Internet of Things

Module 4: Wired

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Is wireless enough?

- Wireless communication isn't appropriate for long distances, low-latency, high reliability, high-bandwidth, etc.

Office building IoT

Mobile and infrastructure backhaul

Service Infrastructure

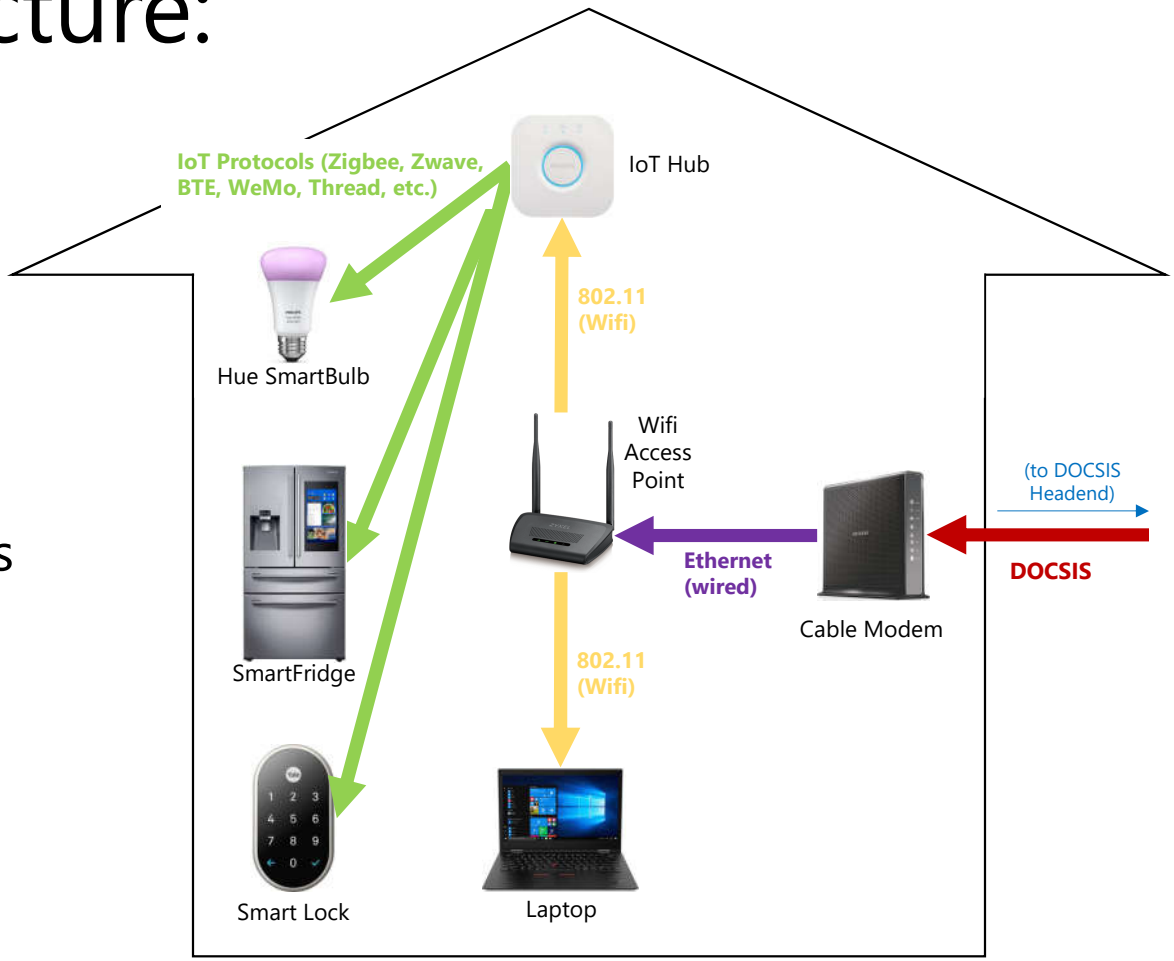
Wired Networking Introduces New Challenges

- First line of defense from public internet
- Larger sizes
- Distributed over wide areas
- High bandwidth, reliability; low-latency

- We will learn technologies used to solve these challenges
  - How to configure and use them

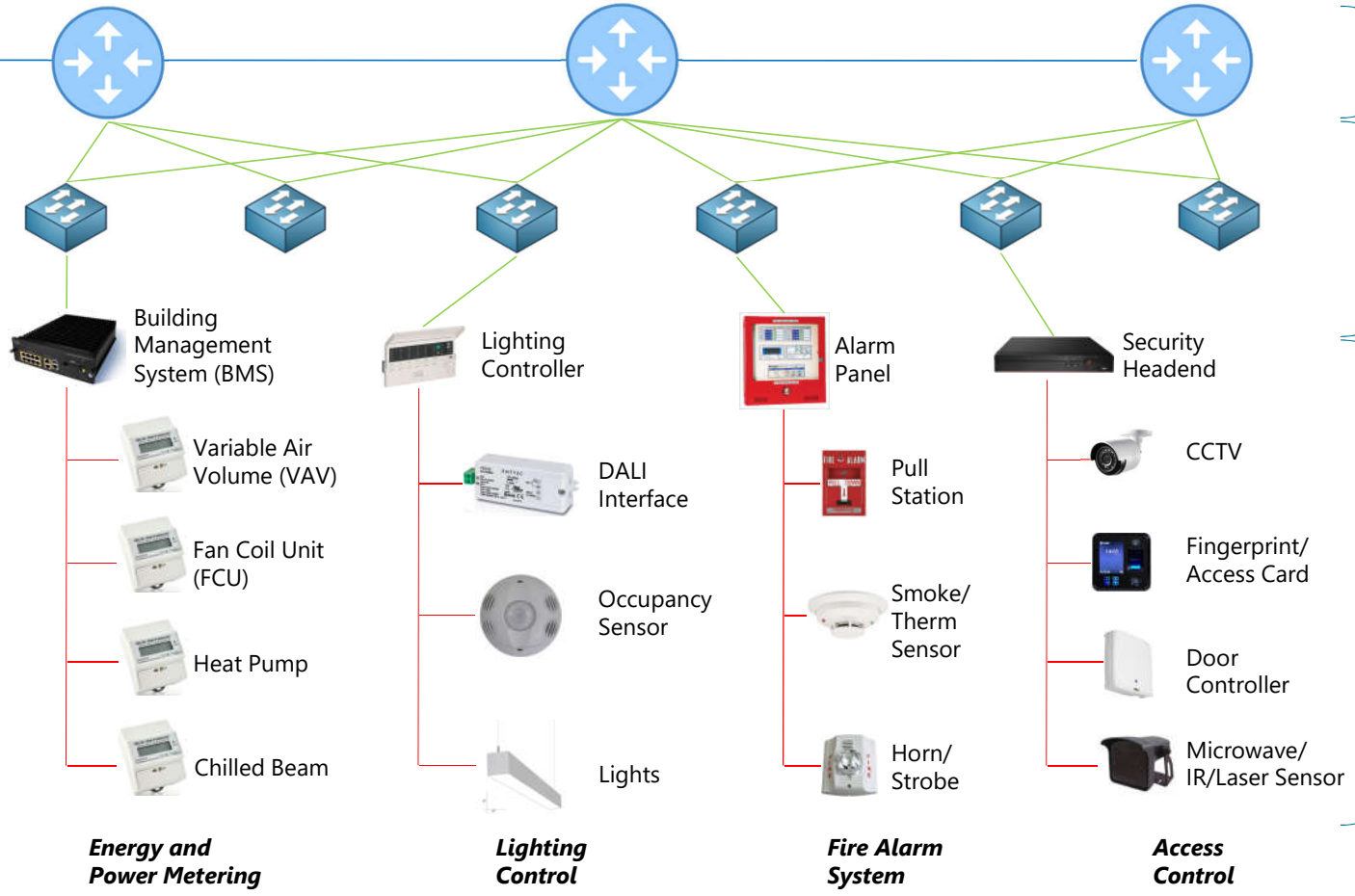
Reference Architecture: Smart Home

- Controlled devices connected to "gateway"/"hub"
- Currently, few accepted industry standards
  - Poor documentation hinders independent development
- Poor release/patching practices lead to security issues
  - Estimated 87% of devices vulnerable

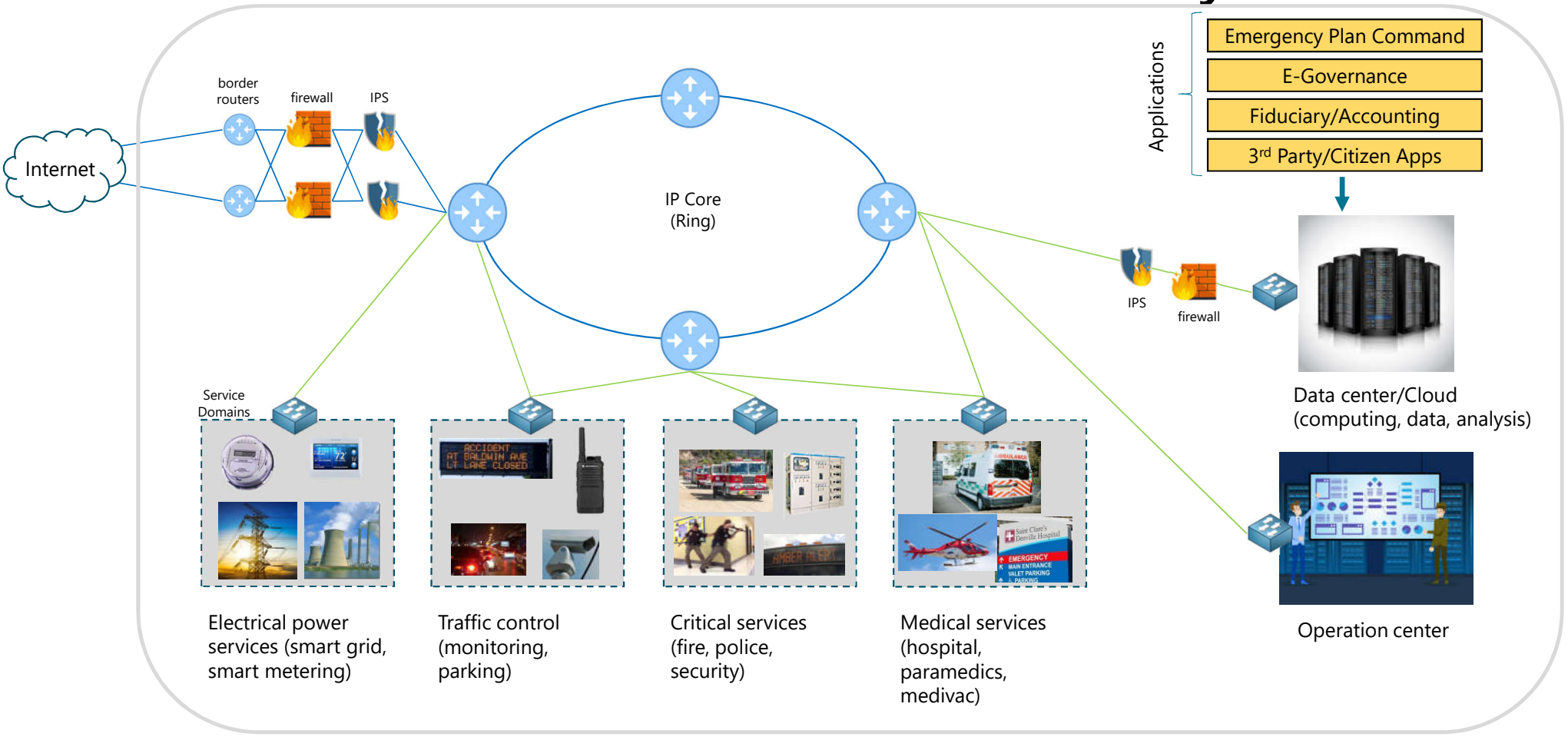


Reference Architecture: Smart Building

Building Operations Center



Reference Architecture: Smart City

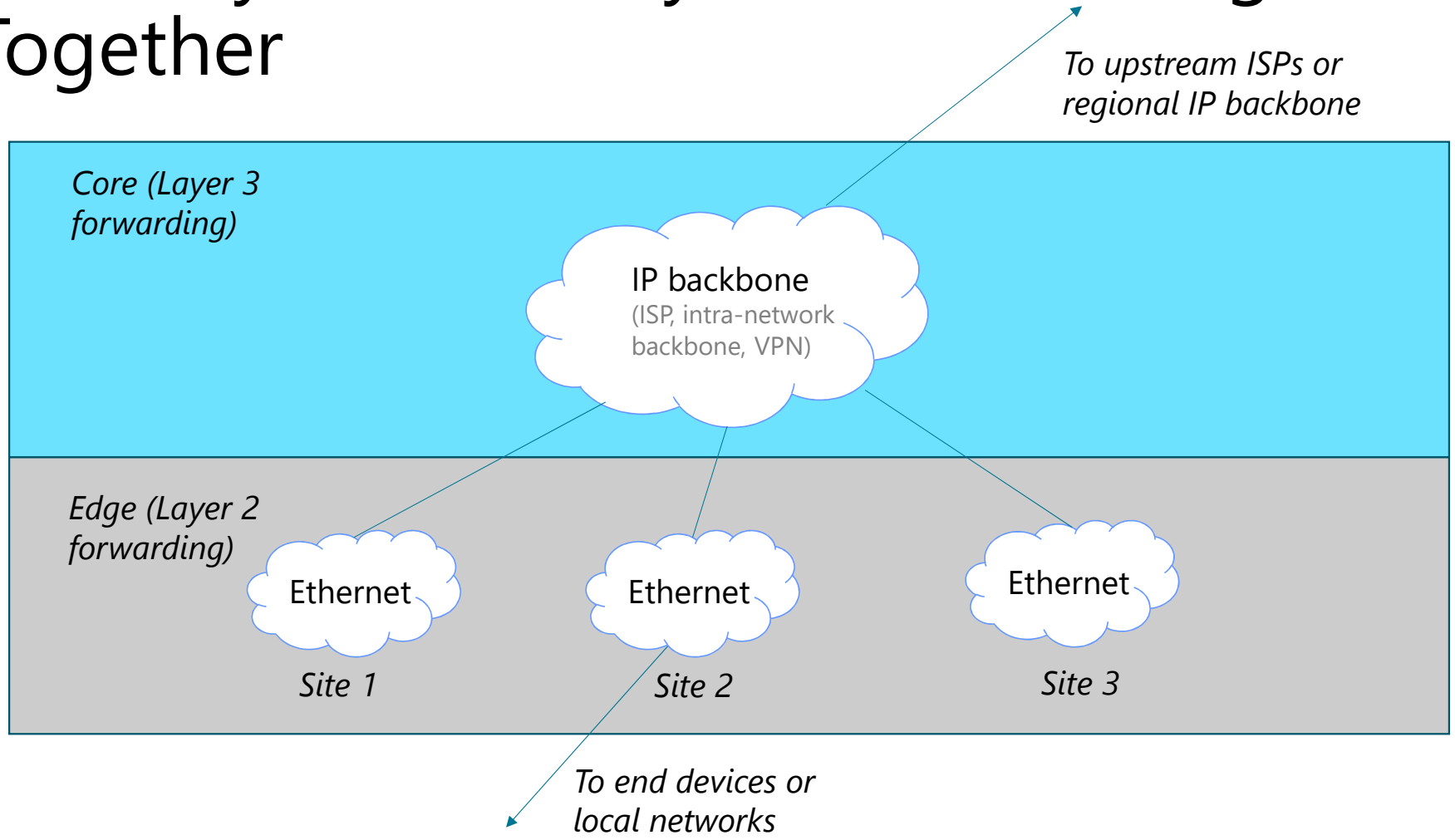


[title: Enterprise Networking]

[week X: video X]

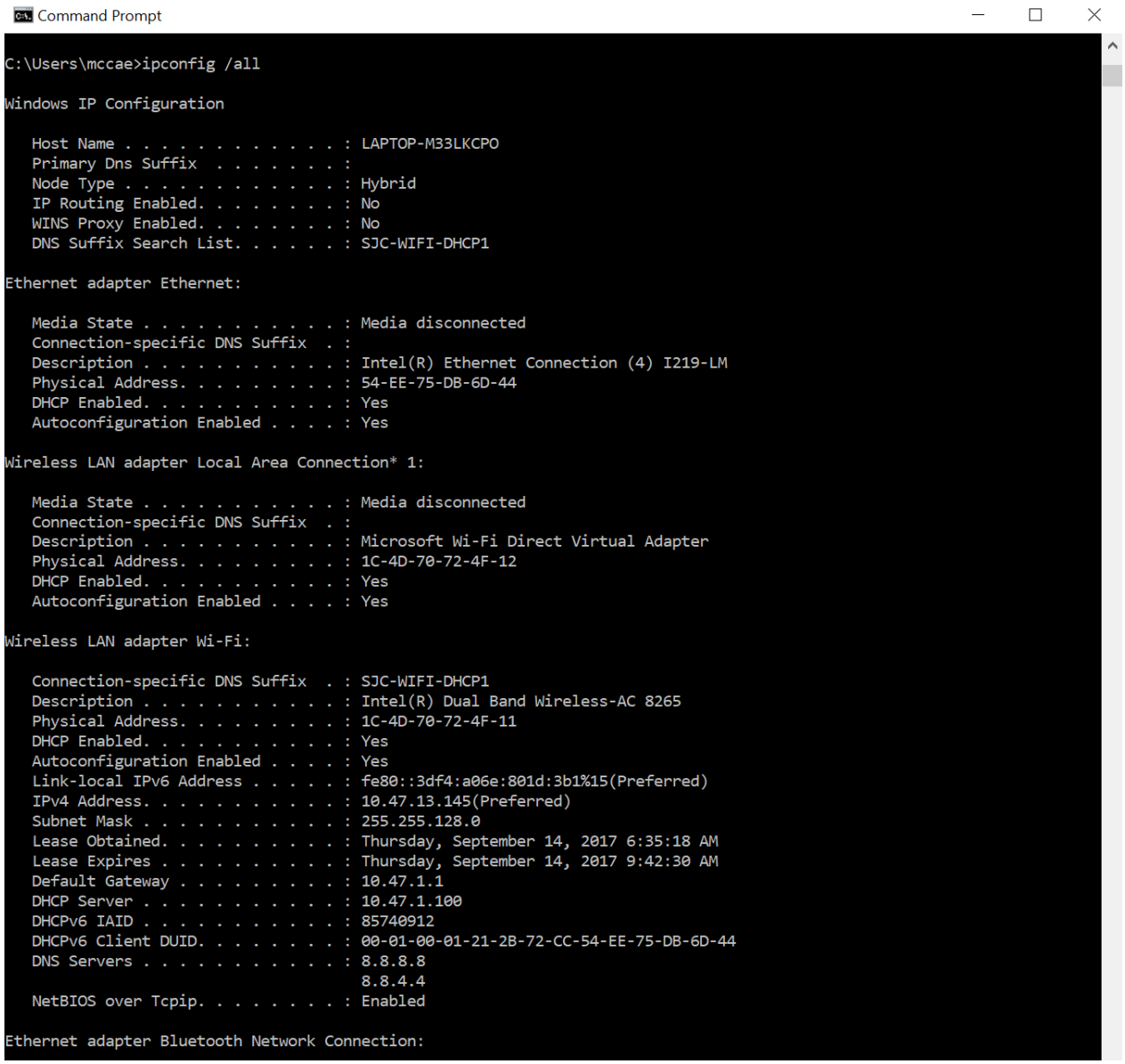
[status: done]

How Layer 2 and Layer 3 Networking Fit Together

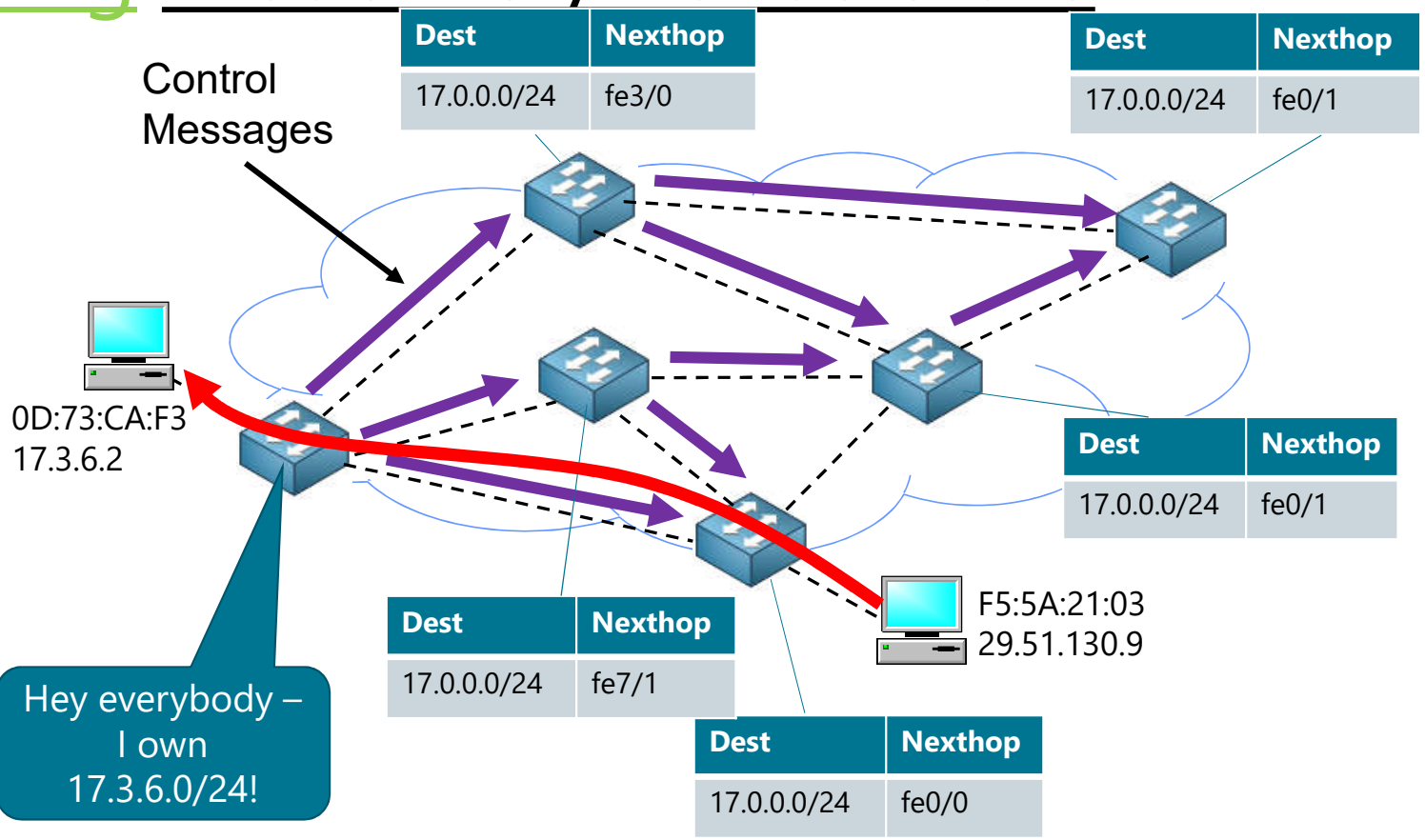


Ethernet (Layer 2) vs. IP (Layer 3) Routing

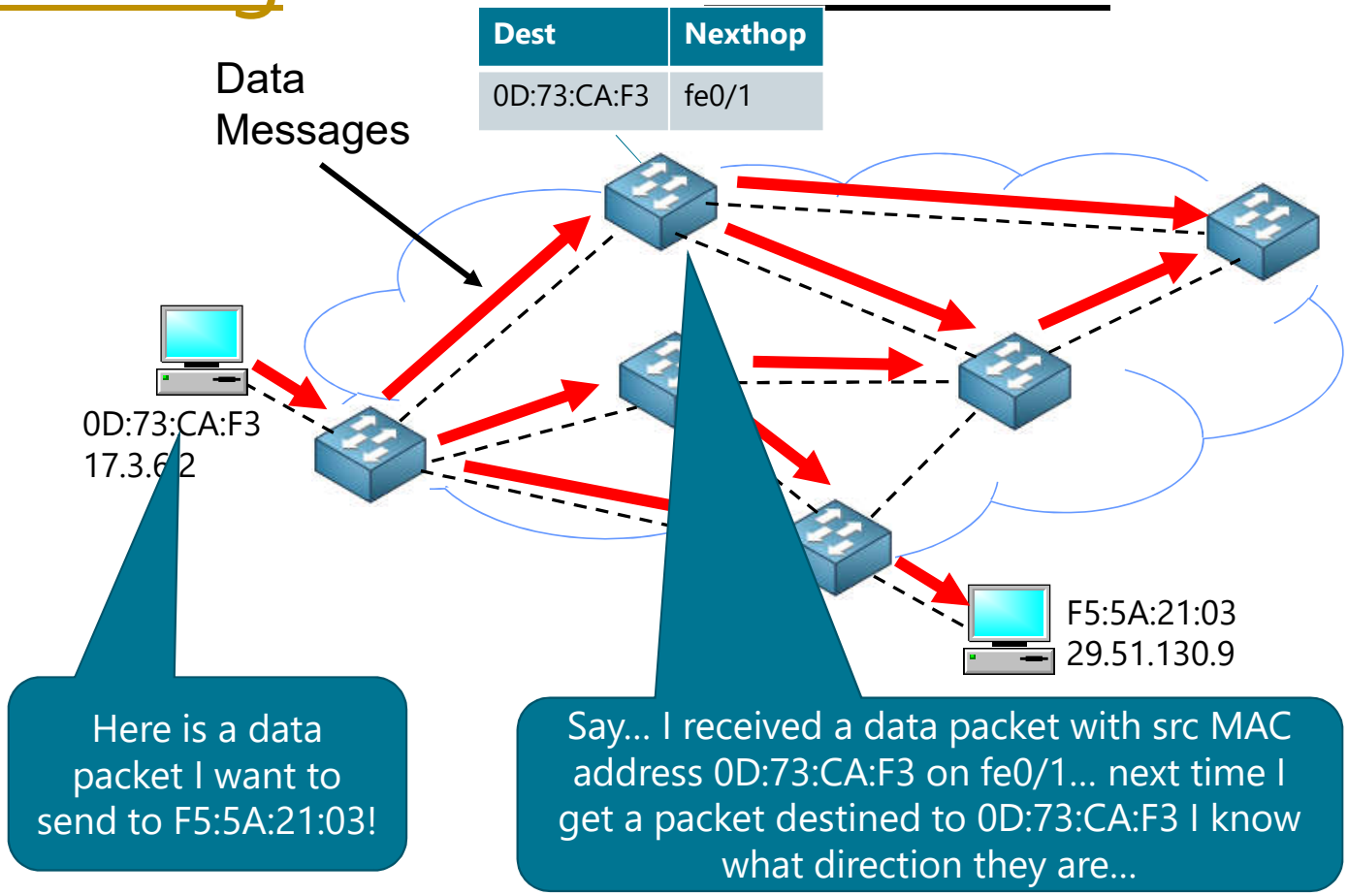
- Ethernet is "plug and play"
  - Easy to build networks
  - May optionally configure ACLs, SSIDs (wireless), spanning tree properties, etc.
- Each host assigned a topology independent MAC address
  - E.g., 00-14-22-01-23-45
- Uses "dumb" flooding (broadcast) to get packets where they need to go
  - Less efficient than link-state (unicast)


Command Prompt
C:\Users\mccae>ipconfig /all
Windows IP Configuration
Host Name . . . . . : LAPTOP-M33LKCPO
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : SJC-WIFI-DHCP1
Ethernet adapter Ethernet:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) Ethernet Connection (4) I219-LM
Physical Address. . . . . : 54-EE-75-DB-6D-44
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Wireless LAN adapter Local Area Connection* 1:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
Physical Address. . . . . : 1C-4D-70-72-4F-12
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Wireless LAN adapter Wi-Fi:
Connection-specific DNS Suffix . : SJC-WIFI-DHCP1
Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
Physical Address. . . . . : 1C-4D-70-72-4F-11
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::3df4:a06e:801d:3b1%15(Preferred)
IPv4 Address. . . . . : 10.47.13.145(Preferred)
Subnet Mask . . . . . : 255.255.128.0
Lease Obtained. . . . . : Thursday, September 14, 2017 6:35:18 AM
Lease Expires . . . . . : Thursday, September 14, 2017 9:42:30 AM
Default Gateway . . . . . : 10.47.1.1
DHCP Server . . . . . : 10.47.1.100
DHCPv6 IAID . . . . . : 85740912
DHCPv6 Client DUID. . . . . : 00-01-00-01-21-2B-72-CC-54-EE-75-DB-6D-44
DNS Servers . . . . . : 8.8.8.8
8.8.4.4
NetBIOS over Tcpip. . . . . : Enabled
Ethernet adapter Bluetooth Network Connection:

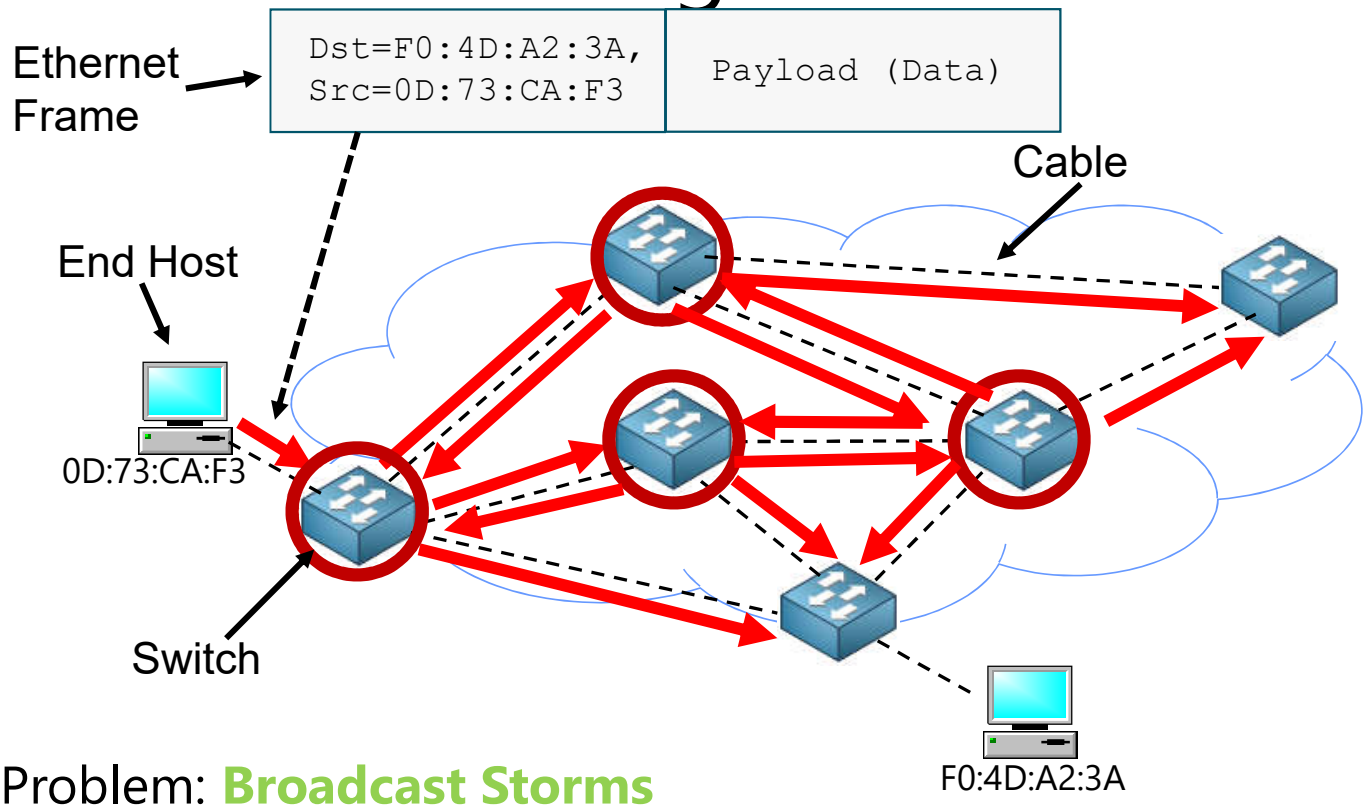
L2 Switching vs. L3 Routing: Routing Proactively Builds State



L2 Switching vs. L3 Routing: Switching Relies on Broadcast

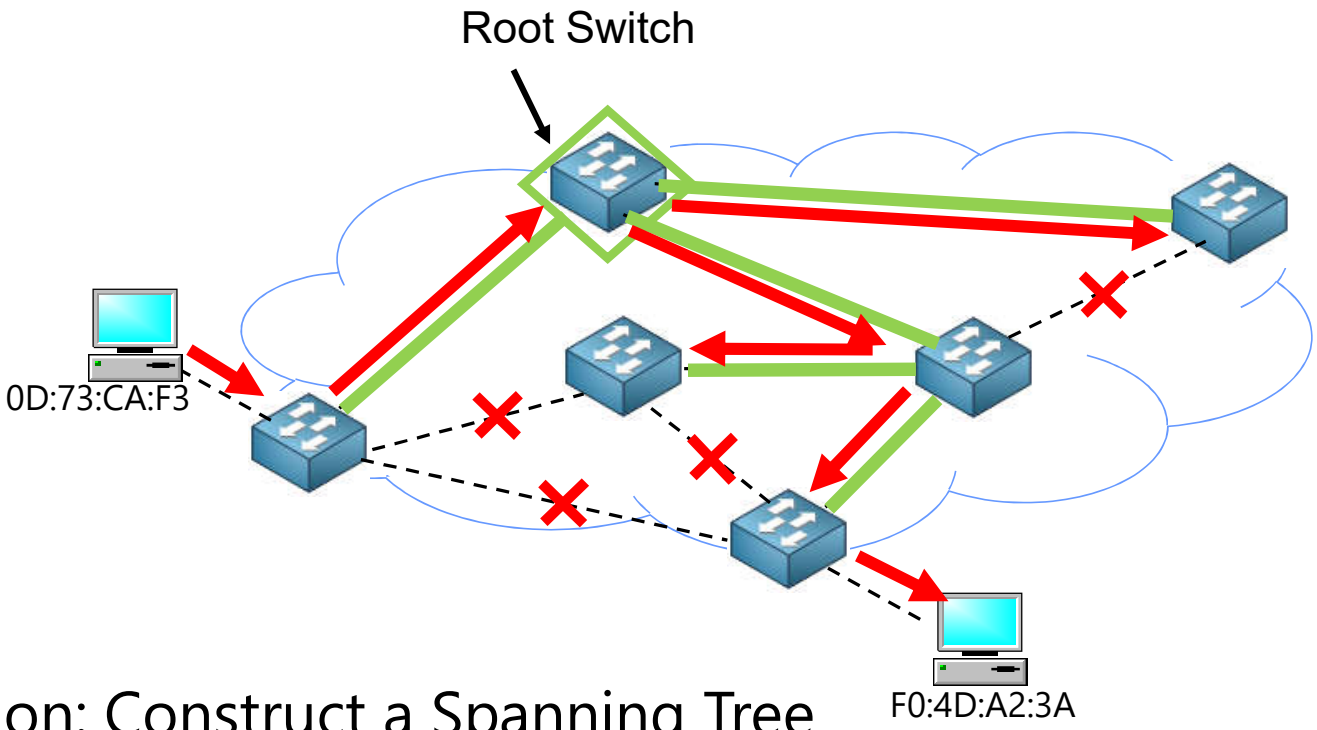


Ethernet Forwarding



- Problem: **Broadcast Storms**
- How to flood with stateless switches?

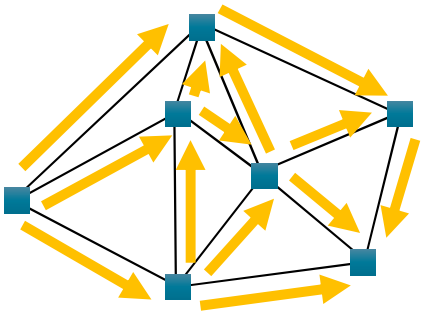
Ethernet Forwarding



- Solution: Construct a Spanning Tree
  - Elect a "root" switch; Root-facing ports are active, others disabled

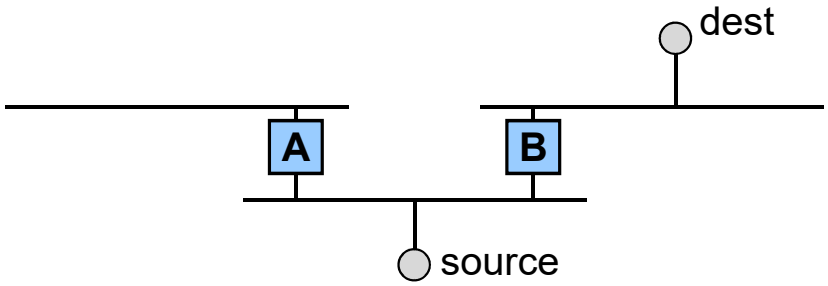
Avoiding Flooding

- Flooding packets throughout network introduces problems
  - Scalability, privacy, resource isolation, lack of access control
- Scalability requirement is growing quickly with advent of IoT
  - Large enterprises: 50k end hosts
  - Data centers: 100k servers, 5k switches
  - Metro-area Ethernet: over 1M subscribers



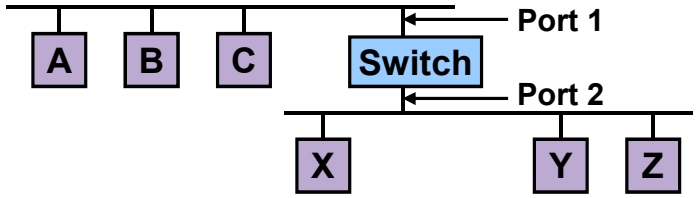
Avoiding Flooding

- Suppose source sends a frame to a destination
  - Which links should a frame be forwarded on?
- Trivial algorithm
  - Forward all frames on all (other) LANs
  - Problem: Wastes bandwidth to flood everything everywhere
- Better approach: "Learn" where hosts are
  - Listen to broadcasts, remember which direction source is



Learning Switches

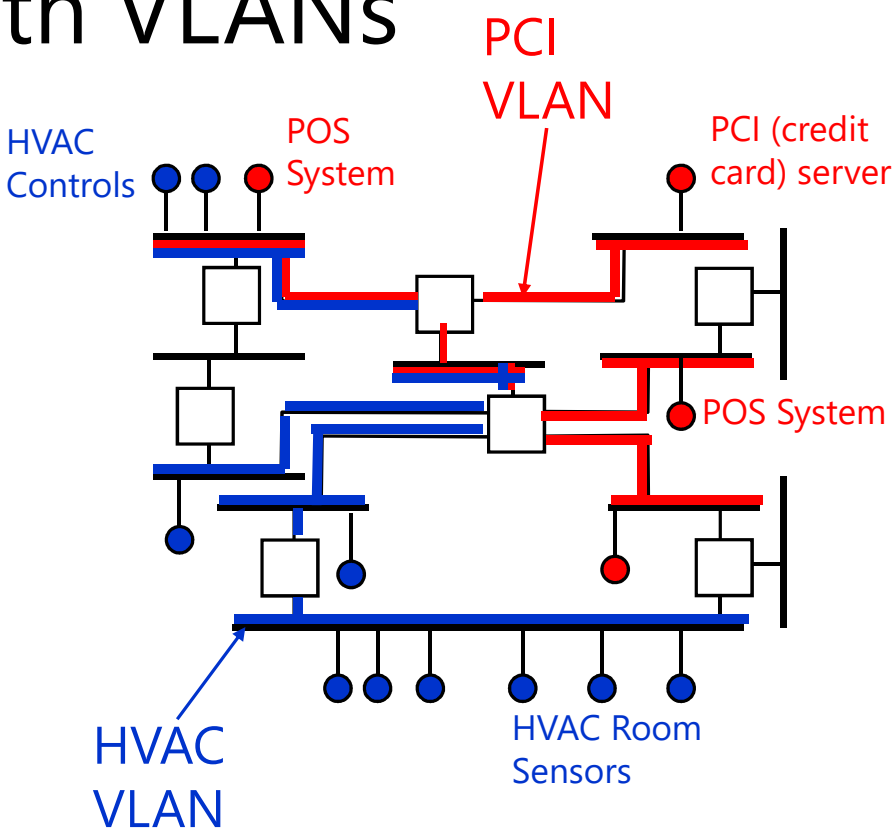
- Switch learns table entries based on source address
  - When receive frame from host on port 1, add host to list of hosts on port 1
  - Time out entries to avoid stale entries
- Table is an "optimization"
  - Improves performance but is not mandatory
- Always forward broadcast frames



Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

Scaling Ethernet with VLANs

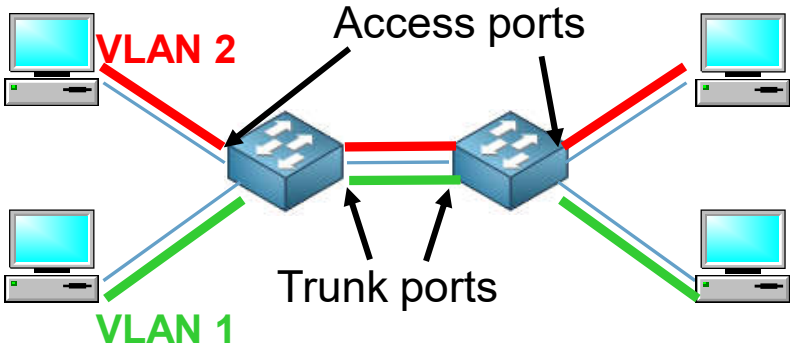
- Divide up hosts into logical groups called **VLANs**
  - Like virtual machines, but for LANs (creates "virtual networks")
  - VLANs isolate traffic at layer 2
- Each VLAN corresponds to IP subnet, single broadcast domain
- Ethernet packet headers have VLAN tag
- Switches forward packet only on subnets on corresponding VLAN



Virtual LANs

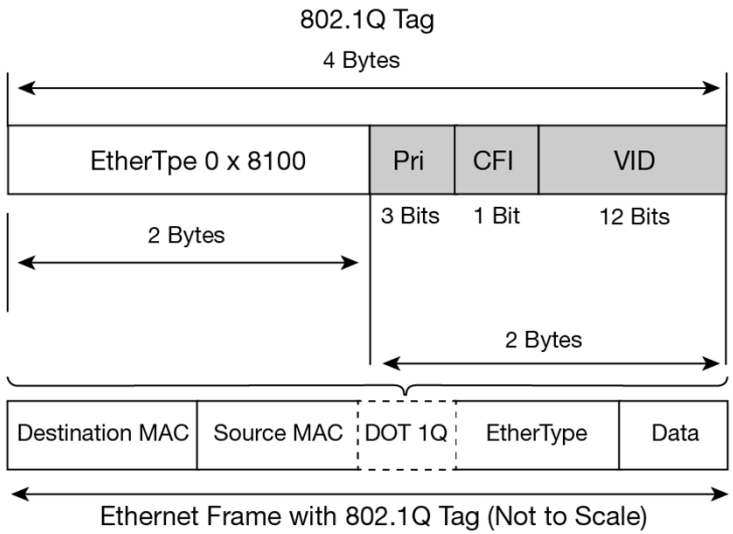
- Downsides of VLANs
  - Are (usually) manually configured, complicates network management
  - Hard to seamlessly migrate across VLAN boundaries due to addressing restrictions
- Upsides of VLANs
  - Limits scope of broadcasts
  - Logical separation improves isolation, security
  - Can change virtual topology without changing physical topology
    - E.g., used in data centers for VM migration

How VLANs Are Configured



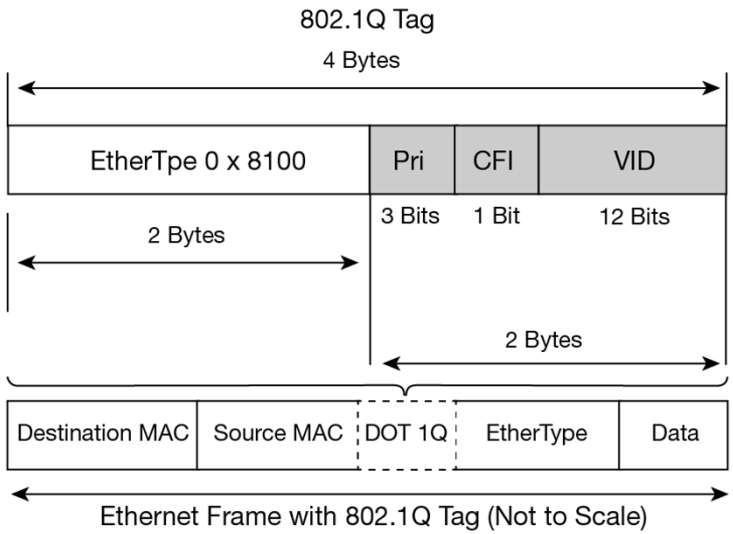
- LAN switches can configure ports as access ports or trunk ports
  - **Access ports** append tags on packets
  - **Trunk ports** can multiplex several VLANs
- VLAN membership typically encoded (statically) in access switch's configuration file

How VLANs Are Implemented



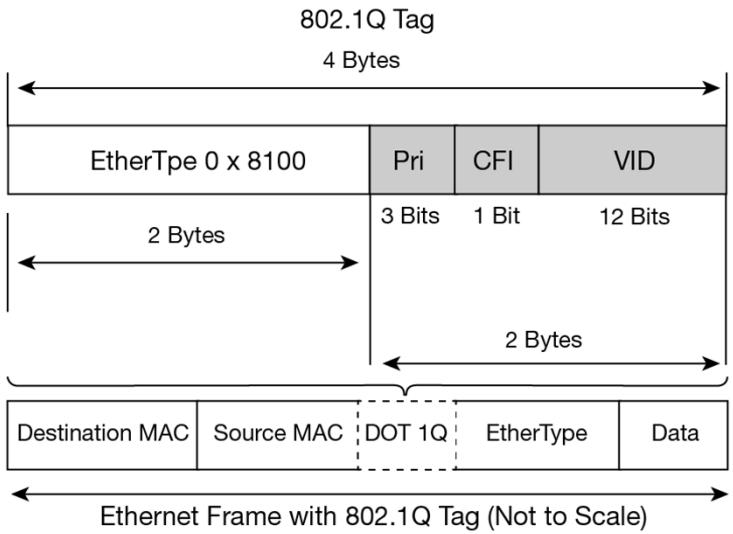
- Packets are annotated with 12-bit **VLAN tags**
  - Defined in VLAN specification (IEEE 802.1Q)
  - Up to **4096** VLANs can be encapsulated within a single VLAN ID

How VLANs Are Implemented



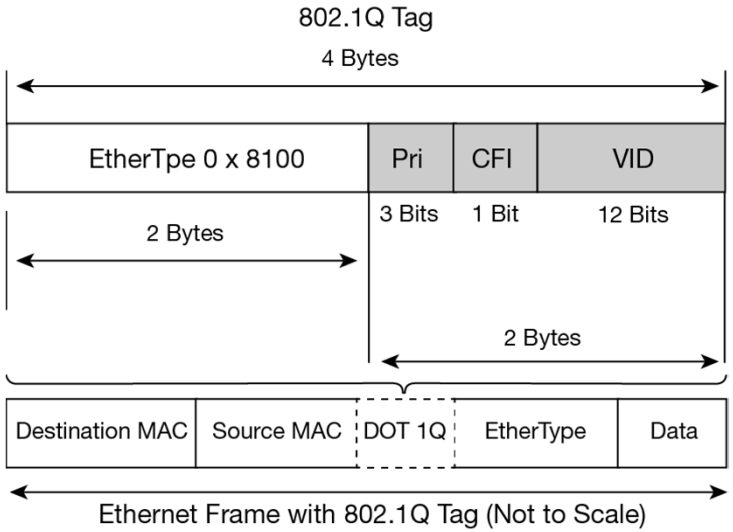
- 802.1Q defines a few other fields too
  - **Ethertype** of 0x8100 instructs switch to decode next 2 bytes as VLAN header
  - 3 bits of priority (like IP ToS)
  - 1 bit for compatibility with token ring

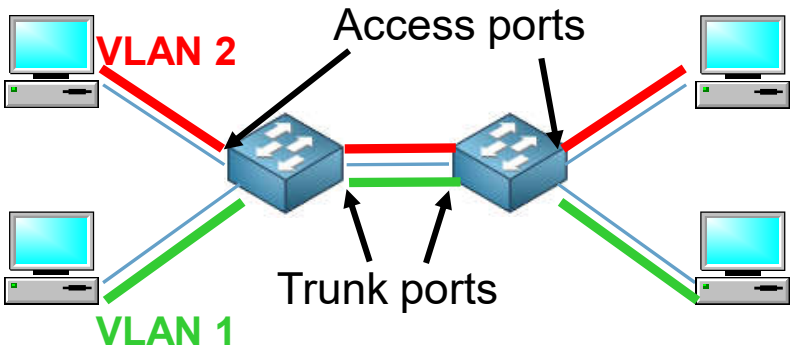
How VLANs Are Implemented



- What if 4096 VLANs aren't enough?
  - **QinQ** (802.1ad) – can encapsulate VLANs within VLANs by stacking VLAN tags
  - Up to 4096 VLANs can be multiplexed within a single VLAN ID→ 4096^2 combinations

How VLANs Are Implemented

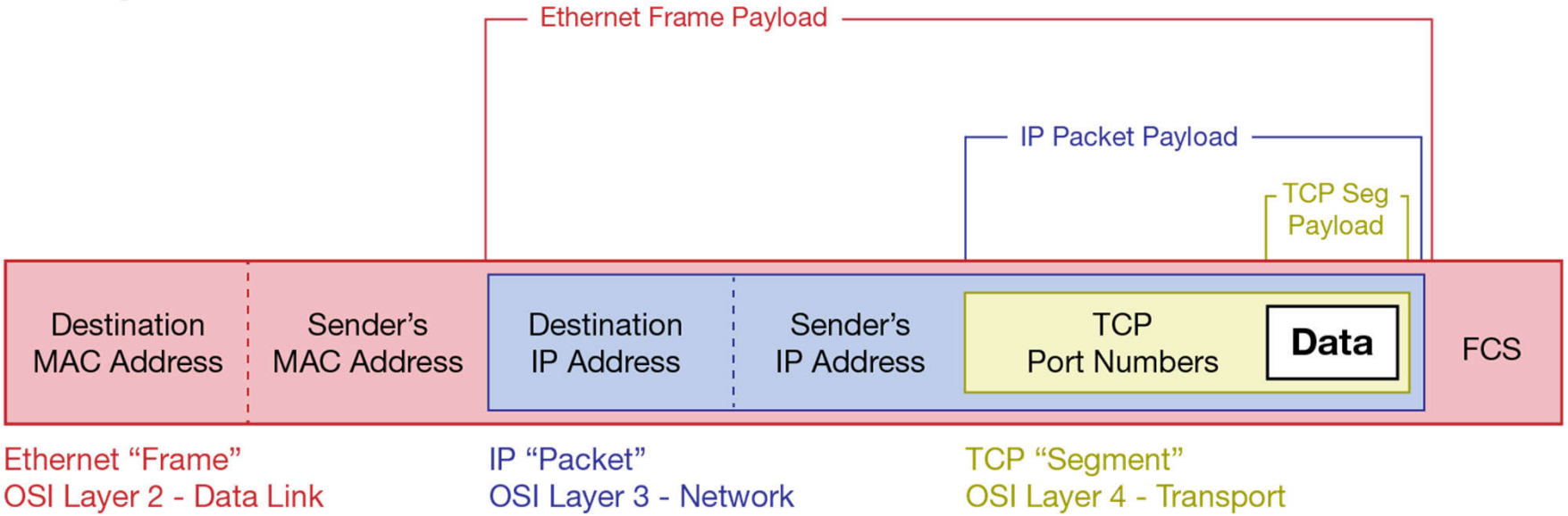




- **Native mode**
  - IEEE likes to make specs that are backwards compatible
  - 802.1Q allows trunk ports to carry both tagged and untagged frames
  - Frames with no tags are said to be part of the switch's **native VLAN**

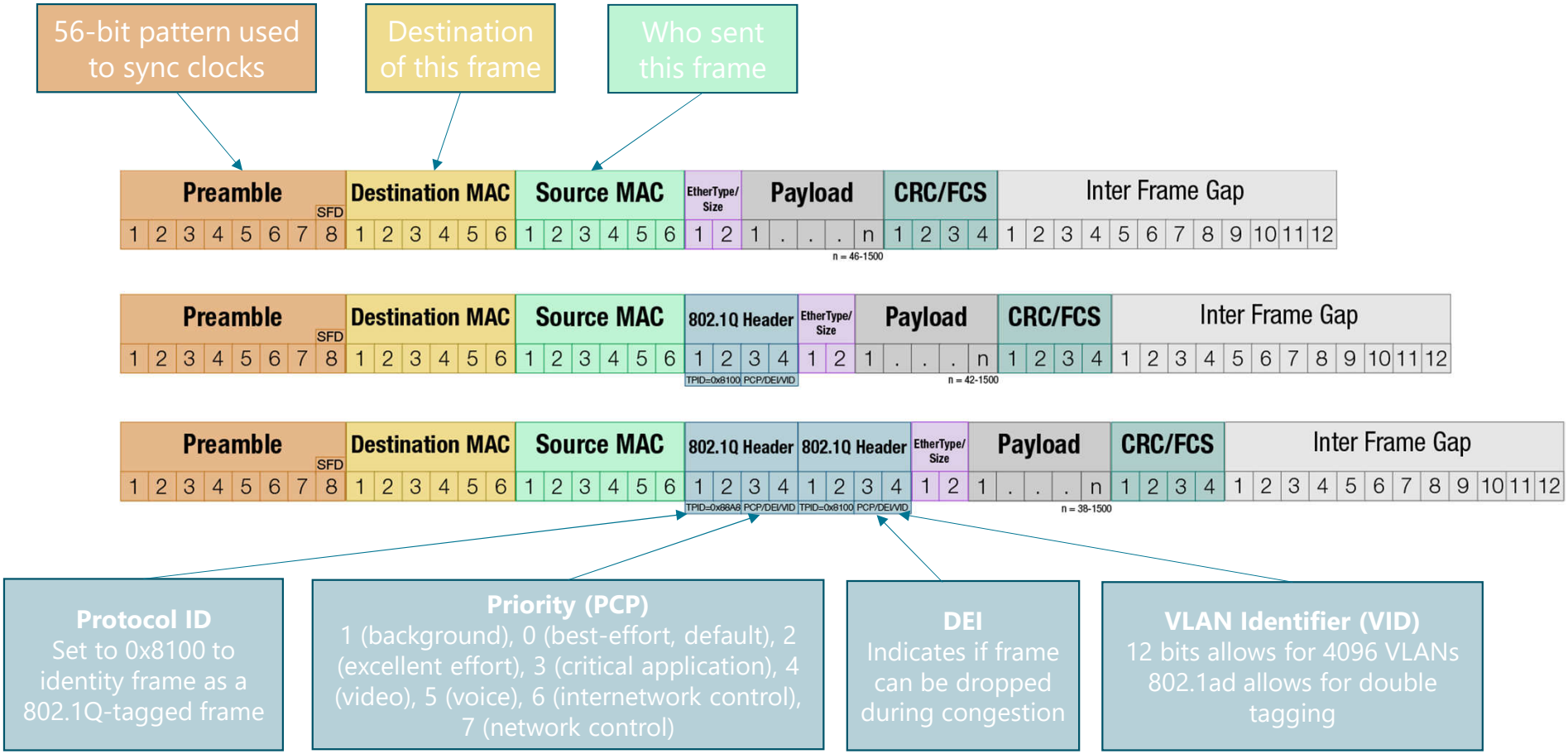
Encapsulation

Encapsulation

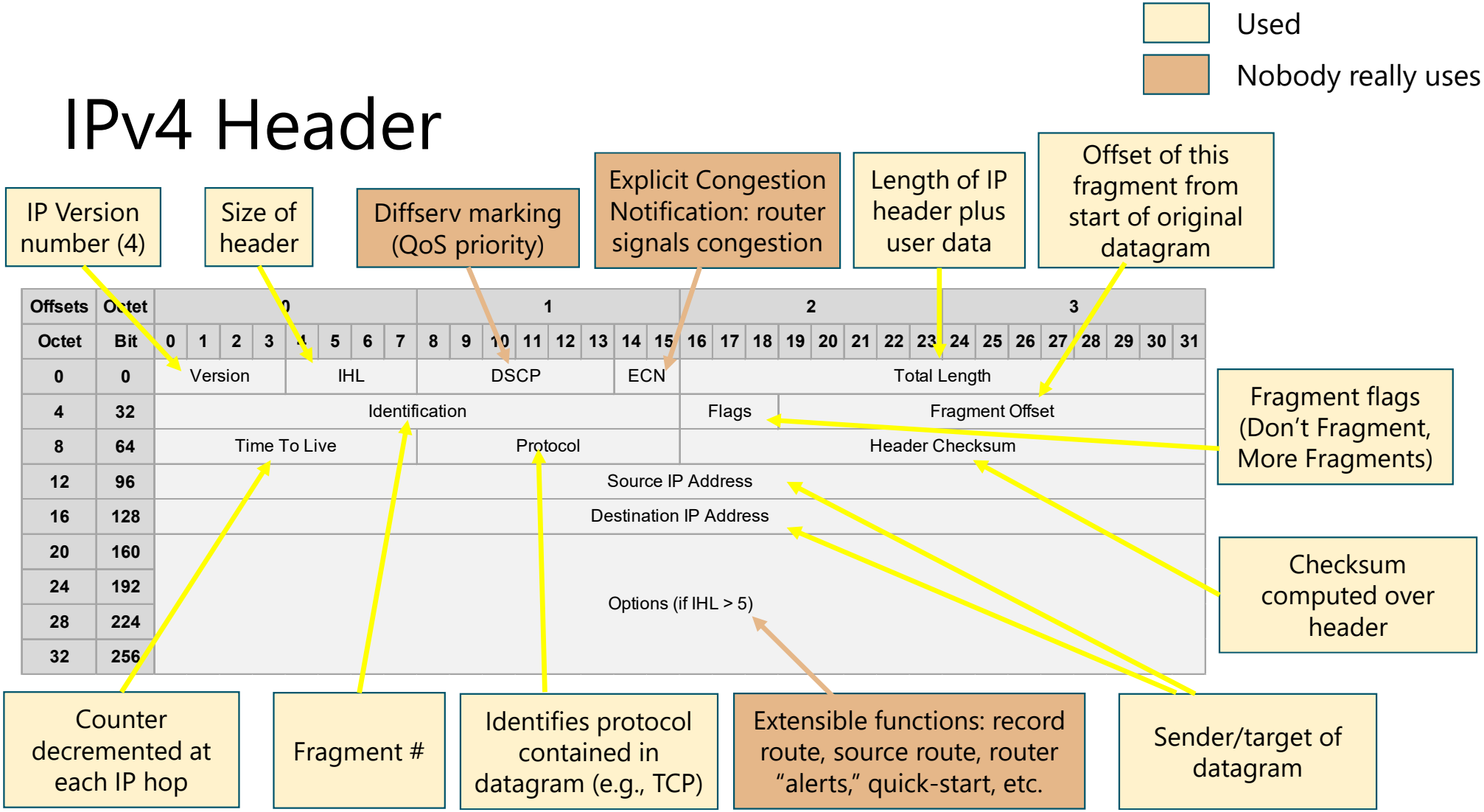


- Each layer of protocol stack encapsulates data passed to it
- Each forwarding layer inspects data only at that encapsulation layer
  - Switching only looks at Ethernet header, routing only looks at IP header, etc.
  - Terminology: "Layer-3 switch," "Layer-4 load balancer," "Layer-7 load balancer"

Ethernet Header



IPv4 Header


Used
Nobody really uses
IP Version number (4)
Size of header
Diffserv marking (QoS priority)
Explicit Congestion Notification: router signals congestion
Length of IP header plus user data
Offset of this fragment from start of original datagram
Fragment flags (Don't Fragment, More Fragments)
Checksum computed over header
Counter decremented at each IP hop
Fragment #
Identifies protocol contained in datagram (e.g., TCP)
Extensible functions: record route, source route, router "alerts," quick-start, etc.
Sender/target of datagram
Offsets
Octet
Bit
0
1
2
3
Version
IHL
DSCP
ECN
Total Length
Identification
Flags
Fragment Offset
Time To Live
Protocol
Header Checksum
Source IP Address
Destination IP Address
Options (if IHL > 5)

TCP Header

"address" of sending application process. Client-side may be ephemeral.

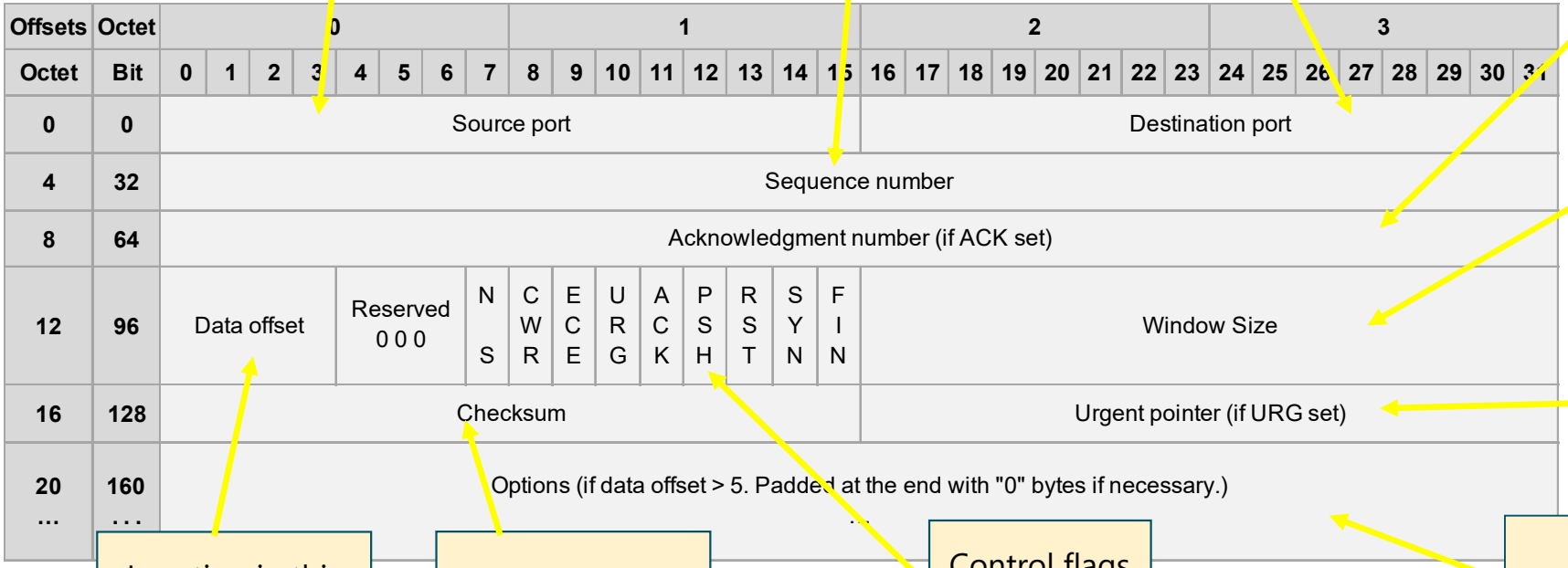
Where the first byte of this packet's data falls within the entire sequence of data set on this connection (byte #). Used for reassembly.

"address" of destination application process

Next byte sender is expecting

Number of bytes sender is willing to receive

Set if segment should be delivered to application immediately



Location in this packet where header ends and data starts

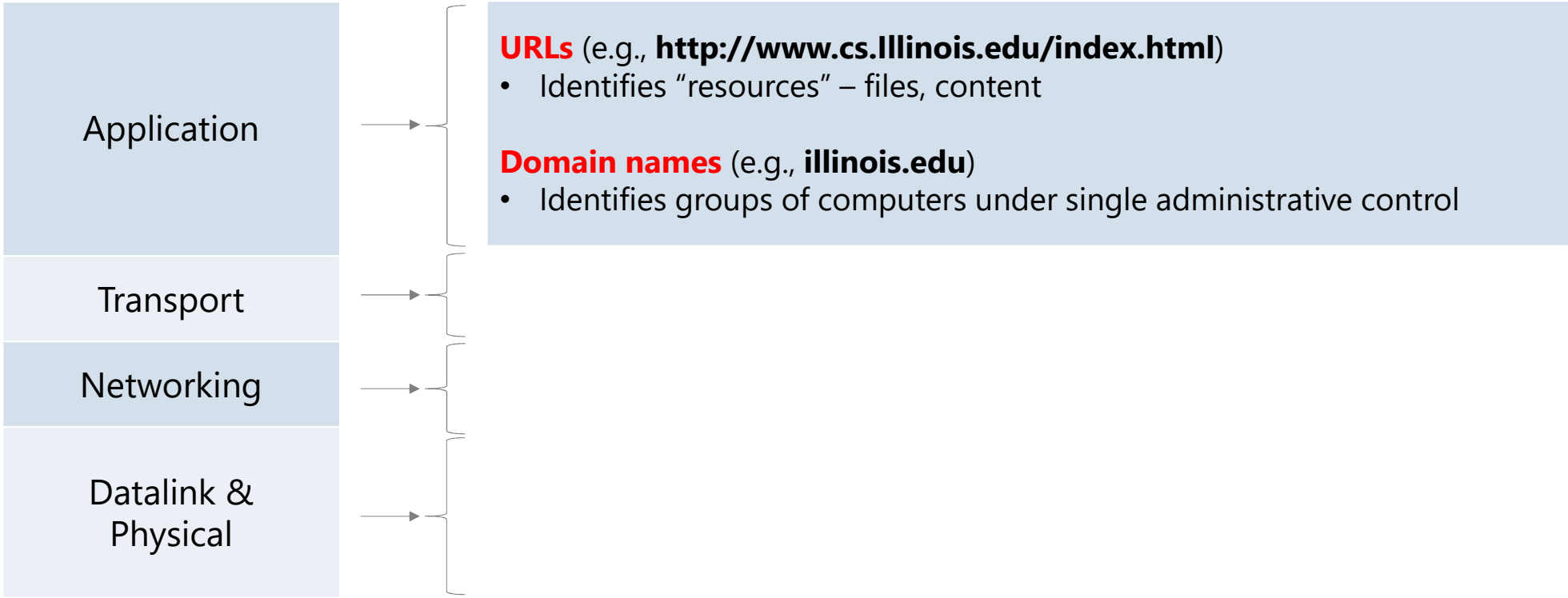
Error check covering header and payload data

Control flags used in TCP protocol operation

Extensible options. Encryption negotiation, multipath data, corruption experienced, etc.

Addressing

Internet Addressing: Different Layers Use Different Addresses



→ *All these addresses are used for end-to-end communication*

Medium Access Control (MAC) Address

- Numerical address associated with a NIC
  - Flat name space of 48 bits (e.g., 00-15-C5-49-04-A9 in HEX)
  - Unique, often hard-coded in the adapter when it is built
- Hierarchical Allocation
  - Organizational Unique Identifier (OUI) – first 24 bits
    - Assigned to vendors (e.g., Dell) by the IEEE
  - Vendor-assigned address – last 24 bits
    - Assigned by the vendor
- Broadcast address (FF-FF-FF-FF-FF-FF)
  - Send the frame to *all* adapters

Internet Protocol (IP) Address

- Numerical address associated with connection point to internet
  - Hierarchical name space of 32 bits (e.g., 186.36.90.1)
  - Network operators make sure they are assigned uniquely
- Hierarchical Allocation
  - ICANN (previously IANA) manages global IPv4 and IPv6 space
  - ICANN delegates blocks (prefixes) of addresses to regional internet registries (RIRs)
    - APNIC, ARIN, LACNIC, RIPE NCC
  - ISPs request blocks from RIRs
  - Enterprises and smaller ISPs request blocks from provider ISPs

Address Discovery

- A host starts up knowing only its MAC address
- It must discover some addressing information before it can communicate with a destination
  1. Its IP address
  2. Destination's IP address
  3. Next-hop MAC address
     a. Destination's MAC address (if destination is local)
     b. First-hop router's MAC address (if destination is not local)

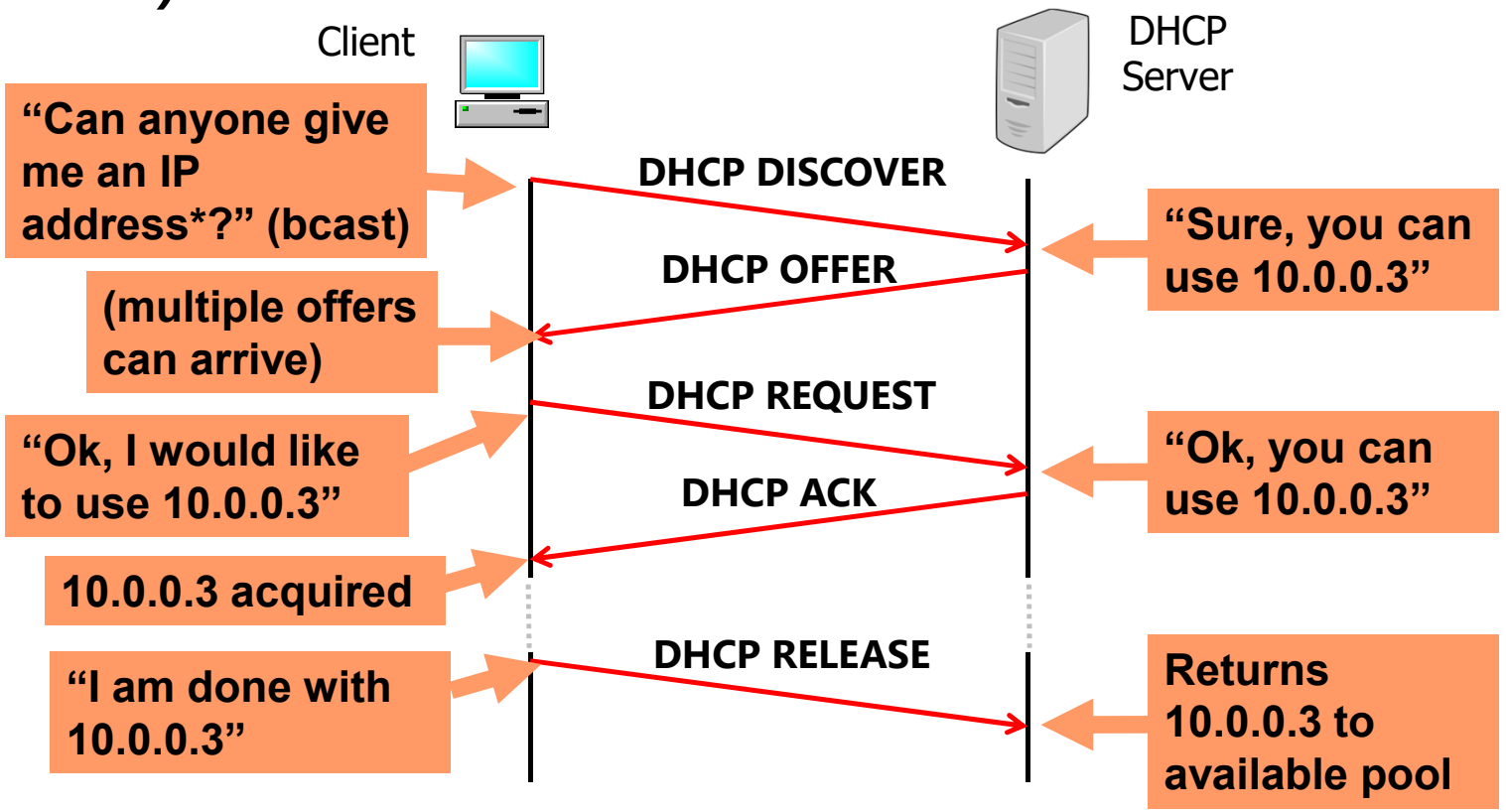
Address Discovery

- How to discover this address information?
- We have some protocols to do this
- **ARP:** discovers MAC address for a given IP address
  - "Who has IP address XX.XX.XX.XX?"
- **DHCP:** automatically assigns IP addresses and other configuration info to network devices
  - "Hey, can someone give me an IP address?"
- These protocols run within a single LAN
  - Leverage broadcast capability of LAN

Dynamic Host Configuration Protocol (DHCP)

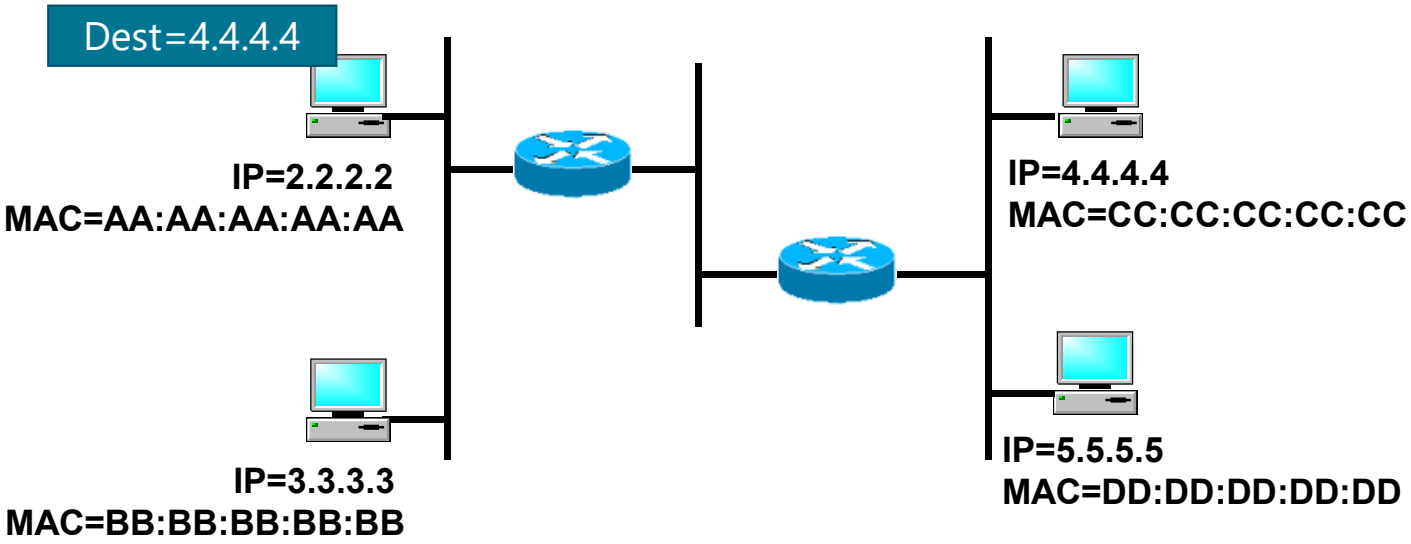
- Automatically configures host to use the network
  - Assigns IP address, subnet mask, DNS server, default gateway
  - Clients listen on UDP port 68, servers on port 67
- Information assigned for a lease time
  - Information cached in host
  - Considered stale and removed after lease time expires
- Defined in RFC 2131

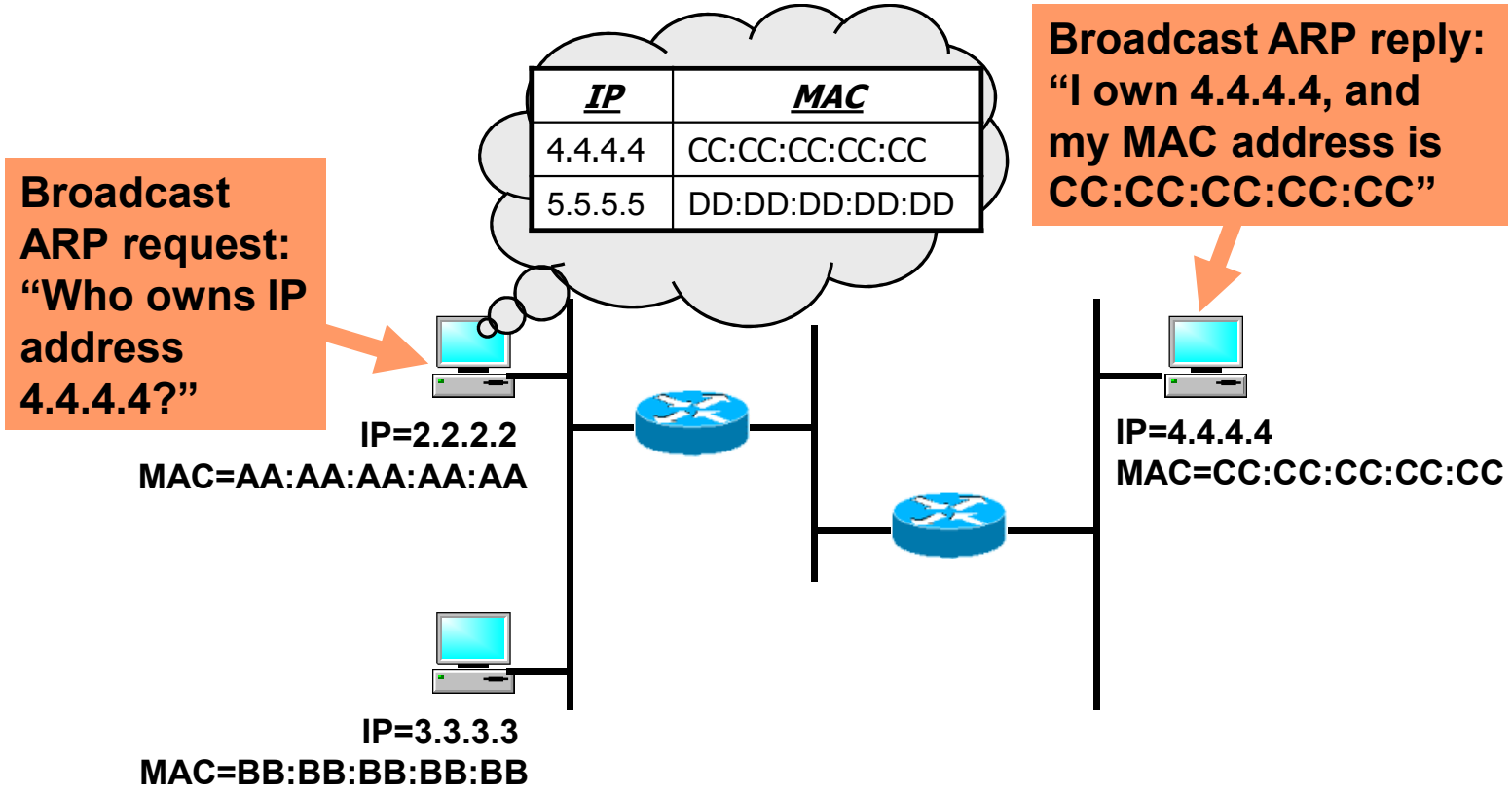
Dynamic Host Configuration Protocol (DHCP)



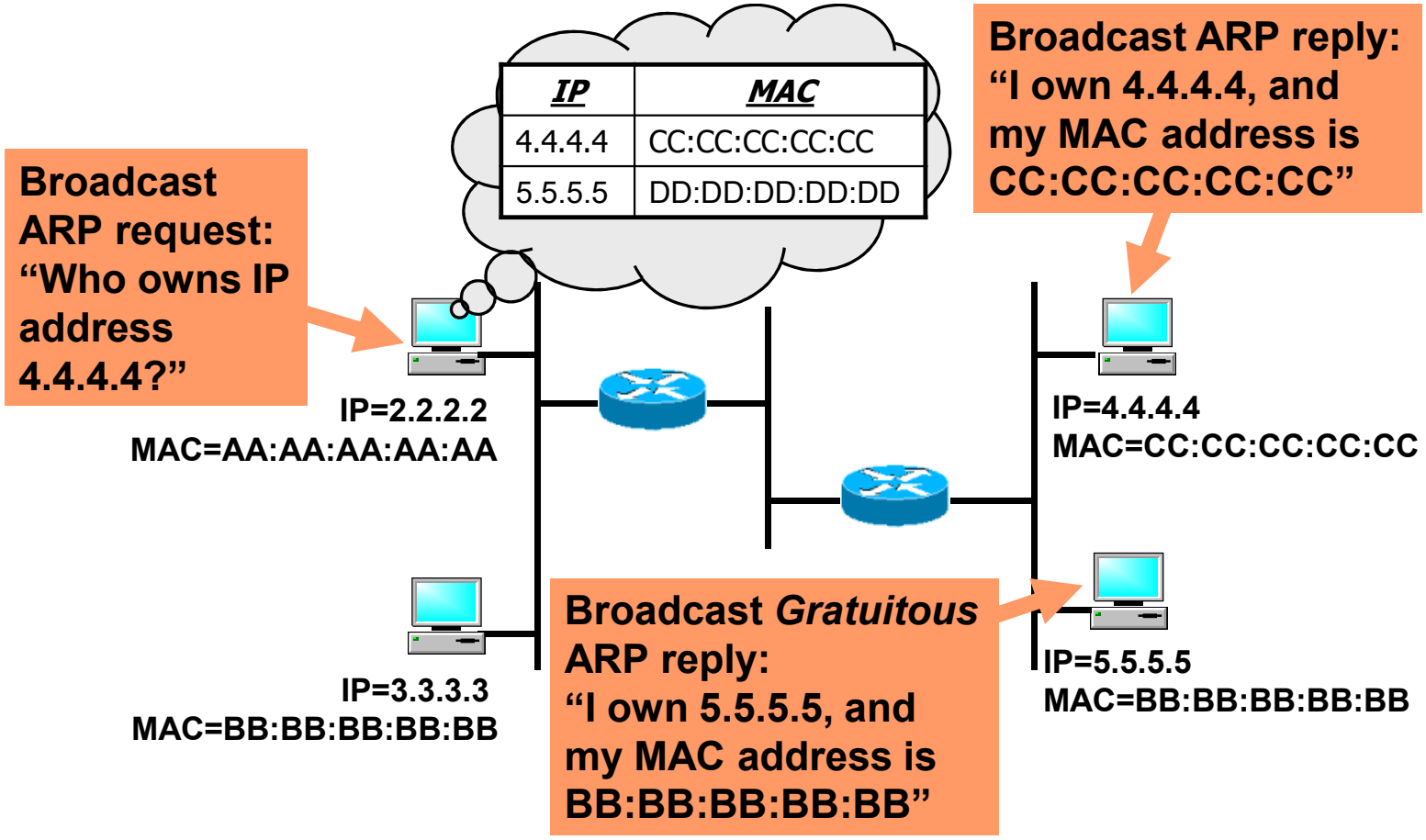
*and other config information

Is DHCP enough?





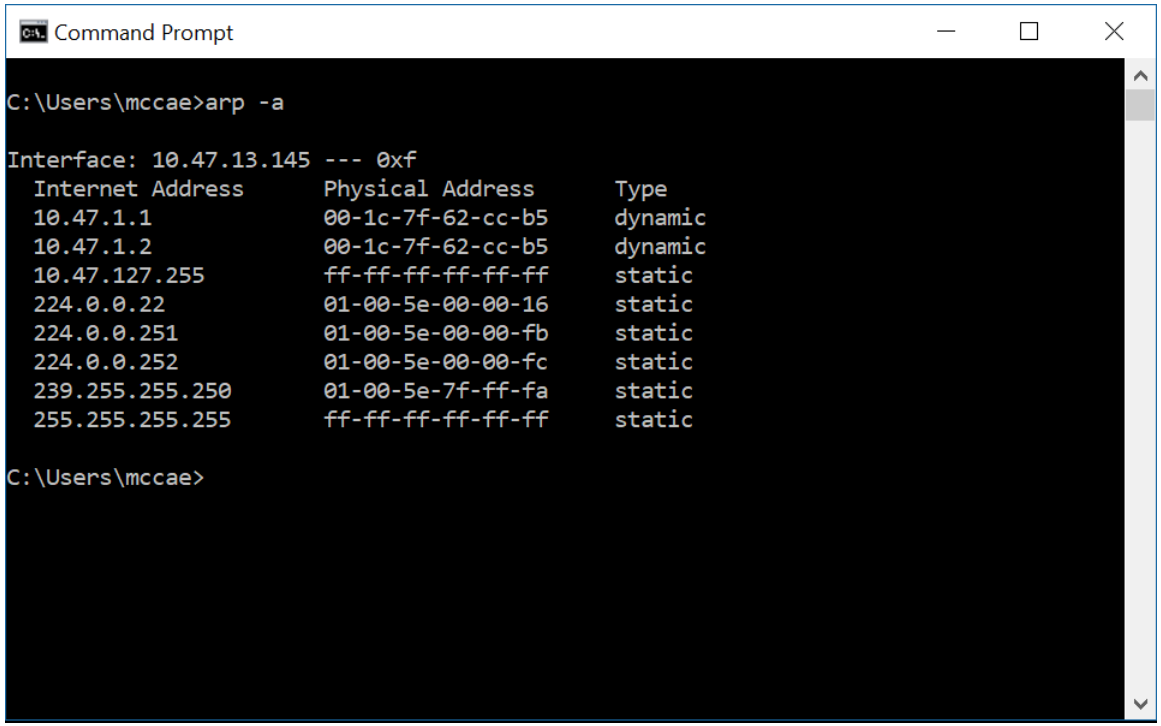
- **ARP:** determine mapping from IP to MAC address
- What if IP address not on subnet?
  - Each host configured with "default gateway," use ARP to resolve its IP address



- What if hosts move?
- **Gratuitous ARP**: tell network your IP to MAC mapping
  - Used to detect IP conflicts, IP address changes; update other machines' ARP tables, update bridges' learned information

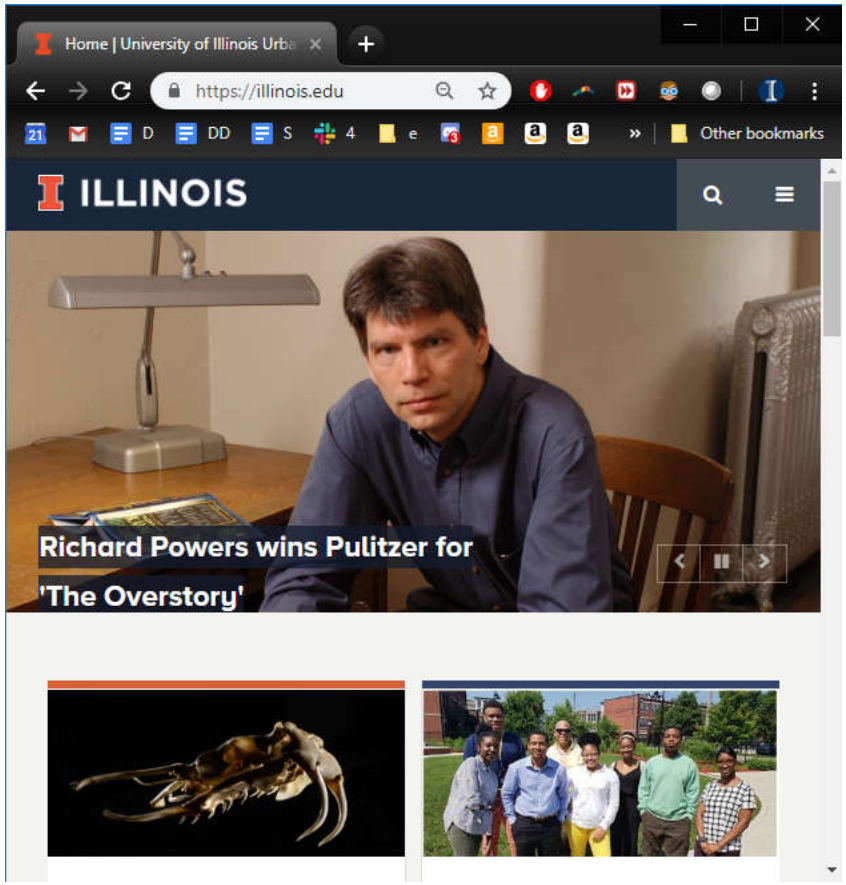
ARP Cache

- End host maintains a cache of learned ARP resolutions:



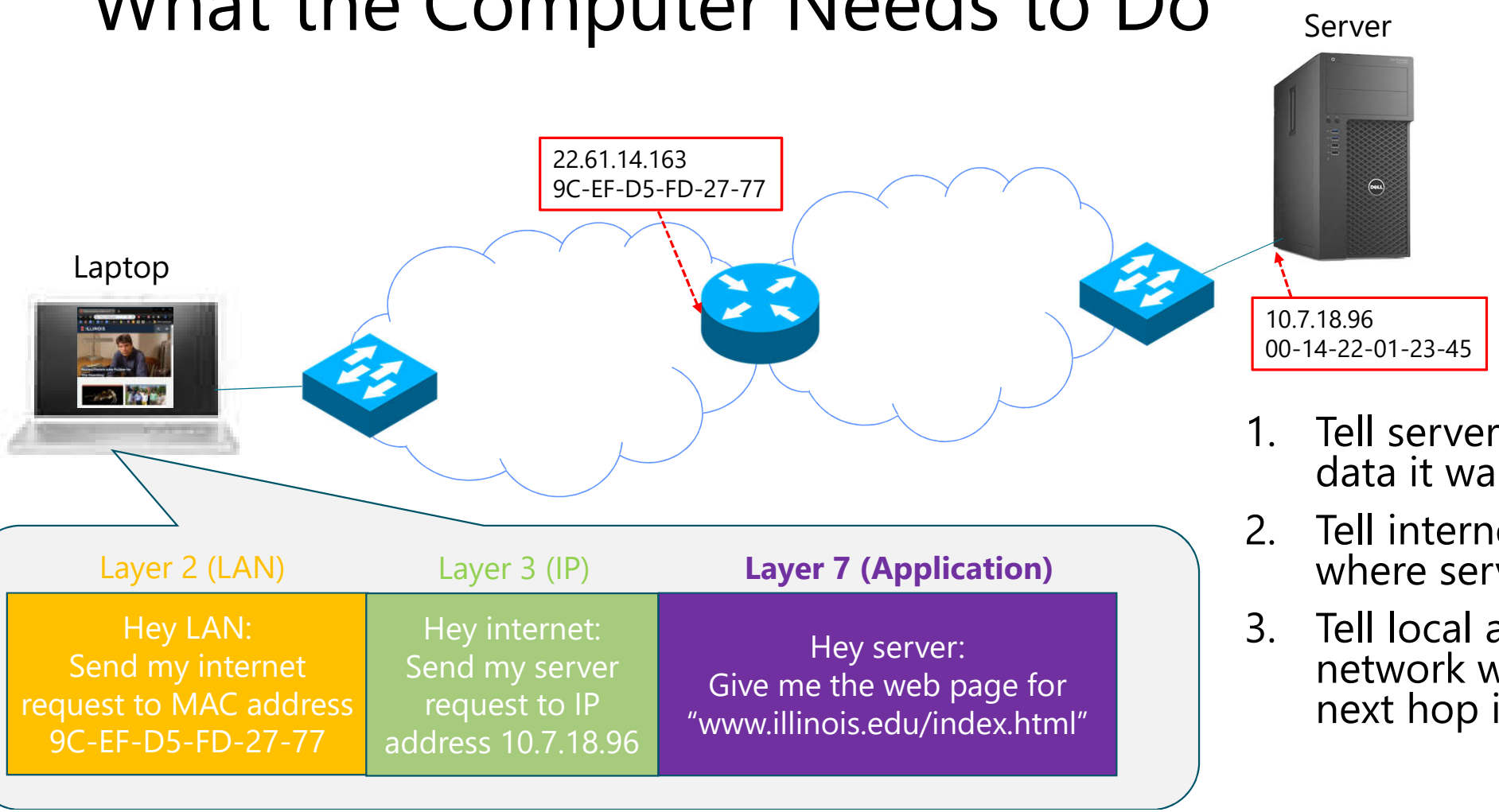
Network Application Programming

What happens inside a browser?



- Browser works with operating system (OS) to construct data packets
- Browser communicates with OS through API ("system calls," or "sockets")

What the Computer Needs to Do

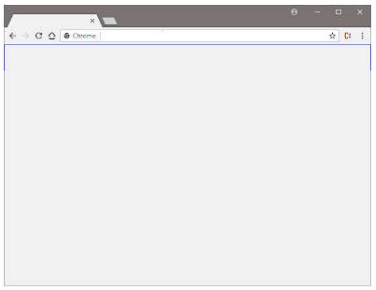


1. Tell server what data it wants
2. Tell internet where server is
3. Tell local area network what next hop is

Layer 7 (Application)

Hey server:
Give me the web page for
"www.illinois.edu/index.html"

Hyper Text Transfer Protocol: Telling a Web Server What to Do



What you want server to do (HTTP Method)

What resource you are referring to (URL)

HTTP Version

"GET /index.html HTTP/1.1" (Client request)

msnbc.com

"HTTP/1.1 200 OK (Server response)

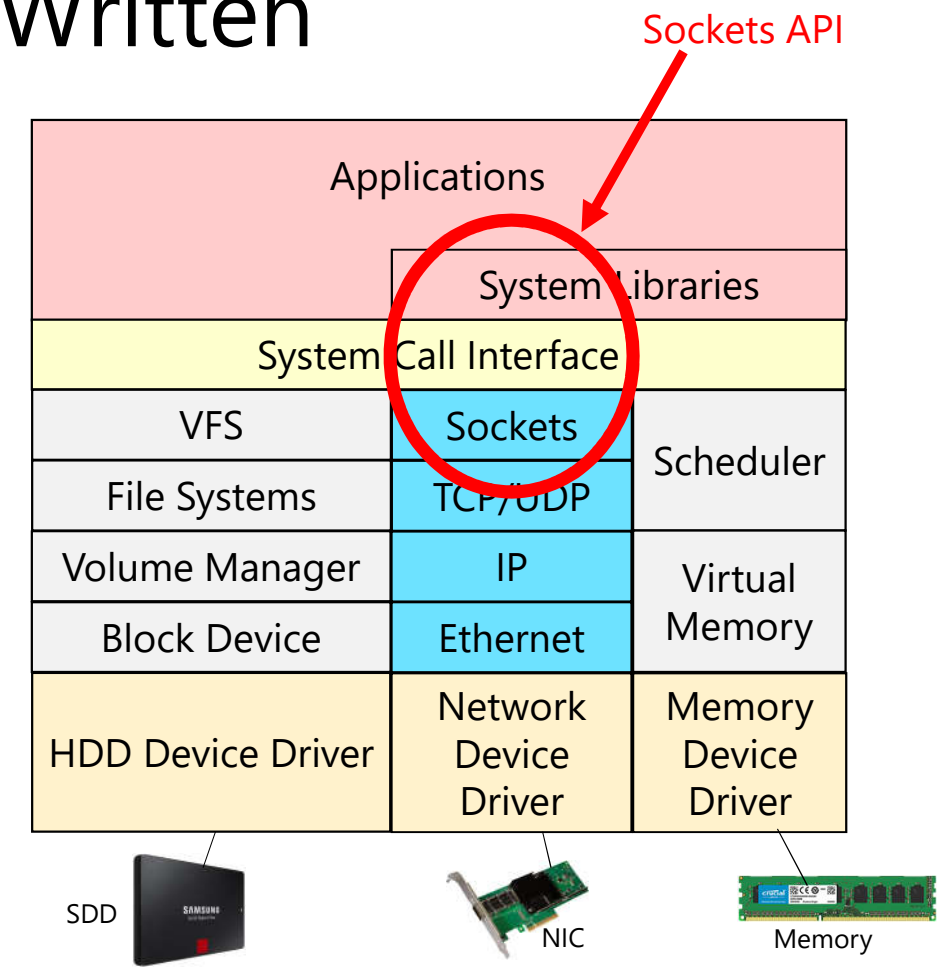
Response Code

HTTP Version

```
<html>
<head>
<title>MSNBC — Breaking News, Top Stories, &amp;amp; Show Clips |
NBC News</title>
<link href="https://nodeassets.nbcnews.com/assets/fonts/shared/fontface.css" rel="stylesheet"/>
<meta name="viewport" content="width=device-width, initial-scale=1"/>
<meta name="apple-itunes-app" content="app-id=1438463967"/>
<a href="https://twitter.com/intent/tweet?text=&amp;amp;via=msnbc&amp;amp;url=
https://www.nbcnews.com/&amp;amp;original_referer=https://www.nbcnews.com/"
target="_blank" rel="noopener noreferrer">
...
```

Sockets Programming: How a Web Browser Is Written

- Sockets API: how applications communicate with the OS networking stack
  - Introduced in 1981 by BSD 4.1
- Implemented as library and set of system calls
- Provides similar interfaces for TCP and UDP



Sockets API: What functions to expose?

- Data structures to store information about connections and hosts
  - IP address and port (`struct sockaddr_in`), socket number (`int`)
- Functions to establish and tear down connections
  - Get available socket number (`socket()`) associates socket with network address (`bind()`)
- Functions to send and receive data
  - Send data string (`send()`), receive data string (`recv()`)

Example Sockets Program: Web Client

```
int main (){
  int sockfd = 0, numbytes = 0;
  char buf[MAXDATASIZE + 1];
  struct hostent* he = NULL;

  if (argc != 2) {
       fprintf (stderr, "usage: client hostname\n");
       exit (1);
  }
  if ((he = gethostbyname ("www.msnbc.com")) == NULL) {
       perror ("gethostbyname");
       exit (1);
  }
```

Socket descriptor: index into per-process table of connections maintained by kernel

Information about host:
Domain name
List of addresses

Performs DNS lookup to get IP address for specified hostname

Example Sockets Program: Web Client

```
if ((sockfd = socket (AF_INET, SOCK_STREAM, 0)) == -1) {
      perror ("socket");
      exit (1);
}
struct sockaddr_in their_addr;
their_addr.sin_family = AF_INET;
their_addr.sin_port = htons (PORT);
their_addr.sin_addr = *((struct in_addr*)he->h_addr);
bzero (&(their_addr.sin_zero), 8);
if (connect (sockfd, (struct sockaddr*)&their_addr,
             sizeof (struct sockaddr)) == -1) {
      perror ("connect");
      exit (1);
}
```

Returns available socket descriptor

Specifies identity of server we will connect to, using data we got from DNS lookup

Forms logical connection to server; initiates TCP 3-way handshake

Example Sockets Program: Web Client

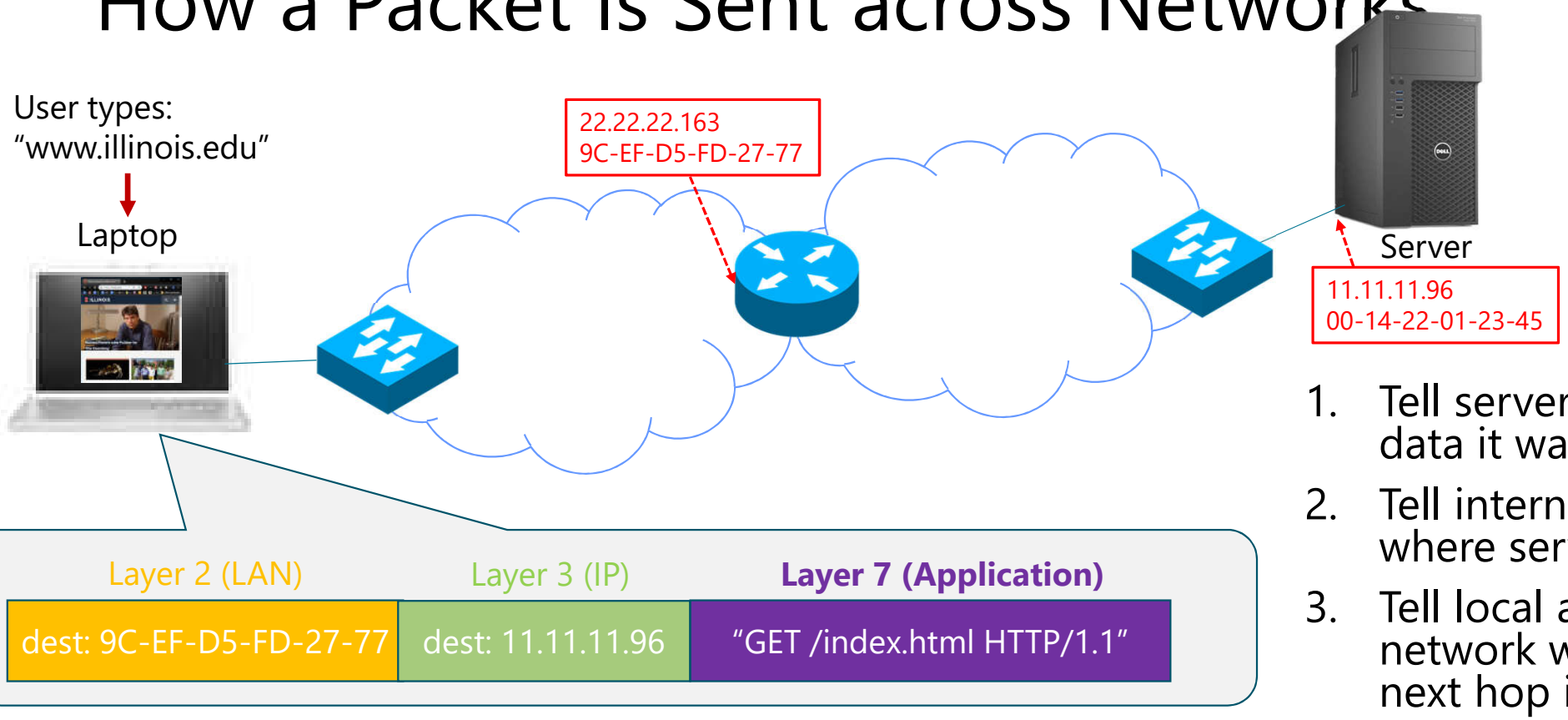
```
  if (send(new_fd,"GET /index.html HTTP/1.1\n\n", 27, 0) == -1) {
       perror("send");
   }


  buf[numbytes] = '\0';
  printf ("Received: %s", buf);
  close (sockfd);
  return 0;
}
```

Sends string of text to server

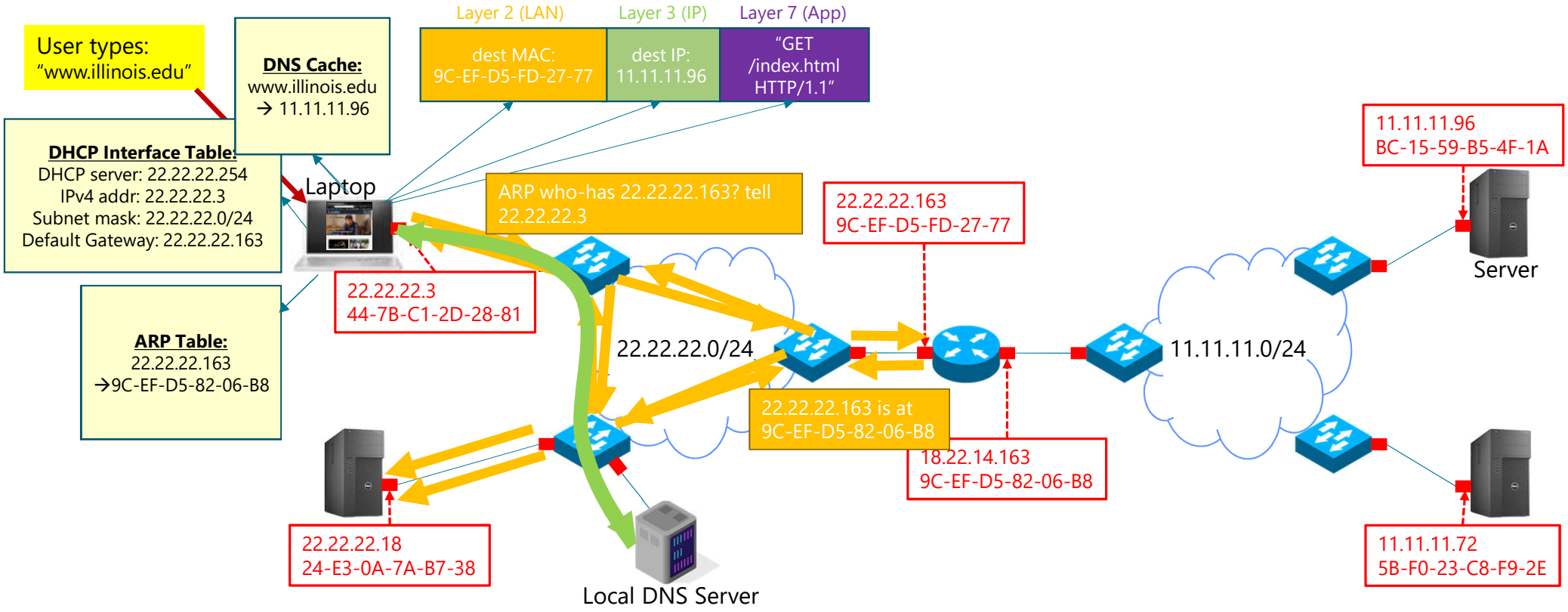
Tell OS we are done with this socket, which will clean up state and tears down any connections (TCP FIN)

Putting Things Together: How a Packet Is Sent across Networks

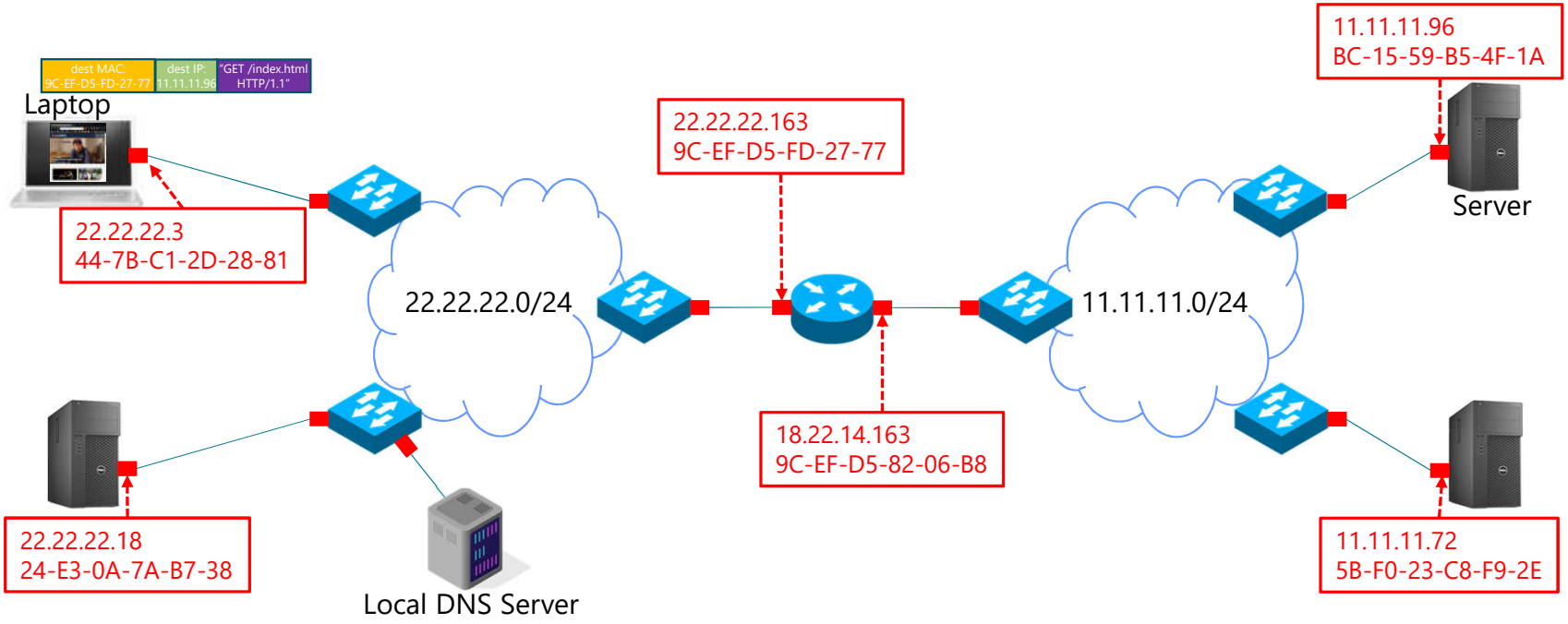


1. Tell server what data it wants
2. Tell internet where server is
3. Tell local area network what next hop is

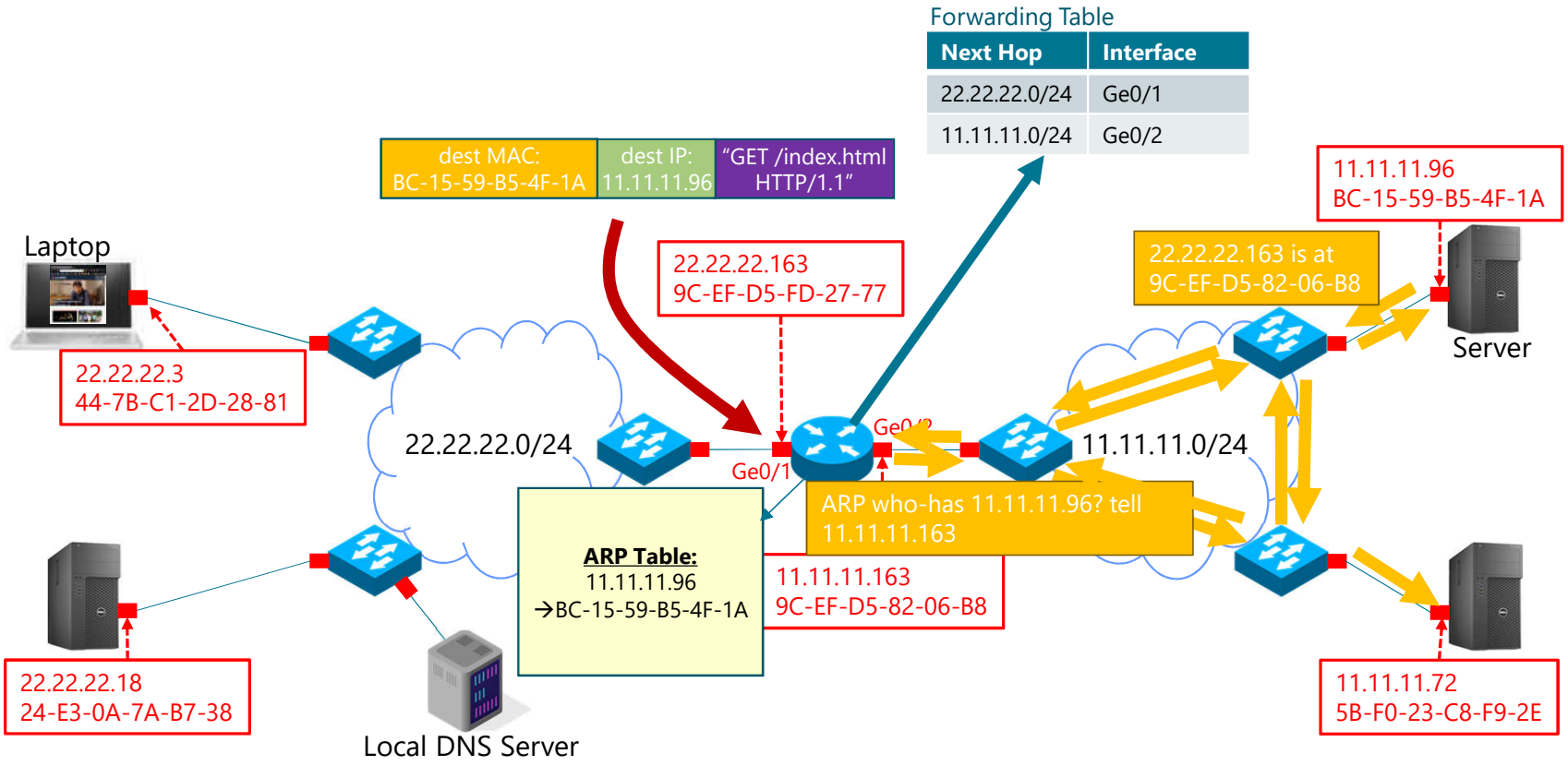
Putting Things Together: How a Packet Is Sent across Networks



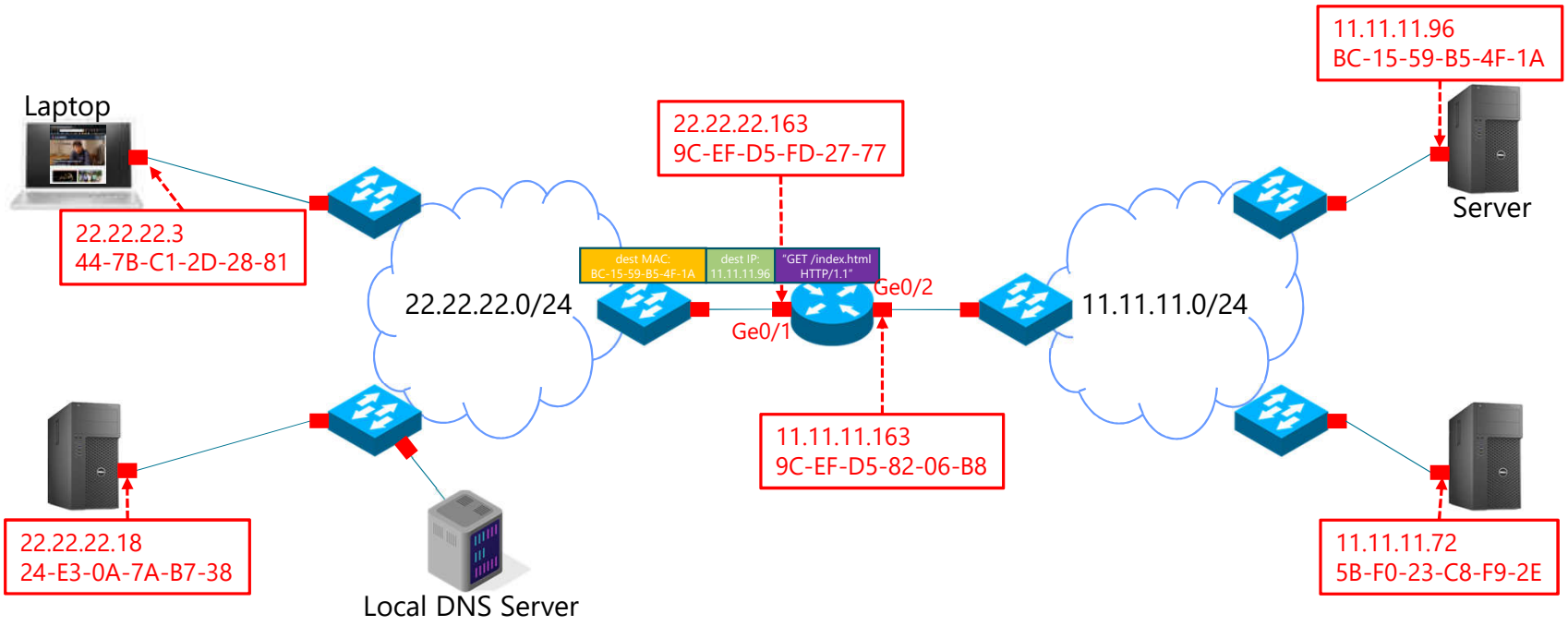
Putting Things Together: How a Packet Is Sent across Networks



Putting Things Together: How a Packet Is Sent across Networks



Putting Things Together: How a Packet Is Sent across Networks



Steps in Creating a Packet (Summary)

- On bootup, use DHCP to discover NIC configuration
- On gethostbyname() resolution request
  - Use DNS lookup to determine IP address
    - Check subnet mask to determine if IP address is on local subnet
    - If so, use ARP if local DNS server's MAC address not in ARP table
    - If not, use ARP if subnet gateway's MAC address not in ARP table
    - Send DNS lookup in UDP packet with corresponding IP and MAC address destinations
    - Return discovered IP address to caller
- On send() call:
  - Use similar steps to above to discover MAC address and send packet

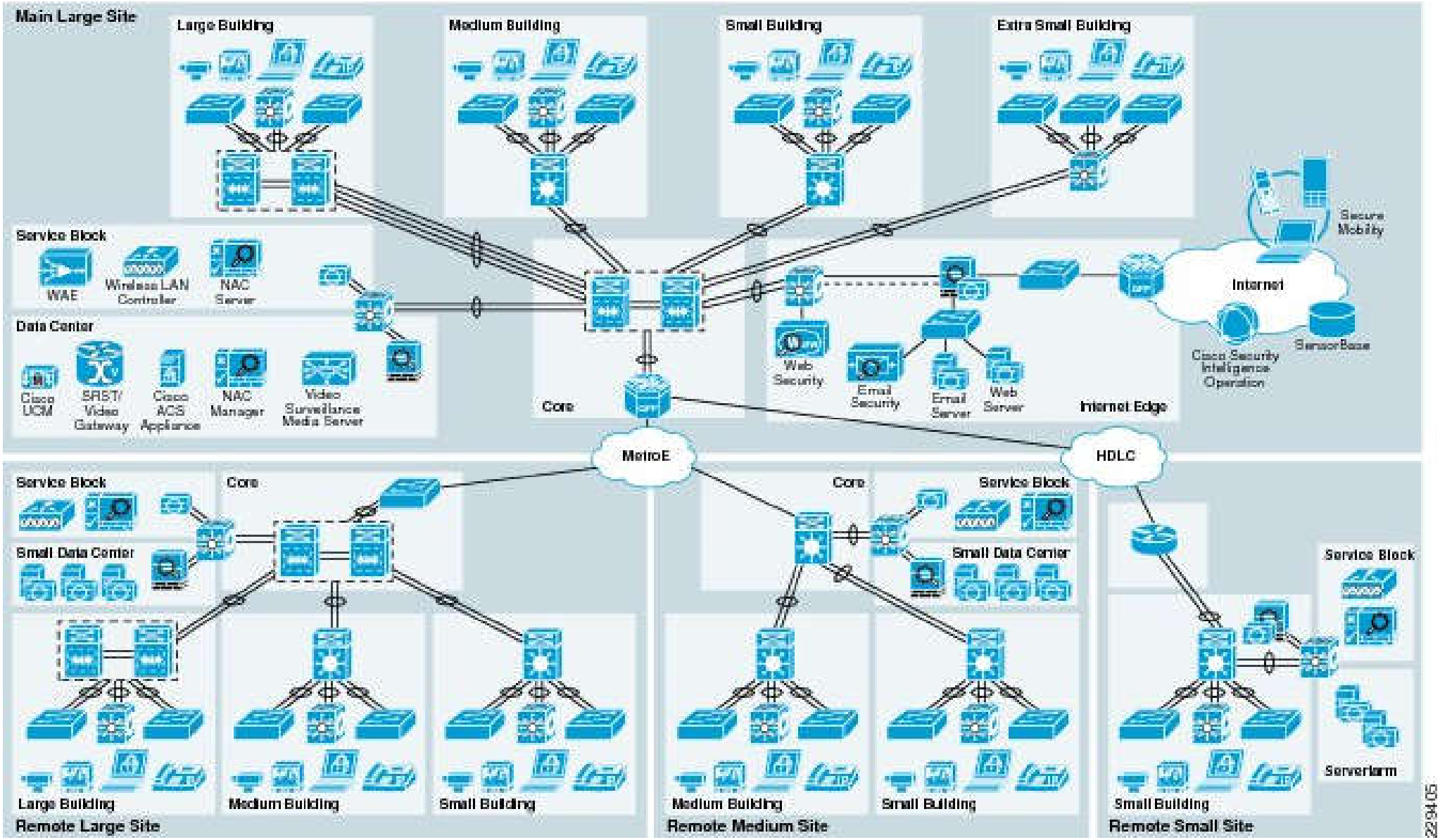
[title: Network Infrastructure]

[week X: video X]

[status: done]

How do you design a network?

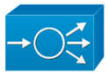
- A network has goals
  - Connect sensors to internet
  - Filter bad traffic
  - Provide voice over IP functionality
- These goals are realized through network design
  - Part of this is done by configuring devices…
  - … but first step is to design the topology
- Network functions are enabled by devices
  - You choose devices that do what you want to happen
  - You put them where you want those things to happen


Main Large Site
Large Building
Medium Building
Small Building
Extra Small Building
Service Block
WAE
Wireless LAN Controller
NAC Server
Data Center
Cisco UCM
SRST/ Video Gateway
Cisco ACS Appliance
NAC Manager
Video Surveillance Media Server
Core
Web Security
Email Security
Email Server
Web Server
Internet Edge
Secure Mobility
Internet
Cisco Security Intelligence Operation
SensorBase
MetroE
HDLC
Service Block
Small Data Center
Core
Large Building
Medium Building
Small Building
Remote Large Site
Core
Service Block
Small Data Center
Medium Building
Small Building
Remote Medium Site
Service Block
Serverfarm
Small Building
Remote Small Site
229405

Variants of Routers

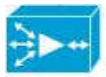

L3 Switch


Router


Load Balancer


Firewall


NAT


WAN Accelerator


Access Point


WLAN Controller


Wireless Router


Wireless Bridge


Satellite


Dish Controller


PBX Switch


Voice Gateway


Voice Router


Voice Switch


Voice Gateway


Reverse Proxy


Intrusion Detection System


Directory Service


Optical Amplifier


Wavelength Router


Label-Switched Router


Transcoder


Access Control Service


DSLAM


BRAS


CMTS

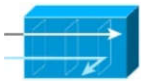

Optical Transport


Fabric Extender

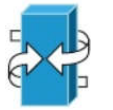

Location Appliance


Softswitch

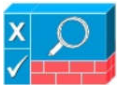

Traffic Analyzer


DDoS Mitigation


Deep Packet Inspection


Signaling Controller


Identity Services Appliance


Network Admission Controller


NAS Appliance


Storage Array

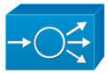
Application Proxy

Device Types

- Think of "functionality" separate from devices
  - Functions like switching and inspection used to be tied to specific device types
  - Functions increasingly available on shared hardware
  - Network function virtualization (NFV) taking this idea further
- Several key device function types
  - **Forwarding** (where to send traffic?)
  - **Processing** (changing contents of traffic)
  - **Translation** (changing format of traffic)
  - **Isolation** (blocking bad traffic)

Device Types: Forwarding


Router


Load Balancer


Voice Switch


Softswitch

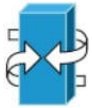

Wavelength Router

- Decide next hop based on header or traffic contents
- May construct forwarding tables in advance to speed lookups
- Run-time:
  1. Determine if self is responsible next-hop for flow
  2. Lookup/determine next hop
  3. Forward out appropriate outbound interface

Device Types: Processing


Application Proxy


Reverse Proxy


Signaling Controller


Location Appliance


Transcoder

- Manipulate or intercept content in flow
- Change data, encrypt, cache, etc.
  - May require application-layer knowledge
- Run-time:
  1. Receive flow
  2. Perform operations on flow (intercept, cache, mark, manipulate headers/data/packets)

Device Types: Translation


NAT


PBX Switch


Wireless Bridge


DSLAM


Optical Transport

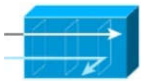
- Convert headers and/or data to different format or proxy data
- May maintain table of translations
- Run-time:
  1. Determine if self is responsible next-hop for flow
  2. Lookup/determine translation
  3. Modify packets

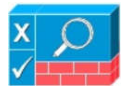
Device Types: Isolation


Firewall


Access Control Service

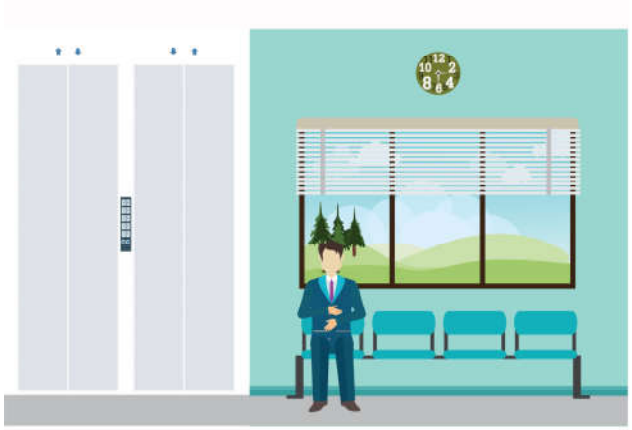

Intrusion Detection System

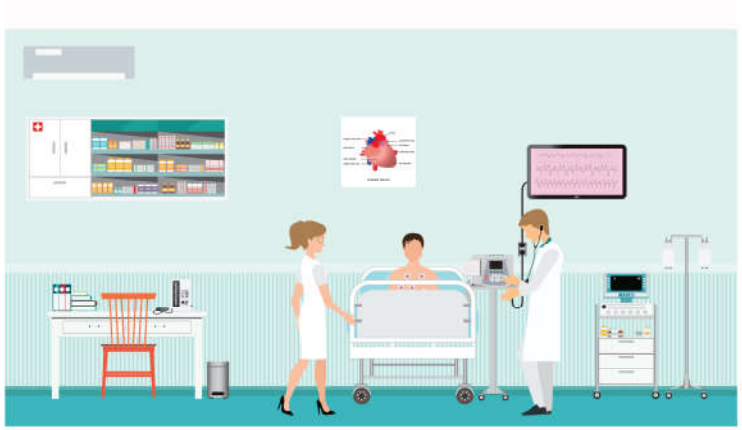

DDoS Mitigation

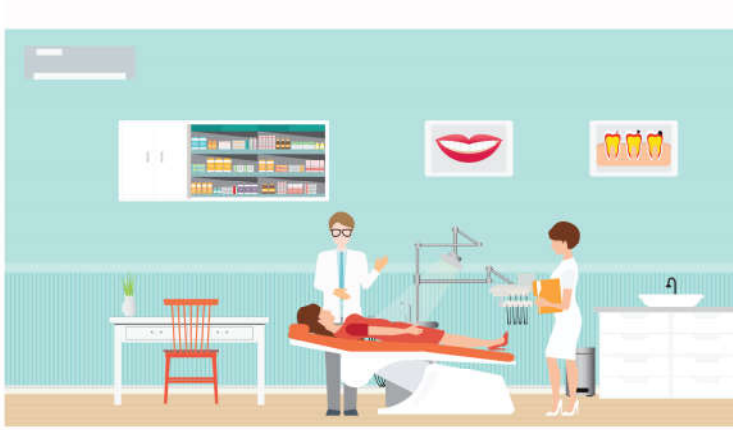

Network Admission Controller

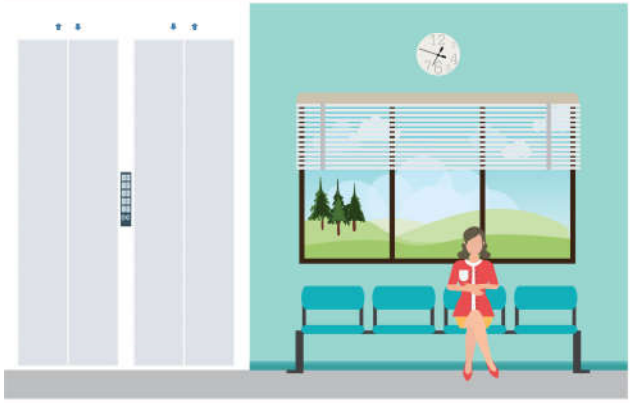
- Prevent information/data/packets from going to certain locations
- Configure access control and prioritization policies in advance
- Run-time:
  1. Select appropriate header fields or content in flow
  2. Traverse policy list, process (usually first) matching rule
  3. Forward or filter traffic appropriately

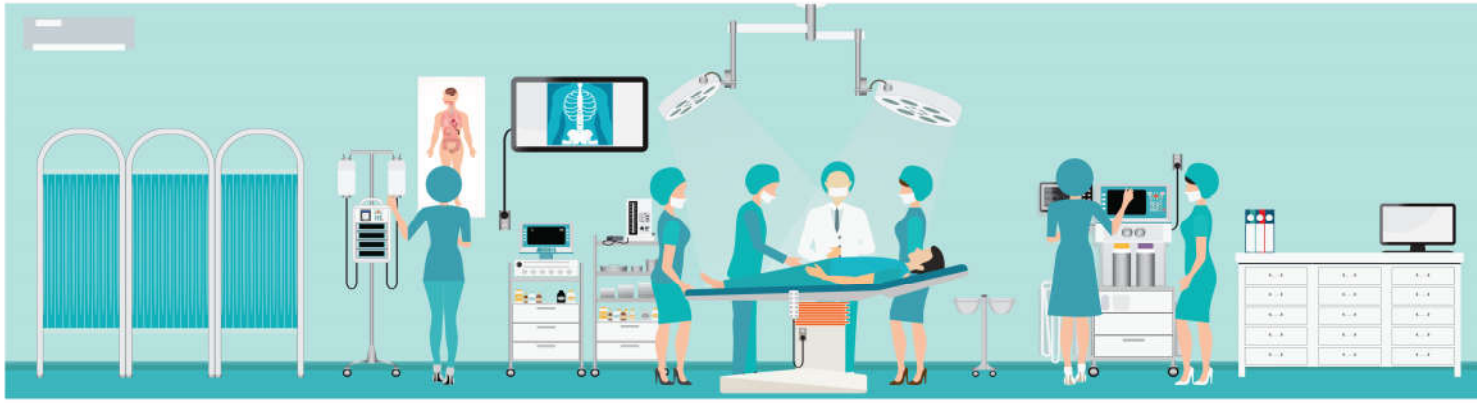
Scenario: Hospital IoT Deployment





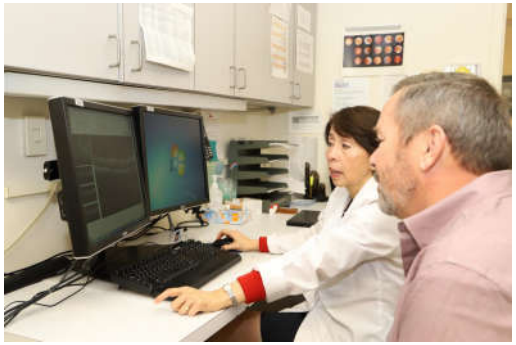




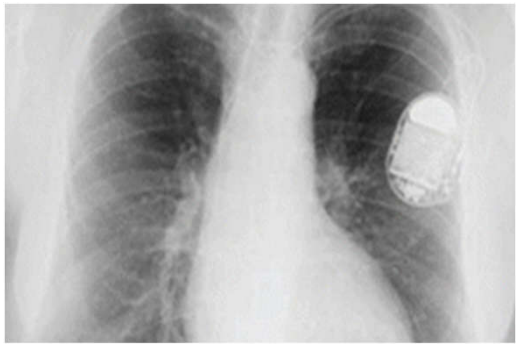


Scenario: Hospital IoT Deployment

Points of Sale (POS)



Medical Computers



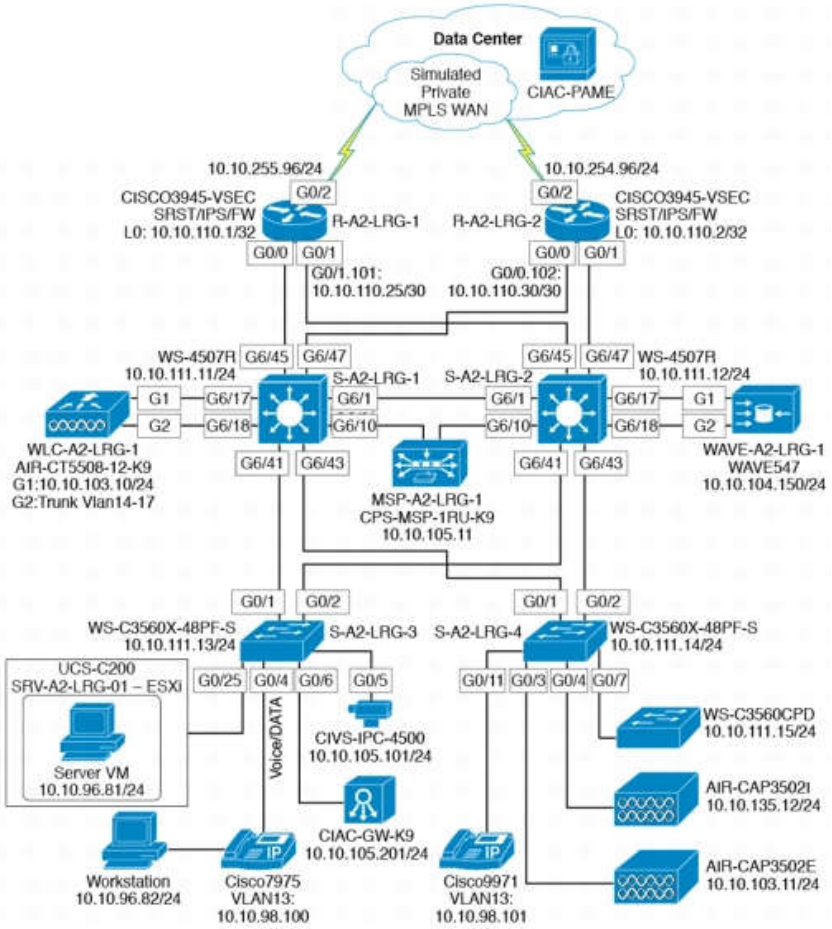
Networked Implants

Patient Monitoring

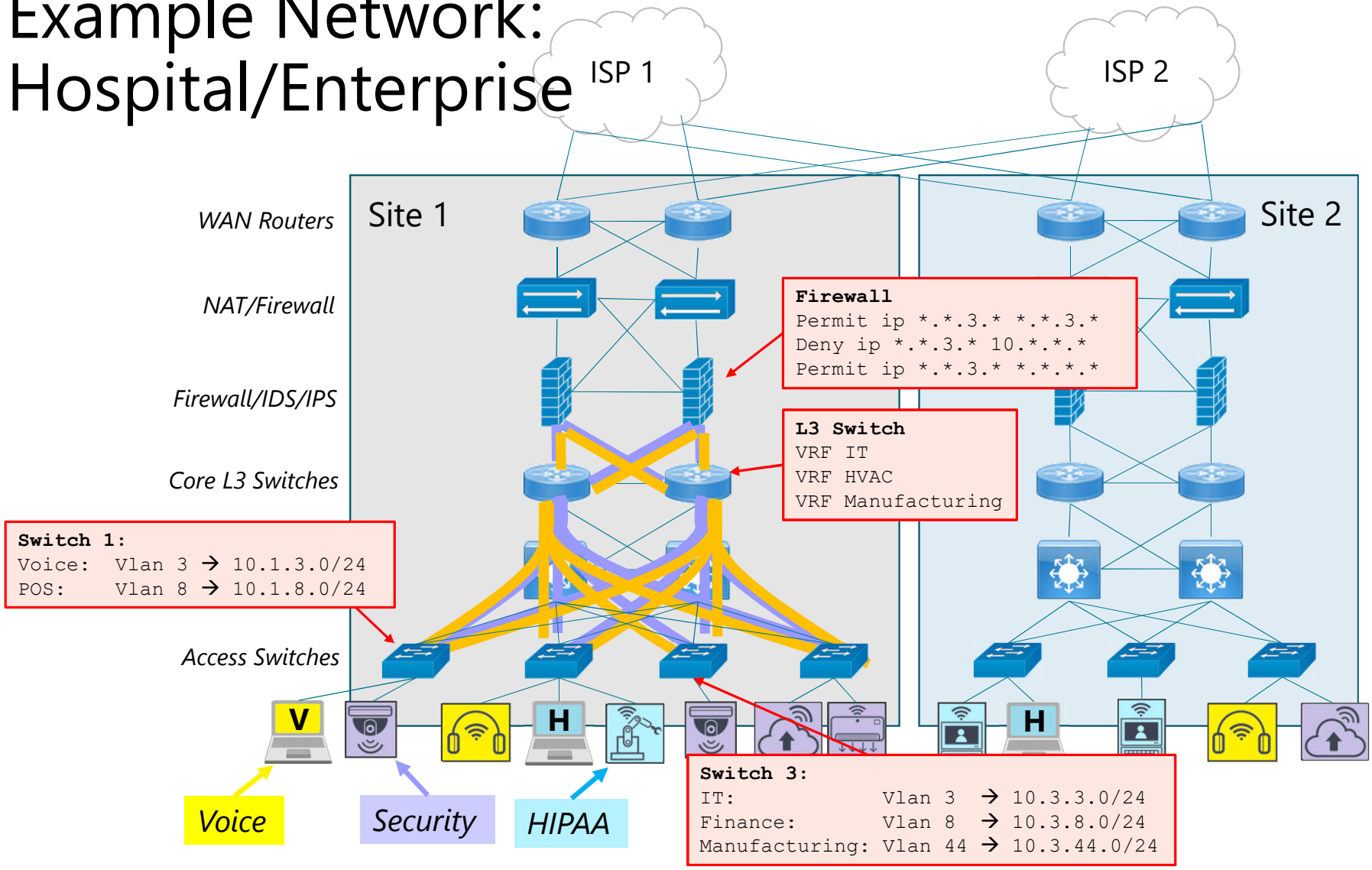
Guest Wireless

Medical Automation and AI

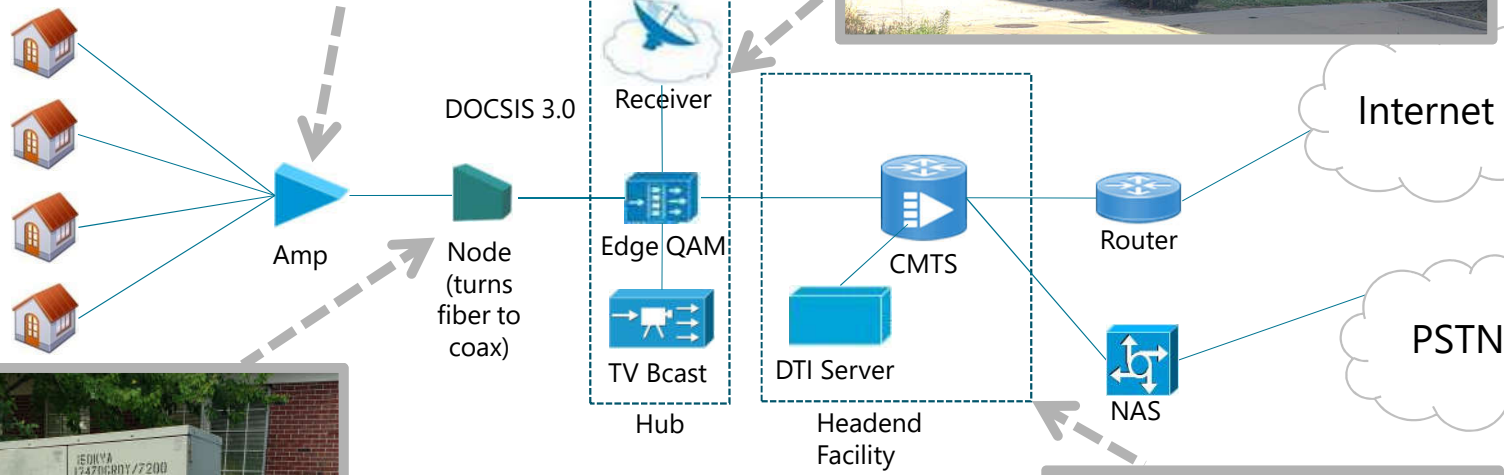
Example Network: Healthcare IoT



Example Network: Hospital/Enterprise



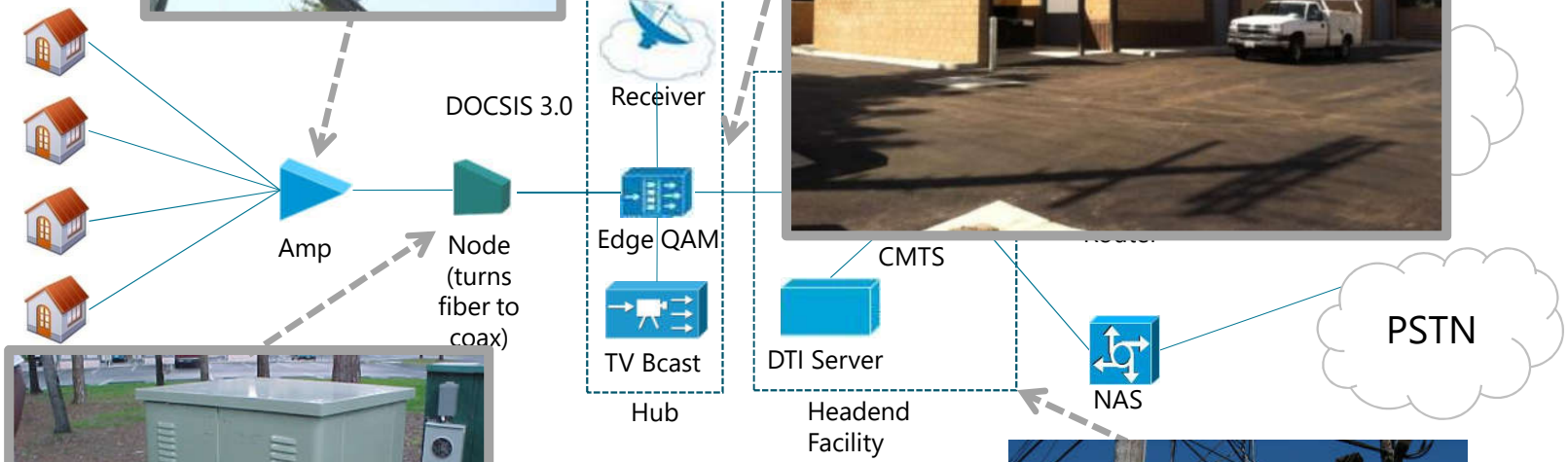
E rk: Broa



k largest in terms of pl
stment

bs, 95600 Nodes, 1.9M oax

Examp k: Brc



largest in terms of phy
tment
s, 95600 Nodes, 1.9M

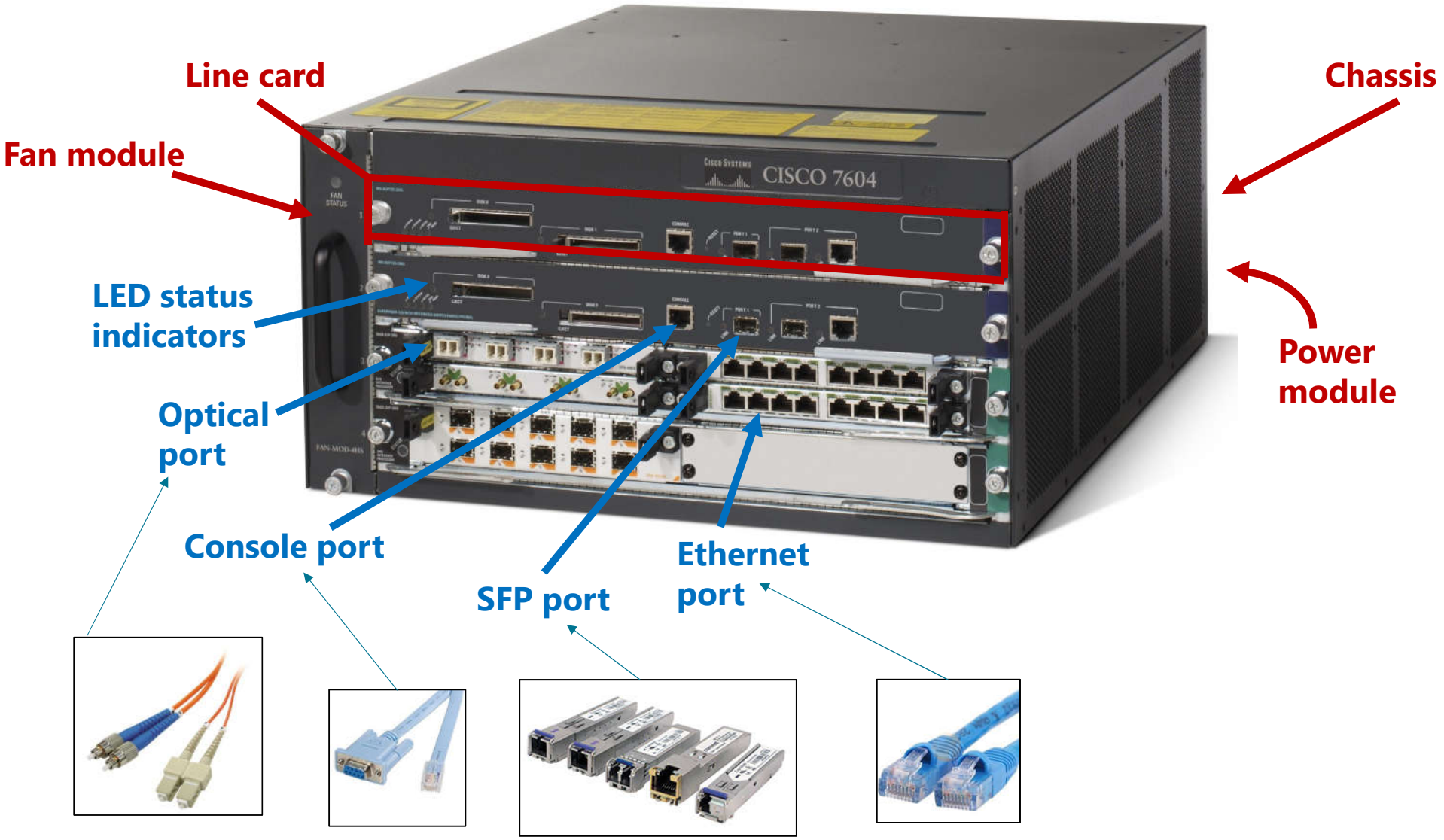
[title: Device Internals]

[week X: video X]

[status: done]

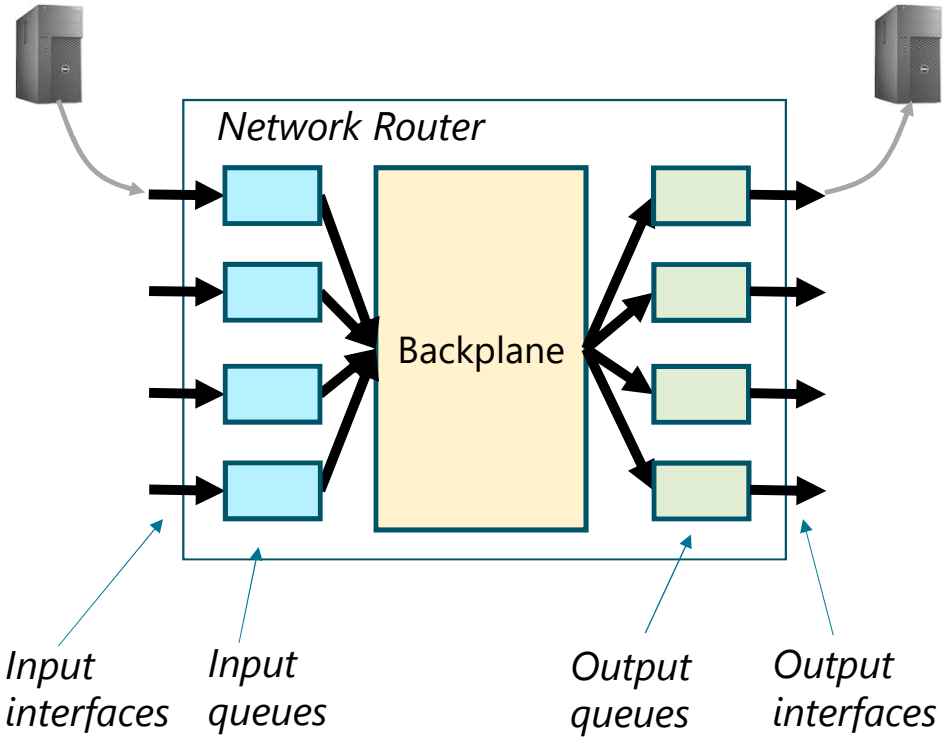
What is a network device?

- Entity used to forward packets
- May be physical or virtual


Line card
Chassis
Fan module
LED status indicators
Optical port
Power module
Console port
SFP port
Ethernet port
CISCO 7604

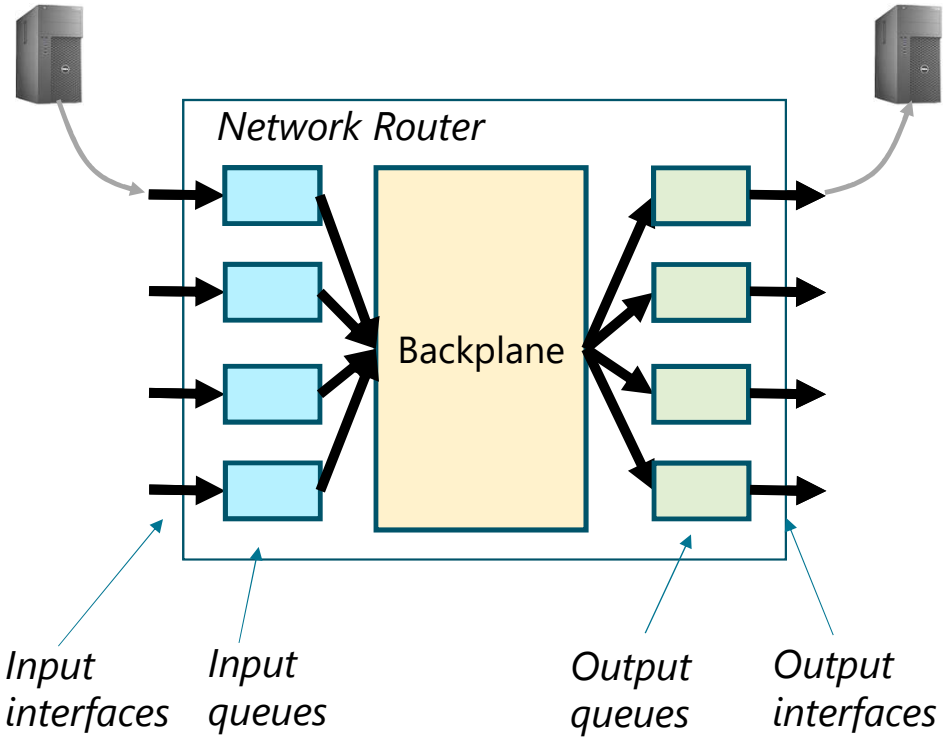
What is a router, exactly?

- Special-purpose computer
  - Traditional computer architecture (CPU, DRAM, bus)
  - Coupled with high-speed packet forwarding hardware (ASIC)
- Router consists of
  - Interfaces where packets arrive
  - Interfaces where packets leave
  - Set of queues to handle bursts of congestion
  - Backplane to forward packets between them



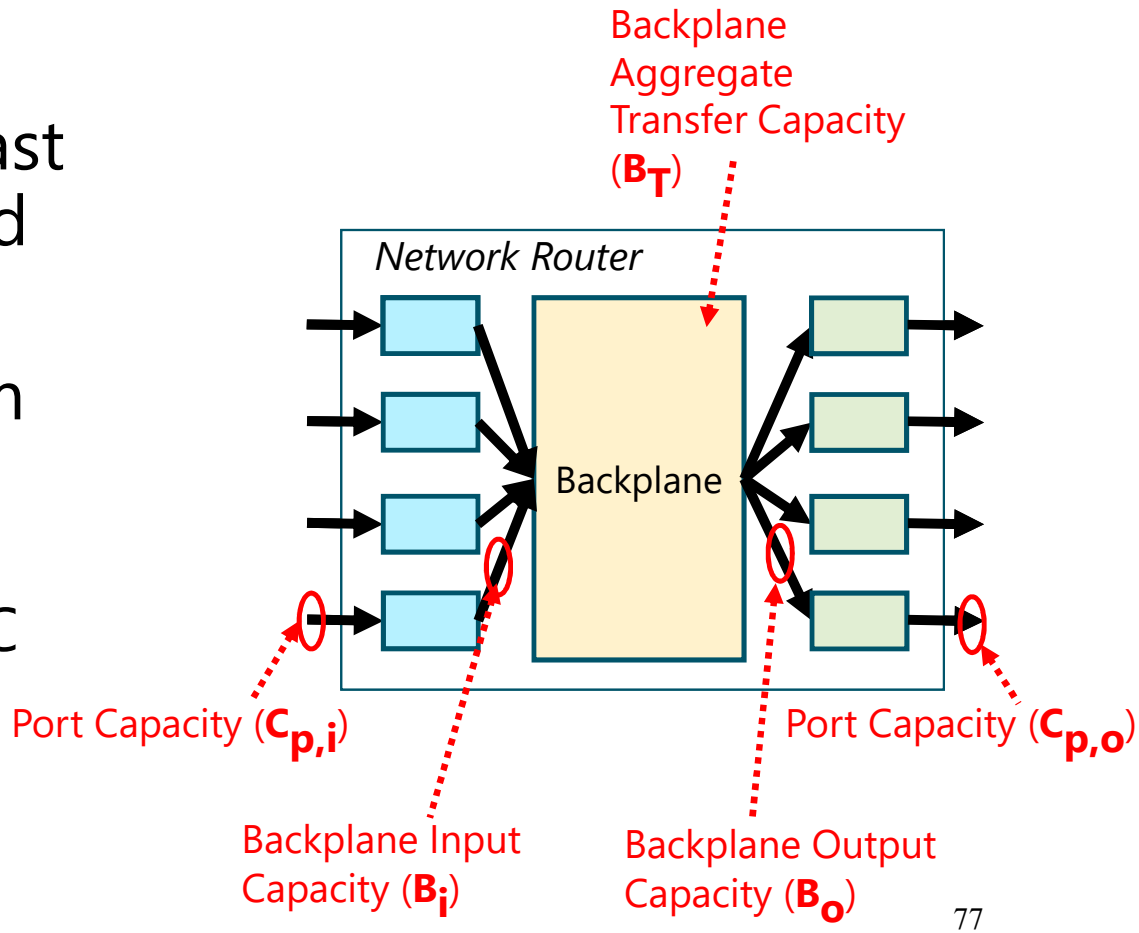
Typical Router Architecture

- Input and output interfaces, connected via a backplane
- Backplane can be implemented in different ways:
  - Shared memory (low-capacity routers, e.g., software routers)
  - Shared bus
  - Switched bus (highest-capacity routers)



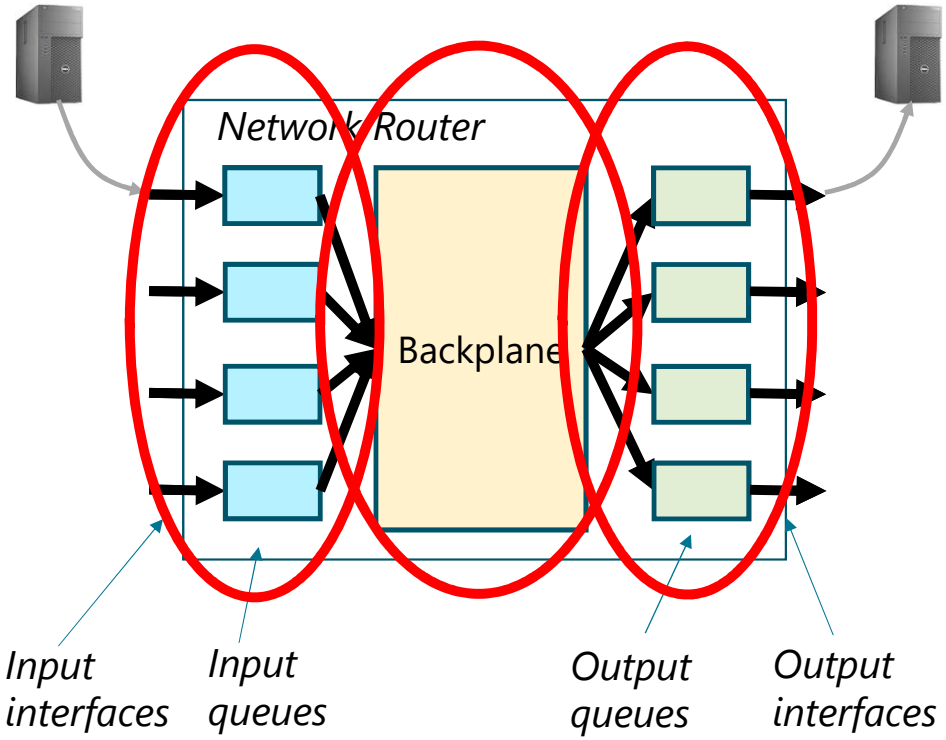
Backplane Speedup

- Backplane needs to be fast enough to handle offered load from all ports
- Router architects perform calculations to provision resources within router
  - Backplane Speedup = B/C
  - Input Speedup = $C_{P,i}/C$
  - Output Speedup = $C_{P,o}/C$



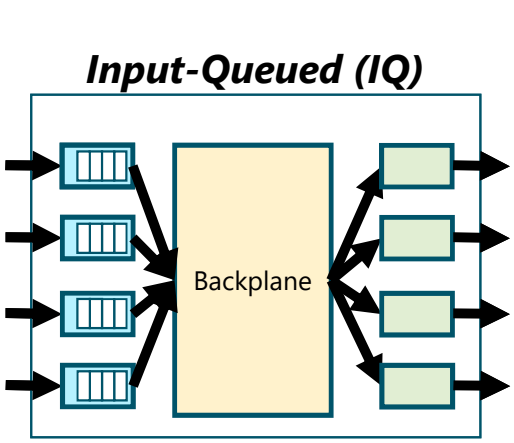
What do pieces of a router do?

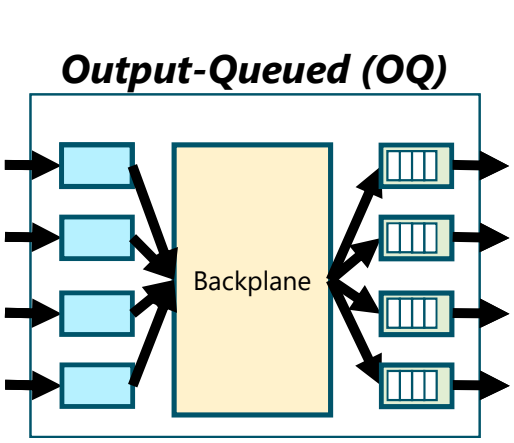
- Input interfaces:
  - Perform lookup (deciding which interface to send out)
  - May enqueue packets
  - May perform scheduling
- Backplane:
  - Move packets from input to output interfaces
- Output interfaces:
  - May enqueue packets
  - May perform scheduling

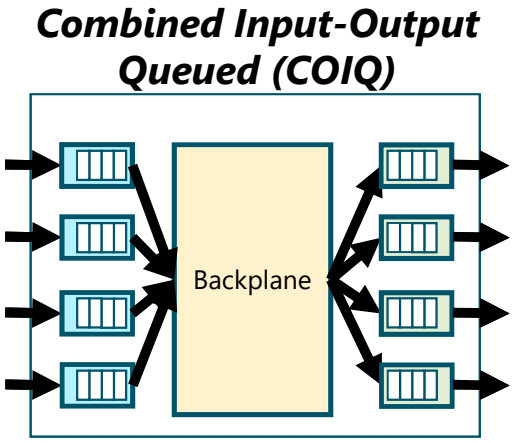


Common Router Architectures

- One key question is where to put queues
  - Need fast memory (expensive)
  - Need to perform complicated functions (deciding which packets to drop)
- Three key architectures:

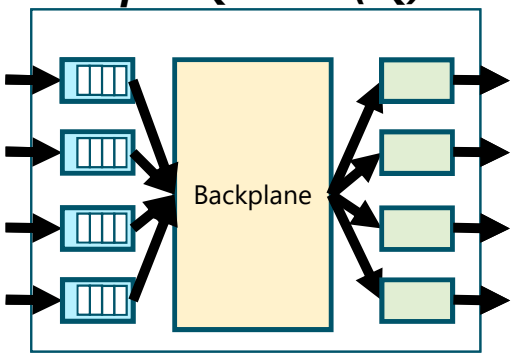






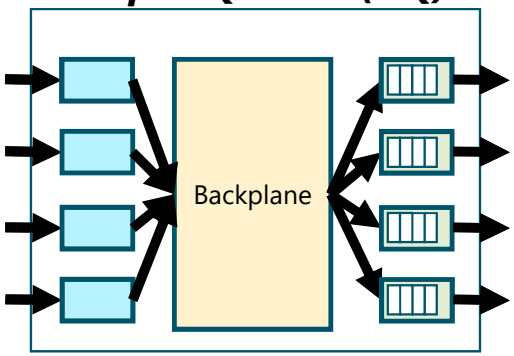
Common Router Architectures

Input-Queued (IQ)



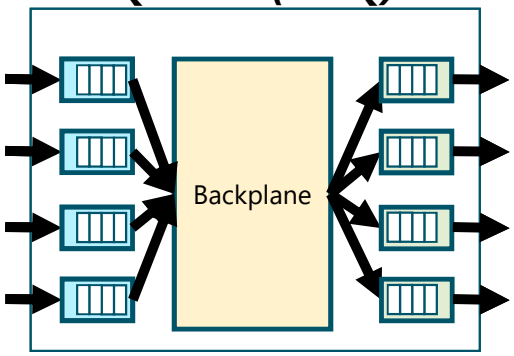
- Queue packets at inputs
- Easy to design
  - Can queue packets if backplane or outputs are overloaded
- Downsides: head-of-line blocking

Output-Queued (OQ)



- Queue packets at outputs
- Problem: What if all inputs send to same output?
  - Requires speedup of [# ports]
  - Not very feasible design

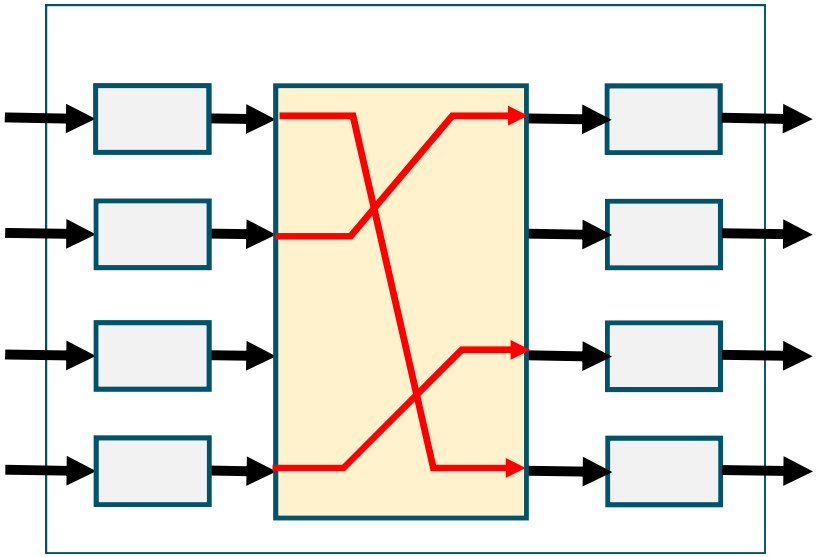
Combined Input-Output Queued (COIQ)



- Queue packets at both inputs and outputs
- Can achieve higher utilization
- Downsides: harder to design algorithms
  - e.g., distributed backpressure

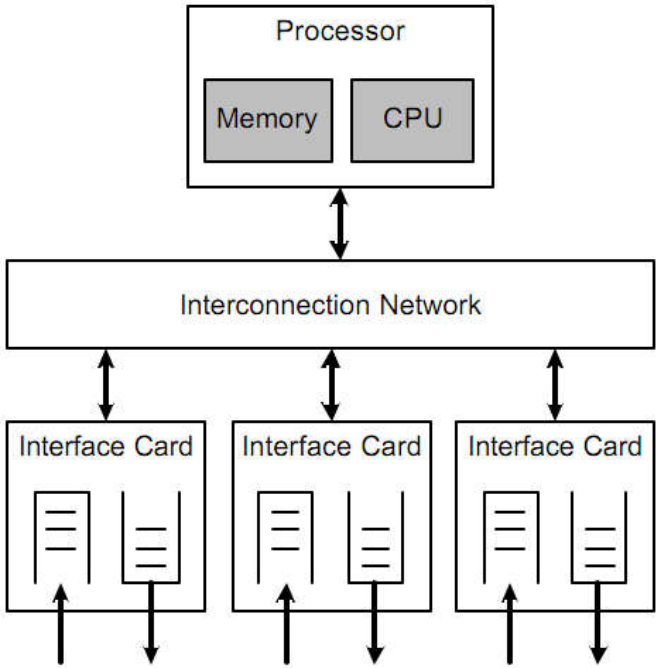
Backplane

- Point-to-point switch inside the router
  - Allows (simultaneous) transfer of packet between any two disjoint pairs of interfaces
  - Like a little network inside a router
- Packets fragmented into fixed-size cells (e.g., 64B)
  - Simplifies hardware, eliminates wastage due to idle cycles



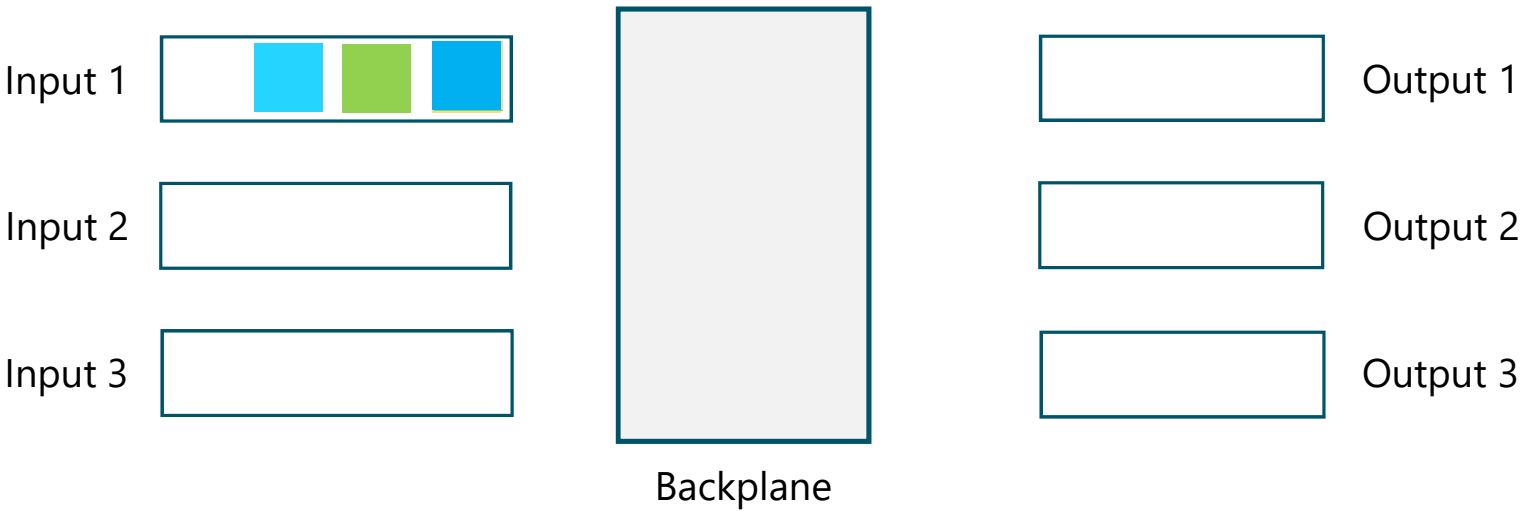
Router Components

- On a PC router:
  - Interconnection network is the PCI bus
  - Interface cards are the NICs (e.g., Ethernet cards)
  - All forwarding and routing is done on a commodity CPU
- On commercial routers:
  - Interconnection network and interface cards are sophisticated, special-purpose hardware
  - Packet forwarding often implemented in a custom ASIC
  - Only routing (control plane) is done on the commodity CPU (route processor)



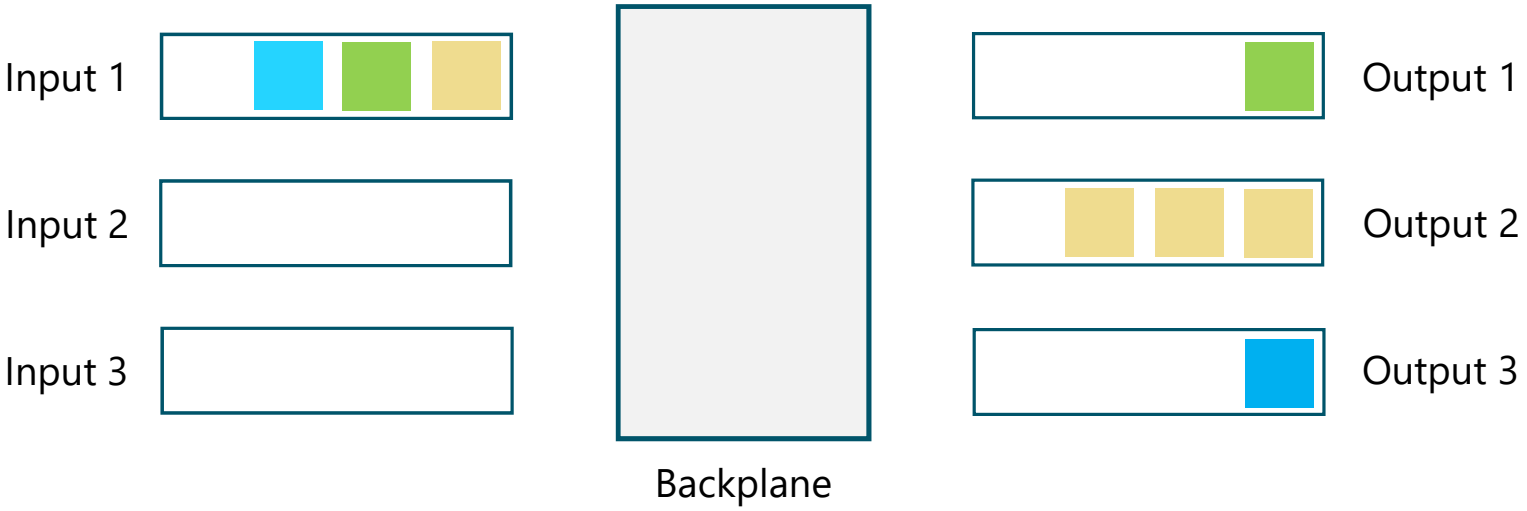

GOODWIN AVENUE
SPRINGFIELD
669 0935

Head-of-Line Blocking



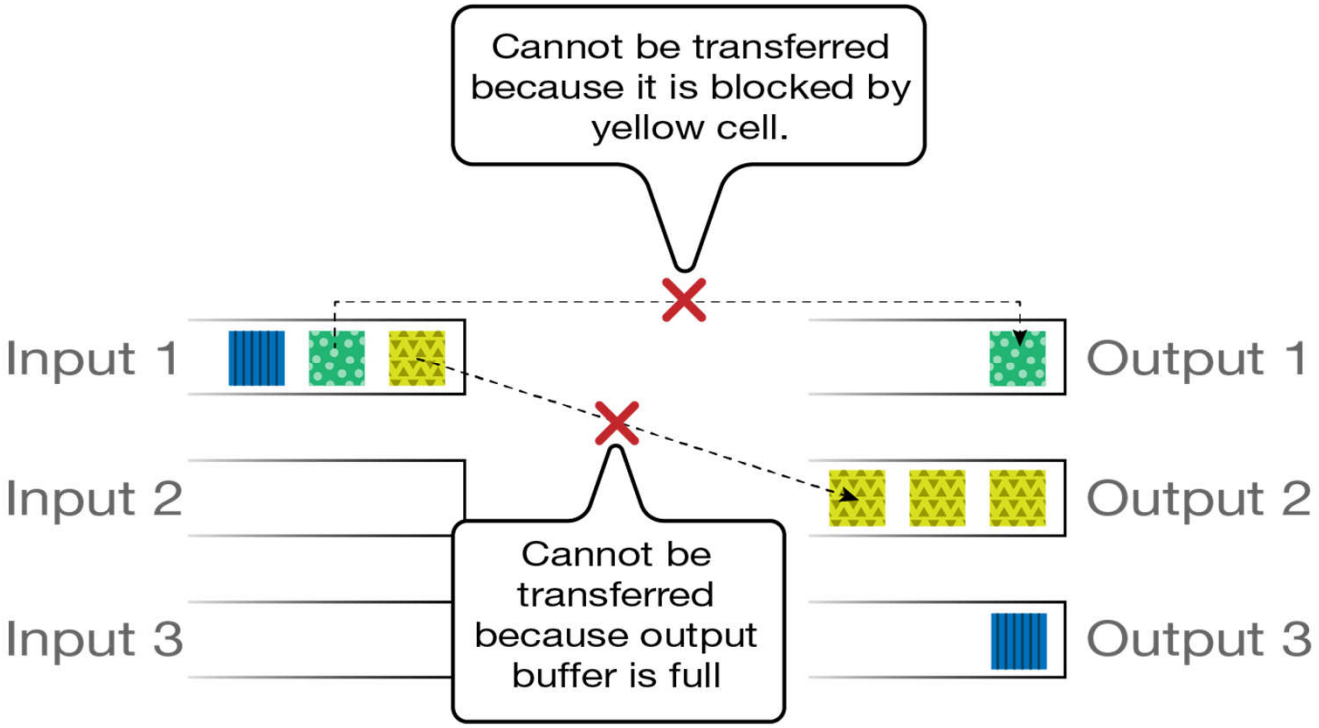
- Cell at the head of the queue can't be sent, blocking cells behind it that could otherwise be sent

Head-of-Line Blocking

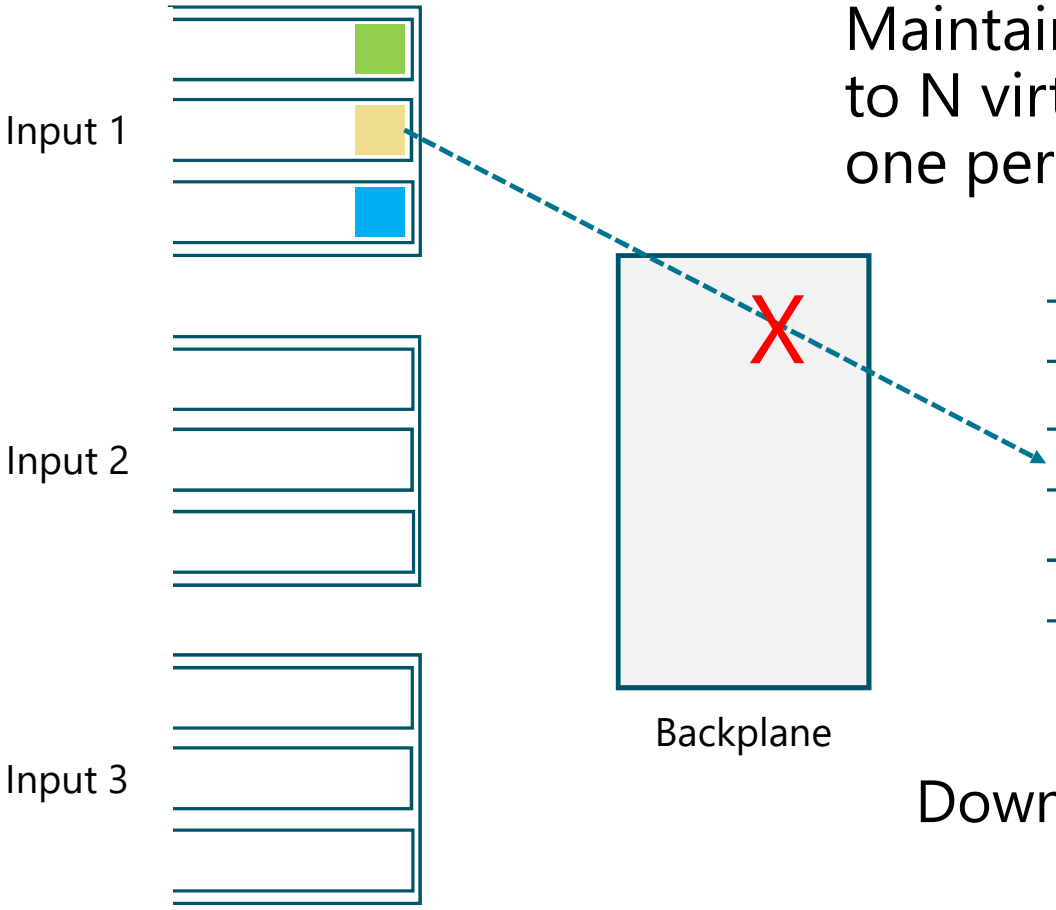


Head-of-Line Blocking

- Cell at the head of the queue can't be sent, blocking cells behind it that could otherwise be sent



Head-of-Line Blocking

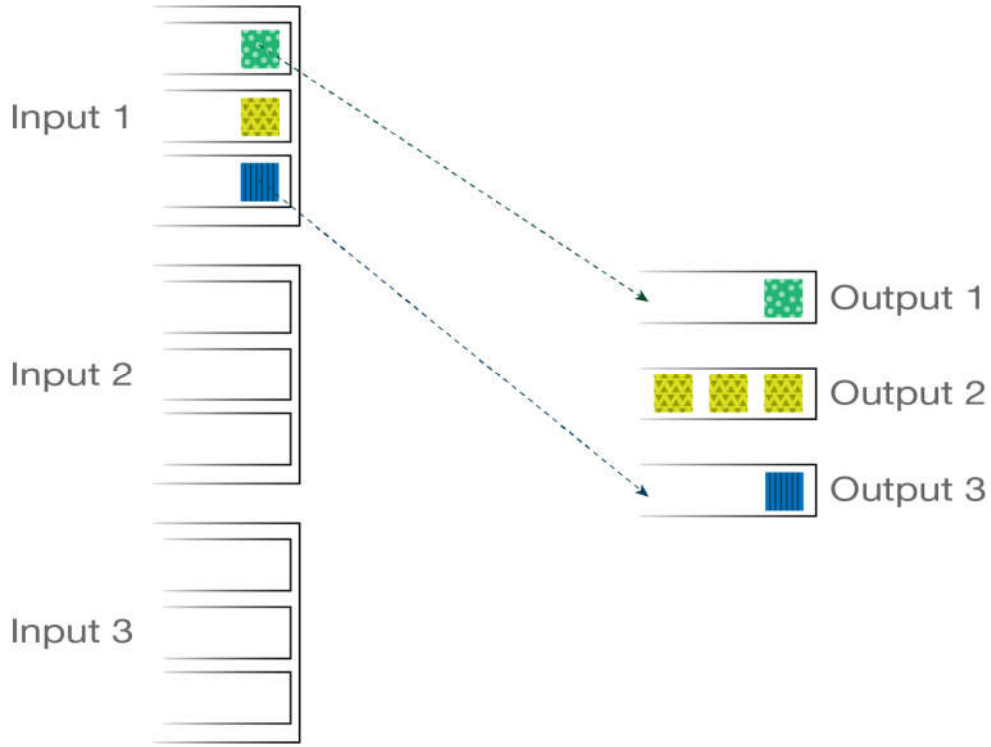


Maintain at each input up to N virtual queues, e.g., one per output

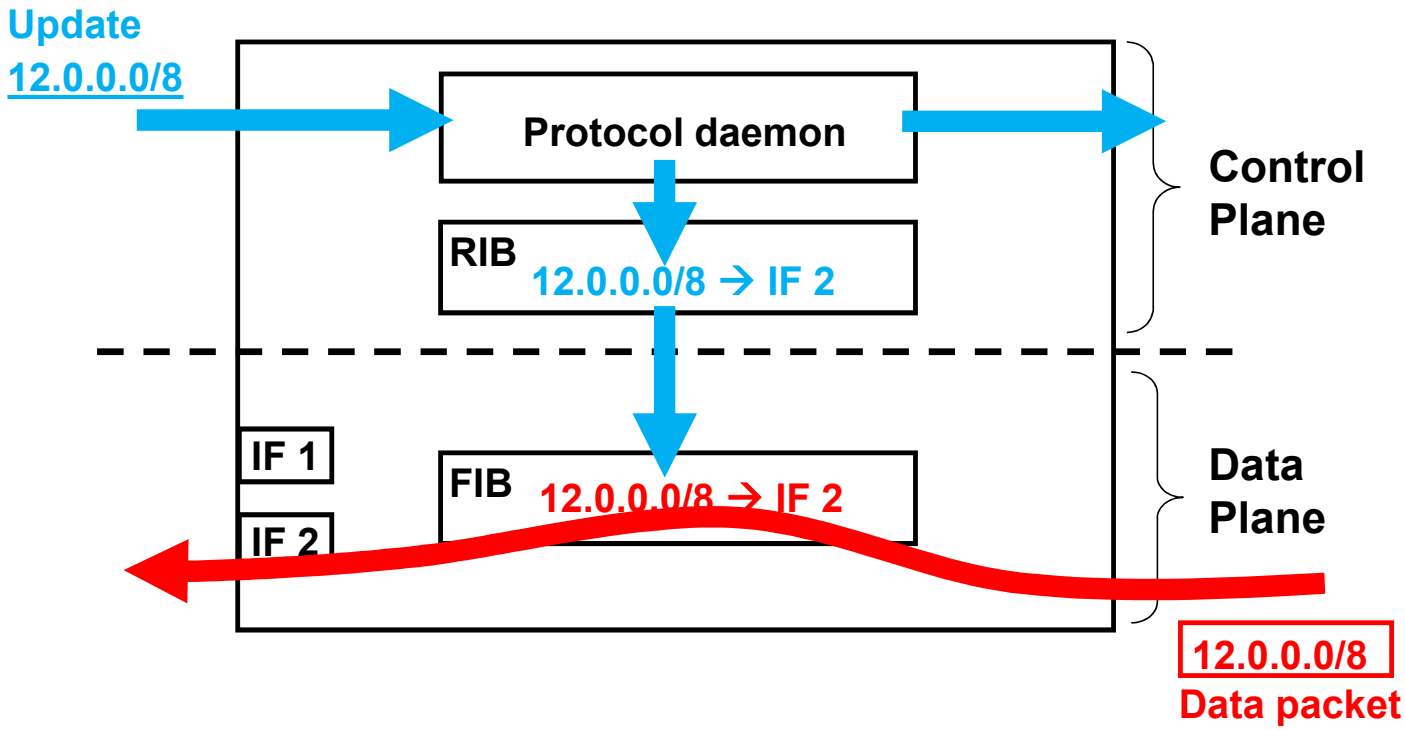
Downsides: increases port cost

Mitigating Head-of-Line Blocking with Virtual Queues

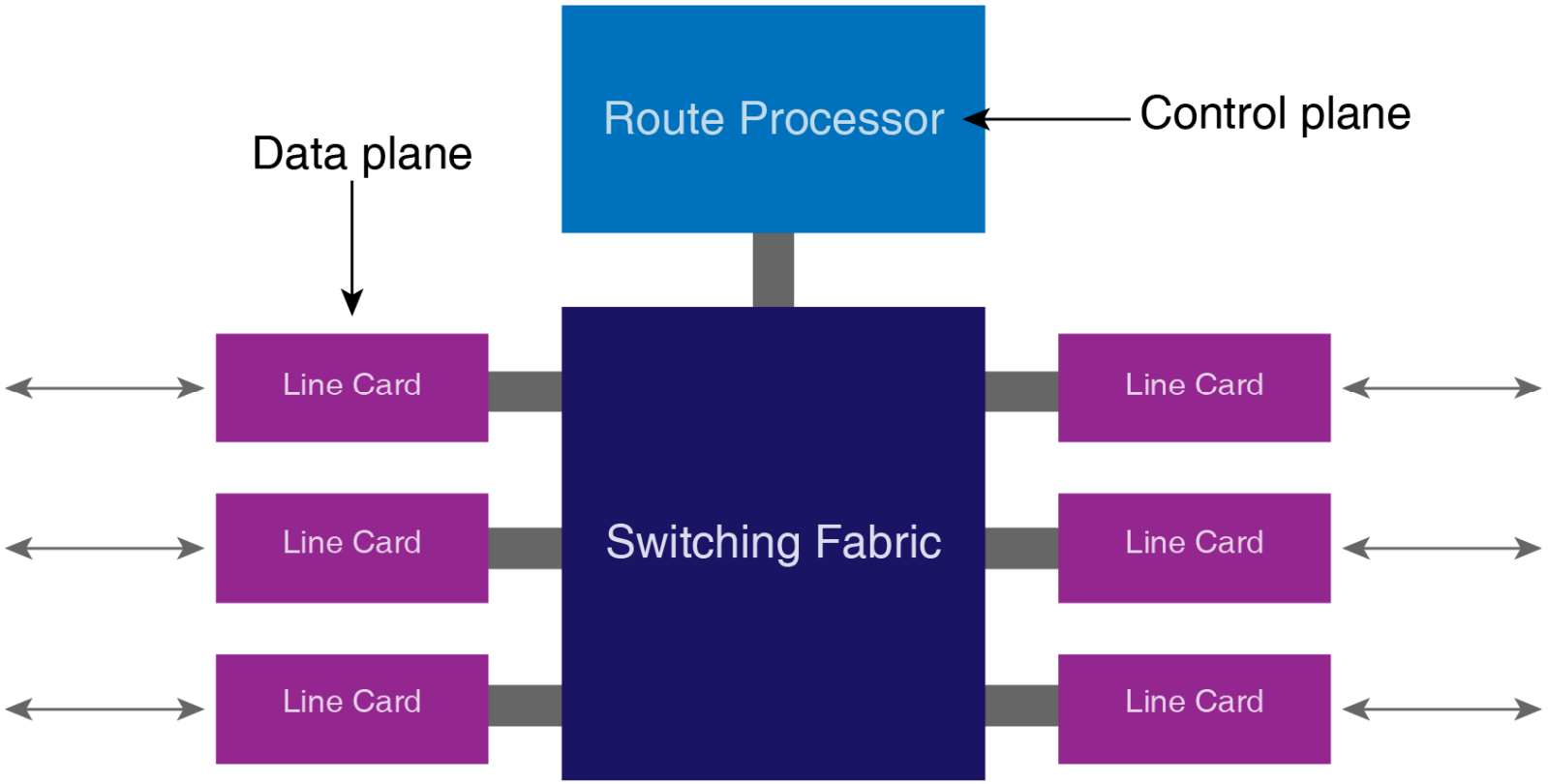
- Maintain at each input N virtual queues, i.e., one per output



How the Control and Data Planes Work Together (Logical View)



Physical Layout of a High-End Router



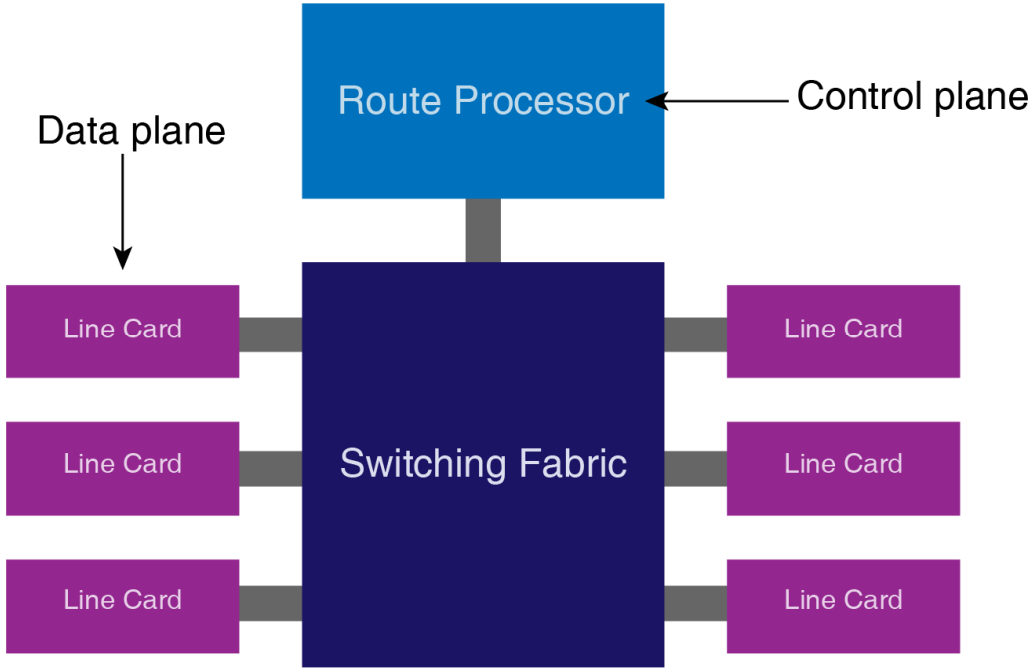
Routing vs. Forwarding

- Routing is done in software (“control plane”)
  - Computing paths the packet will follow
  - Computation performed on router’s CPU
  - Creates forwarding table as output

- Forwarding is (usually) done in hardware (“data plane”)
  - Directs packet to outbound interface
  - Uses forwarding table created by control plane

Routing vs. Forwarding

- Control plane's job:
  - Running routing protocols
  - Maintaining routing table
- On commercial routers, control plane runs on special-purpose "route processor"
  - IP forwarding is distributed across the interface cards



Slotted Chassis

- Large routers are built in a slotted chassis
  - Interface (line) cards are inserted in the slots
  - Route processor is also on a line card
- This simplifies repairs and upgrades of components
  - E.g., “hot-swapping” of components

Router with slotted chassis

Line card

Device Protocols

- Routing: OSPF, BGP, IS-IS…
- Multicast: PIM-SM, DVMRP, PIM-DM, …
- VPN/Tunneling: GRE, MPLS, VXLAN, L2TP, …
- Forwarding: IPv4, IPv6, …
- Spanning tree: STP, RSTP, PVRSTP+, …
- Wireless: 802.11n, 802.11g, 802.11b, ….
- Router Failover: HSRP, VRRP, GSRP, …
- Link Failover: LACP, vPC, PortChannel, …
- QoS: CoS, DSCP, wrr, wred, …
- Administration: SSH, SNMP, …

- Tcp udp, sockets, http and displaying web pages
- Lec 1 is video+syllabus, 2 is Arduino, then ckts

[title: Wiring]

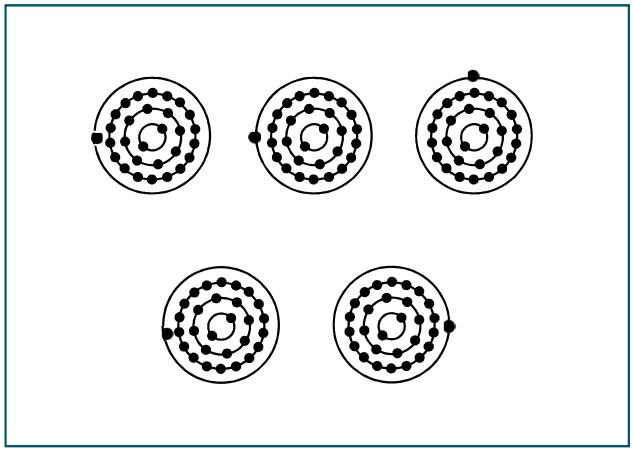
[week X: video X]

[status: done]

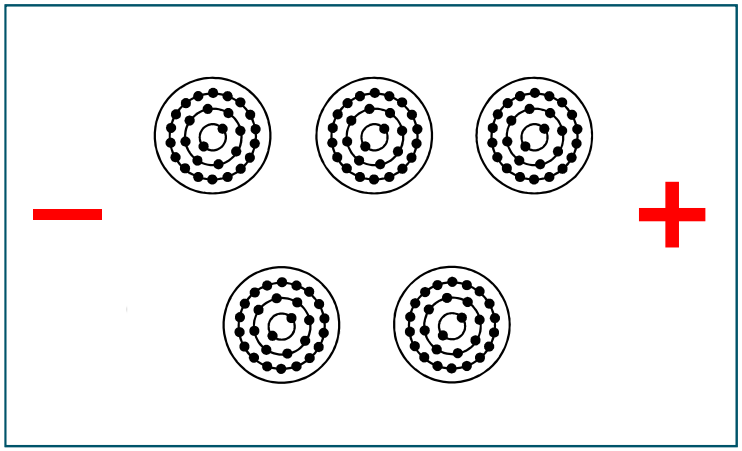
Physical IoT

Background: Electrical Current

- Usually free electrons hop around randomly
- However, outside forces can encourage them to flow in a particular direction
  - Magnetic field, charge differential → this is called **current**
  - We can vary properties of current to transmit information (via waves, like dominos, as electron **drift velocities** are very slow)

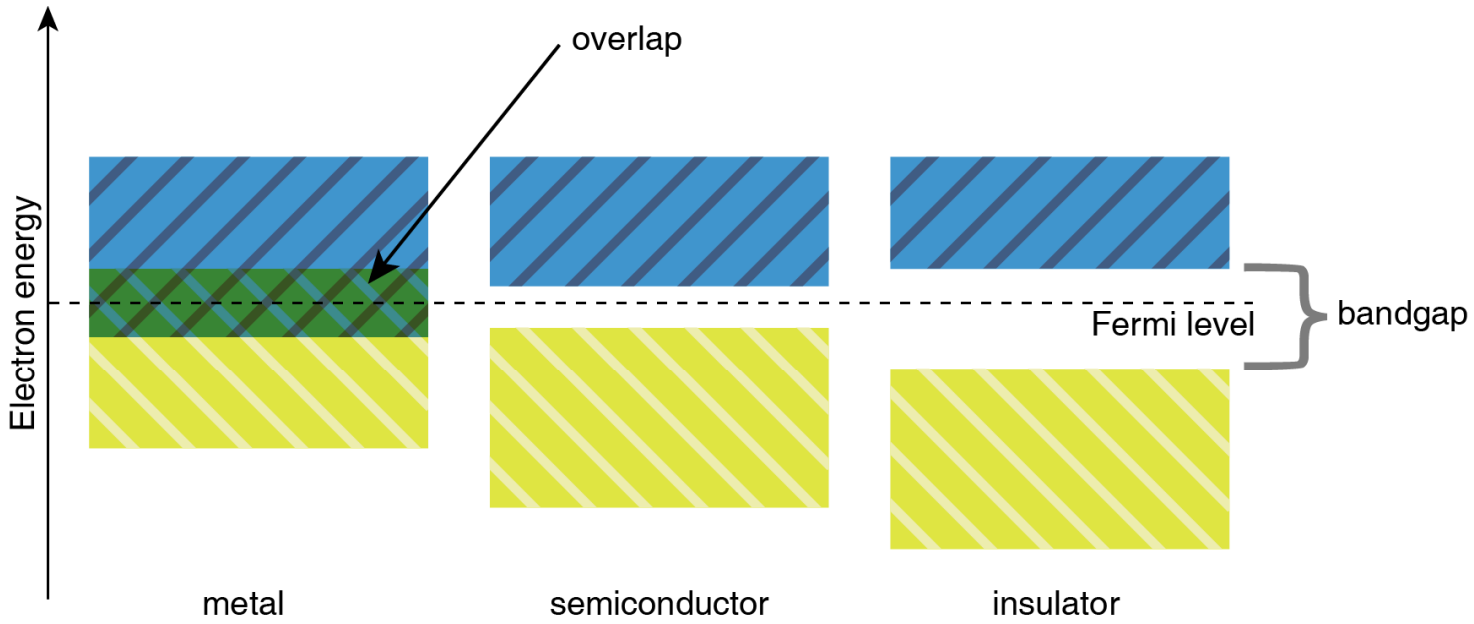


No charge differential



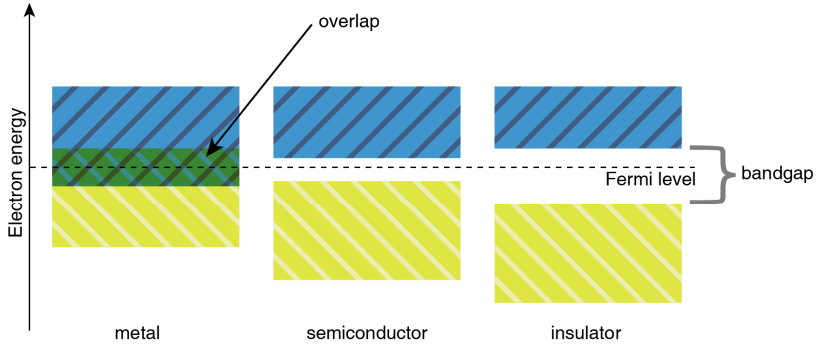
Charge differential

Conductors vs. Insulators



Conductors vs. Insulators

- Conductor: valence electrons wander around easily
  - Used to carry signal in cables
  - Copper, aluminum
- Insulator: valence electrons tightly bound to nucleus
  - Separates conductors physically and electrically
  - Glass, plastic, rubber
- Semiconductor: conductivity between insulator and conductor
  - Used within electronic components, ICs
  - Can be easily made more conductive by adding impurities



Material	Resistivity (ohm m)
Glass	10^{12}
Mica	$9*10^{13}$
Quartz	$5*10^{16}$
Copper	$5*10^{-8}$

Common conductors

Copper

- Cheap, lower strength
- Lower operating temperature (subject to melting)
- Subject to corrosion from air (oxidation) and water
- Better at conducting heat than aluminum (worse at radiating it)
- Good for wiring, flexible parts

Aluminum

- Lower conductivity than copper, harder to handle load surges
- Less malleable; leads to creep/loosening at connection sites, breaks easier, poor choice for flexible wire
- Lightweight, reduced corrosion
- Good for cases, corrode-resistant wiring/pipes/foils, heatsinks

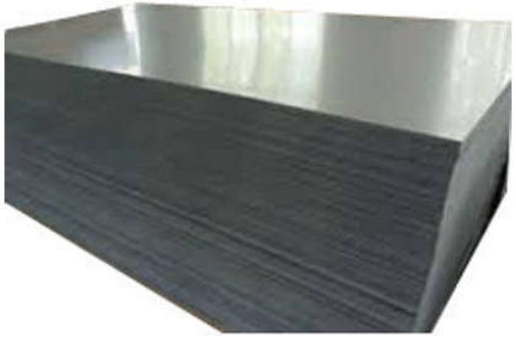
Silver

- Of all metals – best thermal and electrical conductor
- Can withstand high temperatures
- Very expensive. Poor corrosion.
- Very reflective – good for high quality mirrors, optical storage
- Very pliable - good for very thin wires and intricate parts

Common conductors

Nickel

- Improved strength, corrosion resistance. Very high melting point (1453°C). Magnetic
- Tends to react with other materials (catalytic) – good at forming alloys
- Very ductile – good for intricate components
- Higher resistance to electricity and heat.



Tin

- Very soft, malleable, ductile
- Improved durability and strength
- Worse corrosion resistance to acids and alkalis. Good corrosion resistance to water.
- Low melting point
- Good for joining connections (solder), malleable enclosures

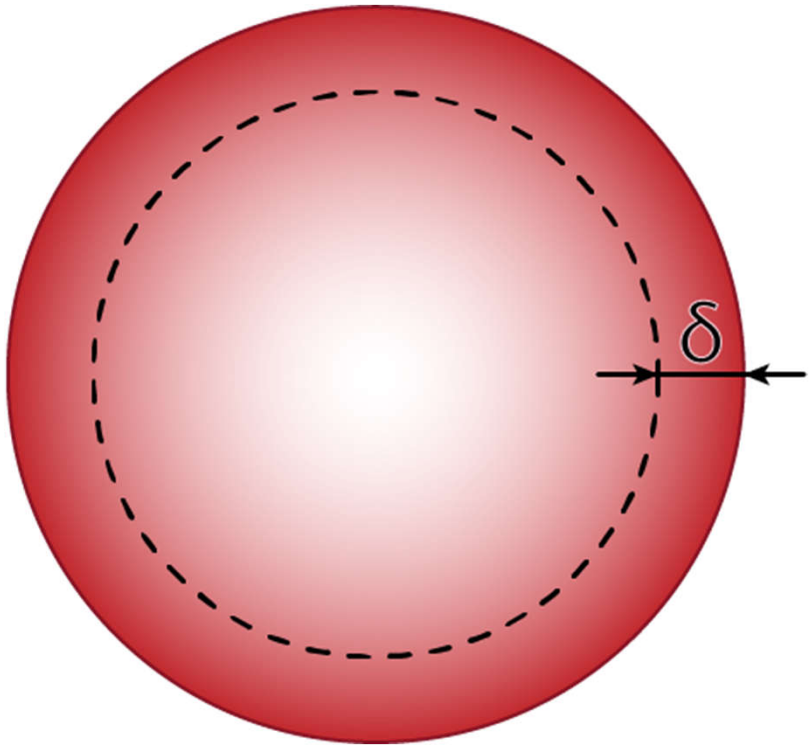
Steel

- Much worse thermal and electrical conductivity than aluminum
- Much harder than aluminum
  - Stronger, heavier
  - Harder to create shapes
- Subject to rust/corrosion when exposed to water
- Good for reinforcing infrastructure: struts, cables, enclosures

Coating Copper to Improve Resilience

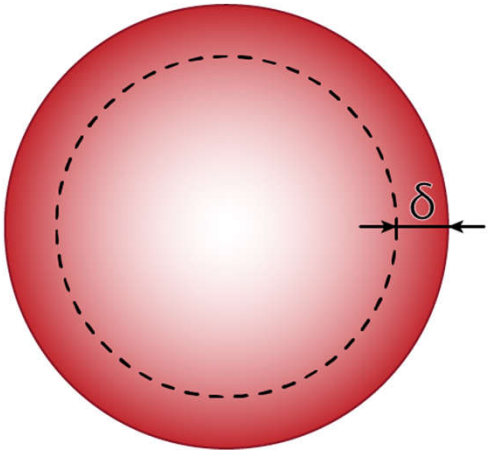
- **Coating** copper can provide additional properties
  - Done by “hot dipping” or electroplating
- **Tinned copper**: corrosion protection, easier to solder
  - Industrial Ethernet deployments, environments exposed to water such as ships
- **Silver plated copper**: better conduction, operation over wider temperature range (-65°C to 200°C). Commonly used in aerospace applications
- **Nickel-plated copper**: corrosion protection, operation over wider temperature range (thick plating can withstand 750 deg C), reduced high-frequency loss

Reducing Resistance from the Skin Effect



Reducing Resistance from the Skin Effect

- Alternating electric current flows mainly at the “skin” of the conductor
  - Due to “turbulent” eddy currents caused by changing magnetic field
- Stranding helps, but not as much as you might think
  - Touching surface area acts like single conductor
  - Individually-insulating strands (Litz wire) helps
- Coating with low-resistance material can leverage this property
  - E.g., silver-tinned copper

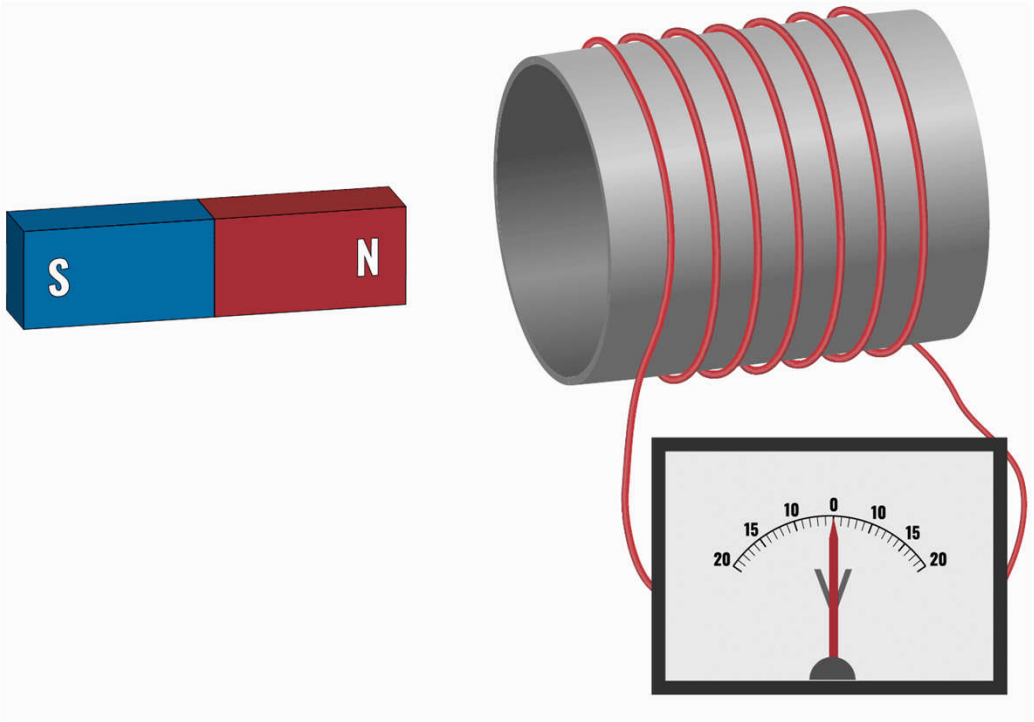


Improving Strength with Stranding

- **Solid** vs. **Stranded** conductors
  - **Solid:** inexpensive and tough, solid seating into jacks and insulation
  - **Stranded:** increased flexibility and flex-fatigue life, increased conductivity

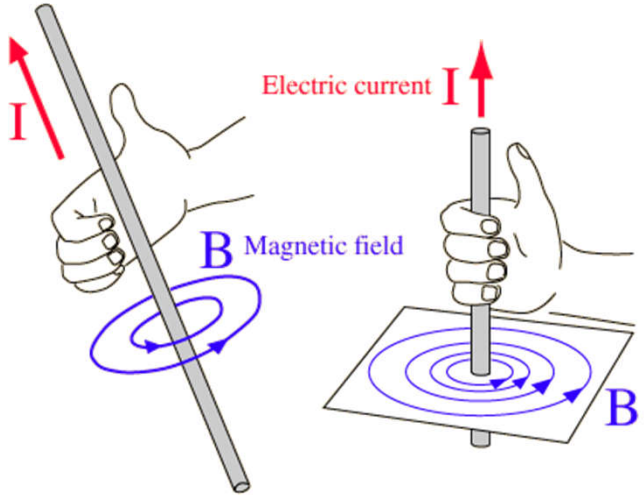
Stranding Types

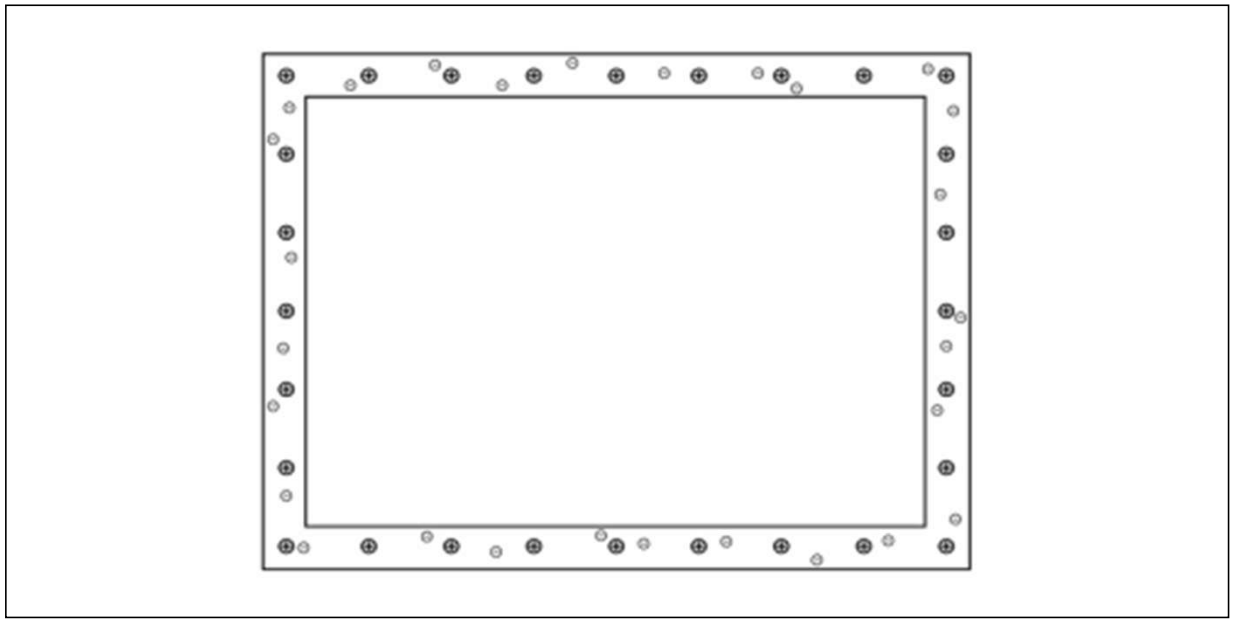
- Stranding type affects wire properties
  - **Bunched:** inexpensive and simple to build, can be bulkier (circle packing problem)
  - **Concentric:**
    - **Unilay:** lighter weight and smaller diameter; greater torsional flex
    - **Contra-helical:** greater mechanical strength and crush resistance; greater continuous flex
    - More twists → improve strength
- Ethernet comes in both solid (plenum and static runs) and stranded (standard, patch panel, etc.)


S
N
20
15
10
0
10
15
20

Noise, Jamming, and Information Leakage

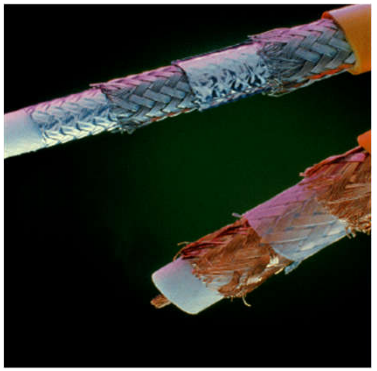
- When you move a conductor through a magnetic field, electric current is induced (electromagnetic induction)
- EMI is produced from other wires, devices
  - Induces current fluctuations in conductor
  - Problem: crosstalk, conducting noise to equipment, etc.

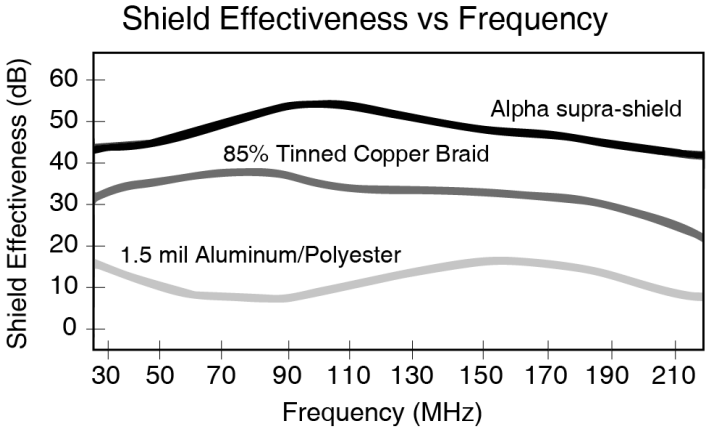


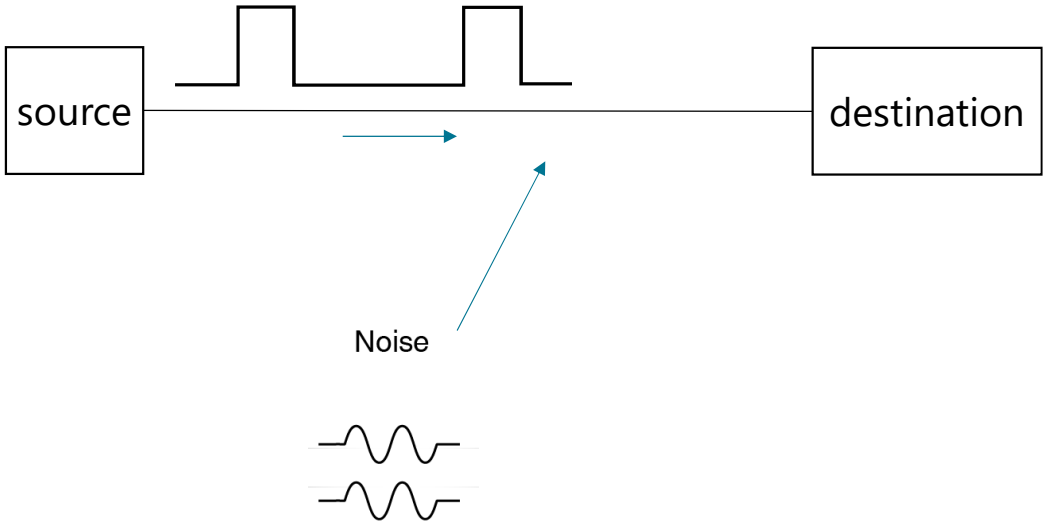


Reducing Noise with Shielding

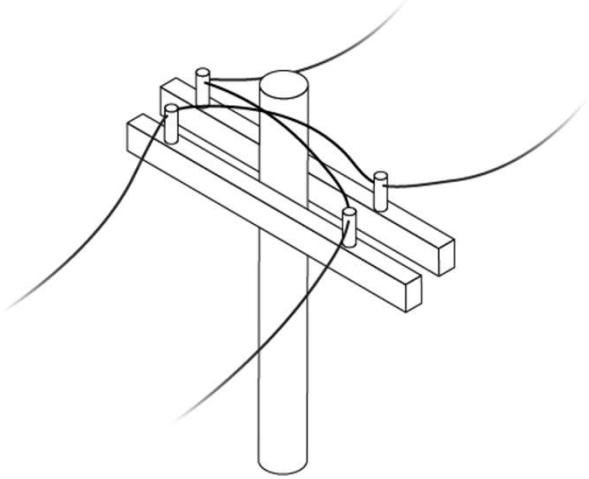
- Enclose insulated conductor with an additional conductive layer (shield)
  - Reflect, absorb (Faraday cage), or conduct EMF to ground
- Metallic foil vs. Braid shield
  - Foil is cheaper but poorer flex lifetime
  - Braid for low freq and EMI, foil for high freq and RFI
  - Foil widely used in commodity Ethernet
  - Combining foil+braid gives best shielding

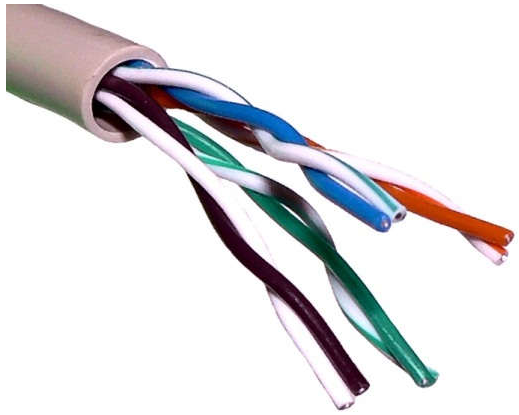





source
destination
Noise

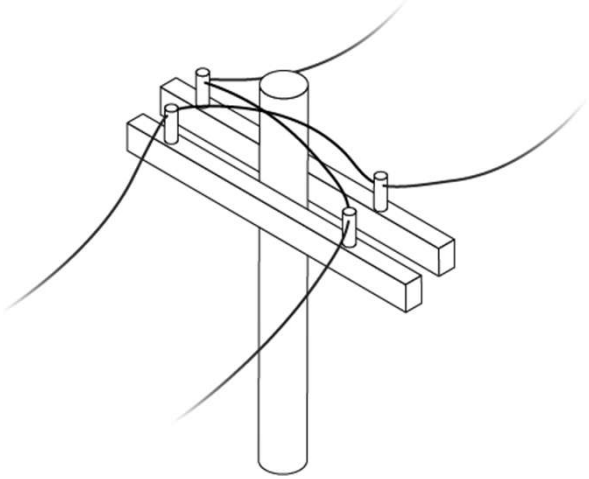
Reducing Noise with Twisted Pairing

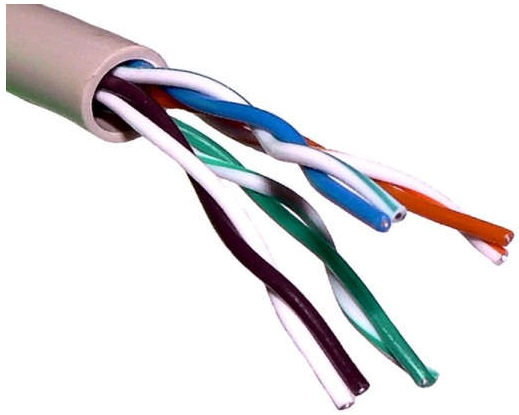




- Differential signaling: transmit complementary signals on two different wires
  - Noise tends to affect both wires together, doesn't change relative difference between signals
  - Receiver reads information as difference between wires
  - Part of Ethernet standard, telegraph wires were first twisted pair

Reducing Noise with Twisted Pairing





- Disadvantages:
  - EMI protection depends on pair twisting staying intact → stringent requirements for maximum pulling tension and minimum bend radius (bonded TP can help)
  - Twisted pairs in cable often have different # of twists per meter → color defects and ghosting on analog video (CCTV)

Insulators

- Insulators separate conductors, electrically and physically
- Avoid air gaps: ionization of air can degrade cable quality
- …


EXIT

Cable Ratings

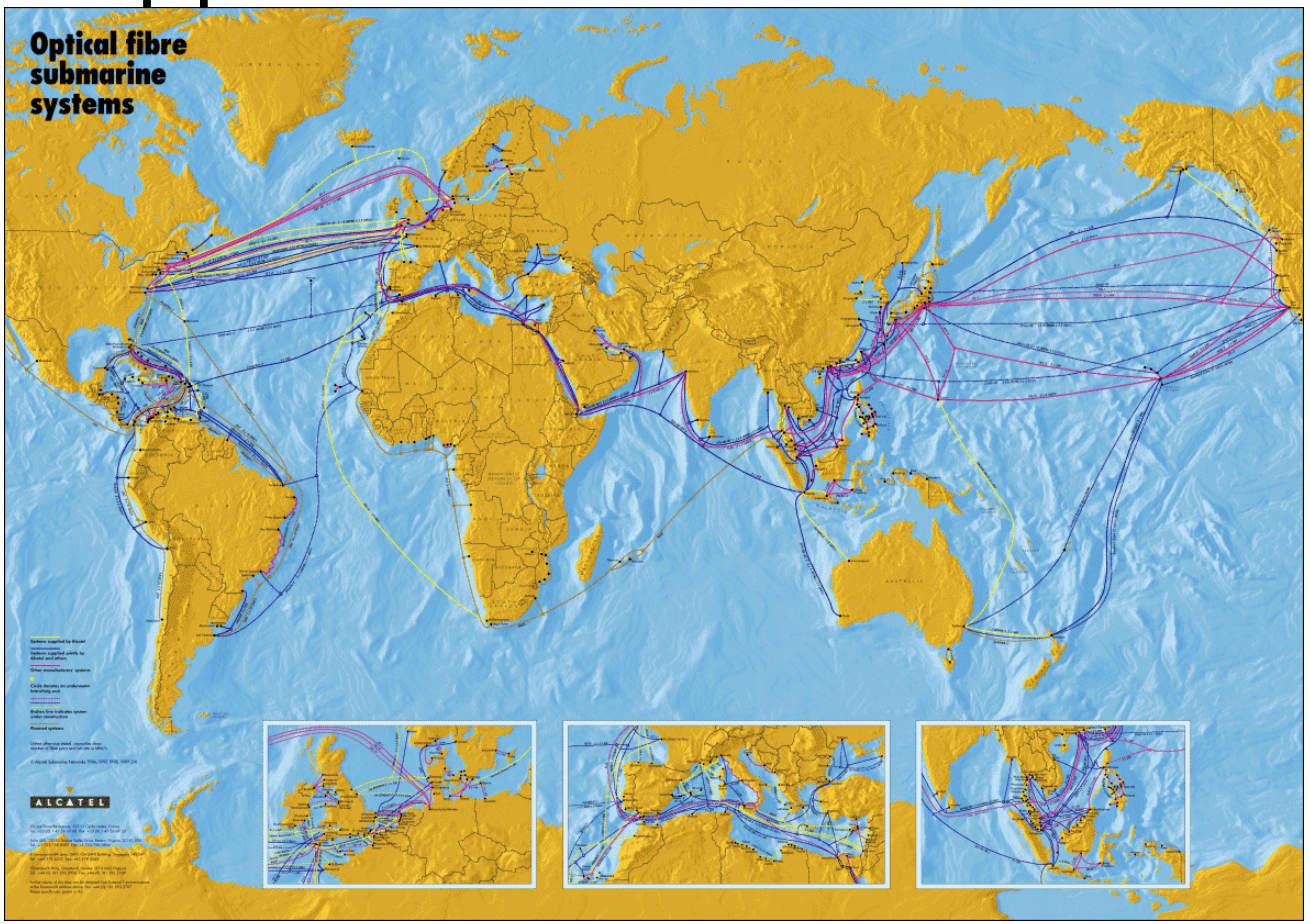
- Plenum rated (toughest rating)
  - National Fire Protection Standard (NFPA) 90A
  - Jacketed with fire-retardant plastic (either low-smoke PVC or FEP)
  - Cables include rope or polymer filament with high tensile strength, helping to support weight of dangling cables
  - Solid cable instead of stranded
  - Restrictions on chemicals for manufacture of sheath → reduced flexibility, higher bend radius, and higher cost

Cable Ratings

- Riser cable: cable that rises between floors in non-plenum areas

- Low smoke zero halogen: eliminates toxic gases when burning, for enclosed areas with poor ventilation or around sensitive equipment

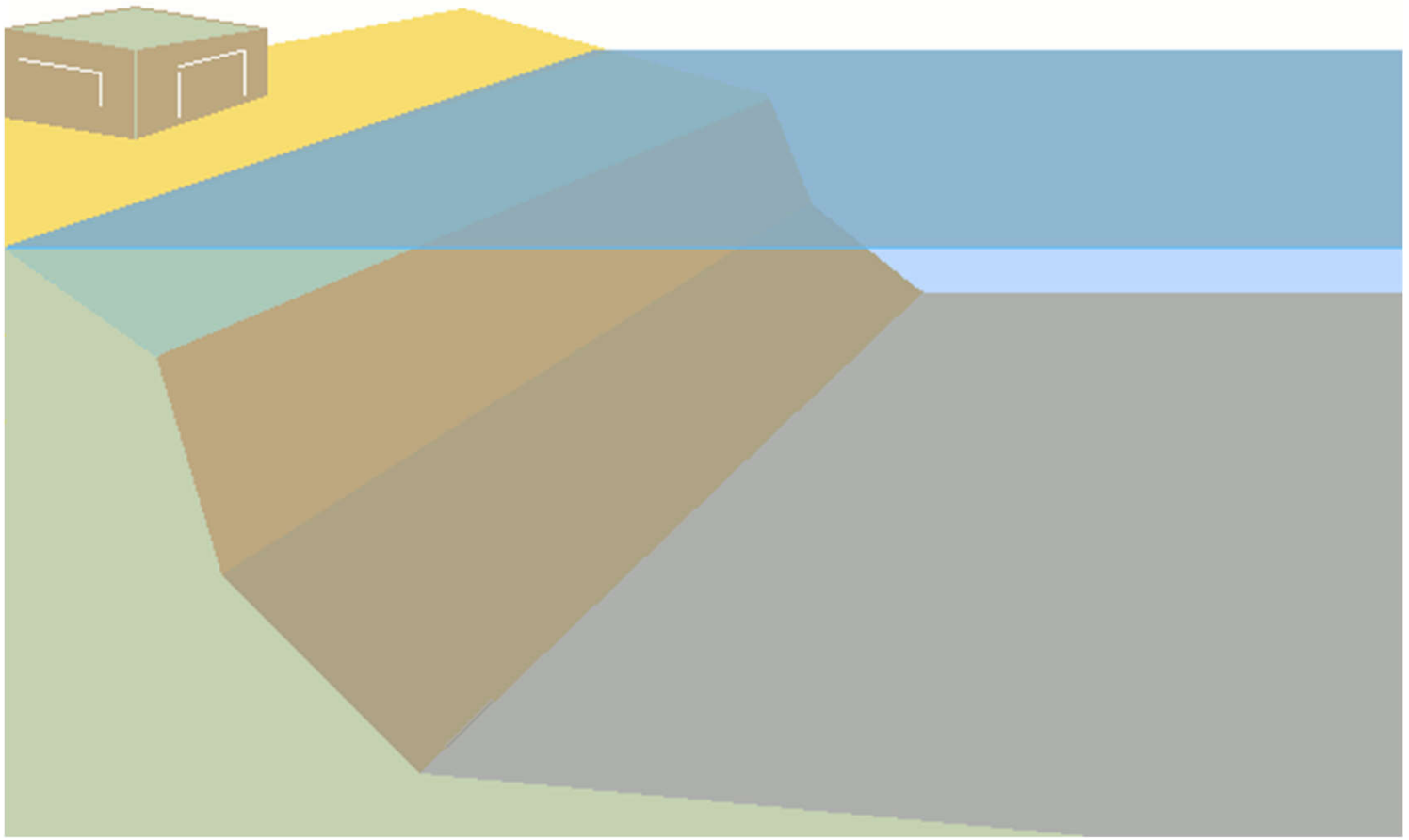
Marine Applications



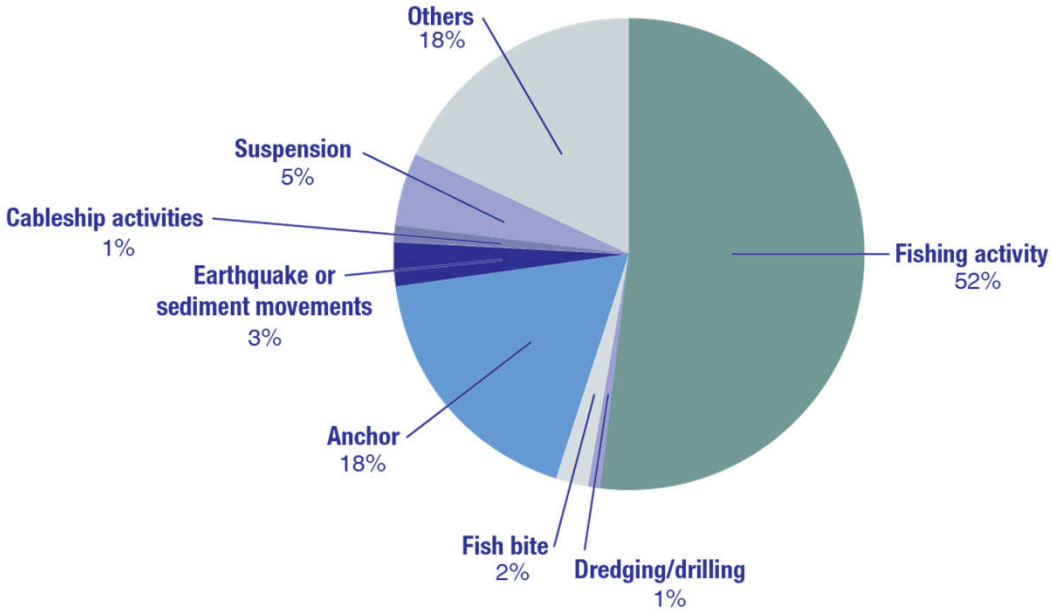
Submarine Cabling


Sea Stallion 433
Sea Stallion 433

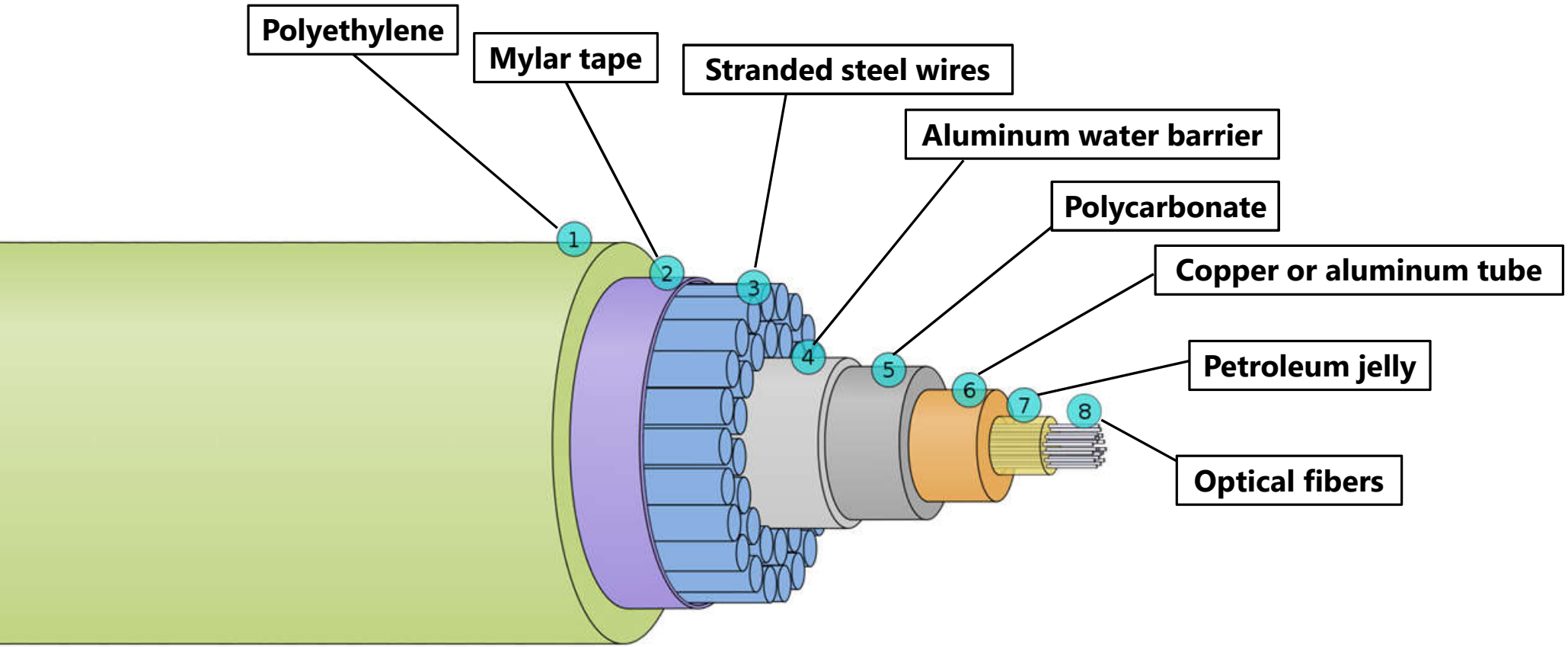
Undersea Cable Laying



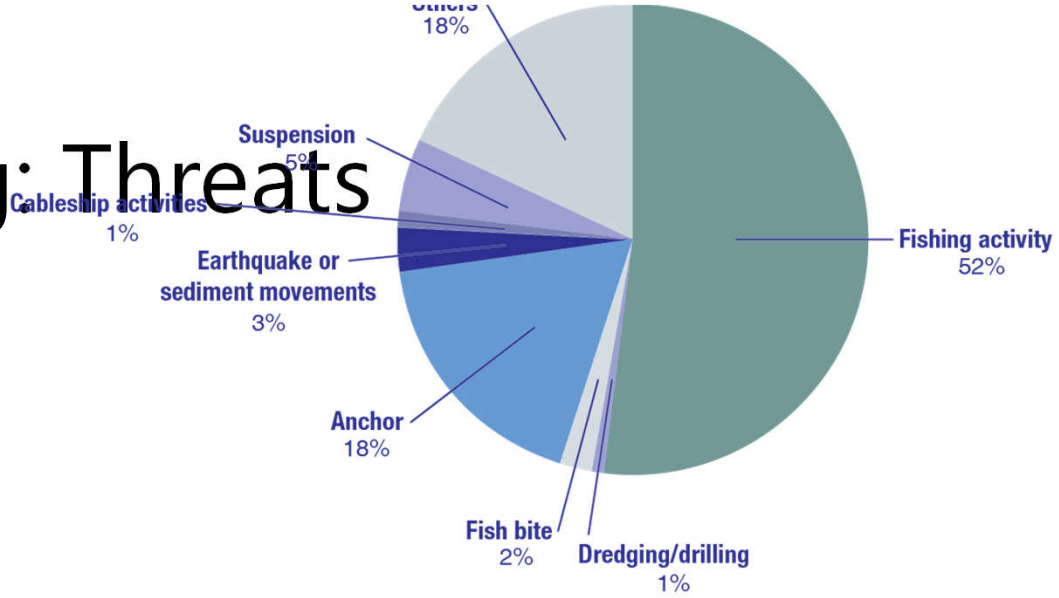
Submarine Cabling: Threats



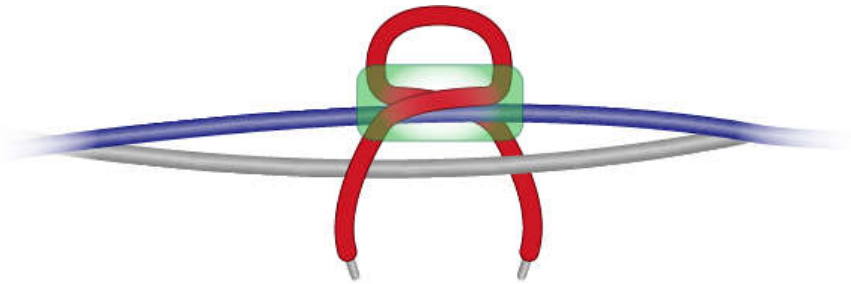
Submarine Cabling: Construction



Cabling: Threats

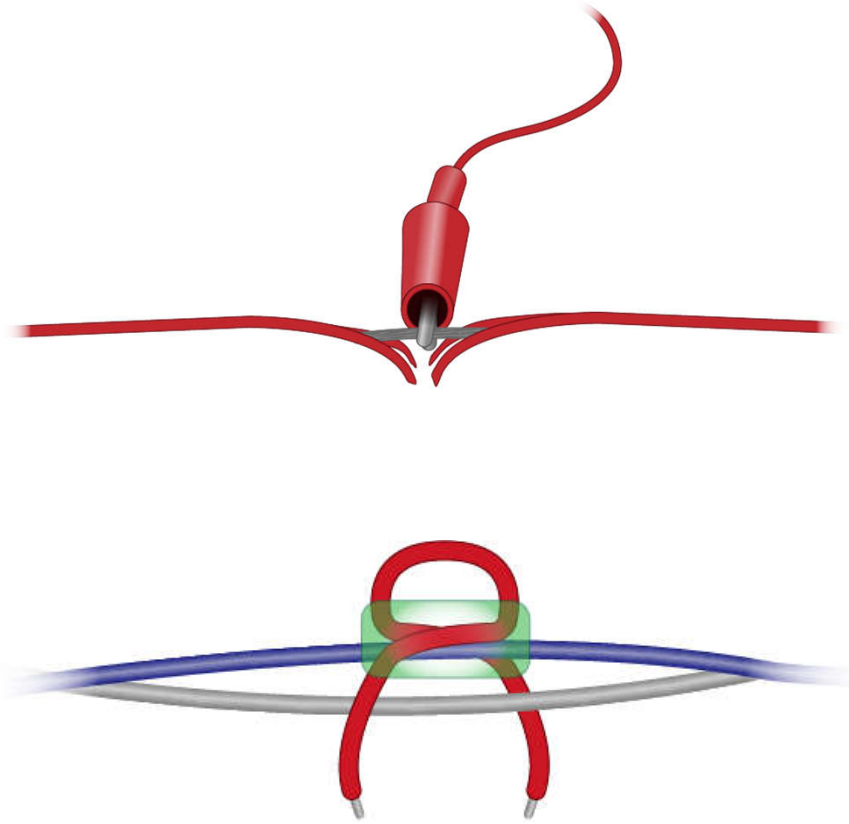


Physical Tapping



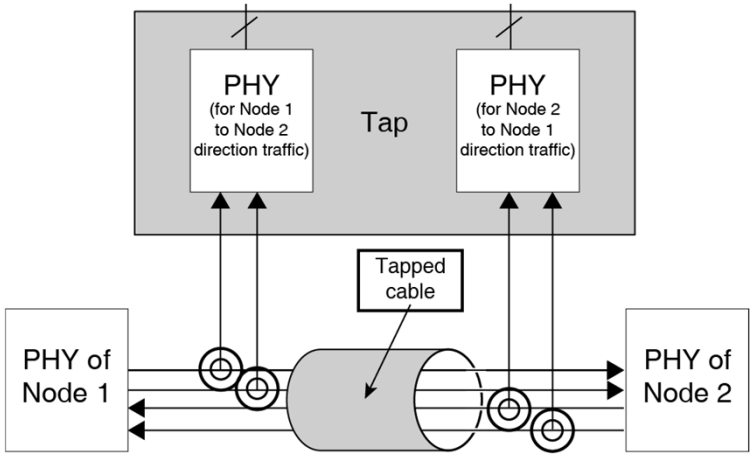
Physical Tapping

- Conductive Taps
  - Form conductive connection with cable
- Inductive Taps
  - Passively read signal from EM induction
  - No need for any direct physical connection
  - Harder to detect
  - Harder to do with non-electric conductors (e.g., fiber optics)



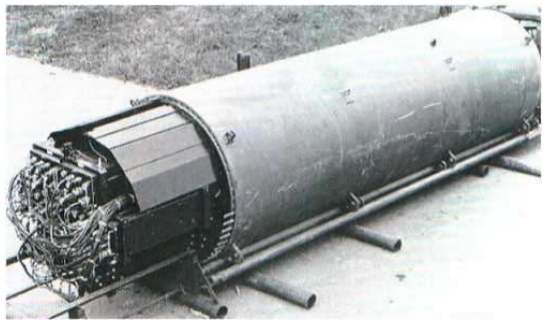
Tapping Cable: Countermeasures

- Physical inspection
- Physical protection
  - E.g., encase cable in pressurized gas
- Use faster bitrate
- Monitor electrical properties of cable
  - TDR: sort of like a hard-wired radar
  - Power monitoring, spectrum analysis



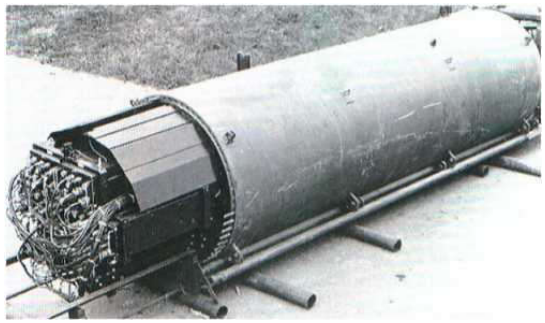
Case Study: Submarine Cable (Ivy Bells)

- 1970: US learned of USSR undersea cable
  - Connected Soviet naval base to fleet headquarters
- Joint US Navy, NSA, CIA operation to tap cable in 1971
- Saturation divers installed a three-foot long tapping device
  - Coil-based design, wrapped around cable to register signals by induction
  - Signals recorded on tapes that were collected at regular intervals
  - Communication on cable was unencrypted
  - Recording tapes collected by divers monthly



Case Study: Submarine Cable (Ivy Bells)

- 1972: Bell Labs develops next-gen tapping device
  - 20 feet long, 6 tons, nuclear power source
  - Enabled
- No detection for over a decade
  - Compromise to Soviets by Robert Pelton, former employee of NSA
- Cable-tapping operations continue
  - Tapping expanded into Pacific ocean (1980) and Mediterranean (1985)
  - USS Parche refitted to accommodate tapping equipment, presidential commendations every year from 1994-97
  - Continues in operation to today, but targets since 1990 remain classified

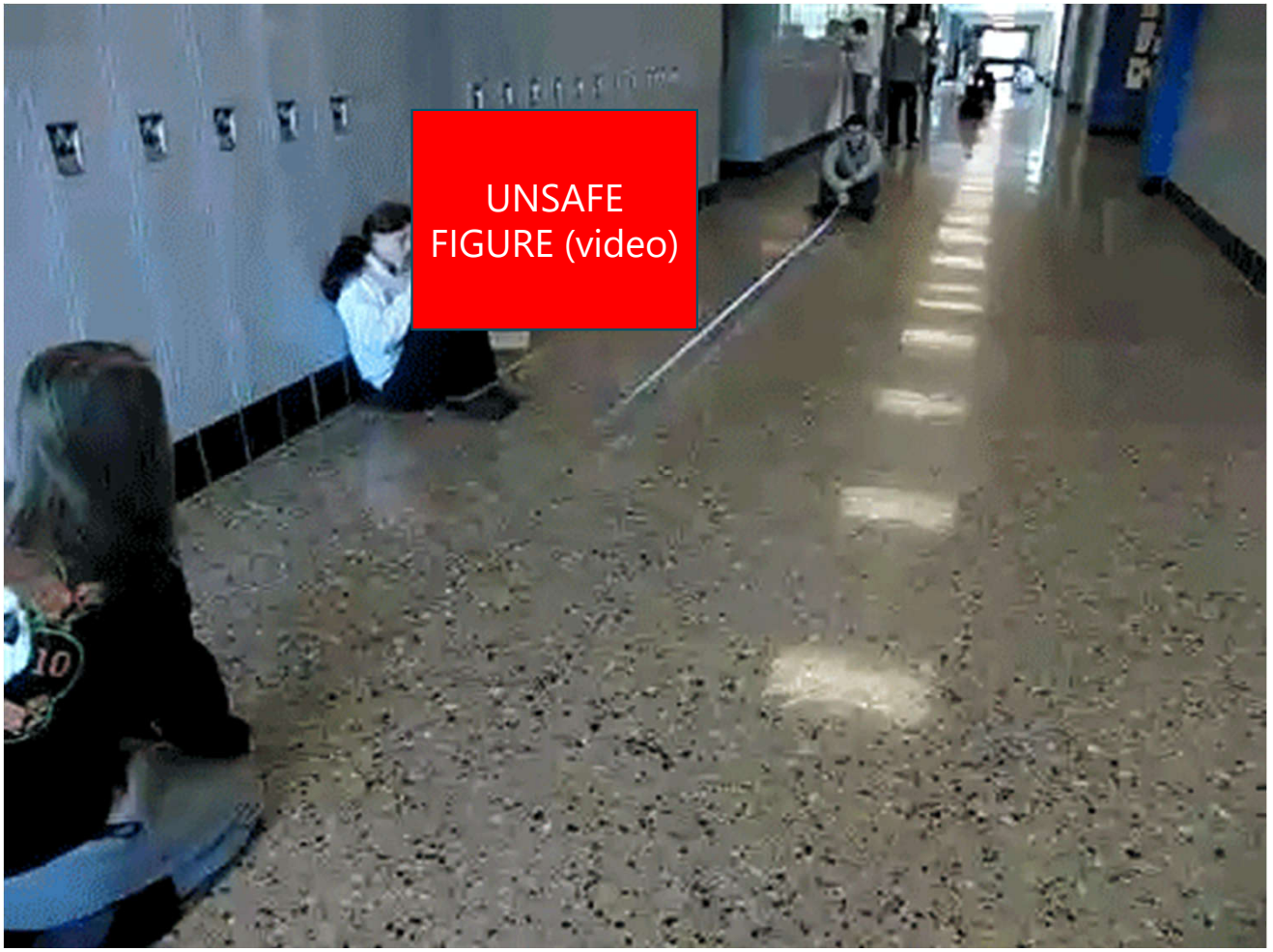


Locating Anomalies with Time-Domain Reflectometry (TDR)

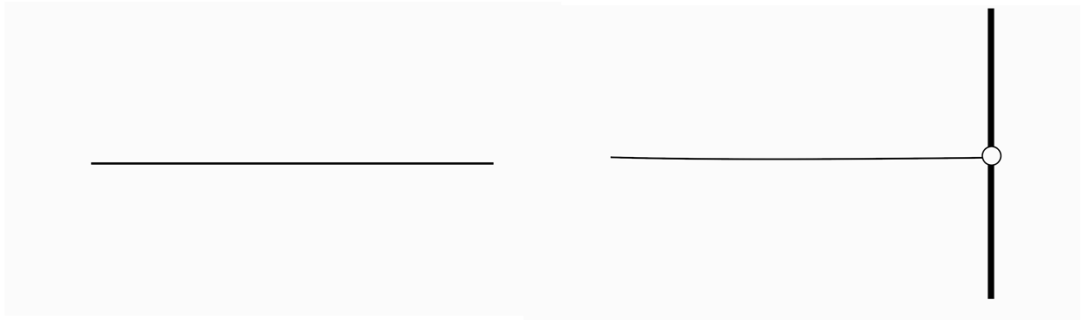
- A tool that can detect and localize variations in a cable
  - Deformations, cuts, splice taps, crushed cable, termination points, sloppy installations, etc.
  - Anything that changes impedance

- Main idea: send pulse down wire and measure reflections
  - Delay of reflection localizes location of anomaly
  - Structure of reflection gives information about type of anomaly

Motivation: Wave Pulse on a String



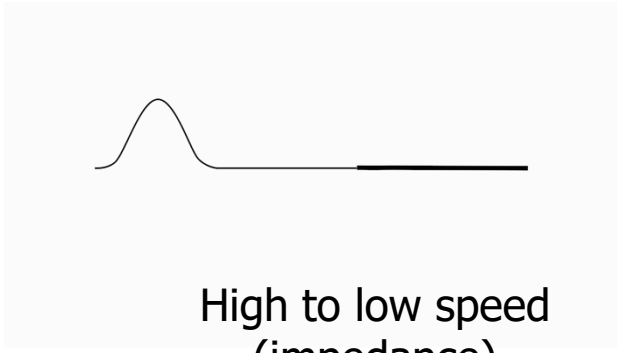
Motivation:
Wave Pulse on a String



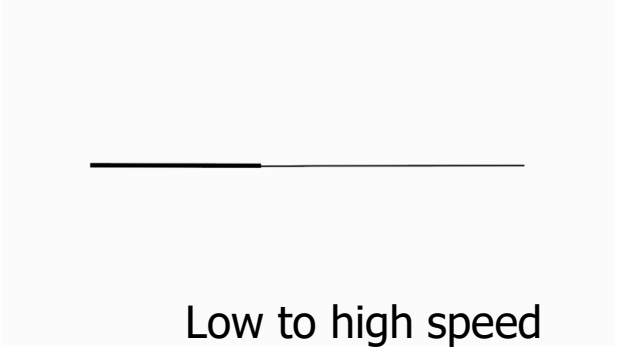
No termination

Reflection from
soft boundary

Reflection from
hard boundary



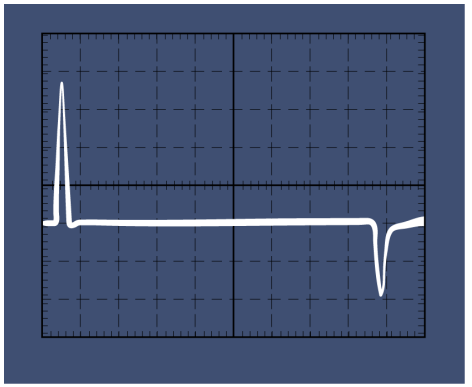
High to low speed
(impedance)



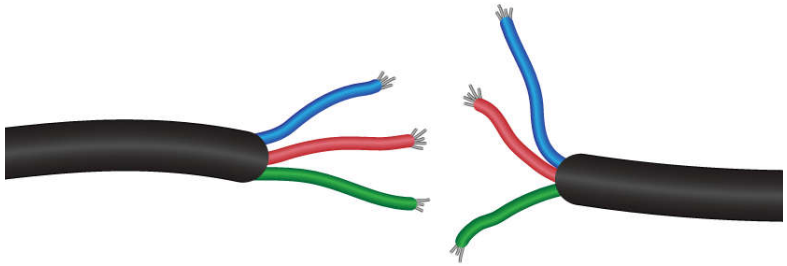
Low to high speed
(impedance)

TDR Examples

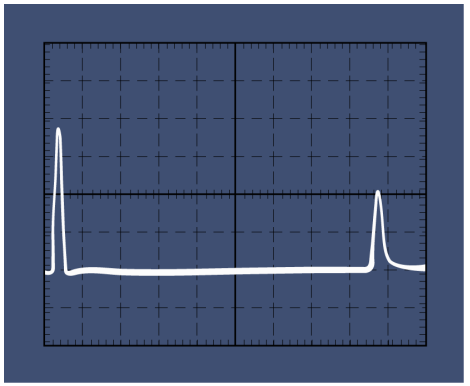
Melted cable (electrical short)



TDR: Inverted reflection

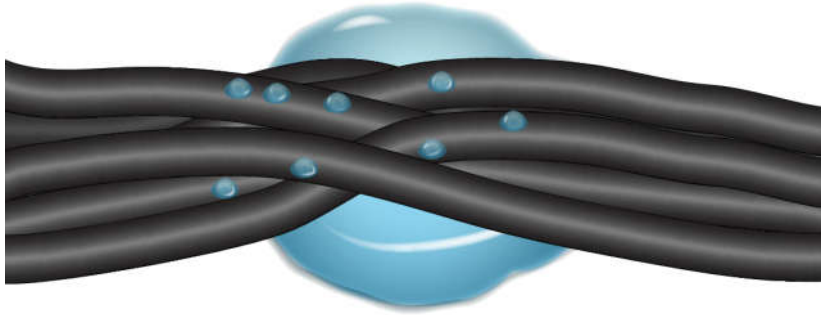


Cut cable (electrical open)

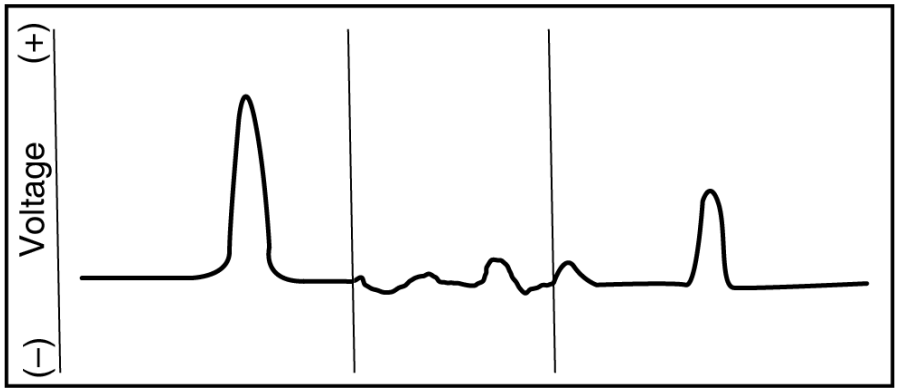


TDR: Reflection

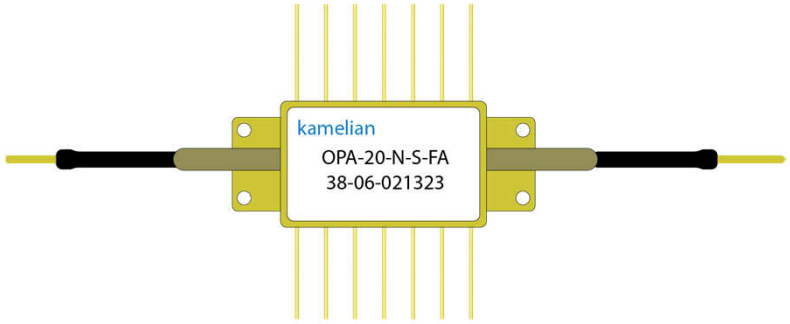
TDR Example: Cable Moisture



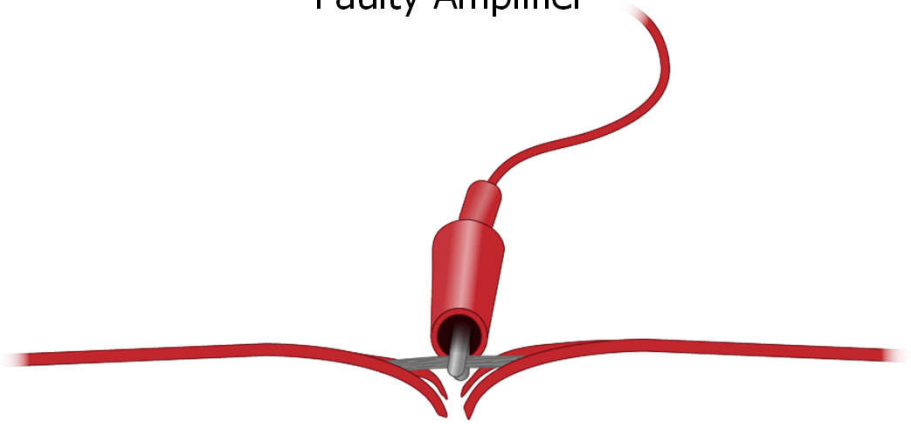
Water-soaked/flooded cable



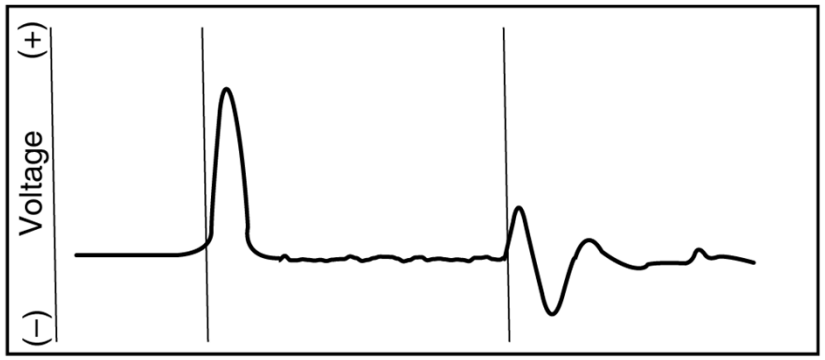
TDR Examples



Faulty Amplifier



Wire Tap



Protection against Wildlife


Rodents

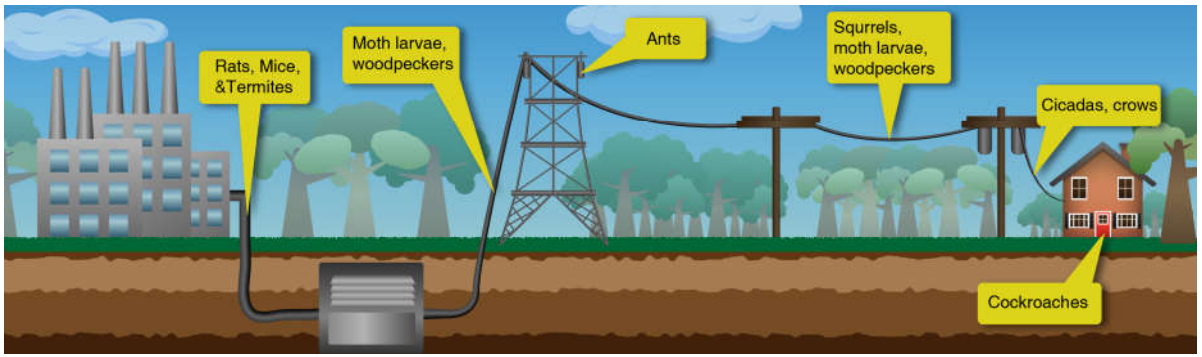

Ants

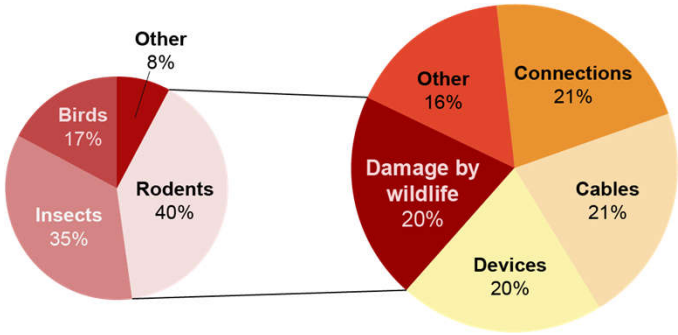

Moths


Cicadas


Crows

Protection against Wildlife





- Rodents (squirrels, rats, mice, gophers)
  - Chew on cables to grind fore teeth to maintain proper length
- Insects (cicadas, ants, roaches, moths)
  - Mistake cable for plants, burrow into it for egg laying/larvae
  - Ants invade closures and chew cable and fiber
- Birds (crows, woodpeckers)
  - Mistake cable for twigs, used to build nests
- Underground cables affected mainly by rats/termites, aerial cables by rodents/moths, drop cables by crows, closures by ants

Countermeasures against Wildlife

- Use high strength sheath cable
  - PVC wrapping stainless steel sheath
  - Performance studies on cable (gnathodynameter)
- Cable wrap
  - Squirrel-proof covers: stainless steel mesh surrounded by PVC sheet
- Fill in gaps and holes
  - Silicone adhesive
- Use bad-tasting cord
  - PVC infused with irritants
  - Capsaicin: ingredient in pepper spray, irritant
  - Denatonium benzoate: most known bitter compound