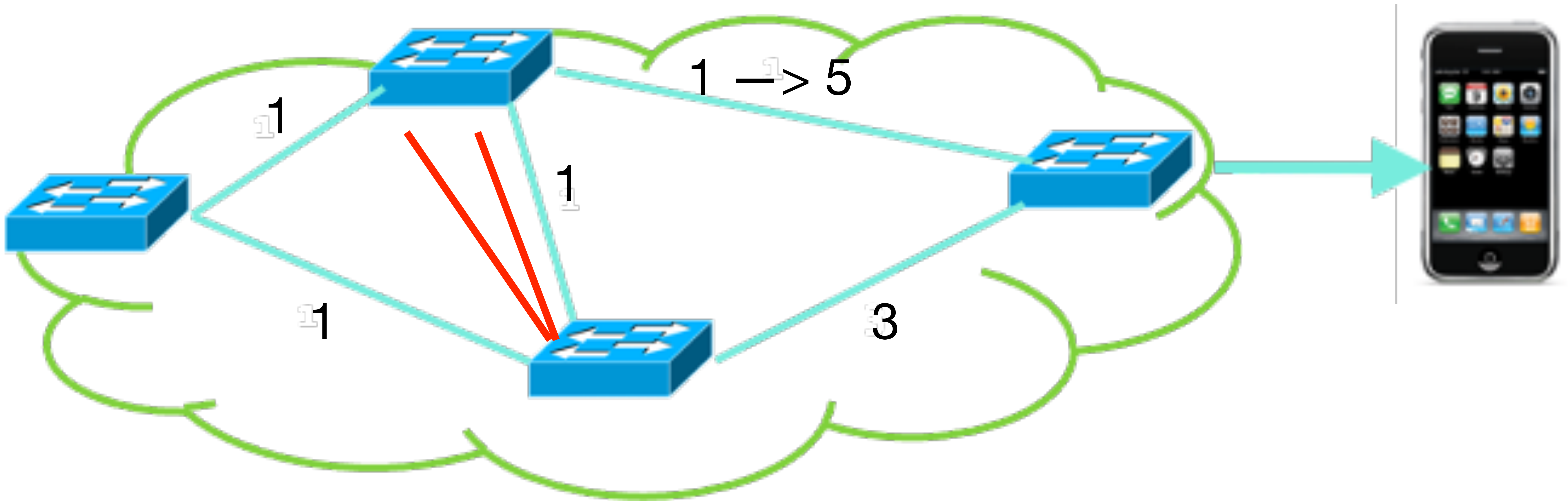
- Change link weights
- … to induce the paths
- … that alleviate congestion



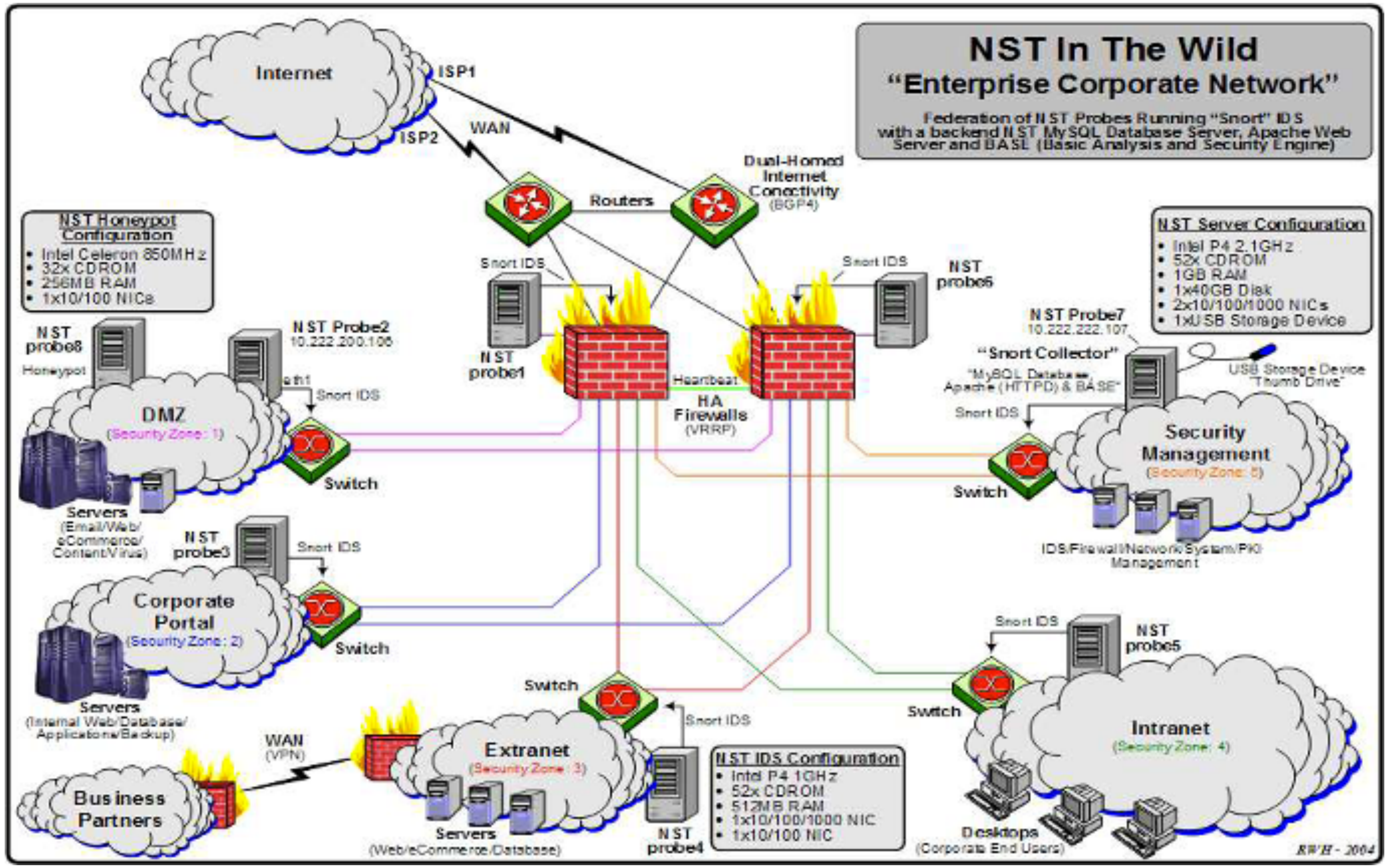
Case #2: Transient Anomalies

Distributed protocol

- Temporary disagreement among the nodes
- … leaves packets stuck in loops
- Even though the changes was planned!



A Lot Messier

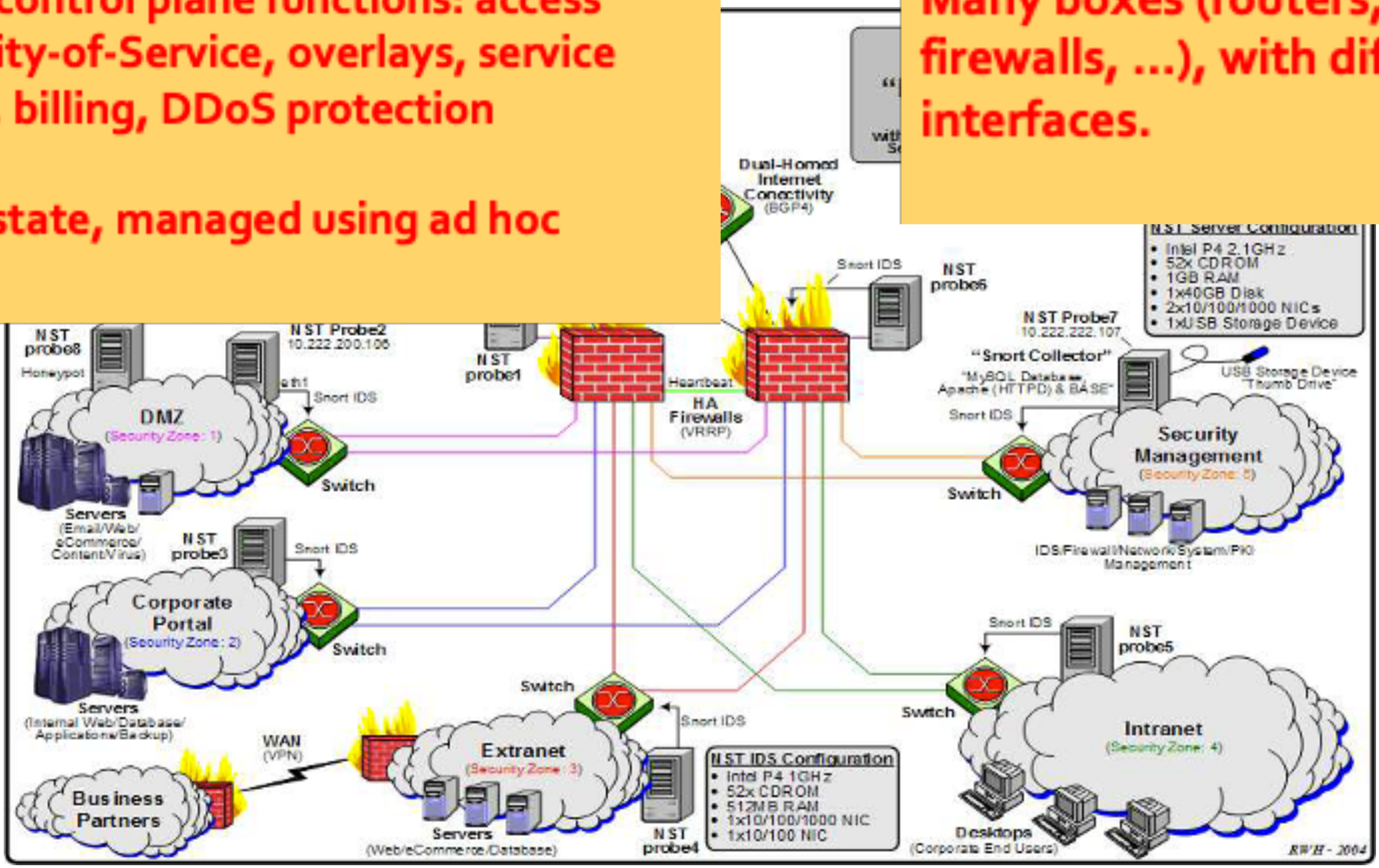


A Lot Messier

Other mgmt/control plane functions: access control, Quality-of-Service, overlays, service interposition, billing, DDoS protection

Non-routing state, managed using ad hoc mechanisms

Many boxes (routers, switches, firewalls, …), with different interfaces.

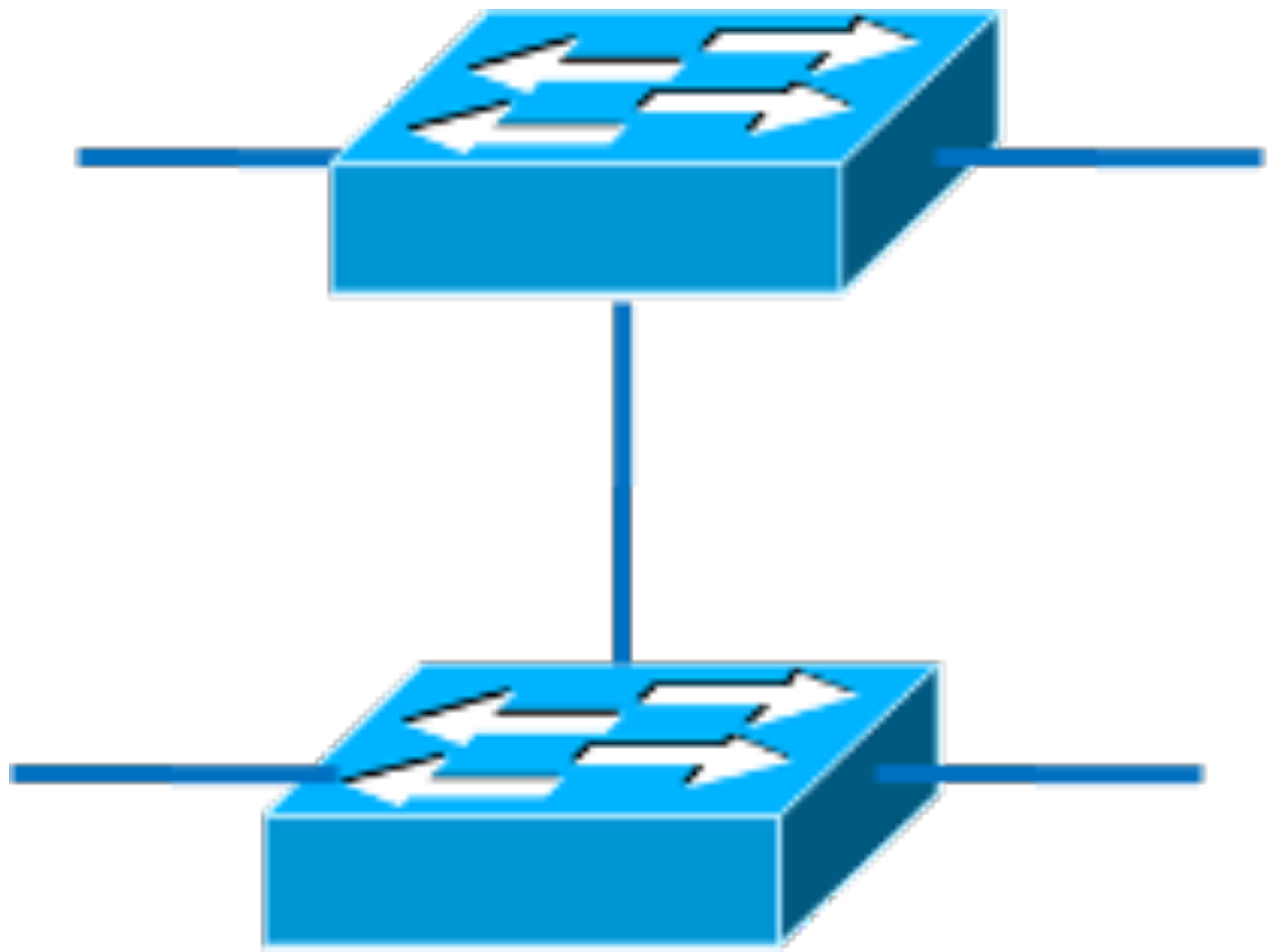


What is the problem?

Closed equipment

- Software bundled with hardware
- Vector-specific interfaces

Distributed nature of control plane



Ad hoc management approaches

Slow protocol standardization

What is the problem?

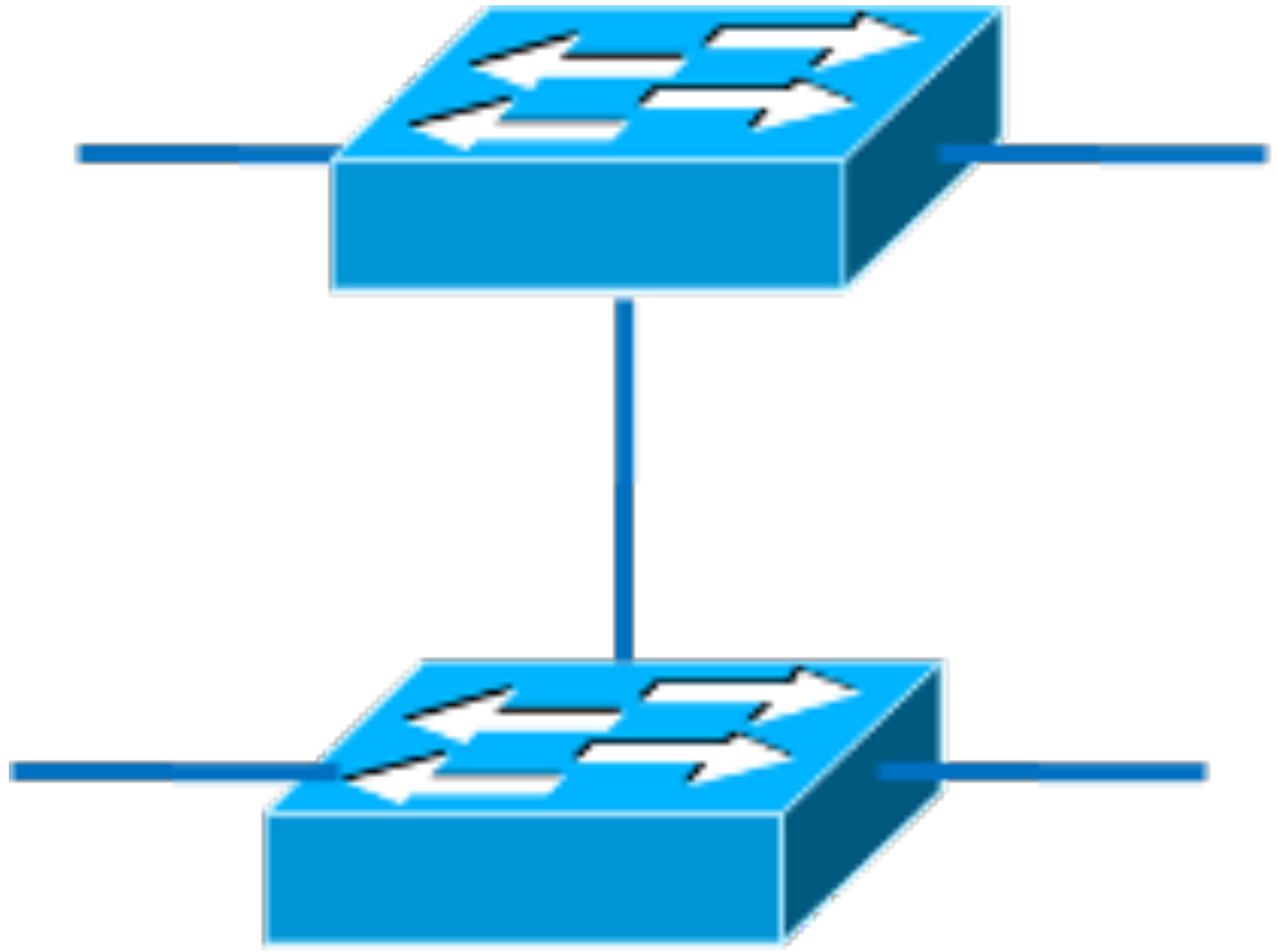
Closed equipment

- Software bundled with hardware
- Vector-specific interfaces

Distributed nature of control plane

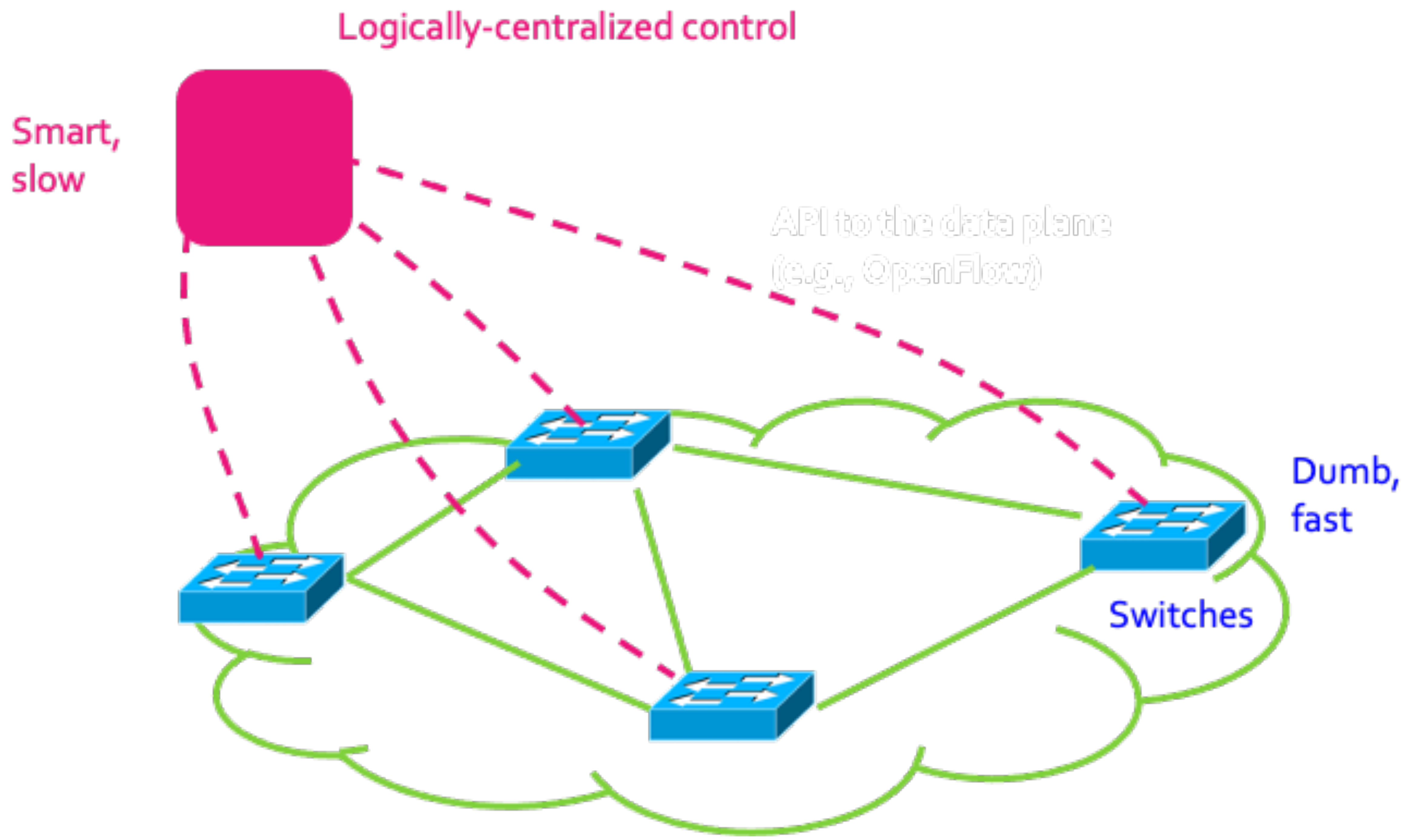
Ad hoc management approaches

Slow protocol standardization



Impacts performance, security, reliability, cost, ..
Innovation is hard

Software Defined Networking



OpenFlow: Enabling Innovation in Campus Networks

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

Jennifer Rexford
Princeton University

Scott Shenker
University of California, Berkeley

Jonathan Turner
Washington University in St. Louis

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
Authors take full responsibility for this article's technical content.
Comments can be posted through CCR Online.

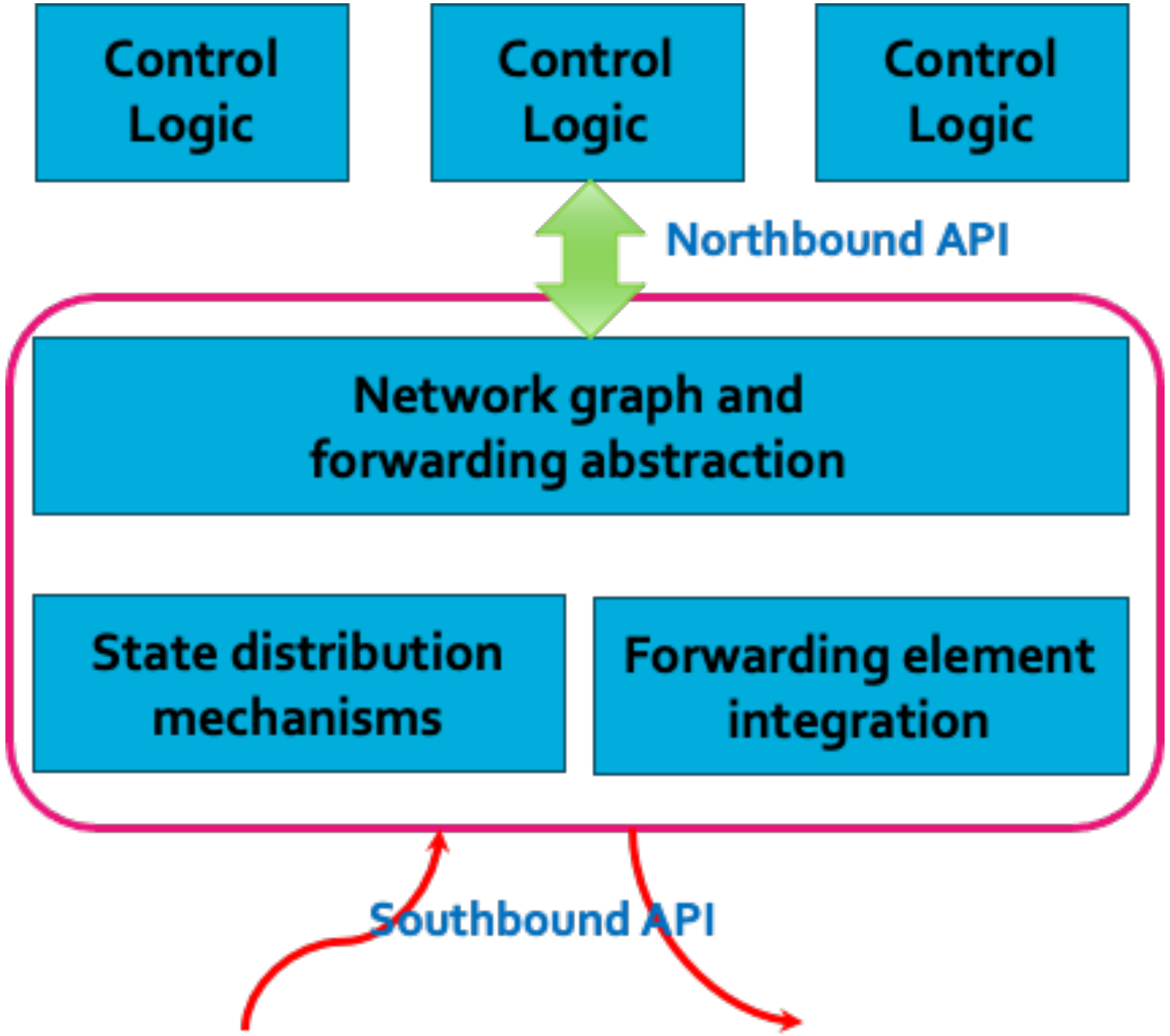
ABSTRACT

This whitepaper proposes OpenFlow: a way for researchers to run experimental protocols in the networks they use every day. OpenFlow is based on an Ethernet switch, with an internal flow-table, and a standardized interface to add and remove flow entries. Our goal is to encourage networking vendors to add OpenFlow to their switch products for deployment in college campus backbones and wiring closets. We believe that OpenFlow is a pragmatic compromise: on one hand, it allows researchers to run experiments on heterogeneous switches in a uniform way at line-rate and with high port-density; while on the other hand, vendors do not need to expose the internal workings of their switches. In addition to allowing researchers to evaluate their ideas in real-world traffic settings, OpenFlow could serve as a useful campus component in proposed large-scale testbeds like GENI. Two buildings at Stanford University will soon run OpenFlow networks, using commercial Ethernet switches and routers. We will work to encourage deployment at other schools; and We encourage you to consider deploying OpenFlow in your university network too.

to experiment with production traffic, which have created an exceedingly high barrier to entry for new ideas. Today, there is almost no practical way to experiment with new network protocols (e.g., new routing protocols, or alternatives to IP) in sufficiently realistic settings (e.g., at scale carrying real traffic) to gain the confidence needed for their widespread deployment. The result is that most new ideas from the networking research community go untried and untested; hence the commonly held belief that the network infrastructure has "ossified".

Having recognized the problem, the networking community is hard at work developing programmable networks, such as GENI [1] a proposed nationwide research facility for experimenting with new network architectures and distributed systems. These programmable networks call for programmable switches and routers that (using *virtualization*) can process packets for multiple isolated experimental networks simultaneously. For example, in GENI it is envisaged that a researcher will be allocated a *slice* of resources across the whole network, consisting of a portion of network links, packet processing elements (e.g. routers) and end-hosts; researchers program their slices to behave as

Controller Architecture



Events from switches

Topology changes,
Traffic statistics,
Arriving packets

Commands to switches

(un)install rules, Query statistics, Send packets

Data-Plane: Simple Packet Handling

Simple packet-handling rules

- Pattern: match packet header bits
- Actions: drop, forward, modify, send to the controller
- Priority: disambiguate overlapping patterns
- Counters: #bytes and #packets

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

Example SDN Applications

Public Demos

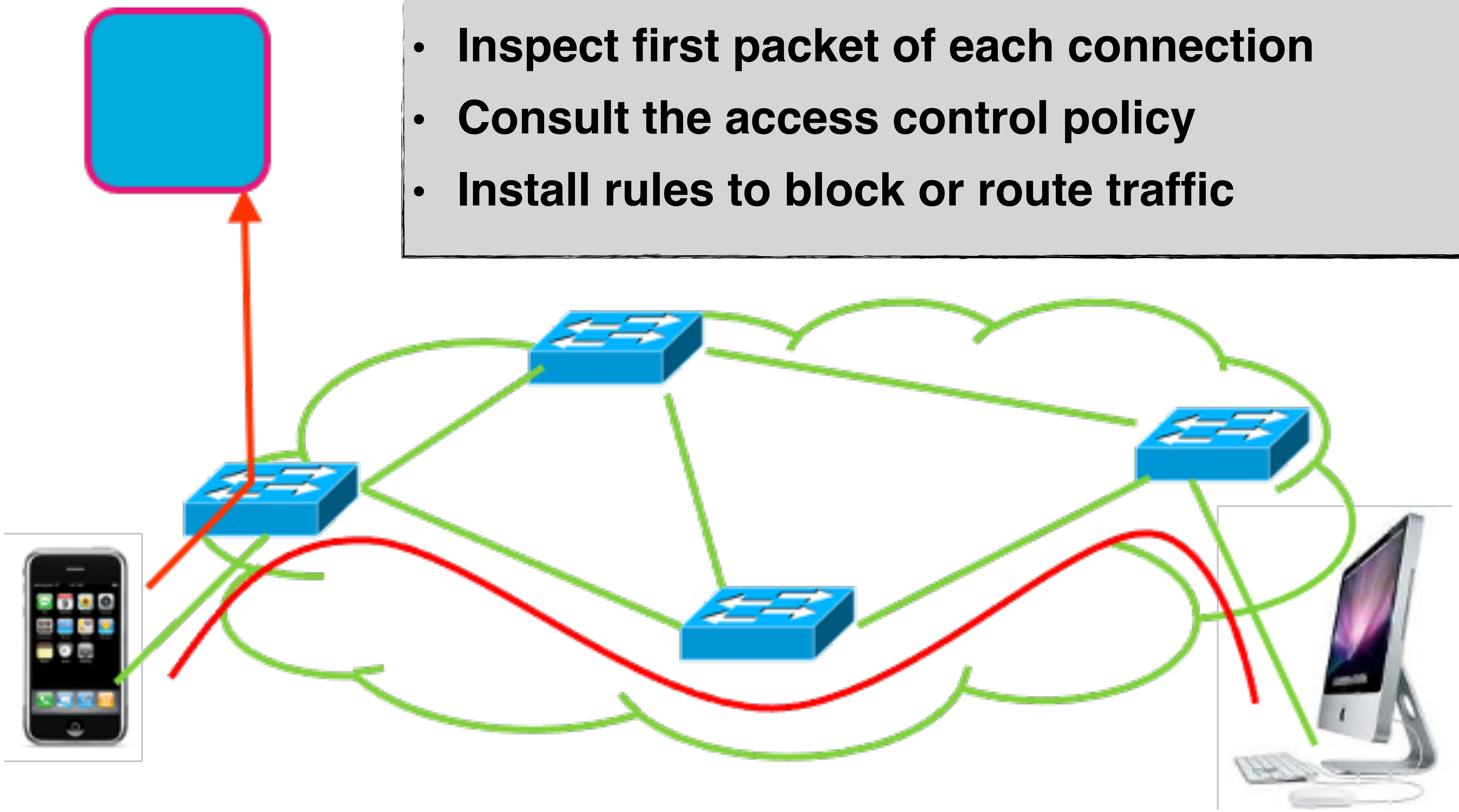
- Dynamic access control
- VM mobility/migration
- Network virtualization
- Load balancing
- Traffic Engineering

Commercial products

- Network virtualization: Nicira/VMWare, Azure, Google, CloudNaaS
- Traffic Engineering: Google's B4, Microsoft's SWAN

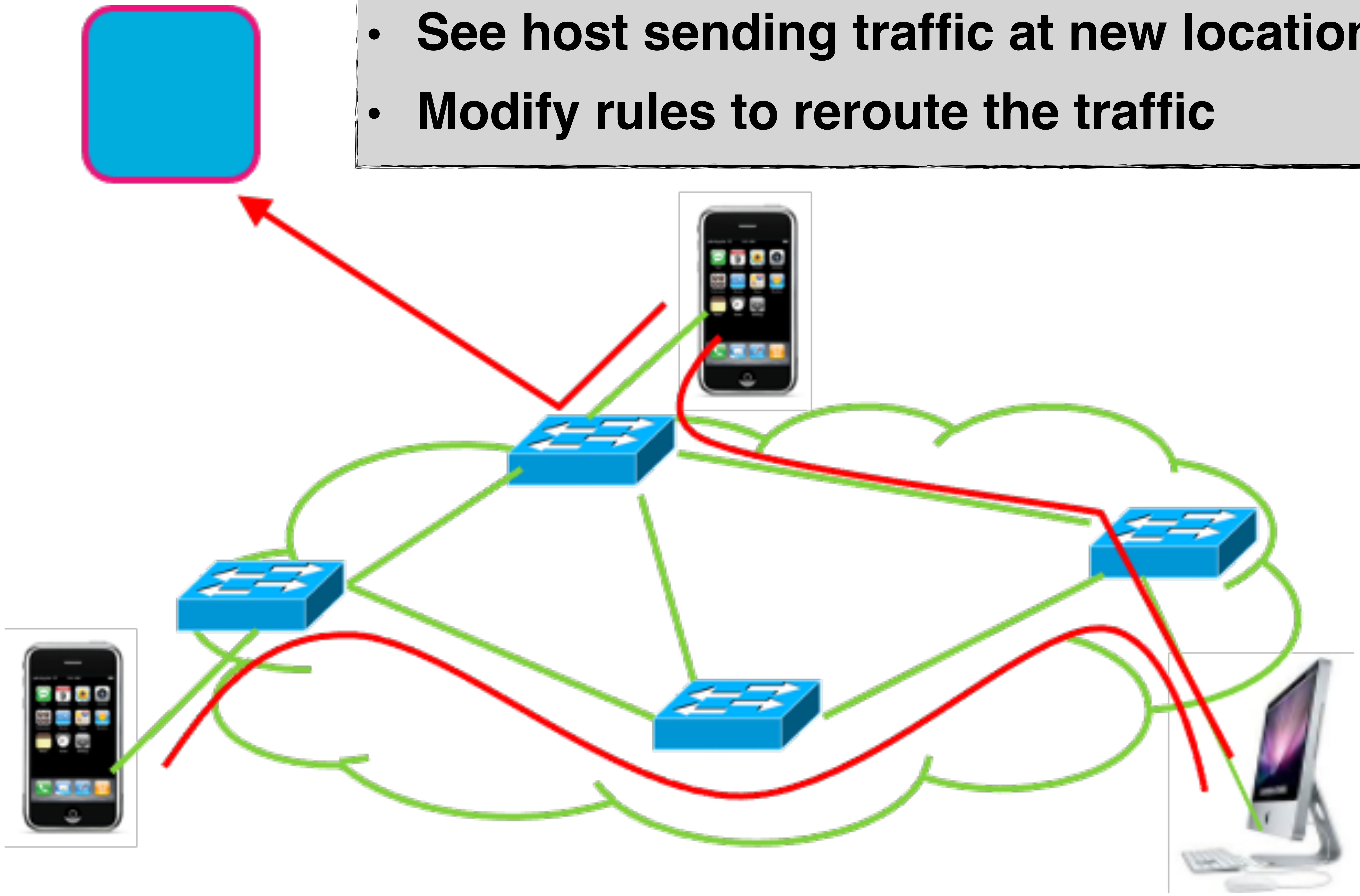
Example #1: Dynamic Access Control

- Inspect first packet of each connection
- Consult the access control policy
- Install rules to block or route traffic



Example #2: Seamless Mobility/Migration

- See host sending traffic at new location
- Modify rules to reroute the traffic



SDN/OpenFlow in the Wild

Open Networking Foundation

- Creating software-defined networking standards

Commercial OpenFlow Switches

- Cisco, HP, NEC, Quanta, Dell, IBM, Juniper, …

Controllers/Languages

- NOX, Beacon, Floodlight, Nettle, ONIX, POX
- Frenetic, MAPLE, Aspera, Pyretic

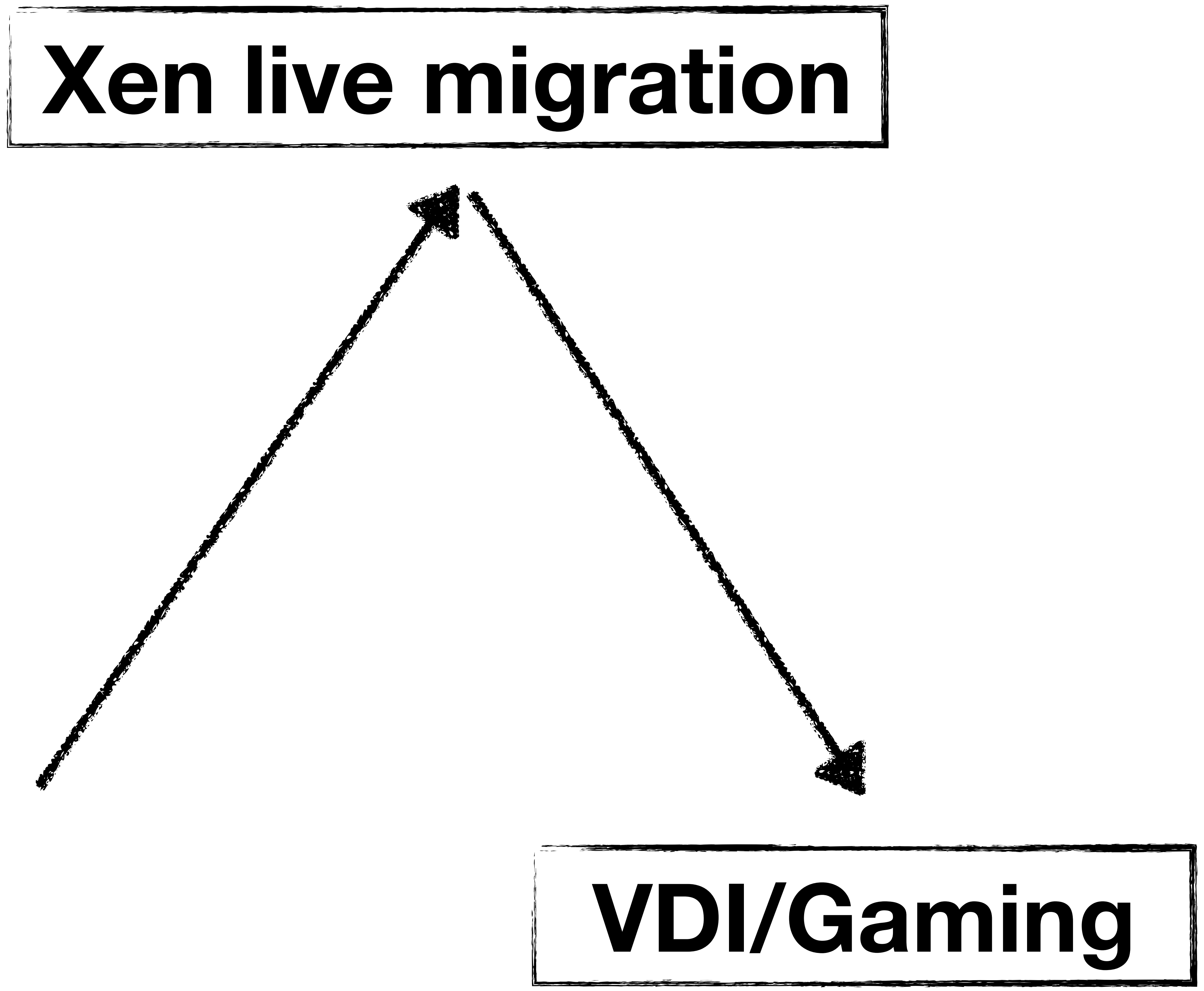
Network deployments

- Many campuses (including us) + commercial deployments

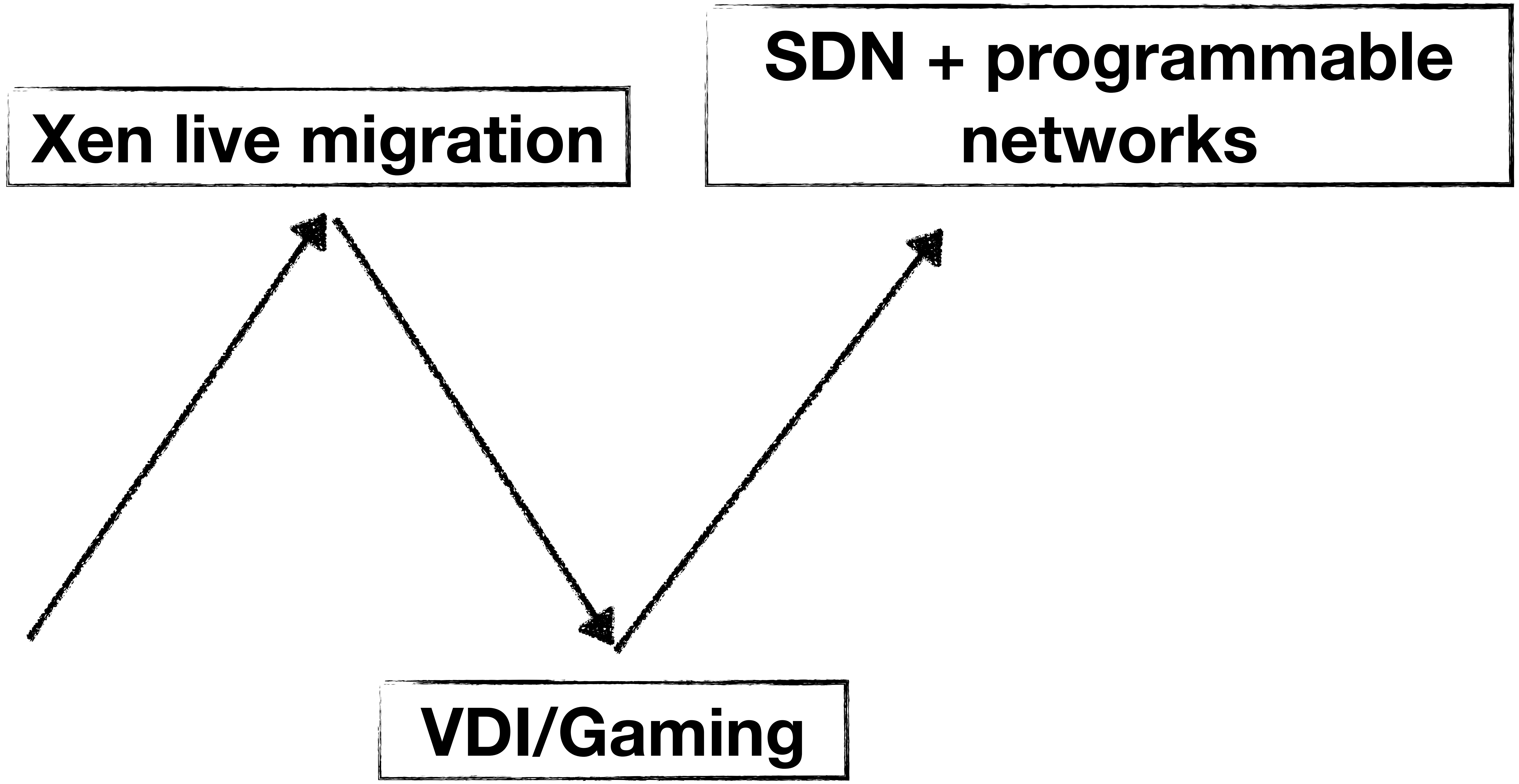
My Lessons

Xen live migration

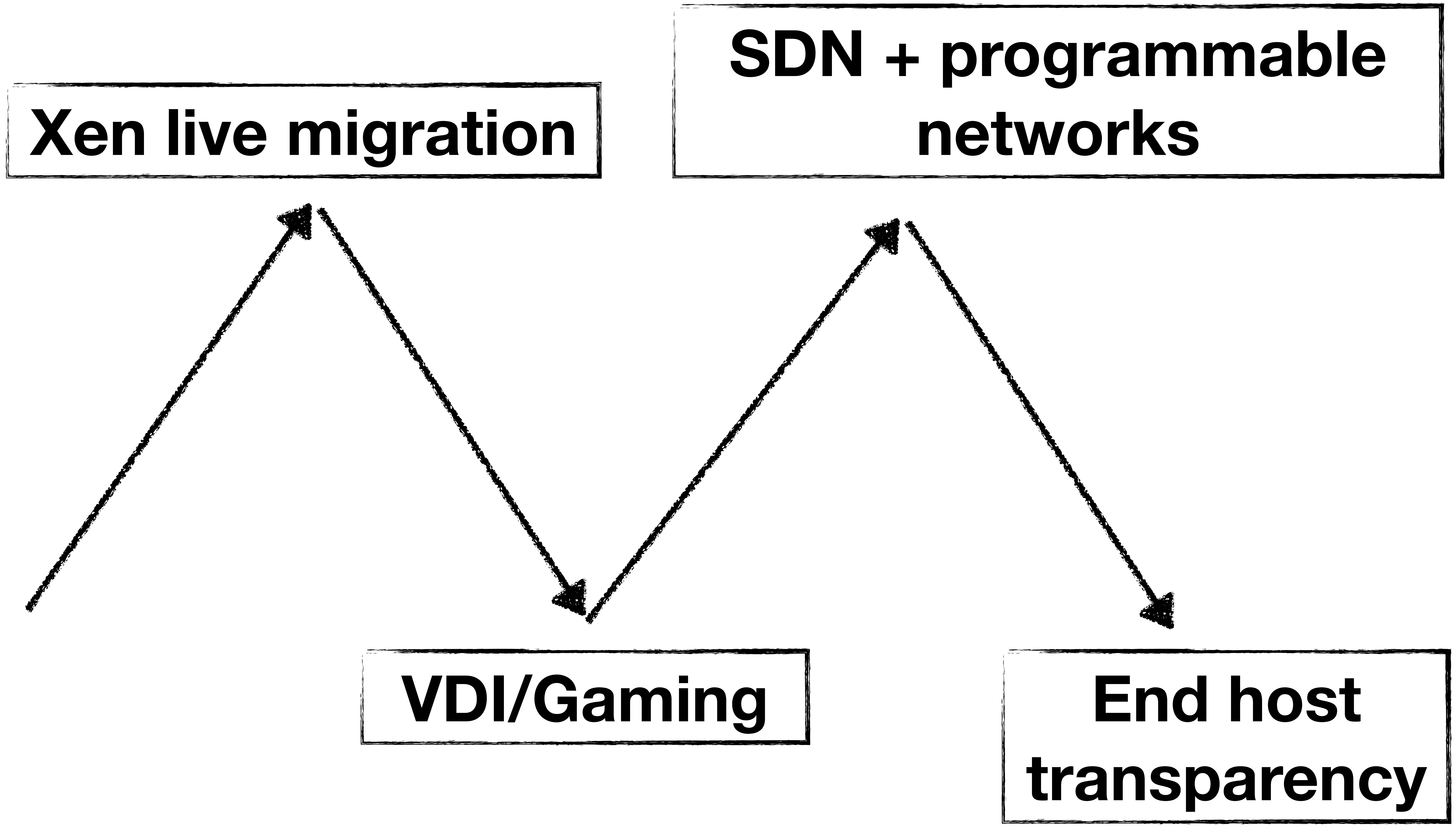
My Lessons



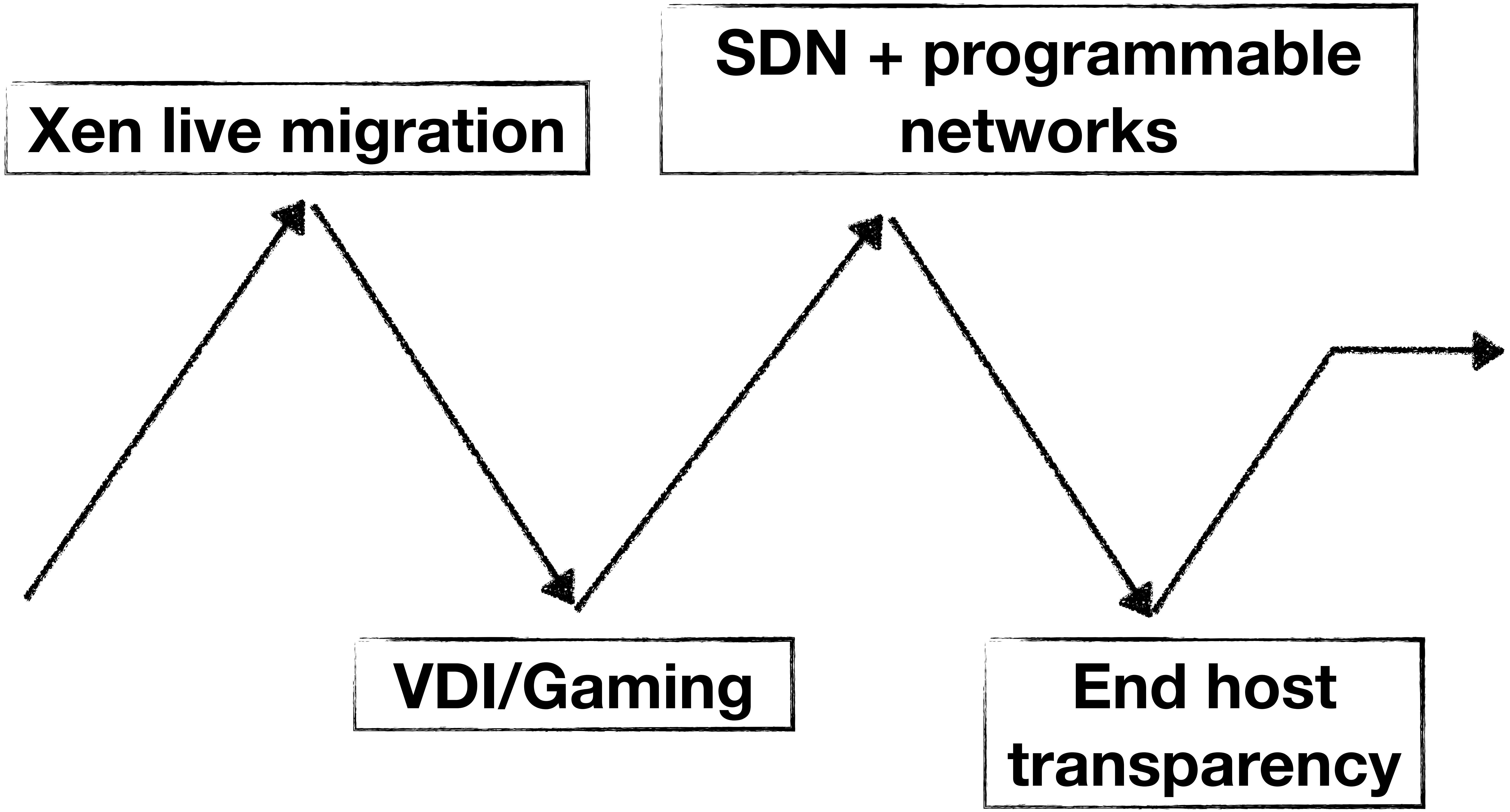
My Lessons



My Lessons


Xen live migration
SDN + programmable networks
VDI/Gaming
End host transparency

My Lessons



Xen live migration
SDN + programmable networks
VDI/Gaming
End host transparency

My Lessons

SDN + programmable

Three questions to ask

- #1: Who deploys the SDN platform?
- #2: Who writes the condition and action policies?
- #3: How much interaction do you need from host applications?

SDN requires elegant distributed system support

- #1: High availability
- #2: Strong consistency guarantee

transparency

IP Router v.s. Ethernet Switch

		IP Router	Ethernet Switch
HW	Exec. Engine	Switching Fabric + Networking processor	Switching Fabric
	Memory	SRAM + Large TCAM	SRAM + TCAM
SW	Layering	Layer 3	Layer 2
	Packet Manipulation	Fragmentation and Reassembly; TTL/ Checksum update	N/A
	Packet Forwarding	IP address	Ethernet address
	Routing	Intra-/Inter-domain	N/A
	NAT	Yes	N/A
	Error Handling	Speak the ICMP protocol	N/A

Terminology

1. Host
2. NIC
3. Multi-port I/O bridge
4. Protocol
5. RTT
6. Packet
7. Header
8. Payload
9. BDP
10. Baud rate
11. Frame/Framing
12. Parity bit
13. Checksum
14. Ethernet
15. MAC
16. (L2) Switch
17. Broadcast
18. Acknowledgement
19. Timeout
20. Datagram
21. TTL
22. MTU
23. Best effort
24. (L3) Router
25. Subnet mask
26. CIDR
27. Converge
28. Count-to-infinity
29. Line card
30. Network processor
31. Gateway
32. Private network
33. IPv6
34. Multicast
35. IGMP
36. SDN

Principle

1. Layering
2. Minimal States
3. Hierarchy

Technique

1. NRZ Encoding
2. NRZI Encoding
3. Manchester Encoding
4. 4B/5B Encoding
5. Byte Stuffing
6. Byte Counting
7. Bit Stuffing
8. 2-D Parity
9. CRC
10. MAC Learning
11. Store-and-Forward
12. Cut-through
13. Spanning Tree
14. CSMA/CD
15. Stop-and-Wait
16. Sliding Window

16. Fragmentation and Reassembly
17. Path MTU discovery
18. DHCP
19. Subnetting
20. Supernetting
21. Longest prefix match
22. Distance vector routing (RIP)
23. Link state routing (OSPF)
24. Boarder gateway protocol (BGP)
25. Network address translation (NAT)

Summary

Today's takeaways

#1: Private IP and IPv6 can solve the address scarcity issue

#2: IP multicast enables better bandwidth utilization, reduces the host/router processing, and allows flexible group service

#3: SDN enables application awareness in the IP layer

Next lecture

- Transport introduction