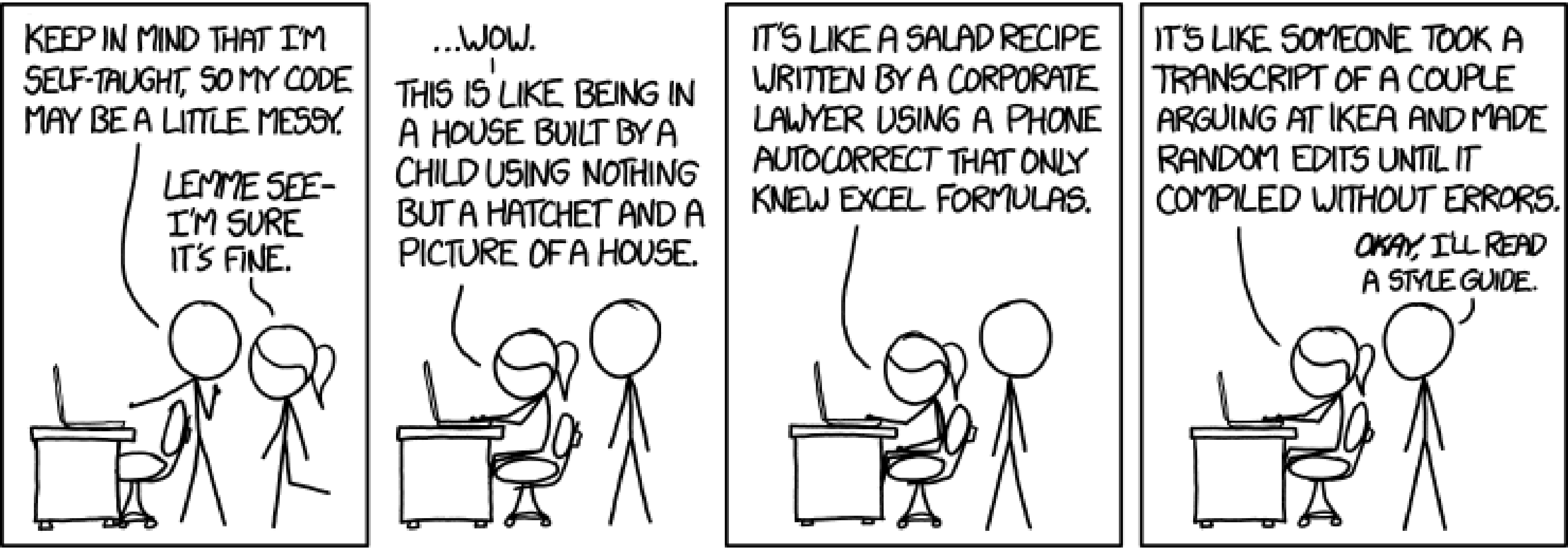


https://xkcd.com/1513/

CS 60:
Computer Networks

Link layer: Data link

Review from last class

Layer 3: routers have two primary functions

1. Forwarding – move packets from input port to output port, quickly!
2. Routing – select the best path from source IP to destination IP

Routing decisions are made:

- ***Within*** an autonomous system:
  - We discussed Open Shortest Path First (OSPF); mentioned others such as RIP and EIGRP
  - Shortest path from router to all other routers in the AS calculated using Dijkstra's algorithm
- ***Between*** autonomous systems: calculate paths between AS'es using Border Gateway Protocol (BGP)

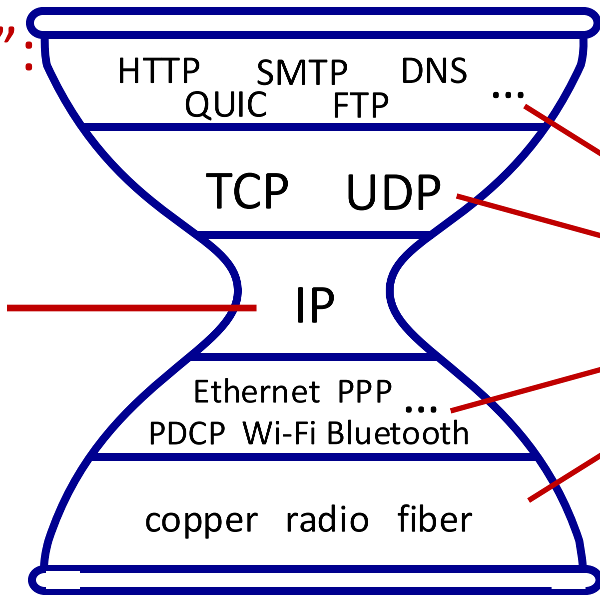
There are two routing paradigms

1. Traditional – each router makes its own decision about selecting routes
2. Software Defined Network – a control plane monitors the network and updates the routing table on each router

The IP hourglass

Internet's "thin waist":

- *One* network layer protocol: IP
- *Must* be implemented by every (billions) of Internet-connected devices
- Each device gets a locally unique IP address



Many protocols in physical, link, transport, and application layers

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Review: Link layer moves frames across a Local Area Network

Conceptual network layers

7) Application	Interacts with application programs to send ***messages*** Applications assigned a port, multiple instances can run (many browser pages) Examples: HTTP, SSH, FTP, SMTP, DNS
4) Transport	Moves ***segments (or datagrams)*** May provide error control, flow control, application addressing (ports) Examples: TCP (connection-oriented), UDP (connectionless) TCP provides sequencing, dropped packet resend, traffic congestion routing
3) Network (IP)	Moves ***packets*** between local area networks (routing) Each computer on the Internet identified by an IP address (IP v4 or v6) Also called Layer 3 or IP layer (ICMP Ping is here)
2) Link (MAC)	Moves ***frames*** within a local area network (switching) Each computer identified by a MAC address on its Network Interface Card (NIC) Also called Layer 2, MAC layer, Data Link layer, or Ethernet layer
1) Physical	How data is physically transmitted • Transmitter converts logical 1 and 0 ***bits*** to electrical/light pulses or phase/amplitude of radio frequency (RF) and sends down wire or over air • Receiver converts electrical/light or RF back to logical 1 and 0 bits

https://www.ibm.com/think/topics/osi-model

Agenda

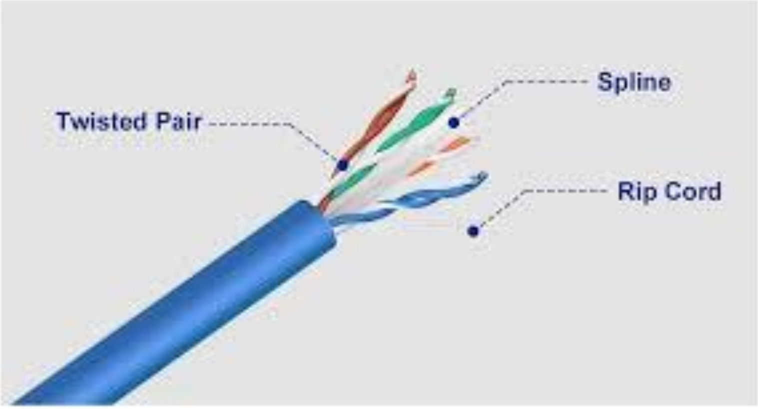
1. Ethernet
2. Putting it all together
3. Error detection/correction
4. Channel sharing

Ethernet is the dominant wired technology used on the Internet

Network infrastructure (routers/switches) is typically connected via wired cable or fiber optic cable

Ethernet is the “dominant” Layer 2 wired LAN technology:

- First widely used LAN technology
- Simpler, cheap
- Kept up with speed race: 10 Mbps – 400 Gbps
- Single chip, multiple speeds
- Full duplex
  - How?



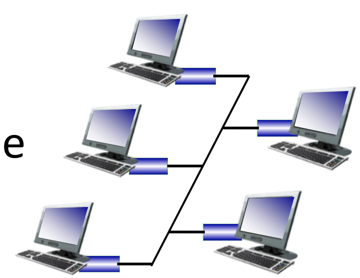
Cable:
2 wires for TX
2 wires for RX

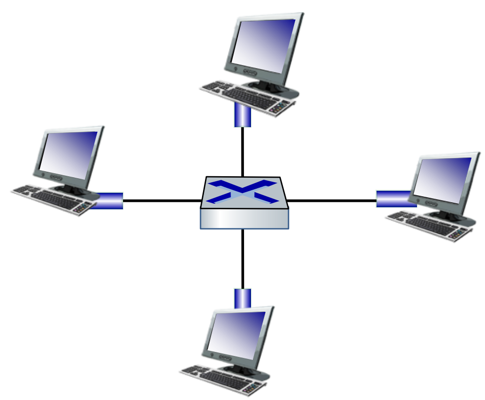
Fiber:
1 fiber for TX
1 fiber for RX

Ethernet's physical topology has moved from a bus to a switched configuration

- Bus: popular through mid '90s
  - All nodes in same collision domain (can collide with each other)
- Switched: prevails today
  - Active link-layer 2 switch logically in the center
  - Point-to-point connection from host to switch
  - Each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)

bus: coaxial cable



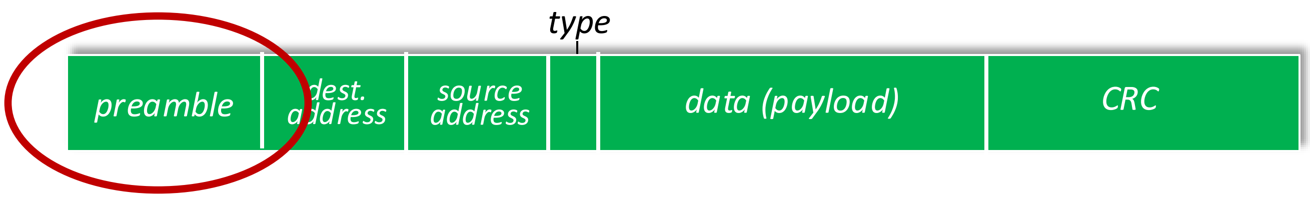


switched

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Ethernet frame structure

Sending interface encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



Preamble:

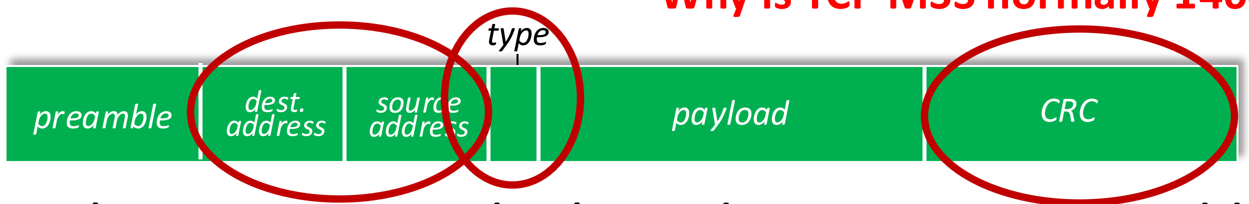
- Used to synchronize receiver and sender clocks
- 7 bytes of 10101010 followed by one byte of 10101011 (last byte is called the Start of Frame Delimiter – SFD)
- After the SFD, the receiver knows “an Ethernet frame is coming next”

Ethernet frame structure (more)

Payload contains IP packet

Ethernet max data size = 1500 bytes (min 46 bytes)

Why is TCP MSS normally 1460?



- Addresses: 6-byte source and 6-byte destination MAC addresses
  - When an adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer
  - Otherwise, adapter discards frame (CPU not interrupted)
- Type: indicates higher layer protocol
  - Mostly IP but others "possible", e.g., Novell IPX, AppleTalk
  - IPv4 type = x0800, IPv6 type = x86DD
  - Used to decode bytes in payload
- CRC: cyclic redundancy check at receiver
  - If error detected: frame is dropped

NIC removes preamble and CRC before passing to layer 2

You will not see these bits in Wireshark

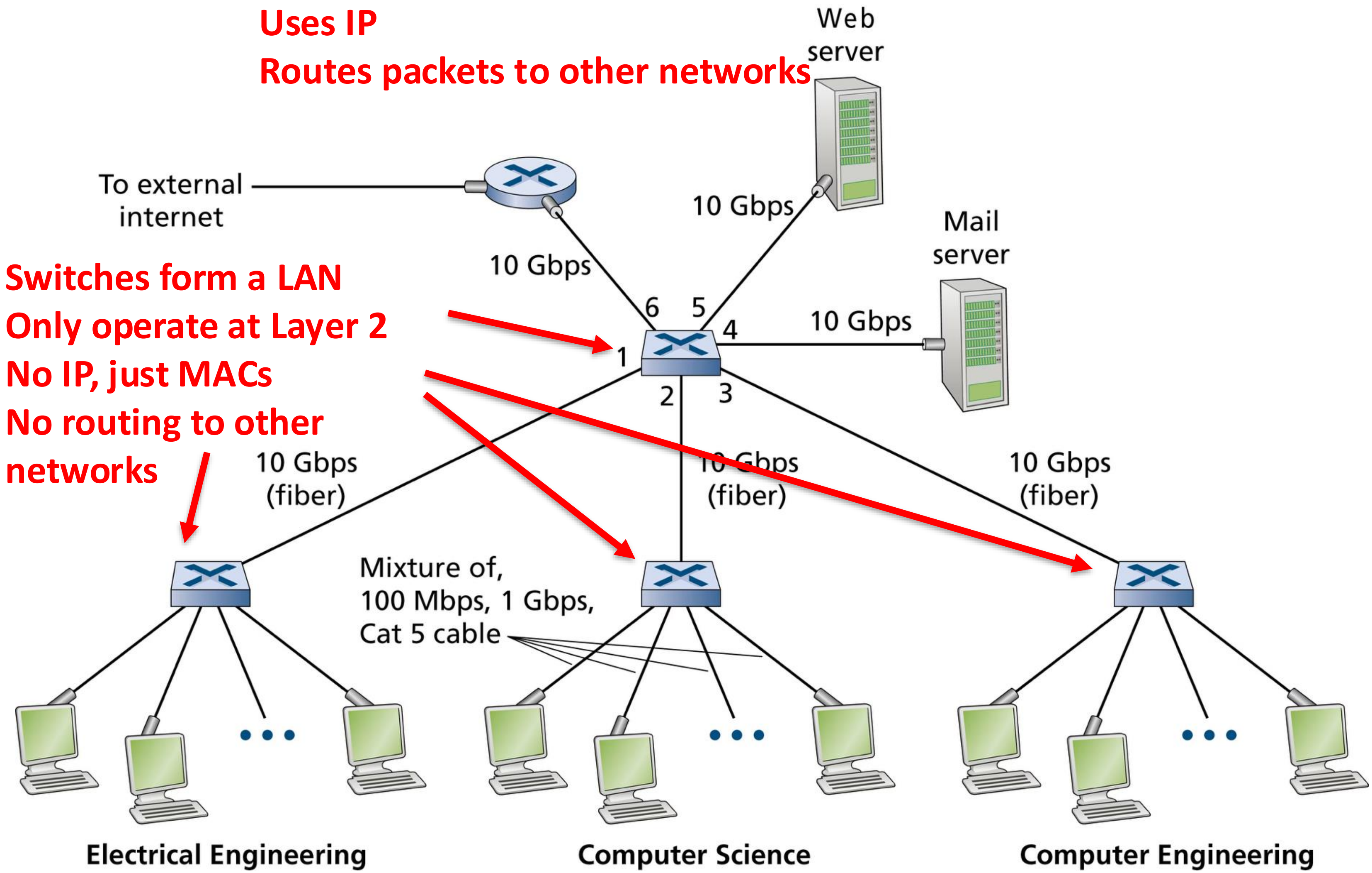
Ethernet is both unreliable and connectionless

- Connectionless:
  - No handshaking between sending and receiving NICs
- Unreliable:
  - Receiving NIC doesn't send ACKs or NAKs to sending NIC
  - Data in dropped frames recovered only if initial sender uses higher layer reliable data transfer (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

Switches (Layer 2) and routers (Layer 3) are typically connected via Ethernet

Router connects LAN to other networks
Uses IP
Routes packets to other networks

Switches form a LAN
Only operate at Layer 2
No IP, just MACs
No routing to other networks



Ethernet switch

- Switch is a link-layer device: takes an *active* role
  - Store and forward Ethernet (or other type of) frames
  - Examine incoming frame's MAC address
    - *Selectively* forward frame to one-or-more outgoing links when frame is to be forwarded
    - Uses CSMA/CD to assess if the link is busy
- Transparent: hosts *unaware* of presence of switches
  - Can think of the switch like a smart cable
- Plug-and-play, self-learning
  - Switches do not **need** to be configured

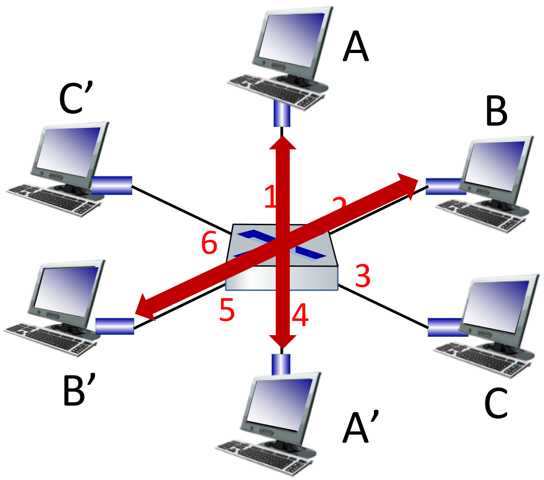
Does a switch have a MAC address?

Hosts never address traffic to the switch so it doesn't strictly *need* one

But, switches often have a MAC address to provide a management interface (ex., VLAN setup)

Switch: multiple simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions

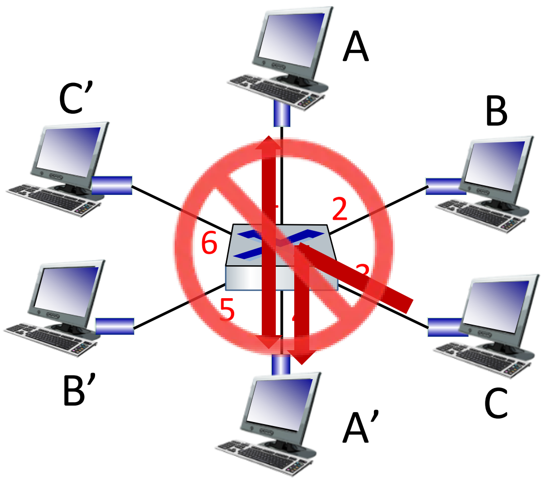


switch with six interfaces (1,2,3,4,5,6)

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Switch: multiple simultaneous transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on *each* incoming link, so:
  - No collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions
  - But A-to-A' and C to A' can *not* happen simultaneously
  - Must buffer



switch with six interfaces (1,2,3,4,5,6)

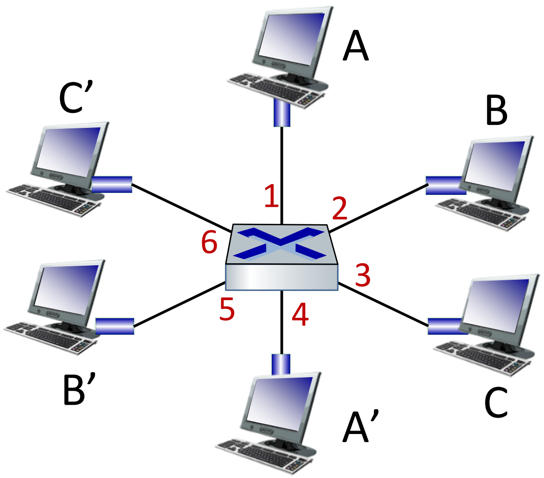
Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Switch forwarding table

Q: How does switch know A’ reachable via interface 4, B’ reachable via interface 5?

A: Each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- Looks like a routing table!



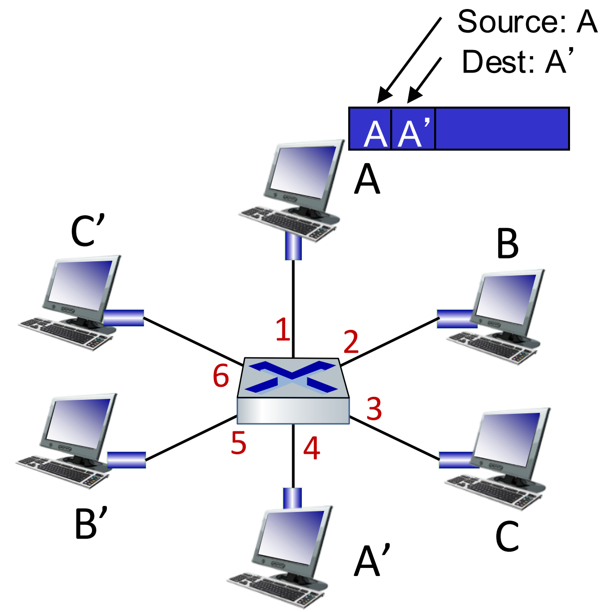
Q: How are entries created, maintained in switch table?

- Something like a routing protocol

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Switch: self-learning

- Switch *learns* which hosts can be reached through which interfaces
  - When frame received, switch "learns" location of sender: incoming LAN segment
  - Records sender/location pair in switch table



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

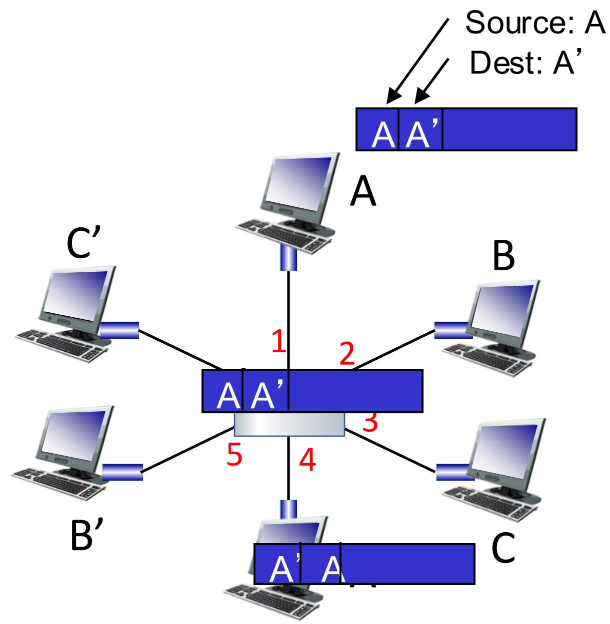
Switch: frame filtering/forwarding

When frame received at switch:

1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. If entry found for destination
   then {
   If destination on segment from which frame arrived
      then drop frame
      else forward frame on interface indicated by entry
   }
   else flood // forward on all interfaces except arriving interface

Self-learning, forwarding: example

- Frame destination, A', location unknown: flood
- Reply: destination A location known: selectively send on just one link



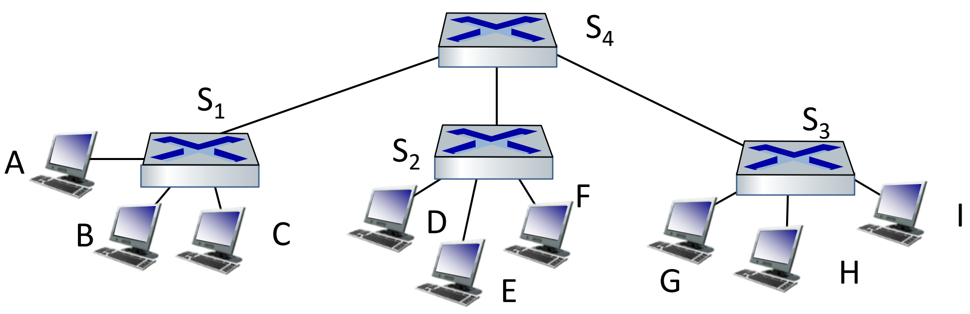
MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Interconnecting switches

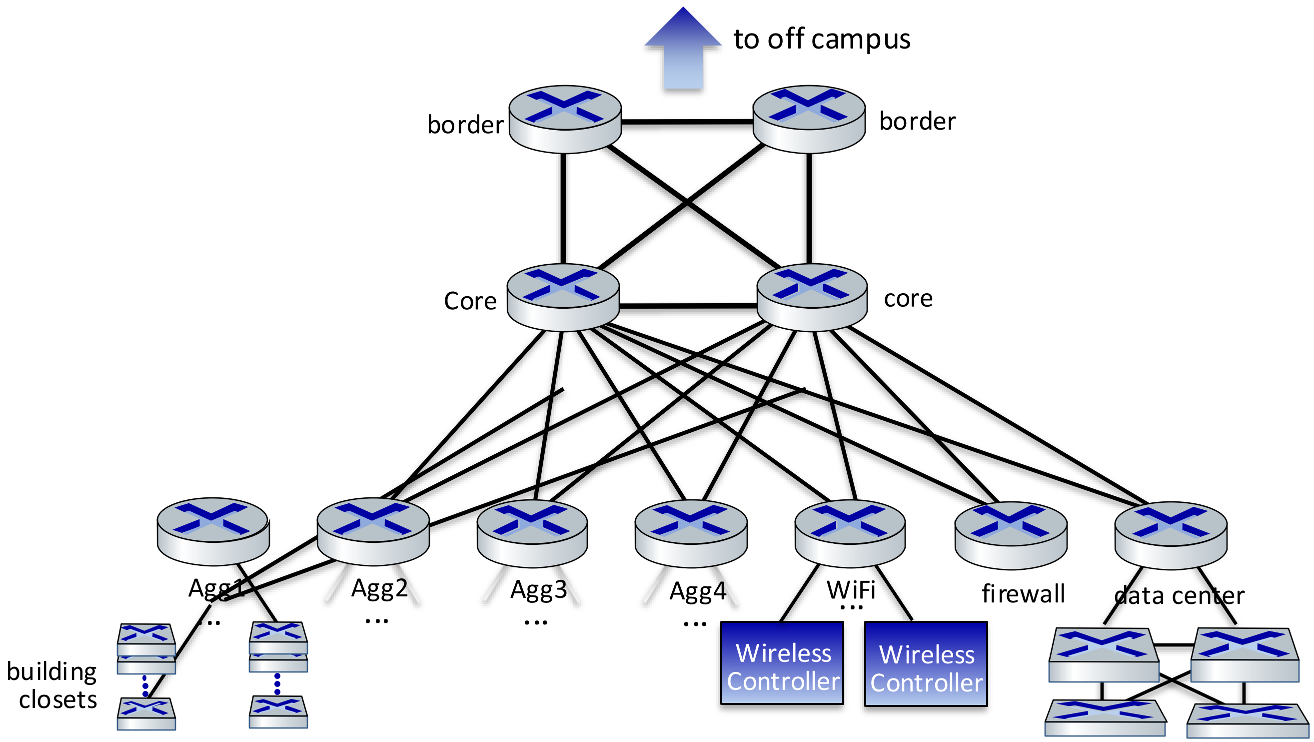
Self-learning switches can be connected together:



Q: Sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

- *A:* Self learning! (works exactly the same as in single-switch case!)

UMass Campus Network - Detail



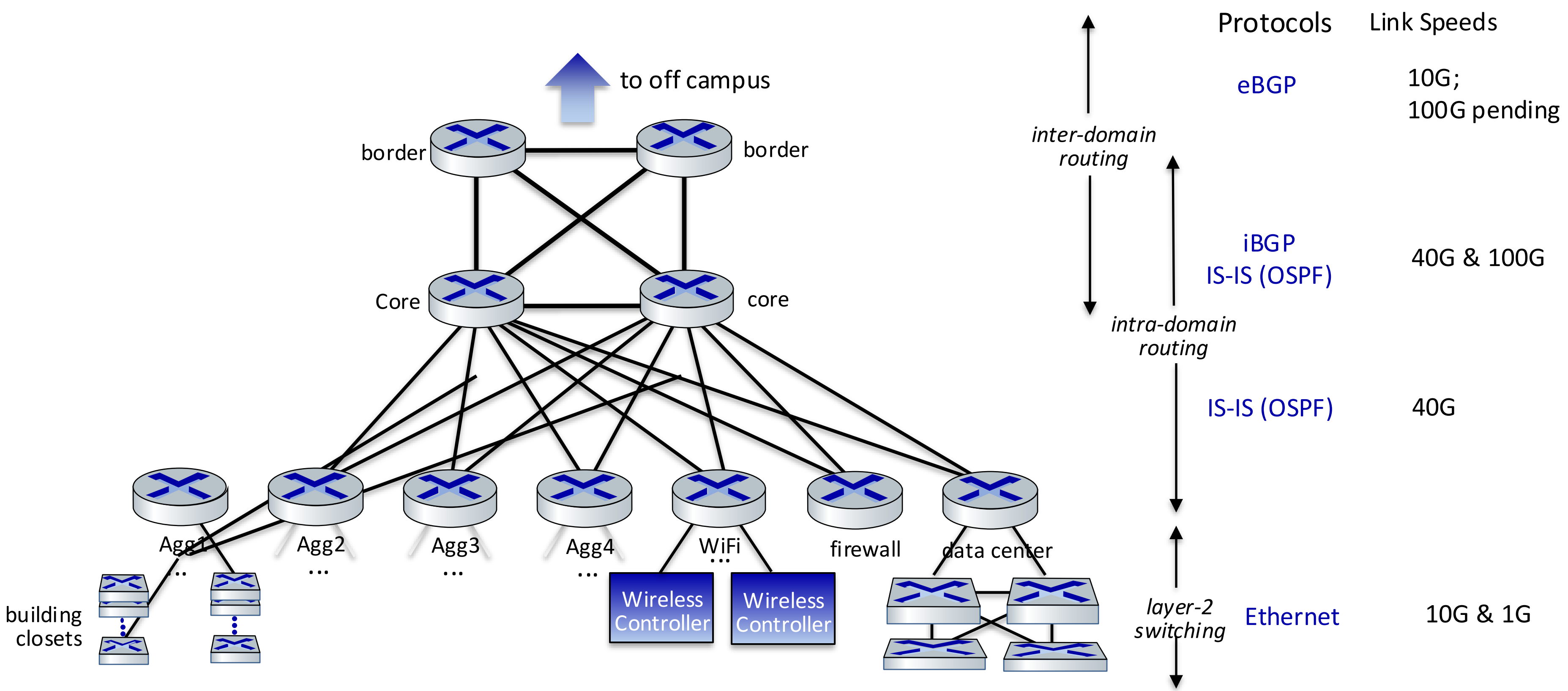
UMass network:

- 4 firewalls
- 10 routers
- 2,000+ network switches
- 6,000 wireless access points
- 30,000 active wired network jacks
- 55,000 active end-user wireless devices

... all built, operated, maintained by ~15 people

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

UMass Campus Network - Detail



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

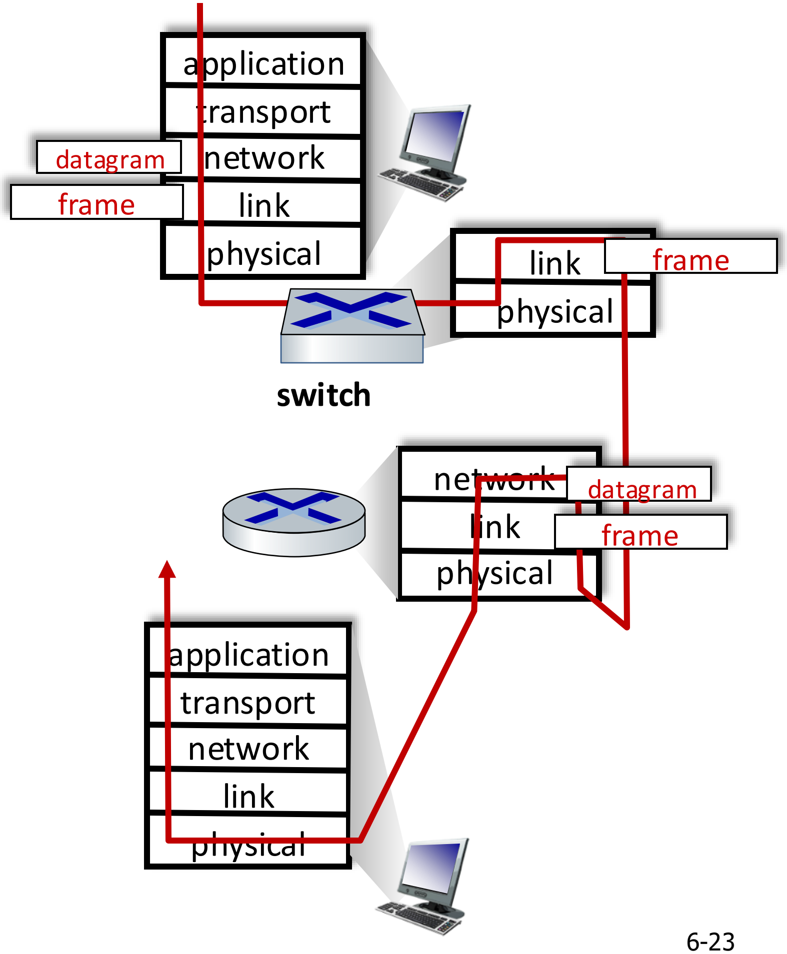
Switches vs. routers

Both are store-and-forward:

- *Routers*: network-layer devices (examine network-layer headers)
- *Switches:* link-layer devices (examine link-layer headers)

Both have forwarding tables:

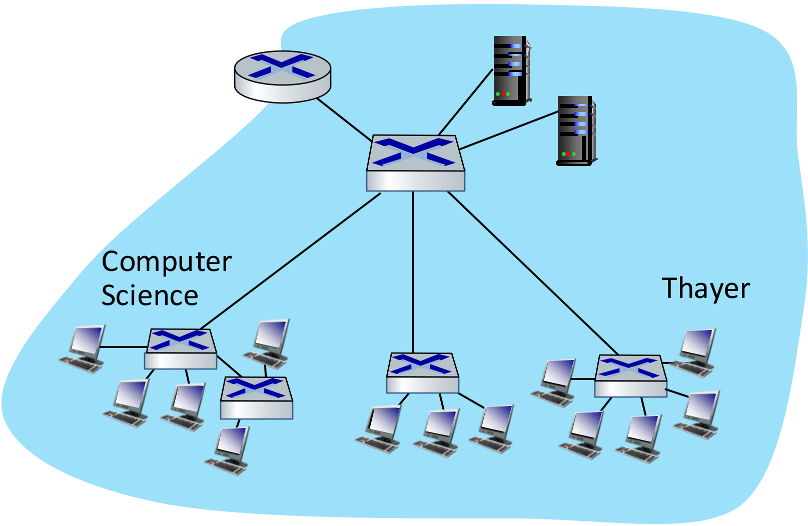
- *Routers:* compute tables using routing algorithms, IP addresses
- *Switches:* learn forwarding table using flooding, learning MAC addresses



6-23

Virtual LANs (VLANs): motivation

*Q: W*hat happens as LAN sizes scale, users change point of attachment?

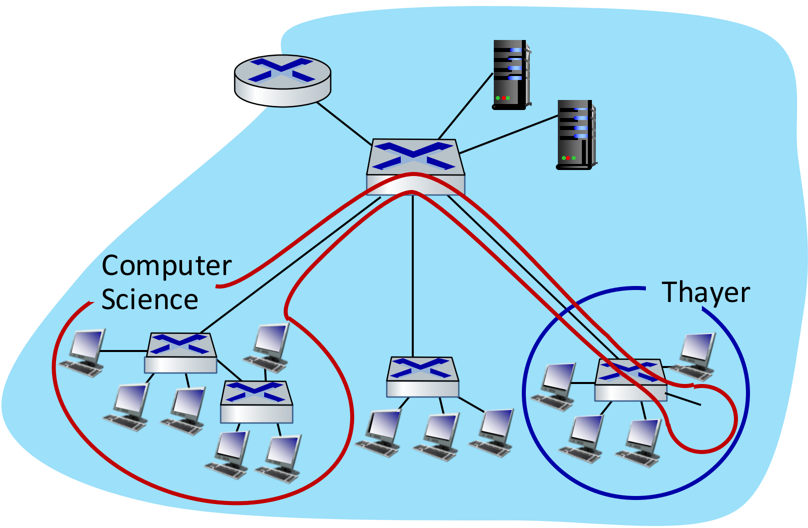


Single broadcast domain:

- *Scaling:* all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- Efficiency, security, privacy, efficiency issues

Virtual LANs (VLANs): motivation

*Q: W*hat happens as LAN sizes scale, users change point of attachment?



Single broadcast domain:

- *Scaling:* all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
- Efficiency, security, privacy, efficiency issues

Administrative issues:

- CS user moves office to Thayer - *physically* attached to Thayer switch, but wants to remain *logically* attached to CS switch

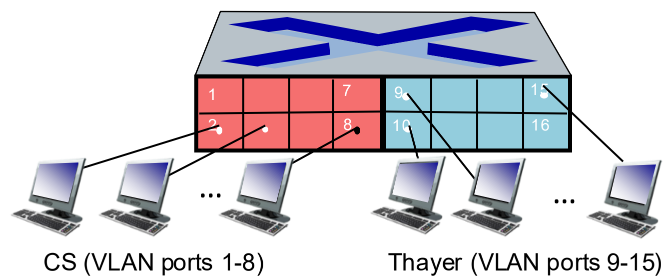
Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Port-based VLANs

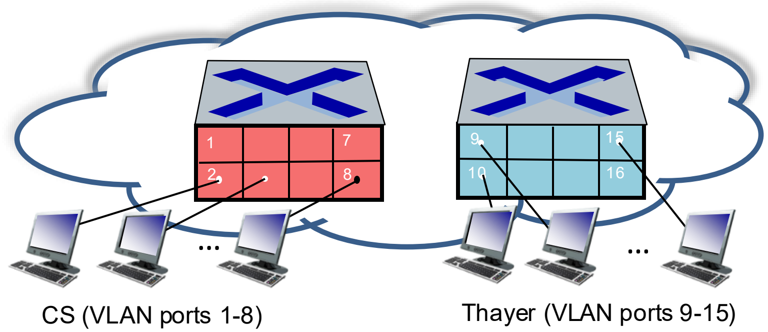
Virtual Local Area Network (VLAN)

Switch(es) supporting VLAN capabilities can be configured to define multiple *virtual* LANS over single physical LAN infrastructure.

Port-based VLAN: switch ports grouped (by switch management software) so that *single* physical switch ……



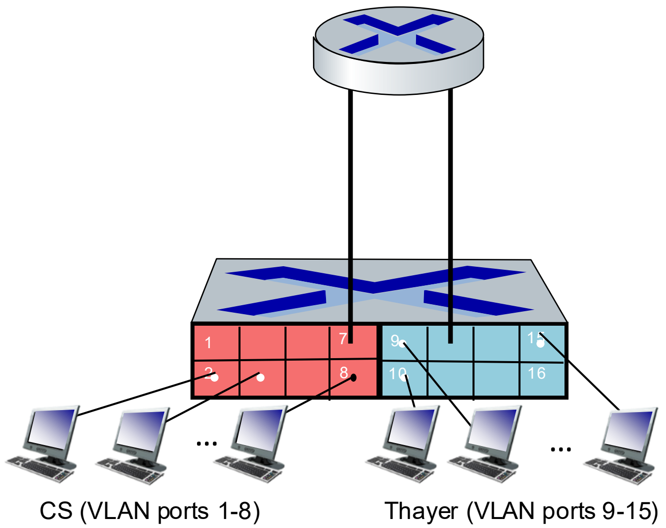
… operates as multiple virtual switches



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Port-based VLANs

- Traffic isolation: frames to/from ports 1-8 can *only* reach ports 1-8
  - Can also define VLAN based on MAC addresses of endpoints, rather than switch port
- Dynamic membership: ports can be dynamically assigned among VLANs
- Forwarding between VLANS: done via routing (just as with separate switches)
  - In practice vendors sell combined switches plus routers



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

At Layer 2 MAC addresses uniquely identify hosts

- 32-bit IP address:
  - *Network-layer* address for interface
  - Used for Layer 3 (network layer) forwarding
  - e.g.: 128.119.40.136

MAC = Media Access Control

- MAC (aka LAN or physical or Ethernet) address:
  - Function: used "locally" to get frame from one interface to another physically-connected interface (same subnet, in IP-addressing sense)
  - 48-bit MAC address (6 bytes) burned in NIC ROM, also sometimes software settable
  - e.g.: 1A:2F:BB:76:09:AD

MACs are meant to be permanent but they can be changed in software I'll assume they are fixed today

Routers have MAC addresses too! (well, their network interfaces do, multiple interfaces = multiple MACs)

hexadecimal (base 16) notation (each "numeral" represents 4 bits)

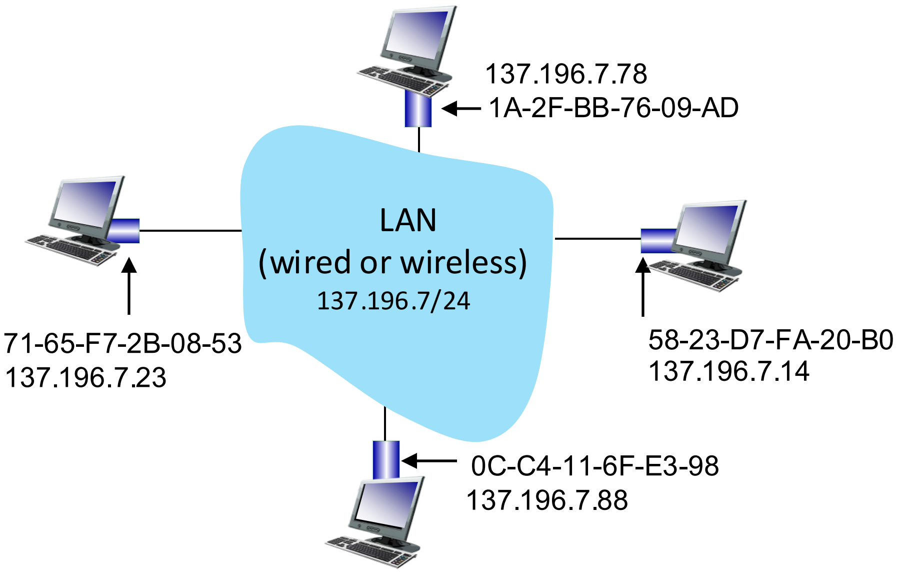
Switches do *not* have MACs! (except for management interfaces)

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

At Layer 2 MAC addresses uniquely identify hosts

Each interface on LAN

- Has unique 48-bit MAC address
- Has a locally unique 32-bit IP address (as we've seen)



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

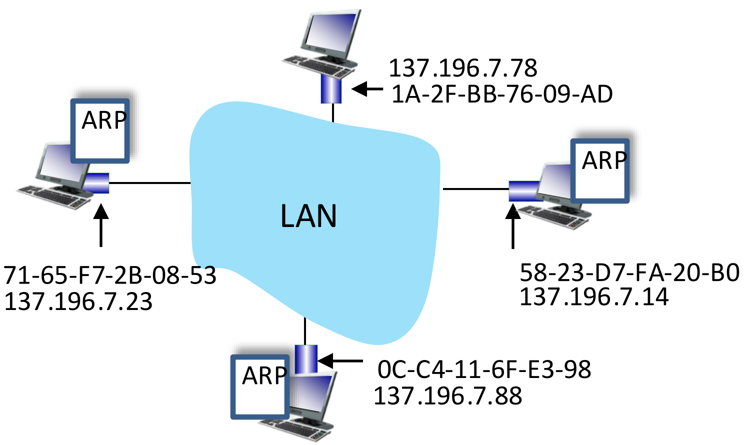
At Layer 2 MAC addresses uniquely identify hosts

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- First three octets (24 bits) identify manufacturer (OUI), last three identify the device
- Analogy:
  - MAC address: like Social Security Number (identity)
  - IP address: like postal address (location)
- MAC flat address: portability
  - Can move interface (MAC) from one LAN to another
  - Recall IP address
    - *Not* portable: depends on IP subnet to which node is attached
    - Hierarchical, first part of IP identifies the network, second identifies the host

ARP: address resolution protocol

Question: How to determine interface's MAC address, knowing its IP address?

IP	MAC	TTL
137.196.7.14	58-23-D7-FA-20-B0	20
137.196.7.88	0C-C4-aa-6F-E3-98	14
137.196.7.23	71-65-F7-2B-08-53	9



ARP table: each IP node (host, router) on LAN has table

- IP/MAC address mappings for some LAN nodes:

  < IP address; MAC address; TTL>
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
- Different from Switch Table which gives the interface where a MAC is attached

ARP protocol in action

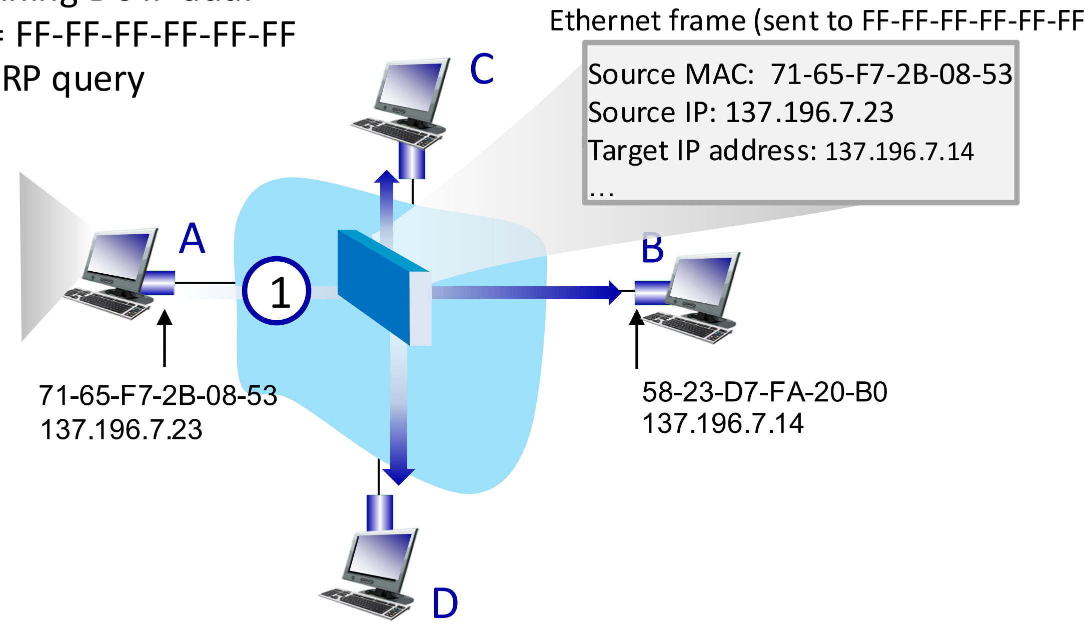
Example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

1. A broadcasts ARP query, containing B's IP addr
   - Destination MAC address = FF-FF-FF-FF-FF-FF
   - All nodes on LAN receive ARP query

ARP table in A

IP	MAC	TTL



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

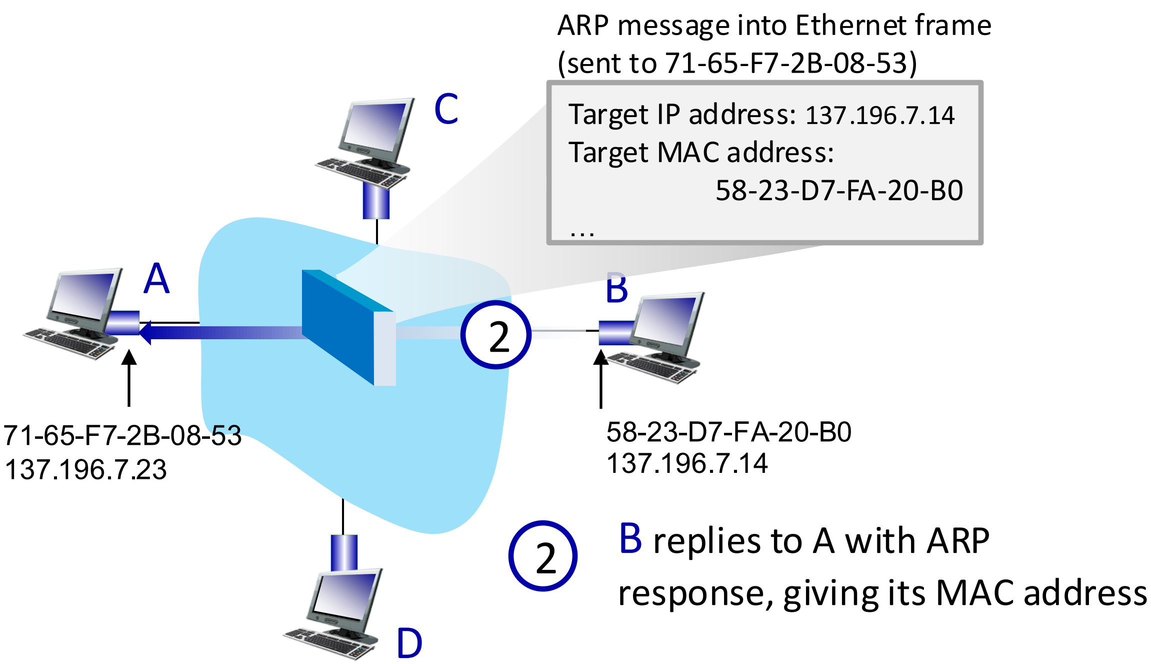
ARP protocol in action

Example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP	MAC	TTL



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

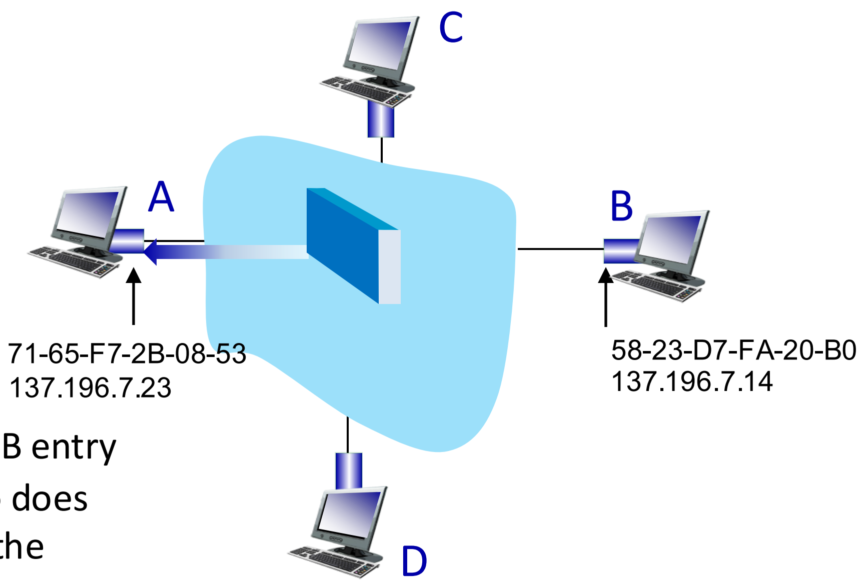
ARP protocol in action

Example: A wants to send datagram to B

- B's MAC address not in A's ARP table, so A uses ARP to find B's MAC address

ARP table in A

IP	MAC	TTL
137.196.7.14	58-23-D7-FA-20-B0	500



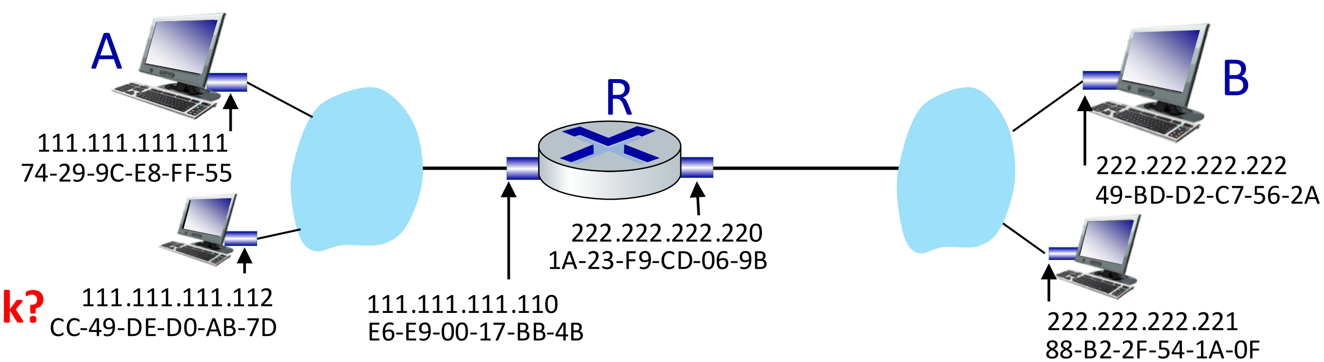
3 A receives B's reply, adds B entry into its local ARP table (so does everyone else that hears the reply!)

ARP table is built automatically
Sys admins do not need to configure them (nice!)

Routing to another subnet: addressing

Walkthrough: *sending a datagram from A to B via R*

- Focus on addressing – at IP (datagram) and MAC layer (frame) levels
- Assume that:
  - A knows B's IP address or gets it from DNS
  - A knows IP address of first hop router, R (how?)
  - A knows R's MAC address (how?)



What is the subnet mask?

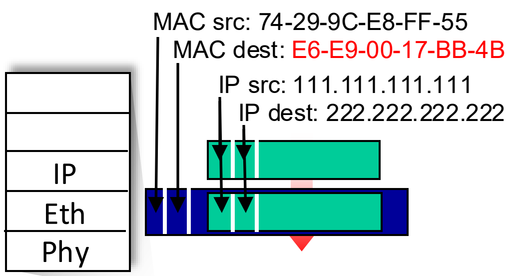
Let's assume 111.111.111.0/24

Note: router has IP address and MAC for each of its interfaces
Left interface IP addr on left subnet
Right interface IP addr on right subnet

Routing to another subnet: addressing

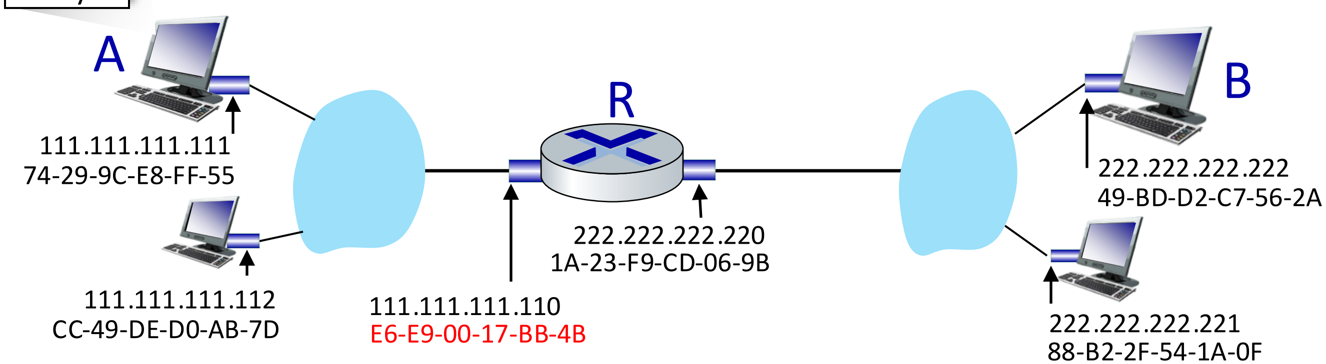
- A creates IP packet with IP source A, destination B
- A creates link-layer frame containing A-to-B IP datagram
  - R's MAC address is frame’s destination

How to tell if dst IP in network?
Use subnet mask
111.111.111.0/24



A looks at dst IP
If in network
- **Use dst MAC (ARP if dst MAC not known)**

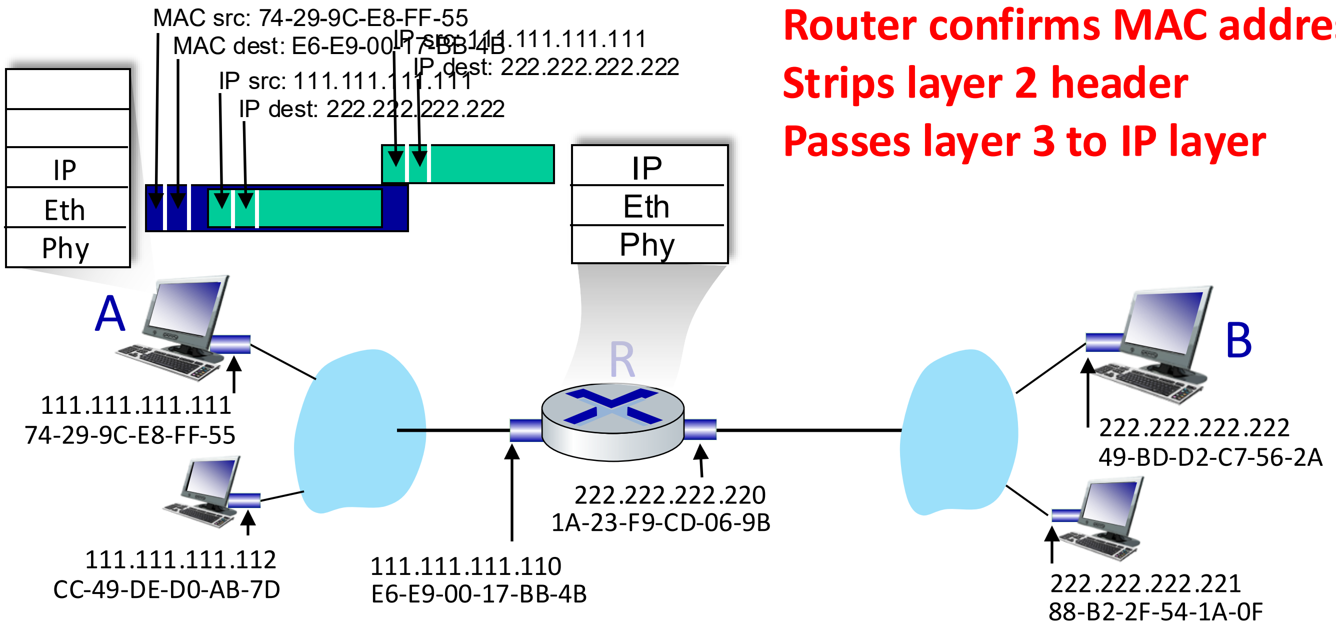
Else
- **Use default gateway router’s MAC**



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Routing to another subnet: addressing

- Frame sent from A to R
- Frame received at R, frame header removed, passed up to IP



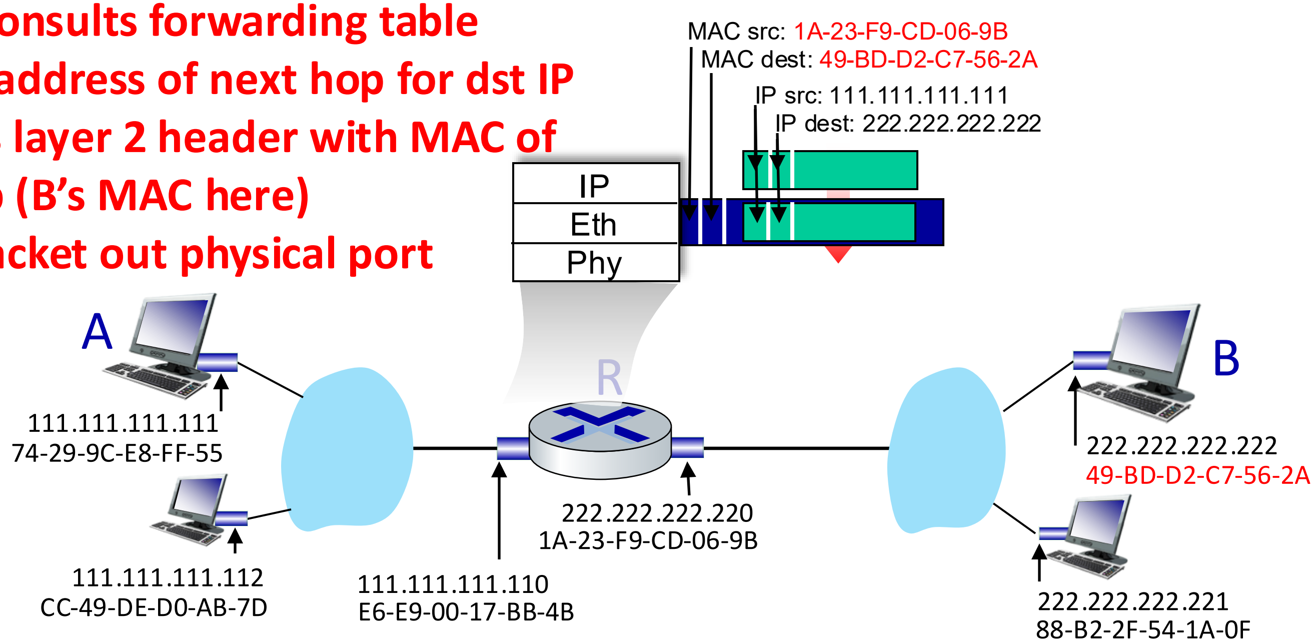
Layer 2
Router confirms MAC address
Strips layer 2 header
Passes layer 3 to IP layer

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Routing to another subnet: addressing

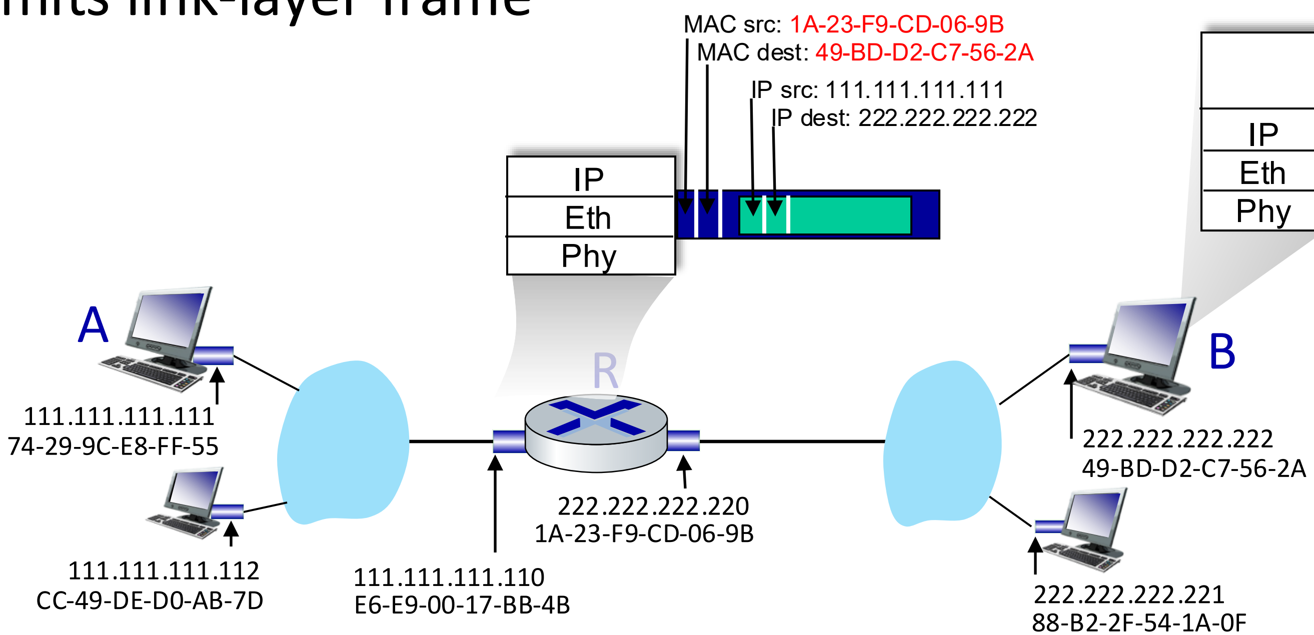
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram Frame destination address: B's MAC address

- **Router consults forwarding table**
- **Finds IP address of next hop for dst IP**
- **Rewrites layer 2 header with MAC of next hop (B's MAC here)**
- **Sends packet out physical port**



Routing to another subnet: addressing

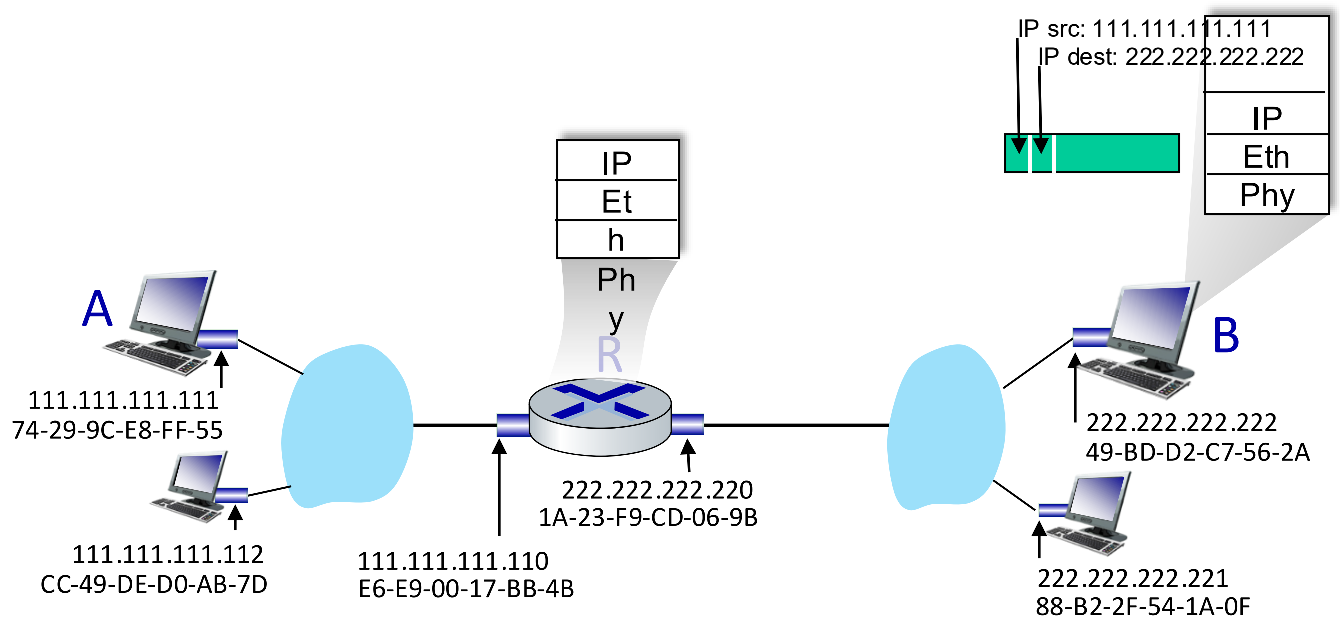
- R determines outgoing interface, passes datagram with IP source A, destination B to link layer
- R creates link-layer frame containing A-to-B IP datagram. Frame destination address: B's MAC address
- Transmits link-layer frame



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Routing to another subnet: addressing

- B receives frame, checks MAC, if match pass to IP layer
- IP layer checks IP address, if match pass to Transport layer
- Transport layer extracts port
- Transport layer passes message to process



Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Agenda

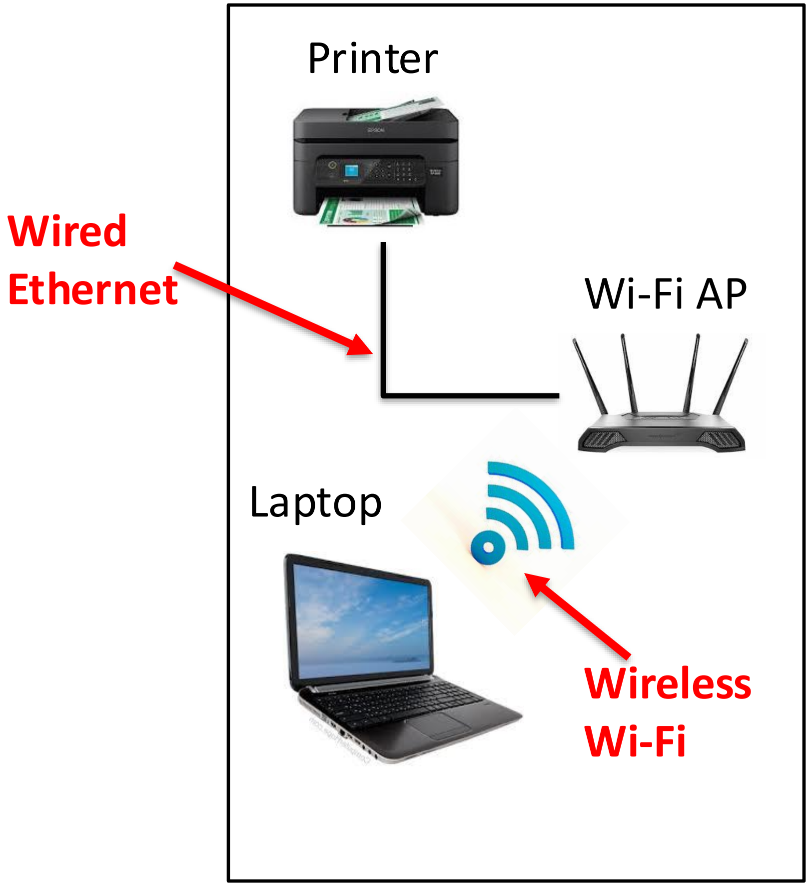
1. Ethernet
2. Putting it all together
3. Error detection/correction
4. Channel sharing

Starting point: your house has a Wi-Fi AP wired to a printer and wireless to a laptop

Networking Concepts: **Will attempt to put 26 networking terms into context**

- **Ethernet/Wi-Fi**

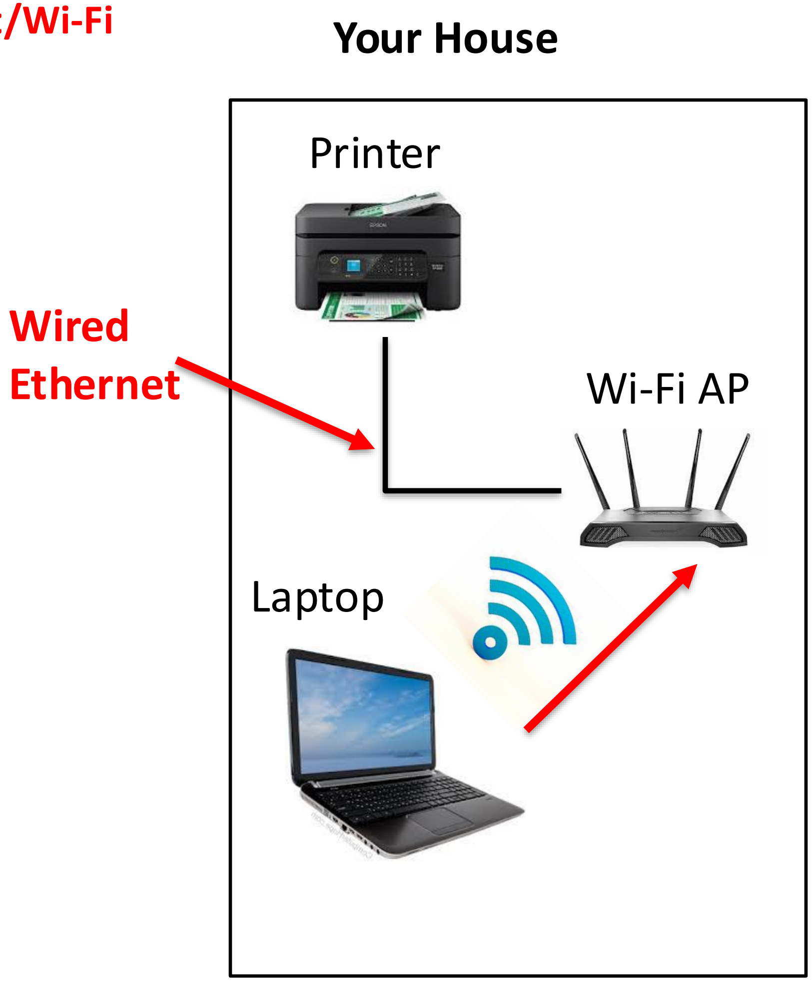
Your House



Starting point: your house has a Wi-Fi AP wired to a printer and wireless to a laptop

Networking Concepts: **Will attempt to put 26 networking terms into context**

- **Ethernet/Wi-Fi**

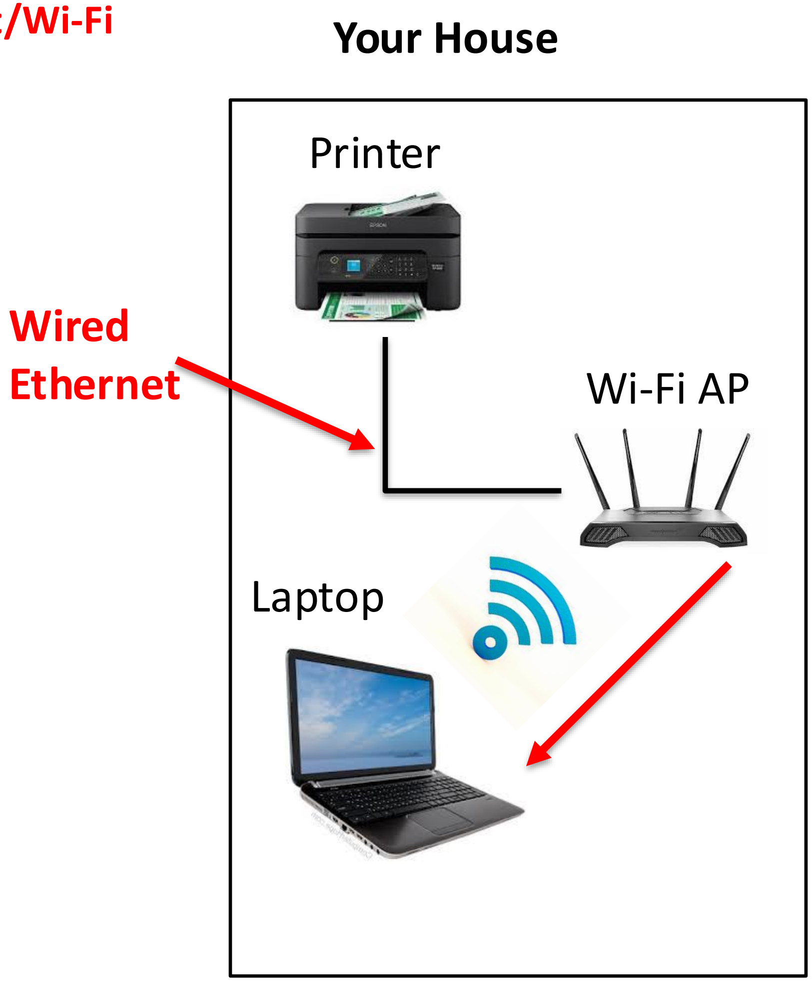


- **Laptop sends wireless Wi-Fi signal to AP**
- **AP extracts Layer 2**
- **AP converts to Ethernet and forwards**

Starting point: your house has a Wi-Fi AP wired to a printer and wireless to a laptop

Networking Concepts: **Will attempt to put 26 networking terms into context**

- **Ethernet/Wi-Fi**

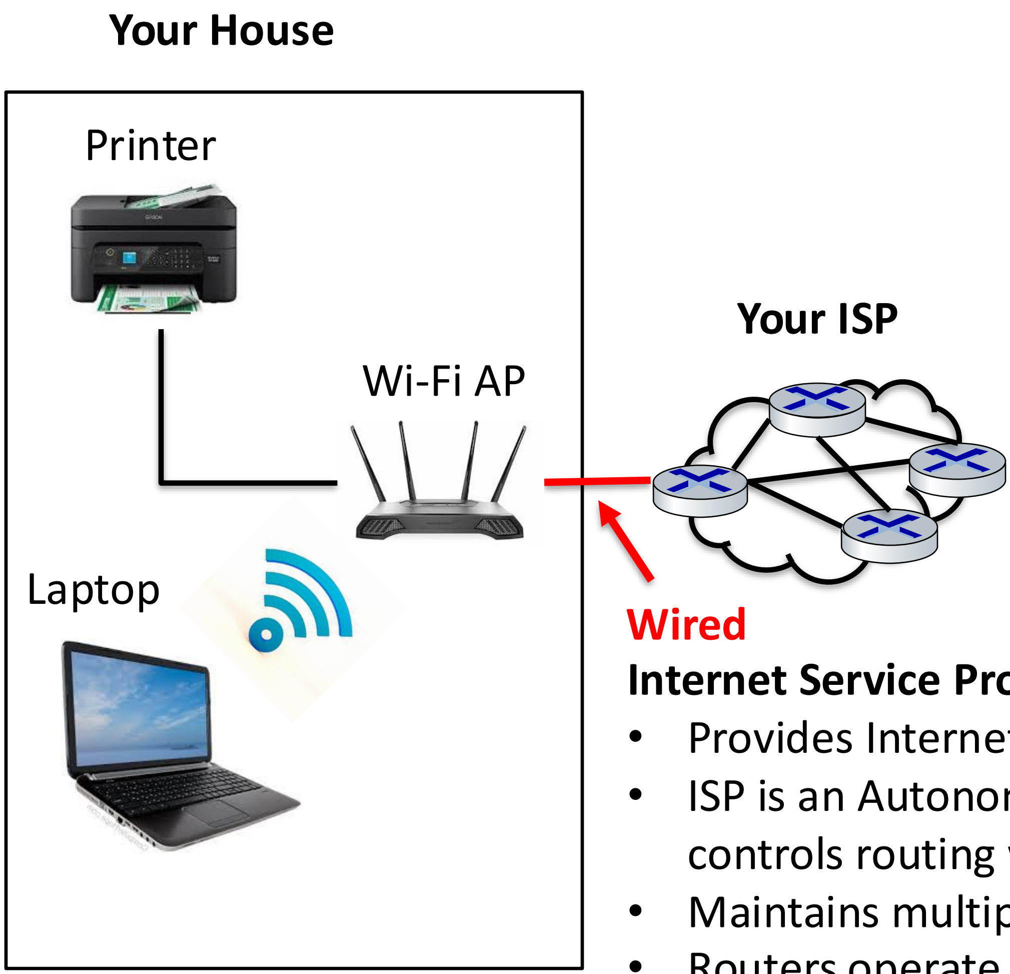


- **AP converts Ethernet to Wi-Fi signal**
- **Sends to Laptop**
- **Laptop extracts Layer 2**

Your Wi-Fi AP is connected to your Internet Service Provider (ISP)

Networking Concepts:

- Ethernet/Wi-Fi
- ISP



Internet Service Provider (ISP)

- Provides Internet access to clients
- ISP is an Autonomous System (AS), controls routing within its network
- Maintains multiple routers
- Routers operate on Layer 3
  - Forwarding
  - Routing

Big picture

Steps:

1. Get routes to Internet sorted out at ISP
2. Get an IP address from ISP and set up NAT

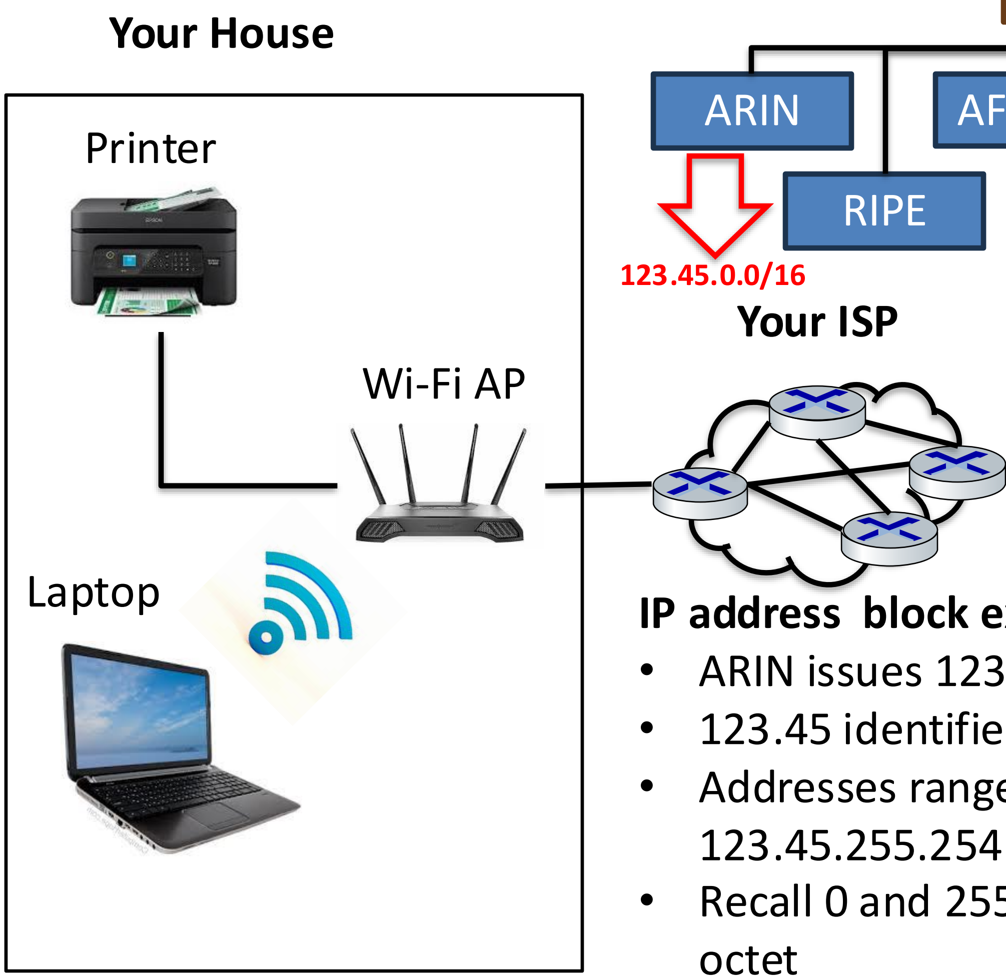
Examples:

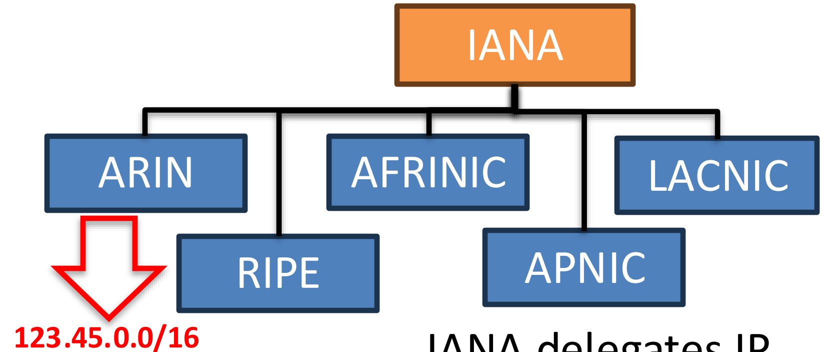
1. Send job from laptop to printer in network
2. Make HTTP web request outside network

Your ISP got a block of addresses from the Internet Assigned Numbers Authority

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address





IANA delegates IP ranges to five Regional Internet Registries (RIRs) worldwide

IP address block example:

- ARIN issues 123.45.0.0/16 to Your ISP
- 123.45 identifies the network
- Addresses range from 123.45.0.1 to 123.45.255.254 in this block
- Recall 0 and 255 are reserved in last octet
- ISP can give each client one of these addresses (covers 65,534 clients)

Your ISP is an AS that calculated the shortest path routes in its network

Networking Concepts:
- Ethernet/Wi-Fi
- ISP
- IP address

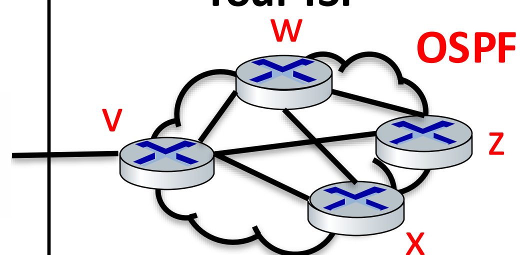
Your House

v's forwarding table

destination	outgoing link
w	(v,w)
x	(v,x)
z	(v,x)

Note: w is something like 123.45.5.0/8

Your ISP



- Each router in the ISP calculates shortest path between every other router in its network
- Might use Open Shortest Path First (OSPF) to calculate routes
- Store results in router's forwarding table

Your ISP is an AS that calculated the shortest path routes in its network

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- **AS**
- **Router**
- **Forwarding**
- **Routing**
- **OSPF**
- **Forwarding table**

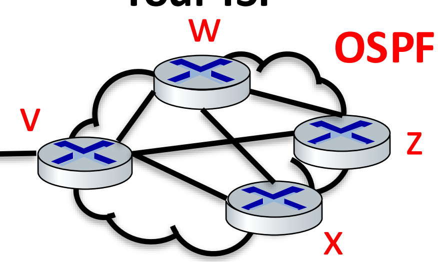
Your House

v's forwarding table

destination	outgoing link
w	(v,w)
x	(v,x)
z	(v,x)

Note: w is something like 123.45.5.0/8

Your ISP

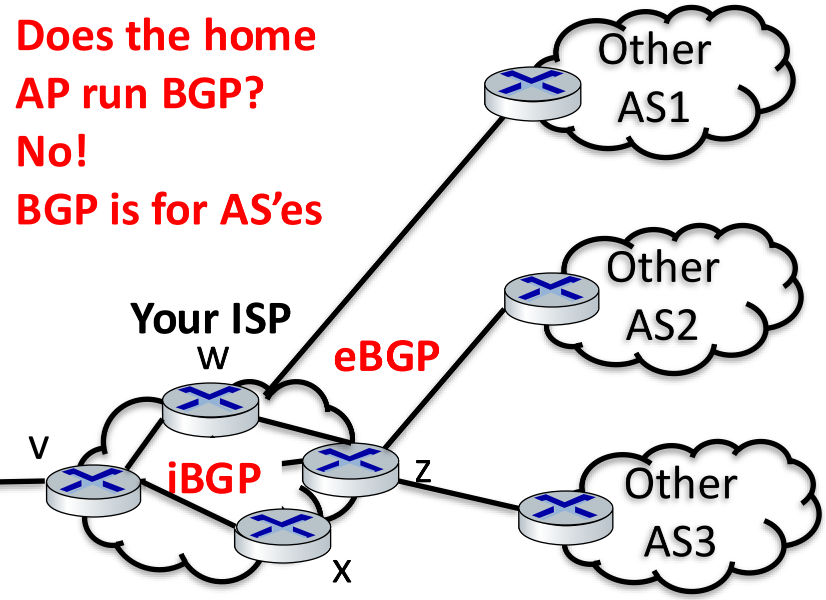


- Each router in the ISP calculates shortest path between every other router in its network
- Might use Open Shortest Path First (OSPF) to calculate routes
- Store results in a forwarding table

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)

Your House

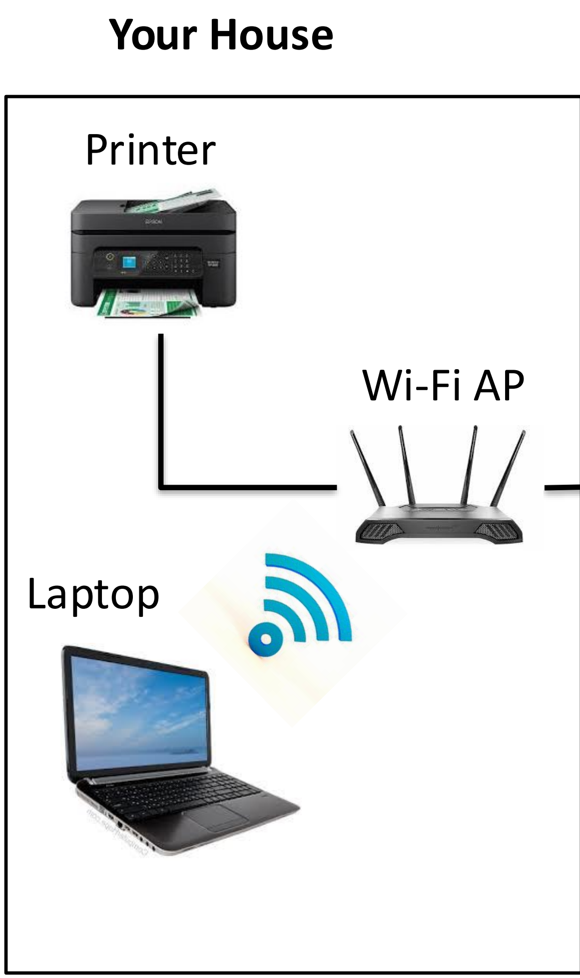


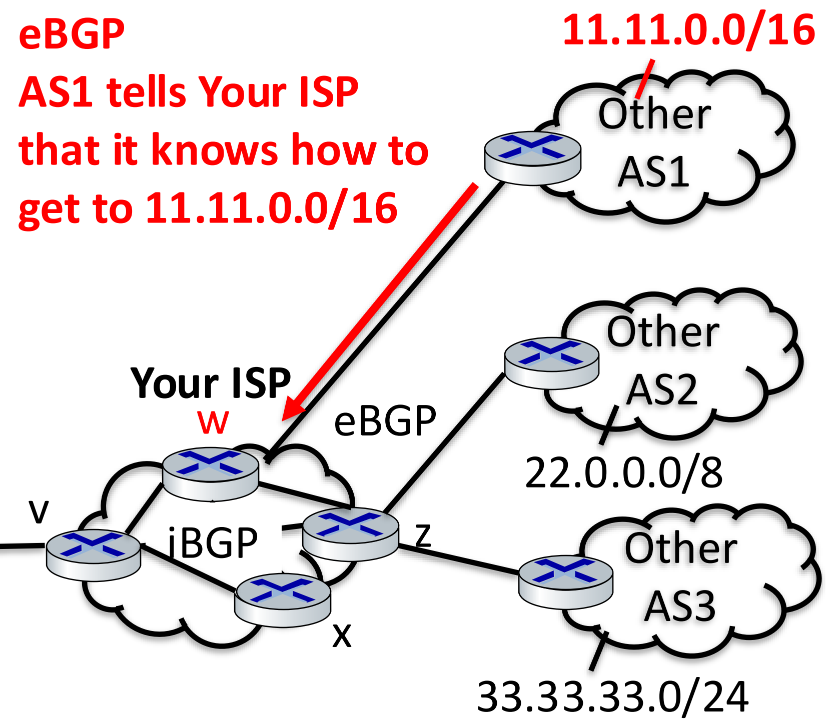
Border Gateway Protocol (BGP)
- BGP is used to connect AS'es
- AS is a network run by large organizations such as ISPs, large businesses, universities, government
- Two flavors of BGP:
  - External BGP (eBGP)
  - Internal BGP (iBGP)

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**





External BGP

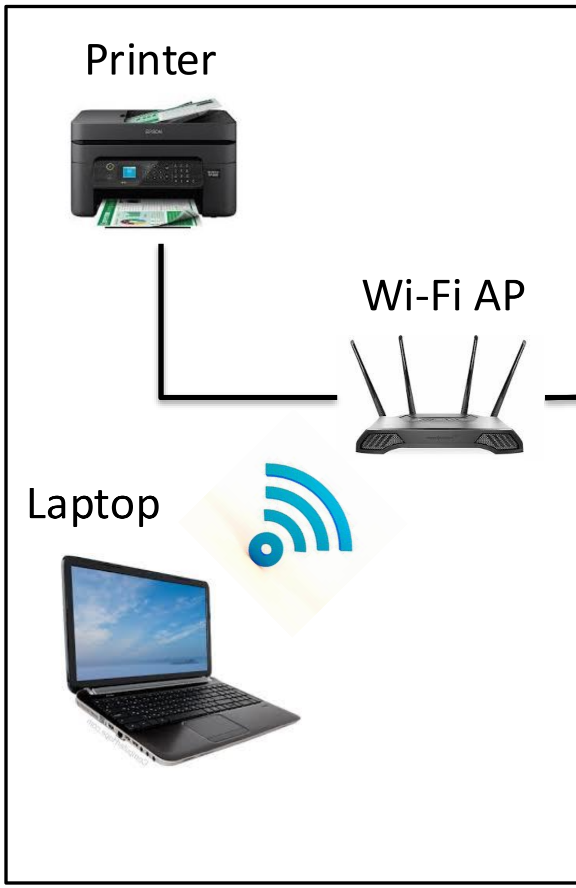
- In eBGP AS'es advertise the networks they know about to neighboring AS'es
- Border routers learn which links they can follow to reach networks connected to other AS'es

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

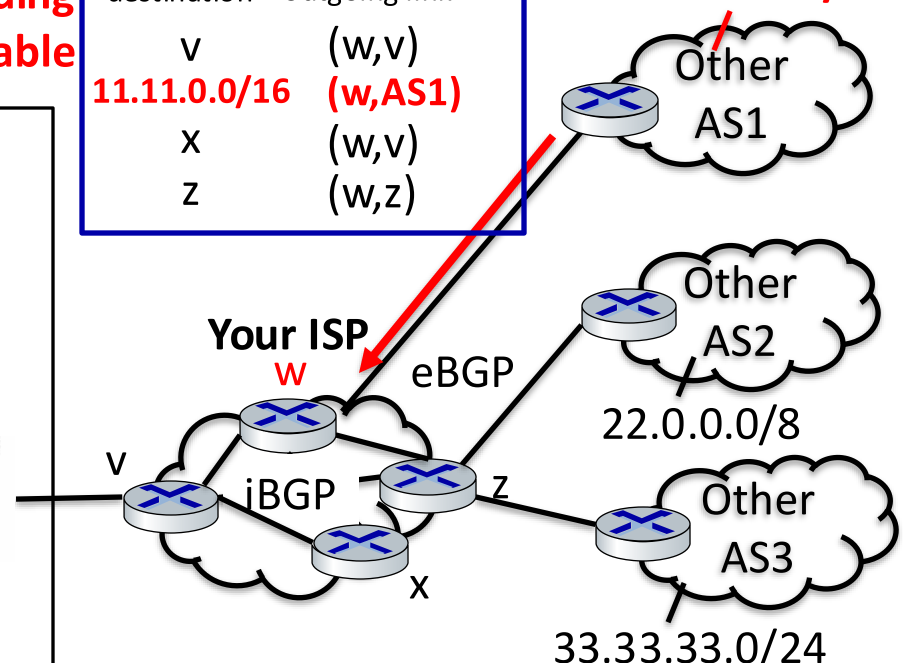
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**

Your House



w forwarding table

destination	outgoing link
v	(w,v)
11.11.0.0/16	**(w,AS1)**
x	(w,v)
z	(w,z)



External BGP (eBGP)

- In eBGP AS'es advertise the networks they know about to neighboring AS'es
- Border routers learn which links they can follow to reach networks connected to other AS'es
- **w notes that it can reach 11.11.0.0/16 via link to other ISP**

Your ISP is connected to other AS'es via Border Gateway Protocol

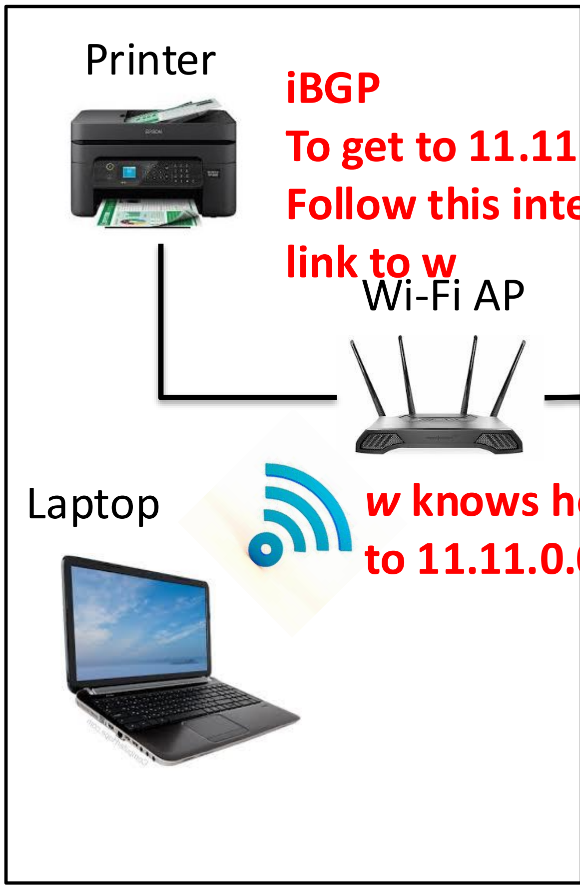
Networking Concepts:

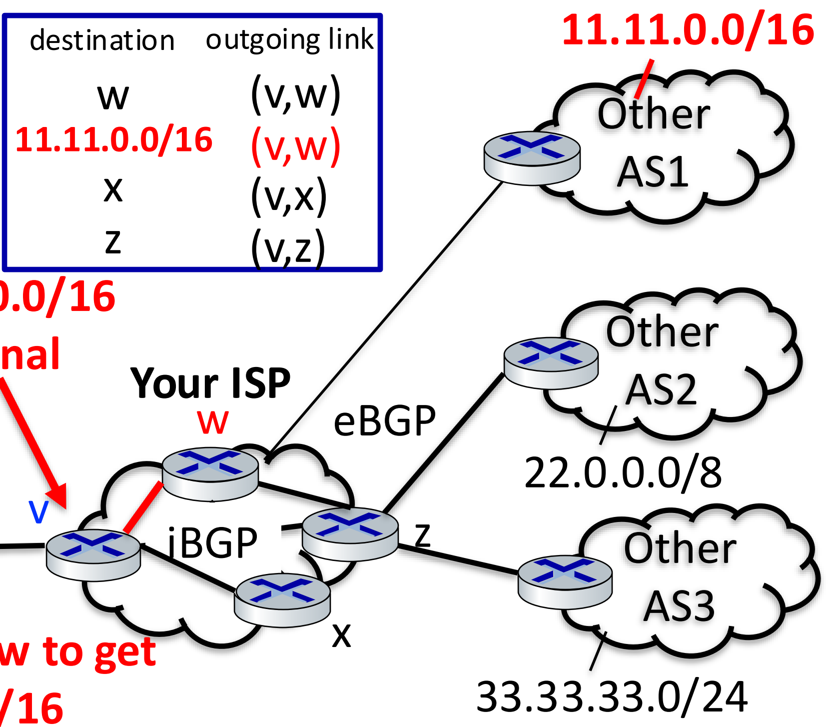
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**

Your House

v forwarding table

destination	outgoing link
w	(v,w)
11.11.0.0/16	(v,w)
x	(v,x)
z	(v,z)





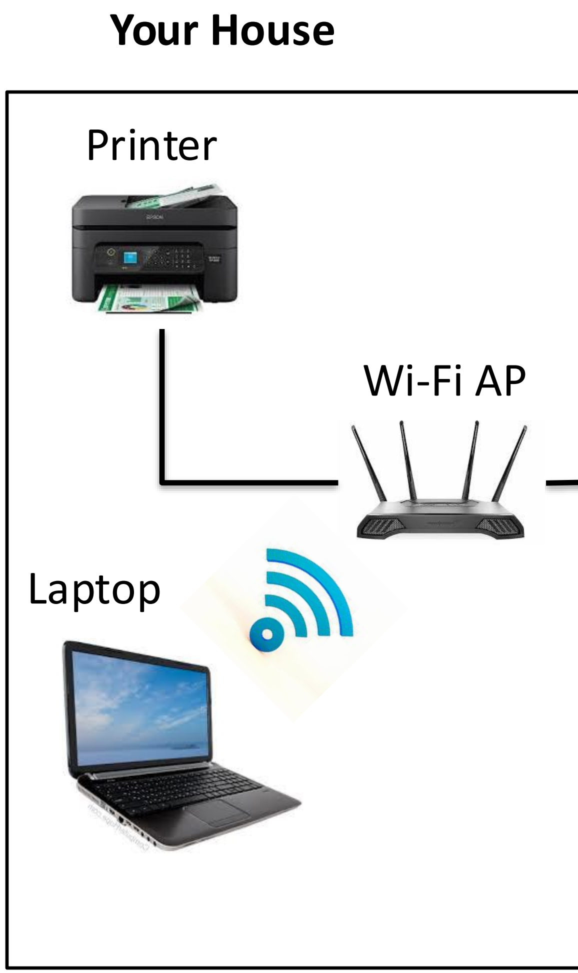
Internal BGP (iBGP)

- Via iBGP *w* tells internal routers about the route to 11.11.0.0./16
- Internal routers make table entry with next hop to get to *w* for 11.11.0.0/16
- From *v* follow link *w* to get to any address of the 65,534 possible hosts on the 11.11.0.0/16 network

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

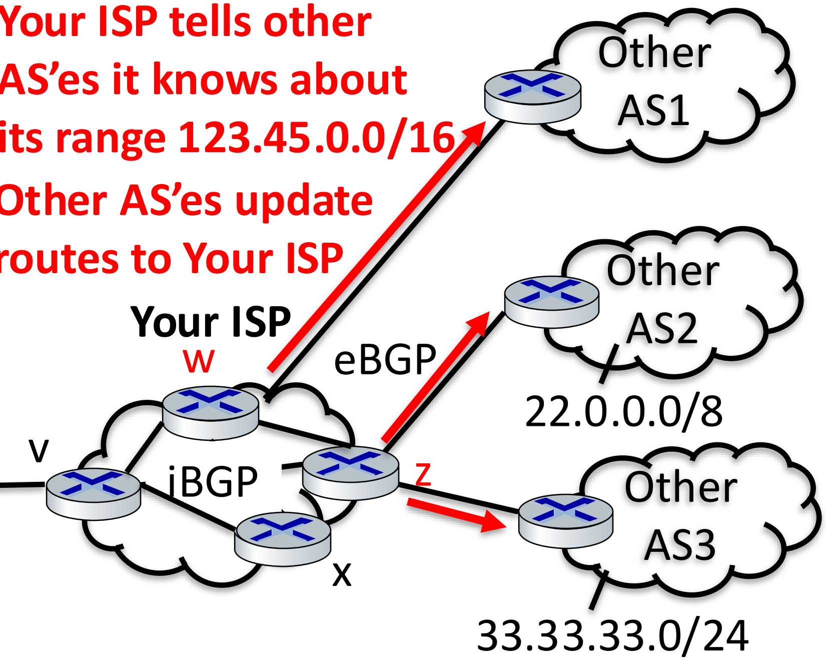
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**



eBGP

Your ISP tells other AS'es it knows about its range 123.45.0.0/16

Other AS'es update routes to Your ISP



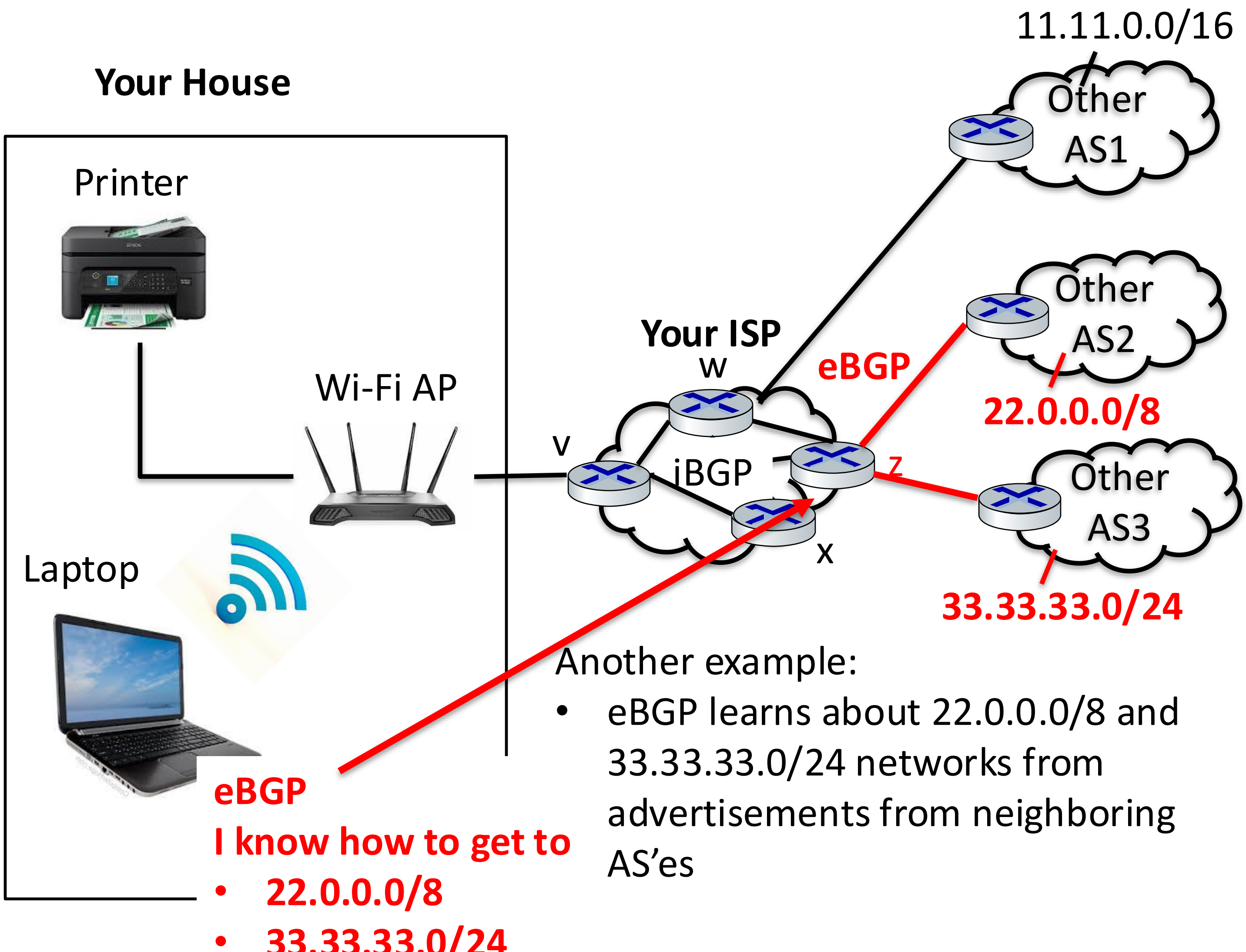
External BGP

- In eBGP AS'es advertise the networks they know about to neighboring AS'es
- Border routers learn which links they can follow to reach networks connected to other AS'es
- **Your ISP tells other AS'es about its range**

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**



Another example:

- eBGP learns about 22.0.0.0/8 and 33.33.33.0/24 networks from advertisements from neighboring AS'es

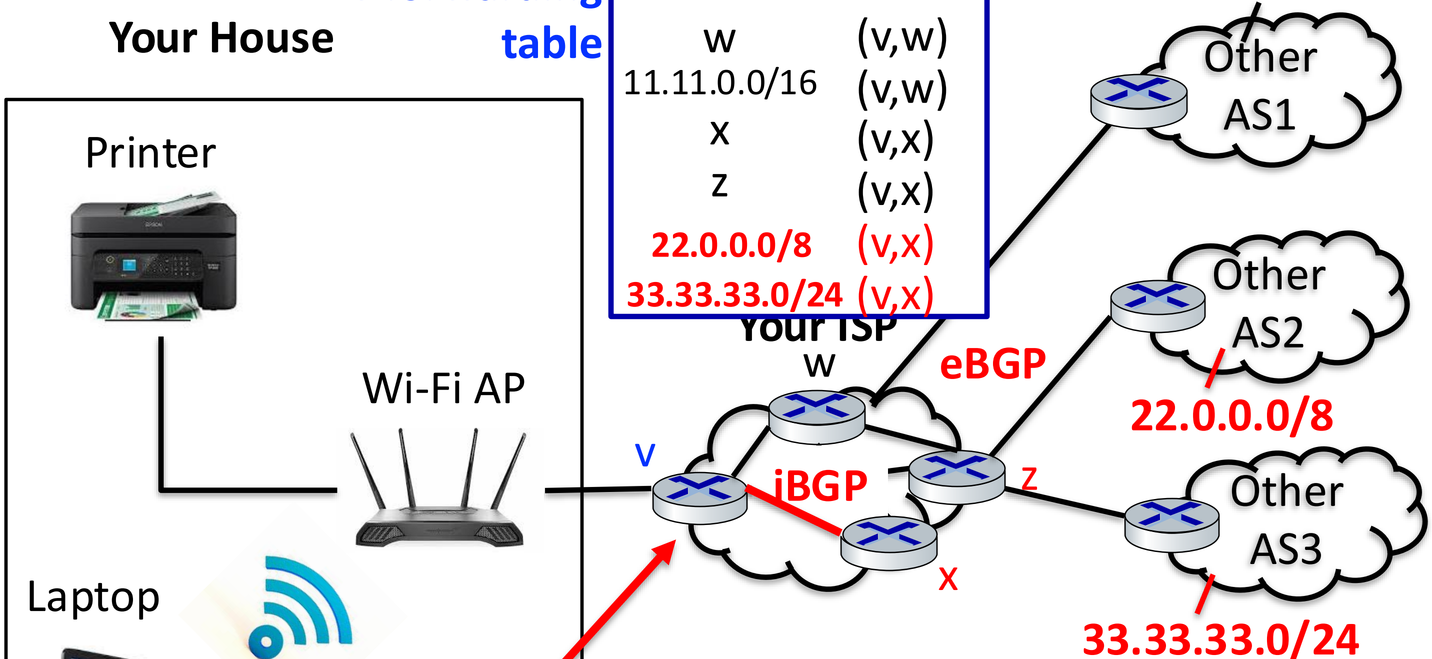
Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**

v forwarding table

destination	outgoing link
w	(v,w)
11.11.0.0/16	(v,w)
x	(v,x)
z	(v,x)
22.0.0.0/8	(v,x)
33.33.33.0/24	(v,x)



iBGP tells routers next hop to get to router that know about other networks

iBGP
To get to 22.0.0.0/8 or 33.33.33.0/24
Follow this link toward z
(v to x was shortest path to z from v)

To get to other networks from v

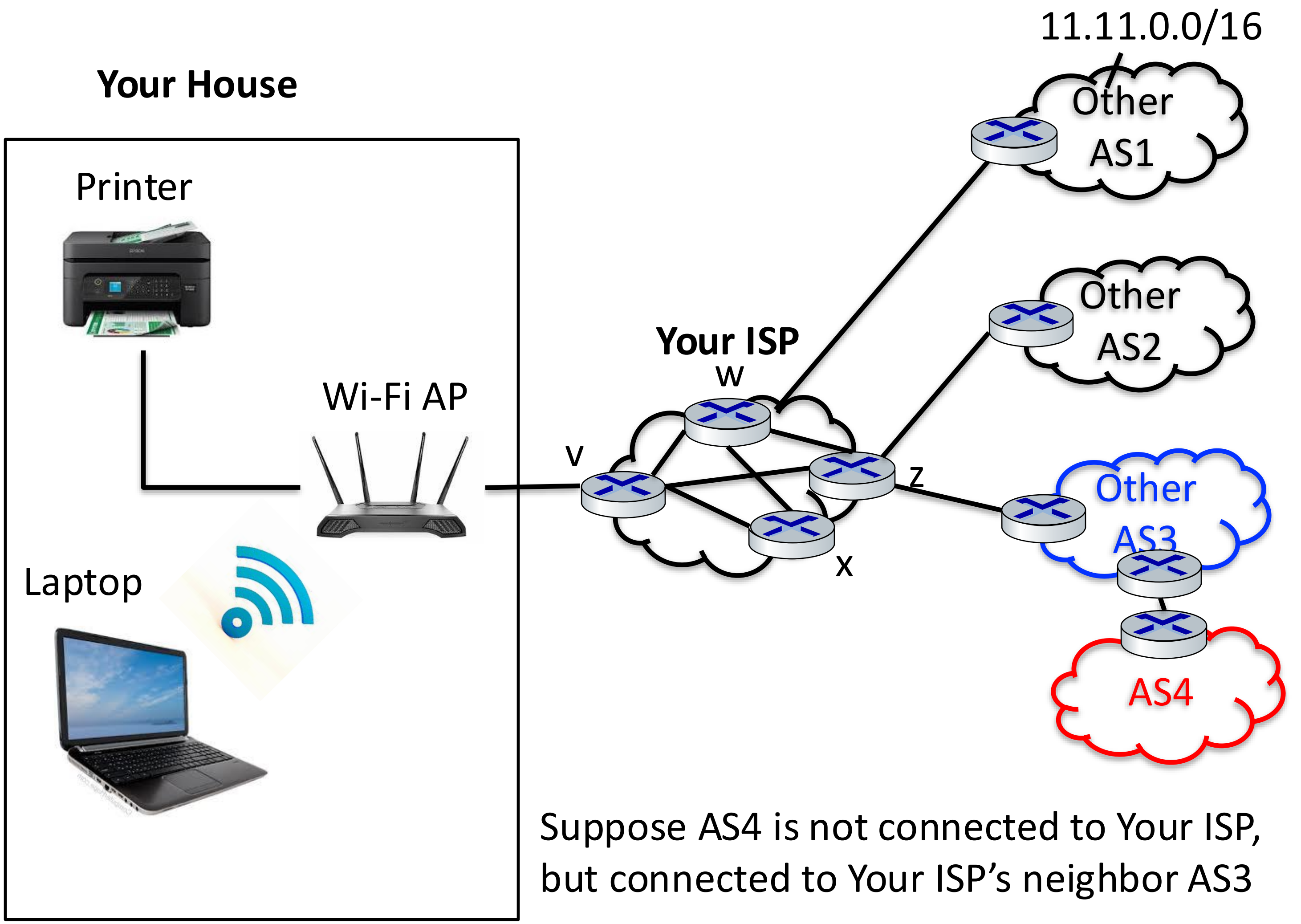
- Hop v from x
- (Then from x to z
- Then from z to other network)

V only cares about next hop!

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**



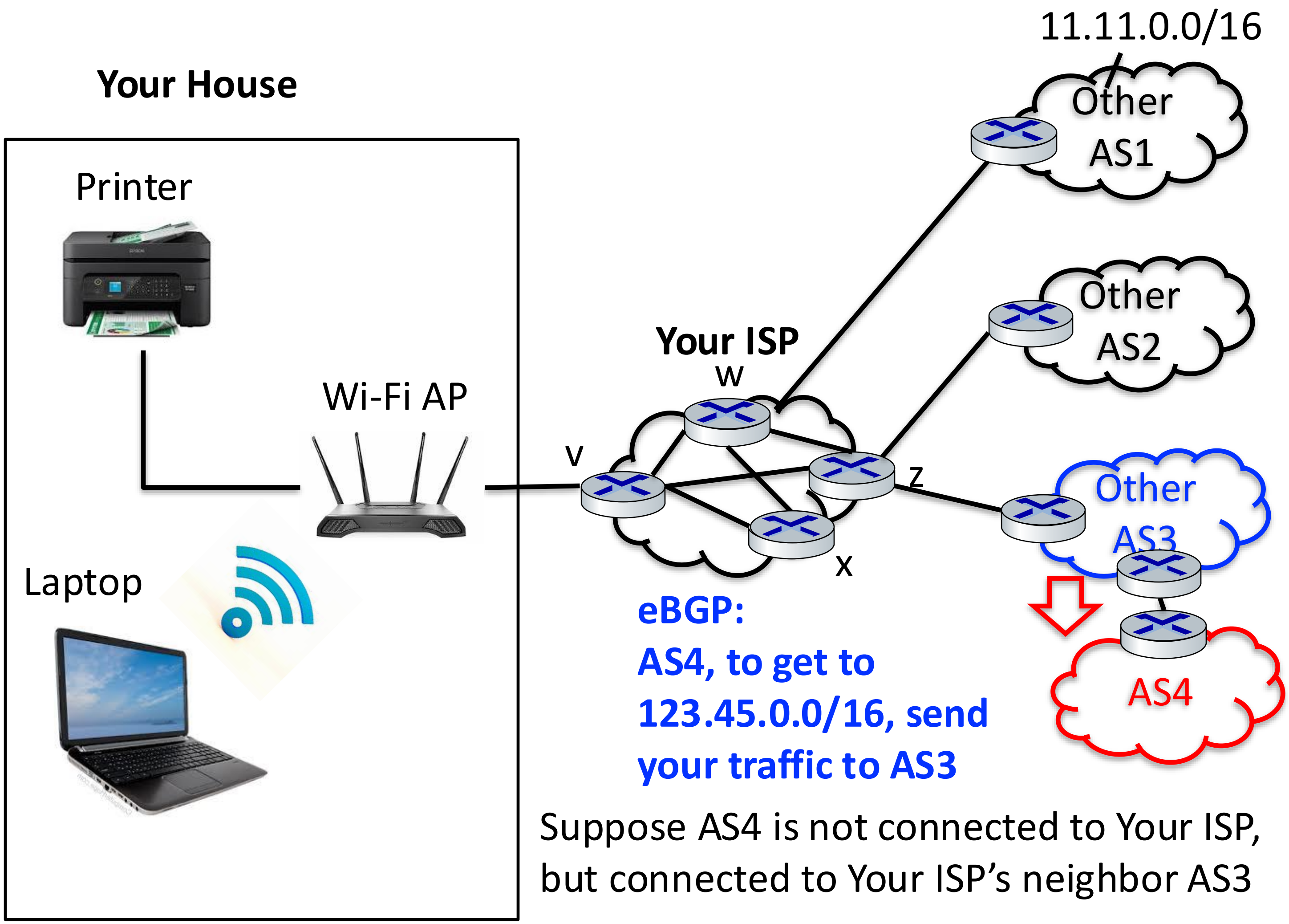
Suppose AS4 is not connected to Your ISP, but connected to Your ISP's neighbor AS3

AS3 tells AS4 to route its traffic through AS3 to get to 123.45.0.0/16

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**



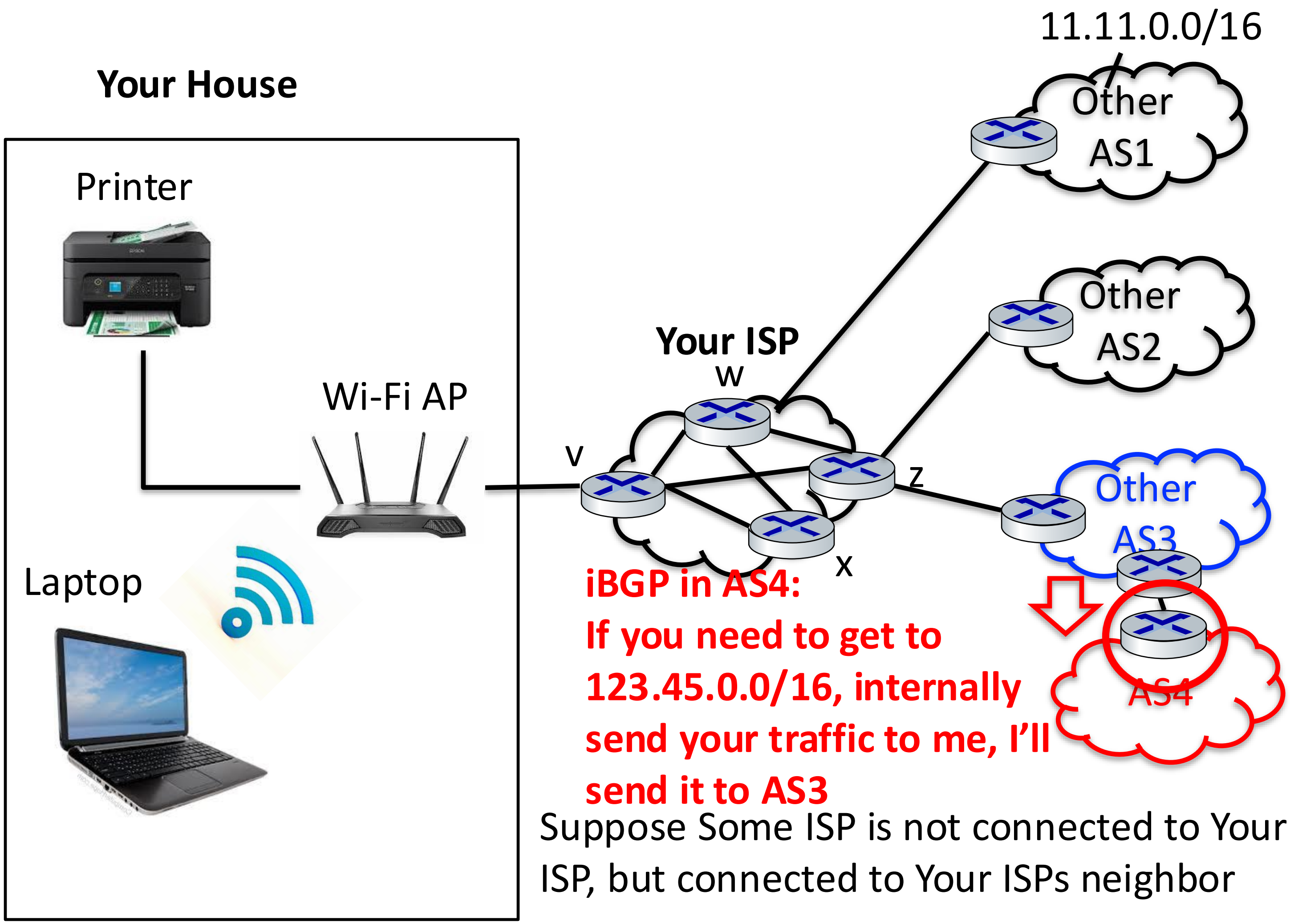
Suppose AS4 is not connected to Your ISP, but connected to Your ISP's neighbor AS3

AS3 tells AS4 to route its traffic through AS3 to get to Your ISP

Your ISP is connected to other AS'es via Border Gateway Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- **BGP (eBGP/iBGP)**



Suppose Some ISP is not connected to Your ISP, but connected to Your ISPs neighbor

Neighbor tells its neighbor to route through it to get to Your ISP

Big picture

Steps:

1. Get routes to Internet sorted out at ISP
2. Get an IP address from ISP and set up NAT

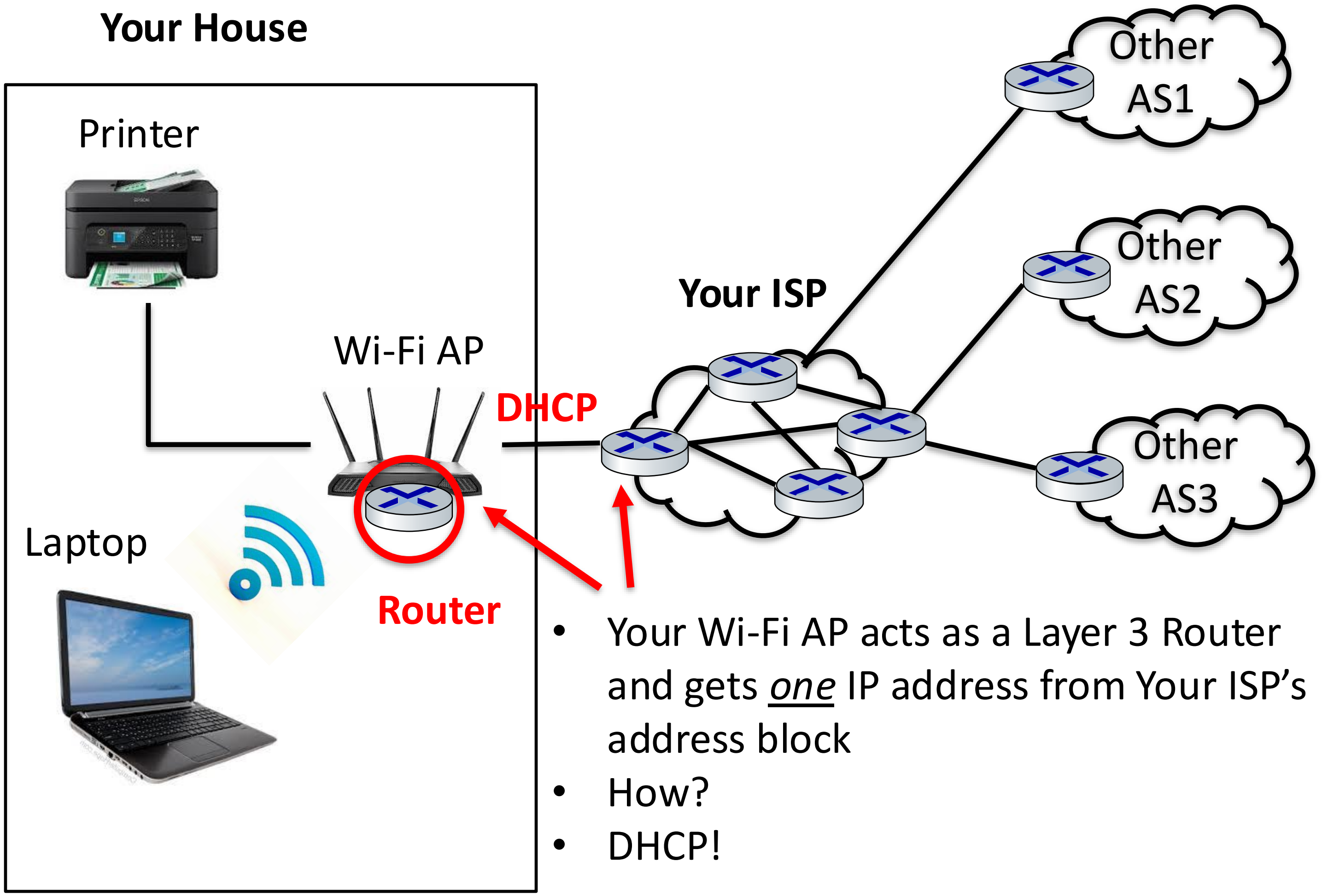
Examples:

1. Send job from laptop to printer in network
2. Make HTTP web request outside network

Your Wi-Fi AP gets an IP address from Your ISP

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP



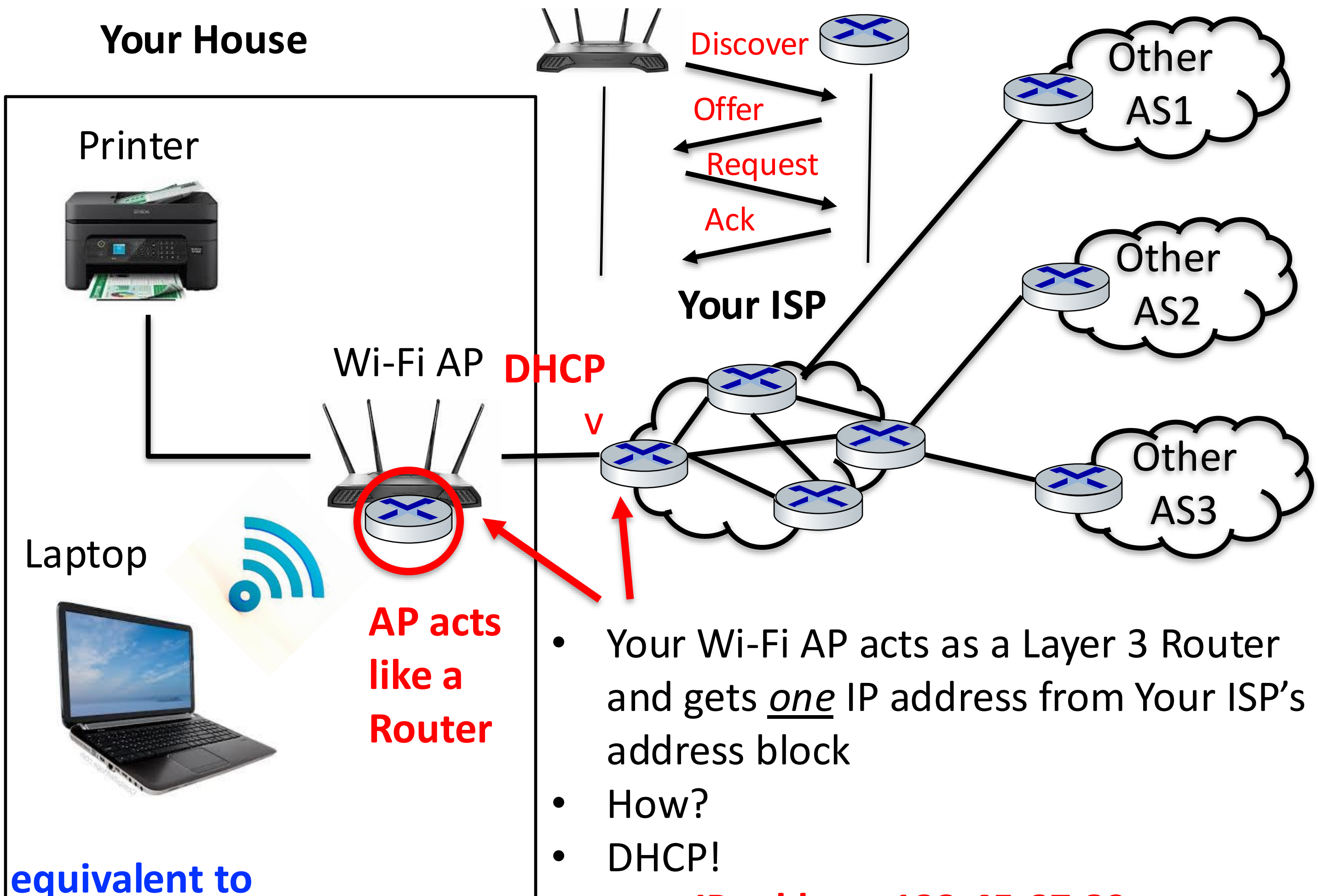
- Your Wi-Fi AP acts as a Layer 3 Router and gets *one* IP address from Your ISP's address block
- How?
- DHCP!

Your Wi-Fi AP gets an IP address from Your ISP

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- **DHCP**
- **Broadcast**

DHCP Discover sent to broadcast Wi-Fi AP doesn't know about DHCP server



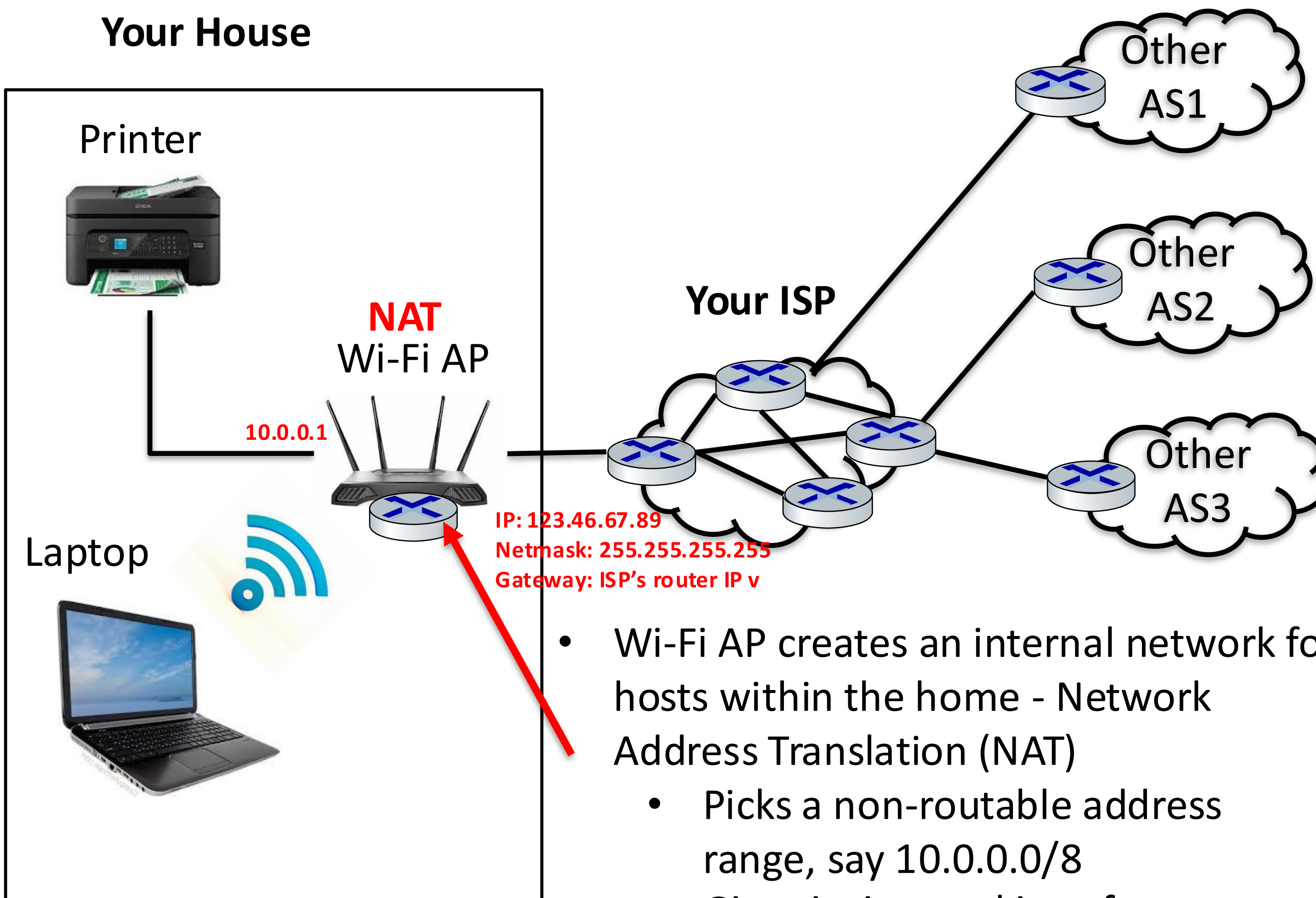
- Your Wi-Fi AP acts as a Layer 3 Router and gets *one* IP address from Your ISP's address block
- How?
- DHCP!
  - **IP address 123.45.67.89**
  - **Network mask 255.255.255.255**
  - **ISP's router *v* as next hop gateway**

Note: the netmask is equivalent to 123.45.67.89/32, just one address! All 32 netmask bits identify the network, no hosts

Your Wi-Fi AP sets up Network Address Translation

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT

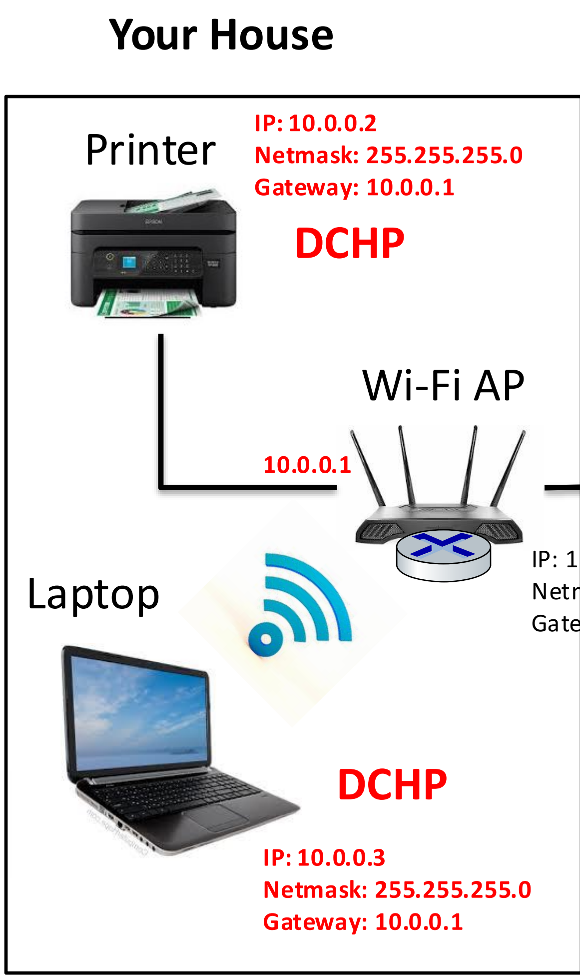


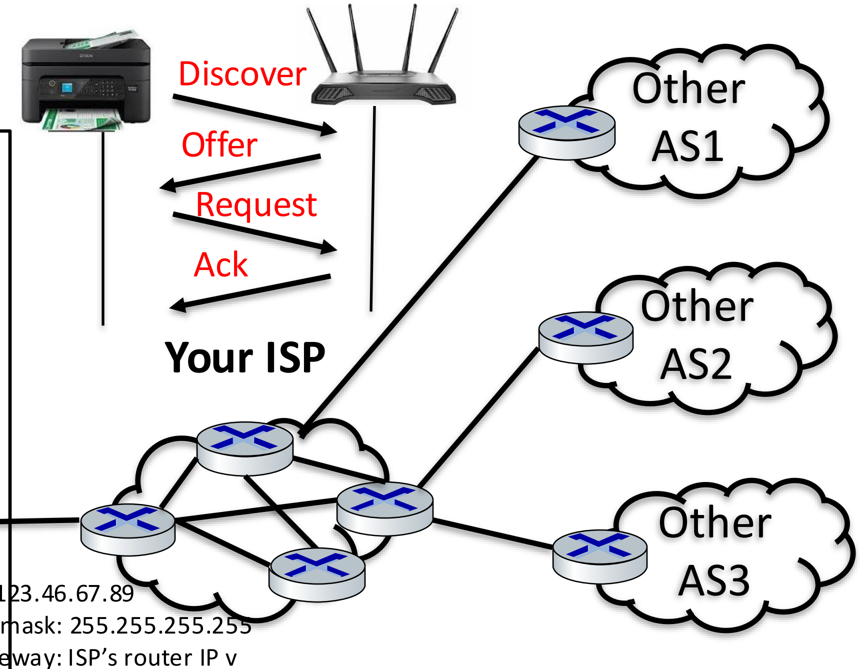
- Wi-Fi AP creates an internal network for hosts within the home - Network Address Translation (NAT)
  - Picks a non-routable address range, say 10.0.0.0/8
  - Gives its internal interface an address on the chosen range
  - 10.0.0.1 here

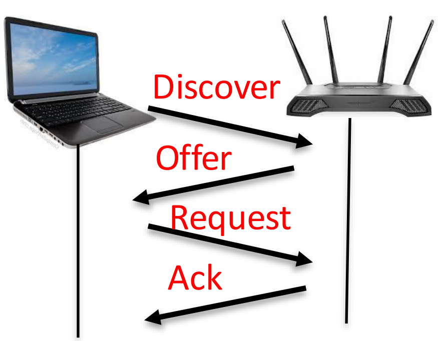
Hosts on the LAN get IP address, netmask, and next hop IP via DHCP

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- **NAT**







- Printer and Laptop IP address get from AP (router) via DHCP
  - IP address (10.0.0.2 and 10.0.0.3)
  - Netmask 255.255.255.0
  - Next hop router's IP: 10.0.0.1

Big picture

Steps:

1. Get routes to Internet sorted out at ISP
2. Get an IP address from ISP and set up NAT

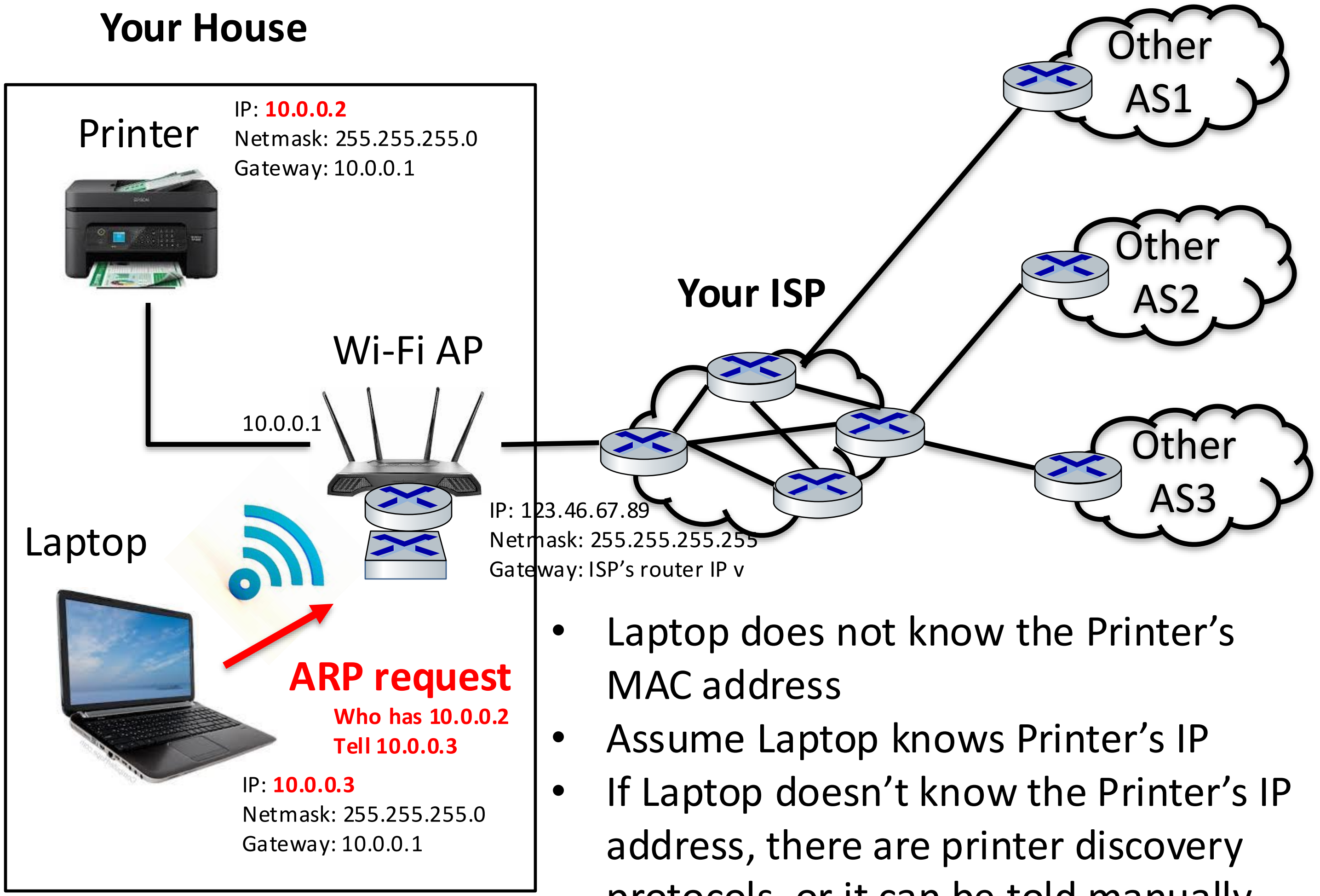
Examples:

1. Send job from laptop to printer in network
2. Make HTTP web request outside network

Laptop wants to send a print job to the printer on the internal network

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP



- Laptop does not know the Printer's MAC address
- Assume Laptop knows Printer's IP
- If Laptop doesn't know the Printer's IP address, there are printer discovery protocols, or it can be told manually
- Laptop sends an ARP message looking for "who has IP address, tell Laptop"

Laptop wants to send a print job to the printer, uses Address Resolution Protocol

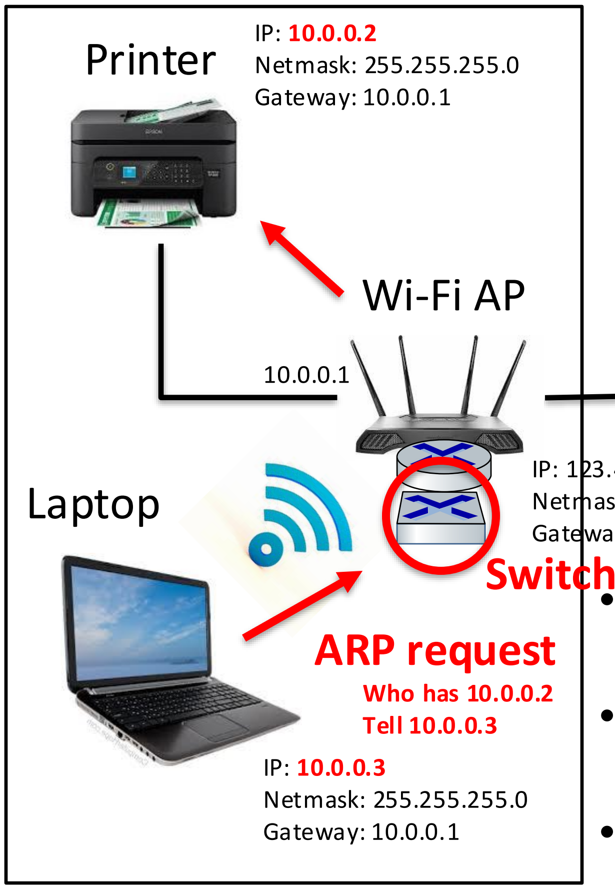
Networking Concepts:

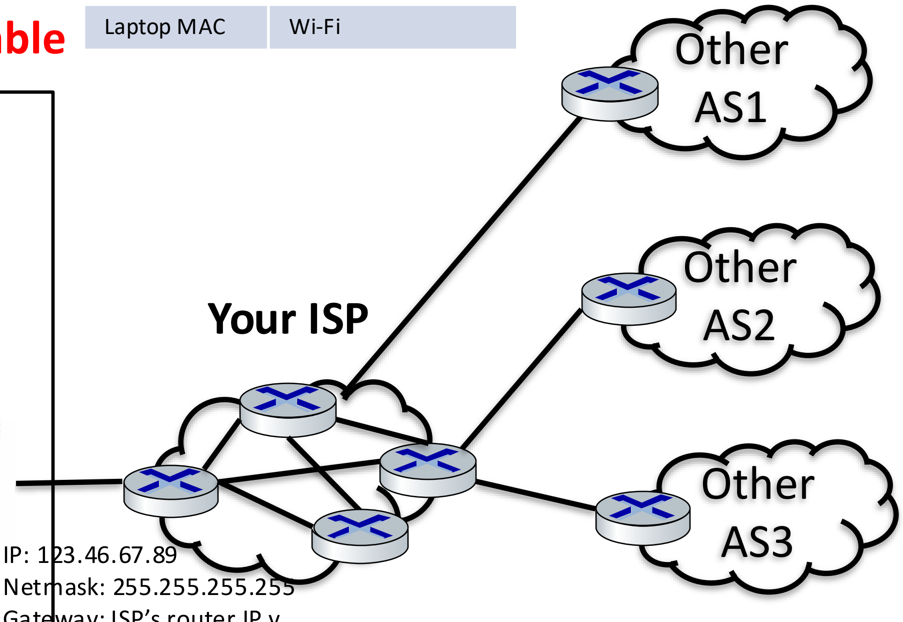
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- **LAN**
- **Switch**
- **MAC address**
- **MAC table**

Your House

AP switch table

MAC	Port
Laptop MAC	Wi-Fi





- AP broadcasts ARP message to all hosts on the Local Area Network (LAN)
- Broadcasts do not leave LAN, so they do not get to ISP! (broadcast is not routed)
- AP is acting like a Layer 2 switch, broadcasting at MAC layer (ff:ff:ff:ff:ff:ff)
- AP updates MAC table, knows Laptop is on Wi-Fi

Laptop wants to send a print job to the printer, uses Address Resolution Protocol

Networking Concepts:

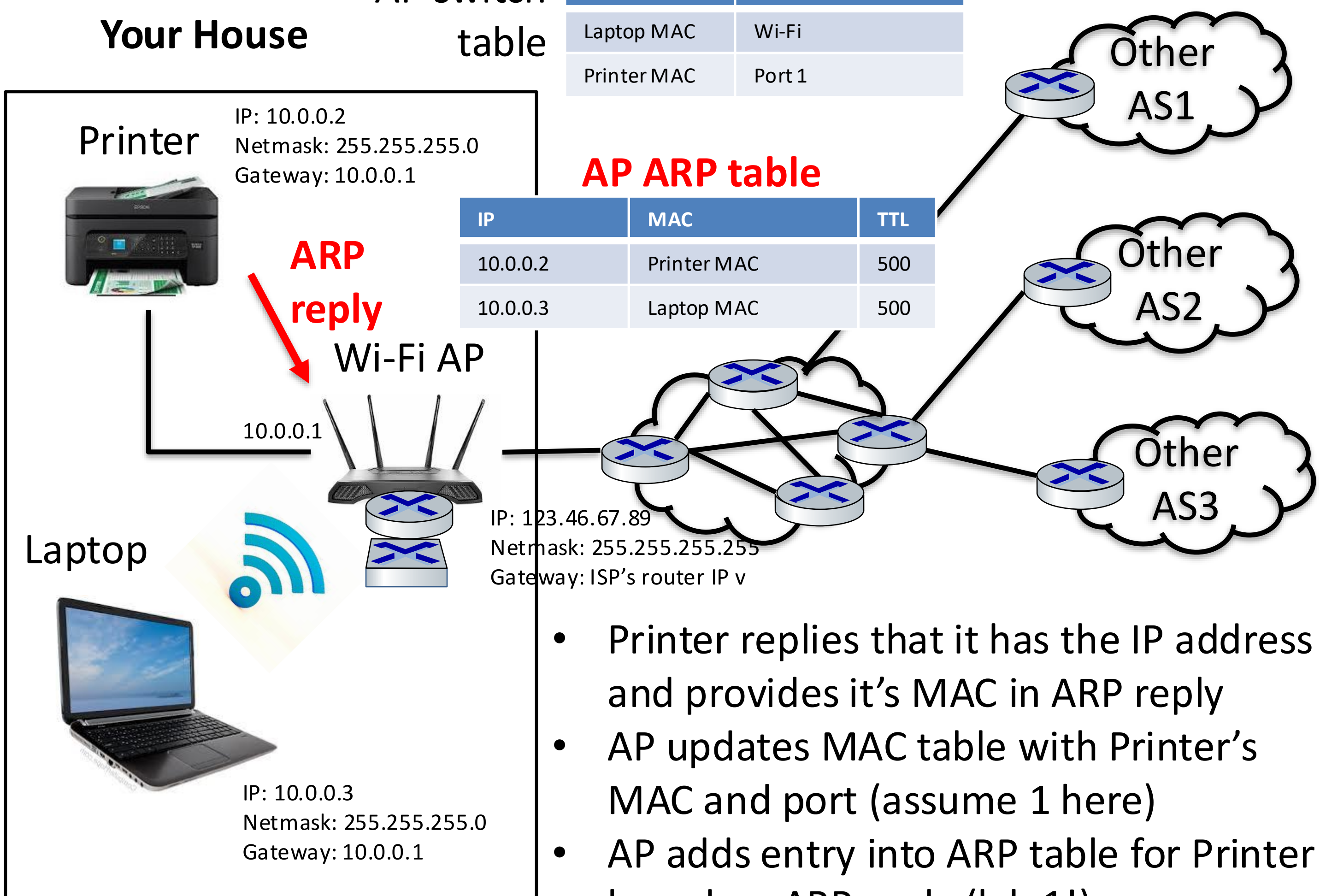
- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- **ARP table**

AP switch table

MAC	Port
Laptop MAC	Wi-Fi
Printer MAC	Port 1

AP ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	500
10.0.0.3	Laptop MAC	500



Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450

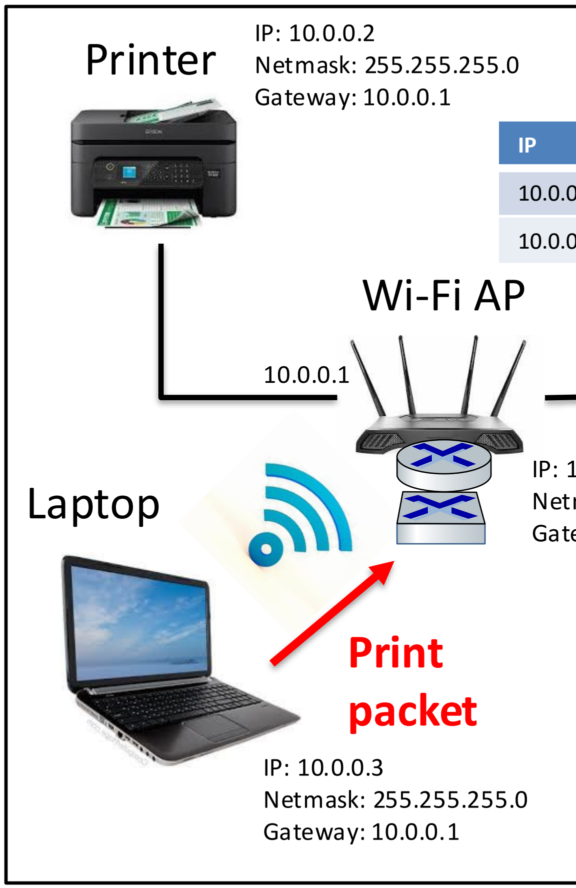
- Printer replies that it has the IP address and provides it's MAC in ARP reply
- AP updates MAC table with Printer's MAC and port (assume 1 here)
- AP adds entry into ARP table for Printer based on ARP reply (lab 1!)
- AP sends reply to Laptop over Wi-Fi
- Laptop updates ARP table

Laptop wants to send a print job to the printer, uses Address Resolution Protocol

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- **Netmask**

Your House

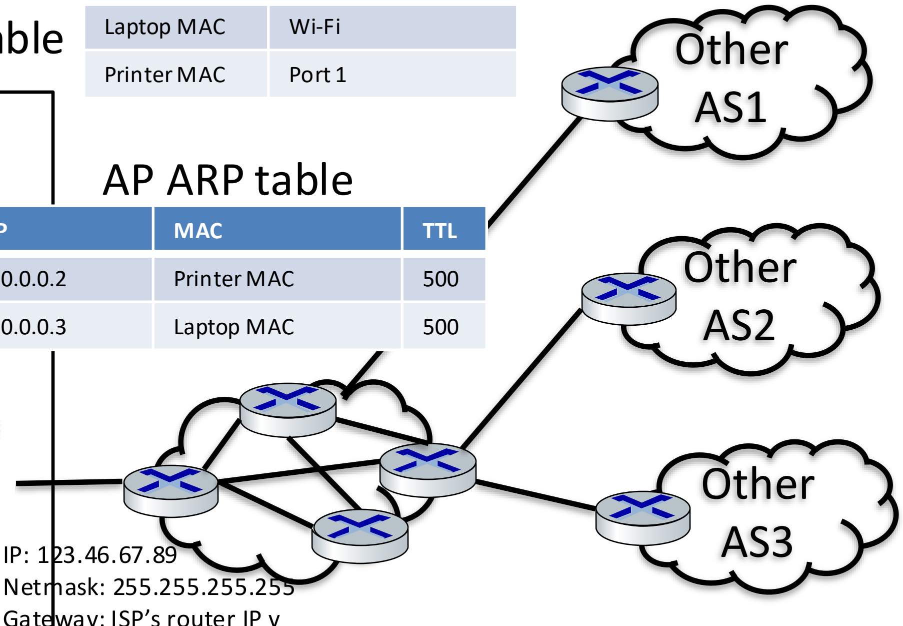


AP switch table

MAC	Port
Laptop MAC	Wi-Fi
Printer MAC	Port 1

AP ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	500
10.0.0.3	Laptop MAC	500



- Laptop creates print job sends it to AP
  - Layer 2: Laptop MAC, Printer MAC
  - Layer 3: src 10.0.0.3, dst 10.0.02
  - Layer 4: TCP or UDP
  - Layer 7: print job message
- Send via Wi-Fi to AP

Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450

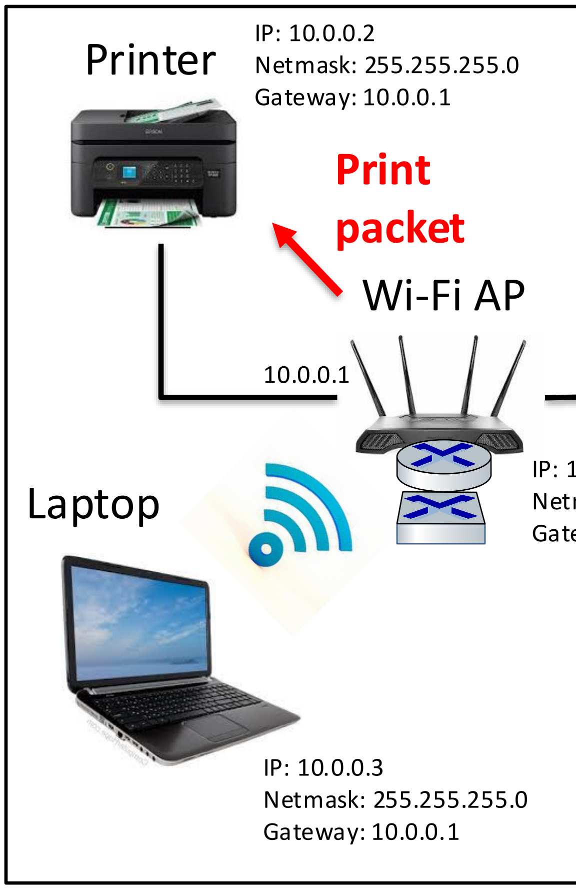
Laptop wants to send a print job to the printer, uses Address Resolution Protocol

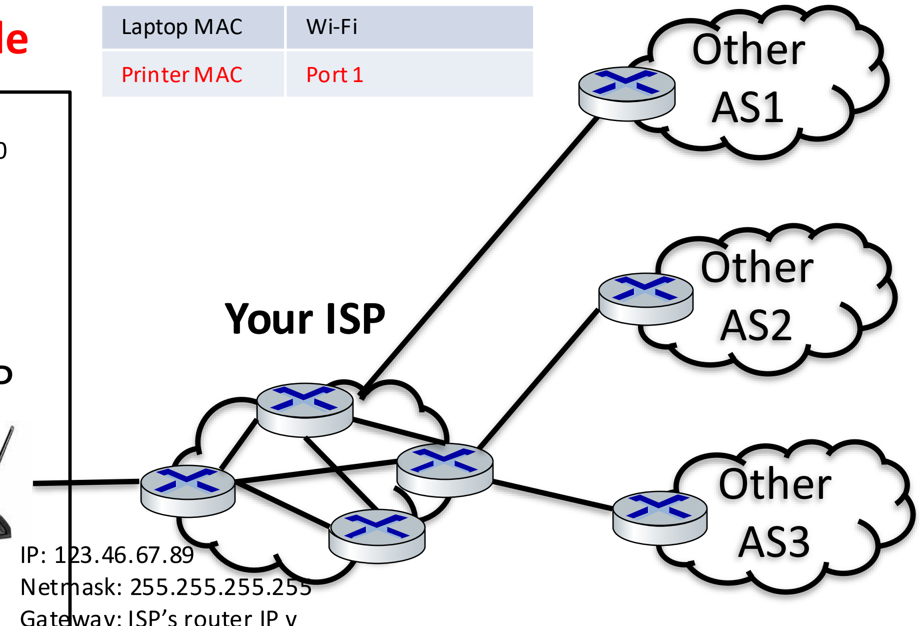
Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- **Netmask**

Switch table

MAC	Port
Laptop MAC	Wi-Fi
Printer MAC	Port 1





Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450

- AP PHY layer decodes radio waves to bits
- AP Switch checks dst IP address, realizes address *is* on its private network (e.g., 10.0.0.0/8) using netmask
- AP forwards to Printer using port matching Printer's MAC in Switch table
- No routing needed
- Routing happens when leaving LAN

Big picture

Steps:

1. Get routes to Internet sorted out at ISP
2. Get an IP address from ISP and set up NAT

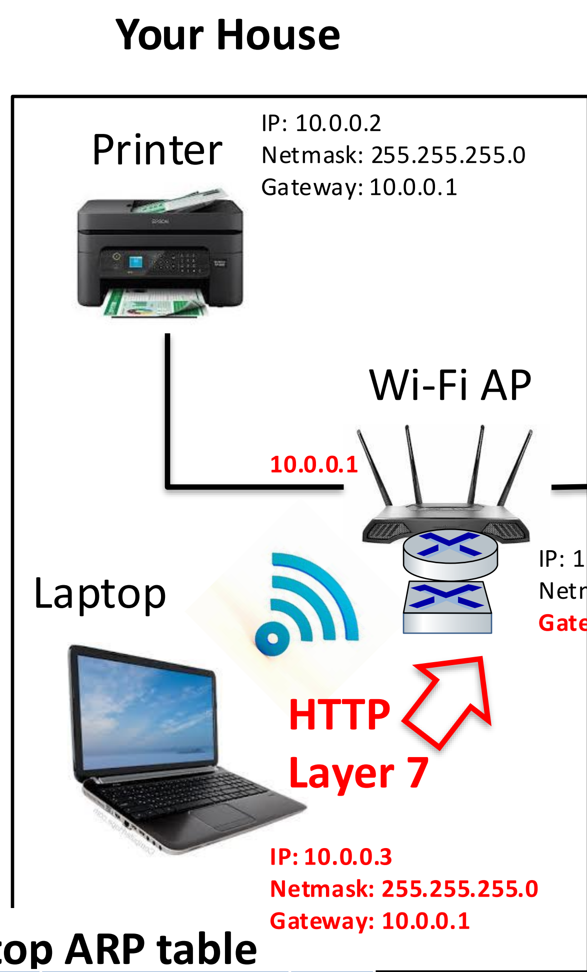
Examples:

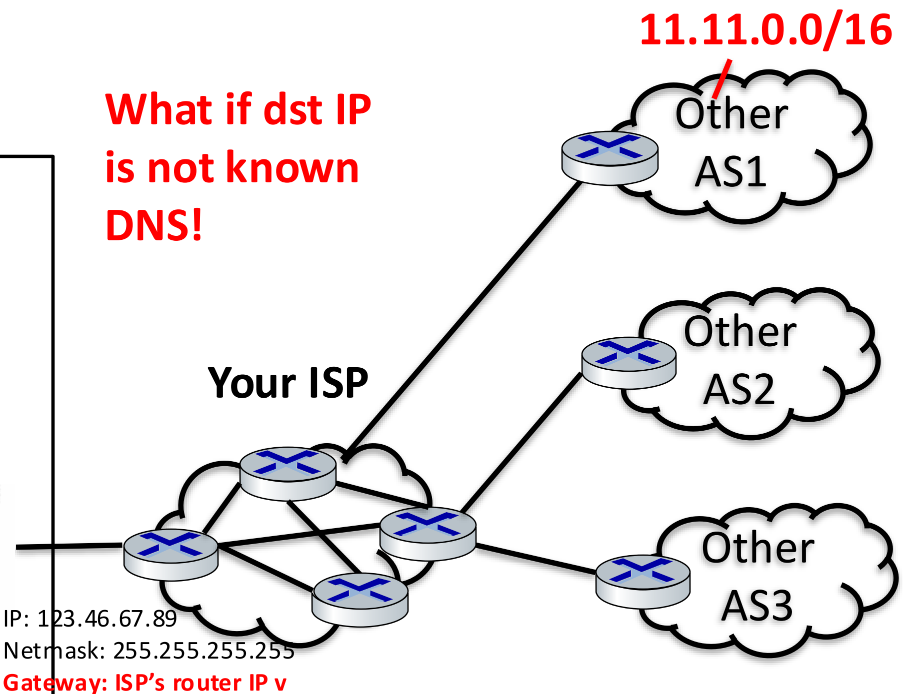
1. Send job from laptop to printer in network
2. Make HTTP web request outside network

Laptop makes request HTTP for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- **HTTP**
- **DNS**





Laptop ARP table

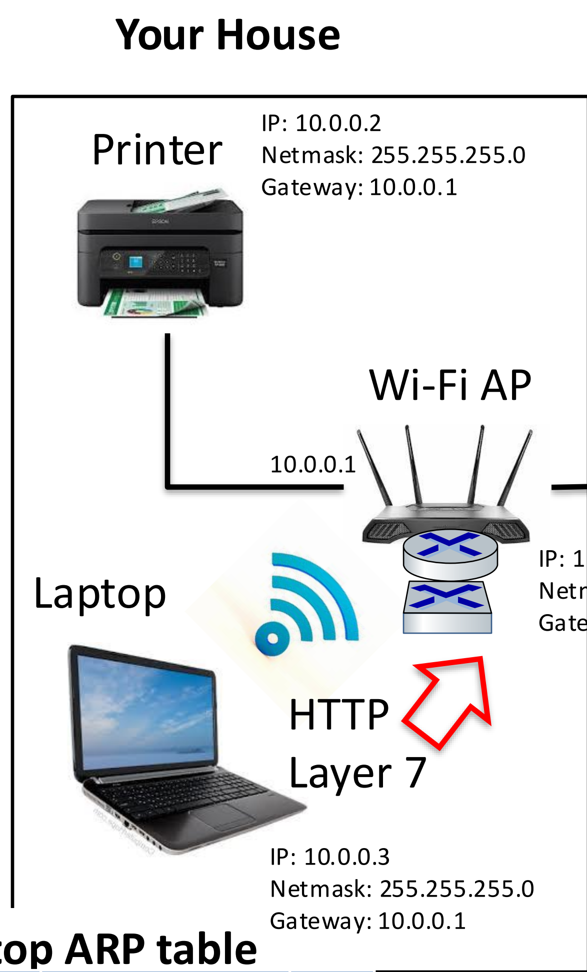
IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

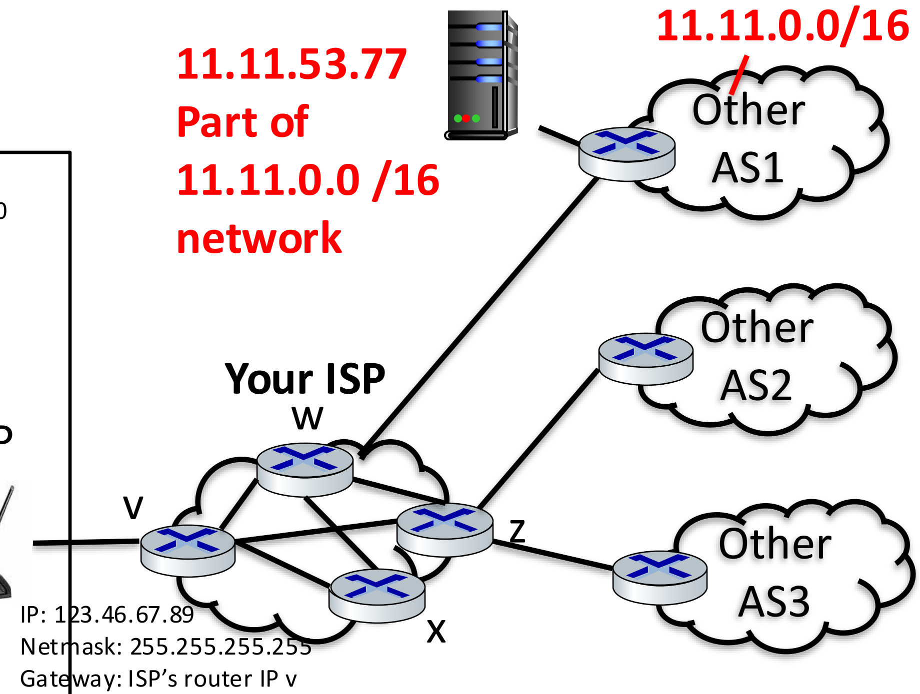
- Laptop makes HTTP Layer 7 web page request to web server at 11.11.53.27
- Laptop looks at dst IP (11.11.53.27), realizes *not* on 10.0.0.0/8 network
- Laptop makes ARP request to get Gateway router's MAC at IP 10.0.0.1
- AP (gateway router) replies with MAC
- Laptop updates ARP table

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- **HTTP/TCP**
- **DNS**





Laptop ARP table

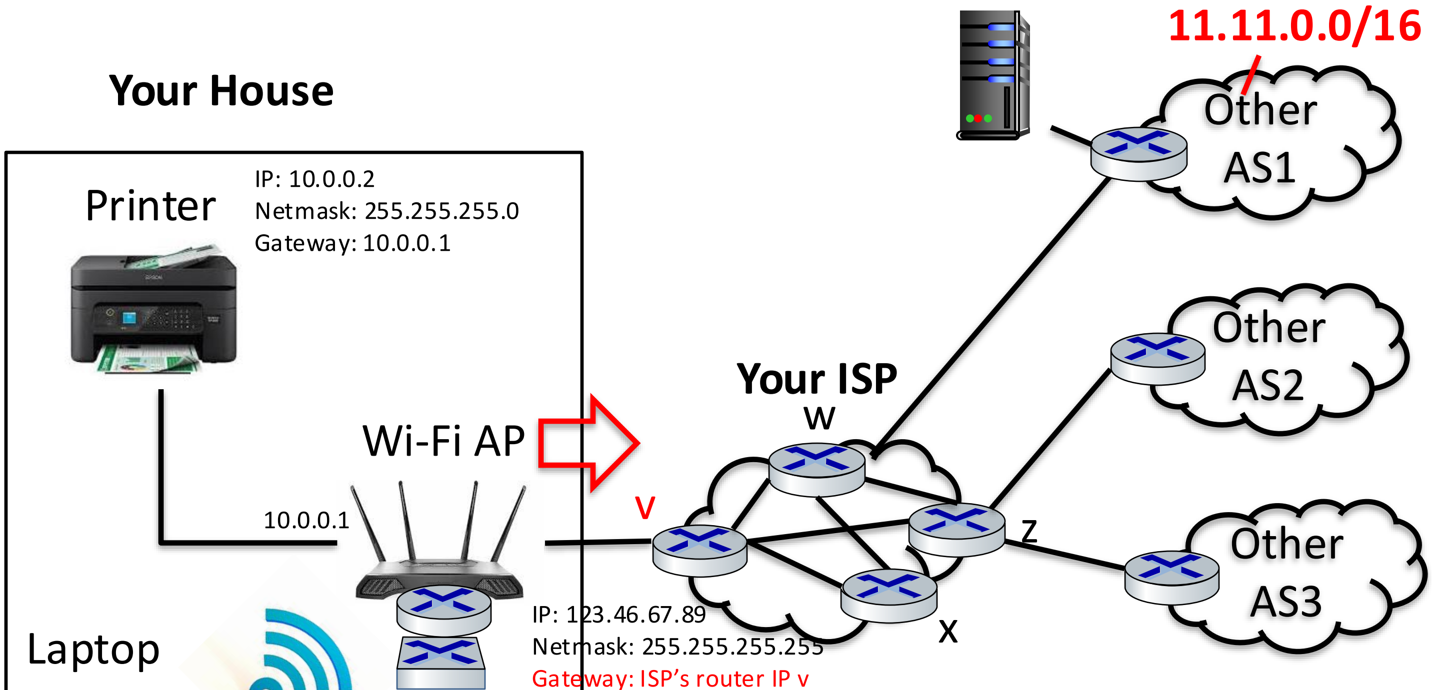
IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

- Laptop formats HTTP Layer 7 GET message
  - Layer 2: Laptop MAC, AP MAC
  - Layer 3: src 10.0.0.3, dst 11.11.53.27
  - Layer 4: TCP sport rnd, dport 80
  - Layer 7: HTTP GET message
- Sends packet
- Packet received by AP

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



Laptop ARP table

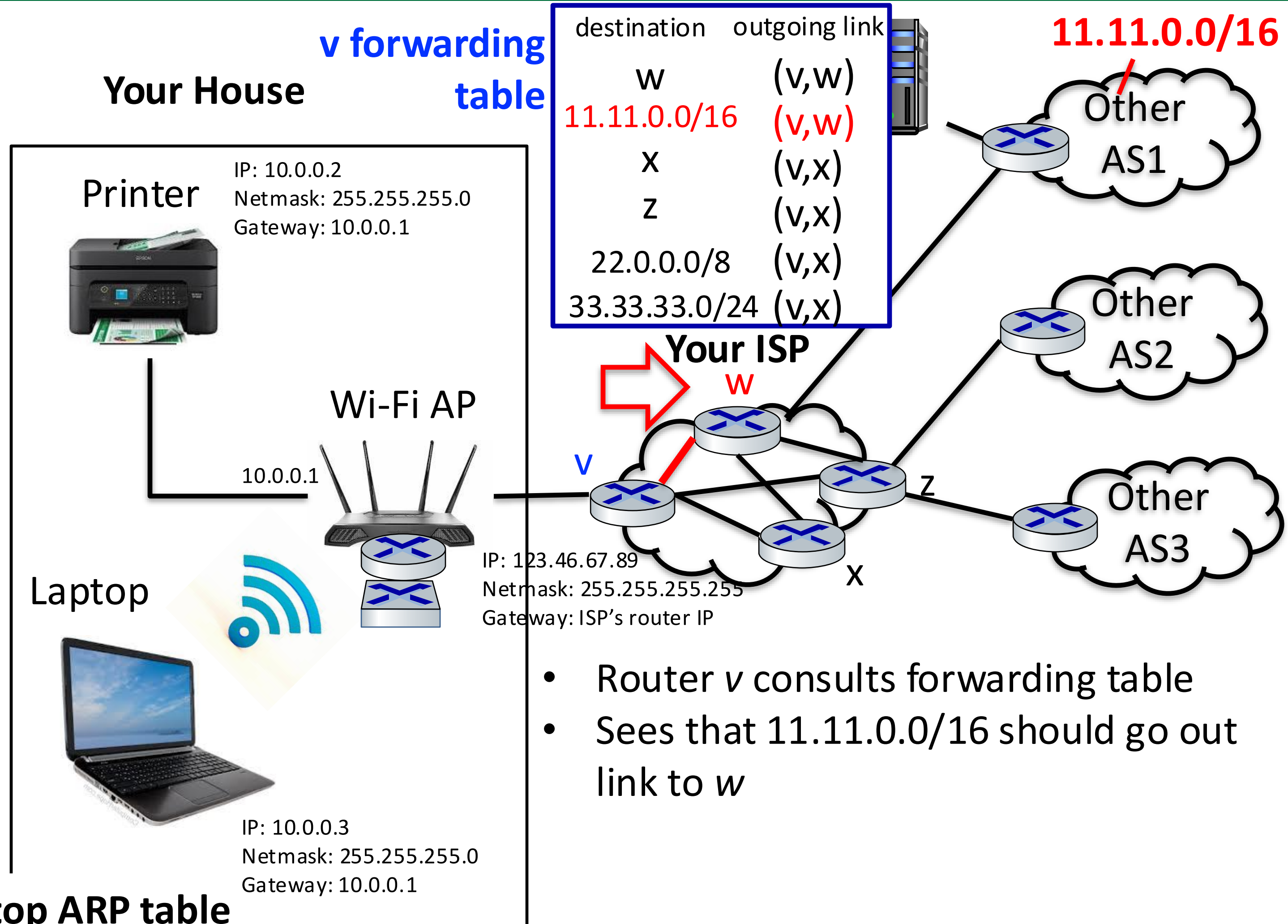
IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

- AP receives at Layer 2
- Checks MAC matches AP's MAC
- At Layer 3 AP sees that dst IP is not on the local network, forward to gateway
- Rewrites Layer 2 headers with gateway's MAC address (Your ISP router v) as dst MAC
- Route packet to ISP router v

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



destination	outgoing link
w	(v,w)
11.11.0.0/16	(v,w)
x	(v,x)
z	(v,x)
22.0.0.0/8	(v,x)
33.33.33.0/24	(v,x)

- Router *v* consults forwarding table
- Sees that 11.11.0.0/16 should go out link to *w*

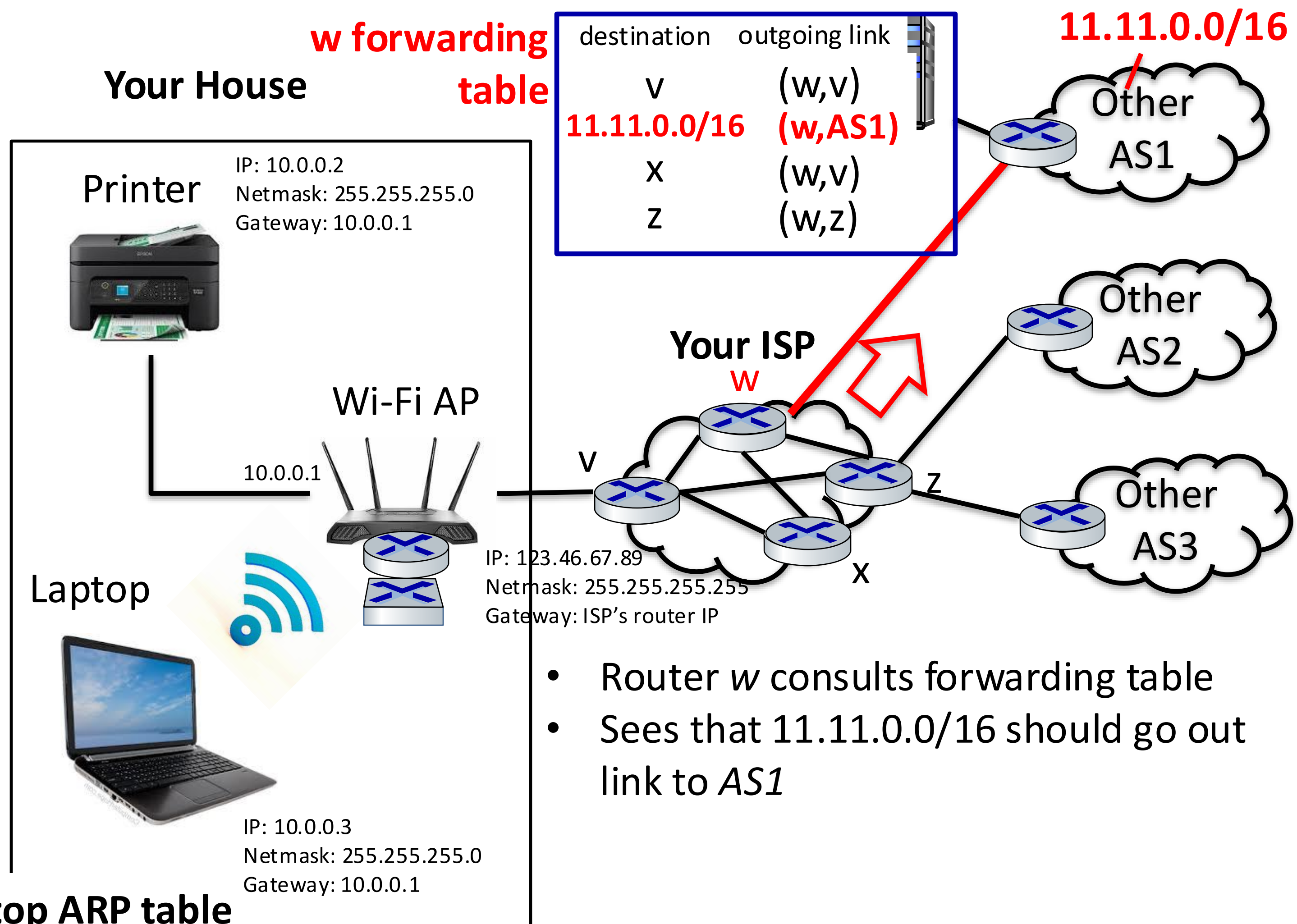
Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



- Router *w* consults forwarding table
- Sees that 11.11.0.0/16 should go out link to *AS1*

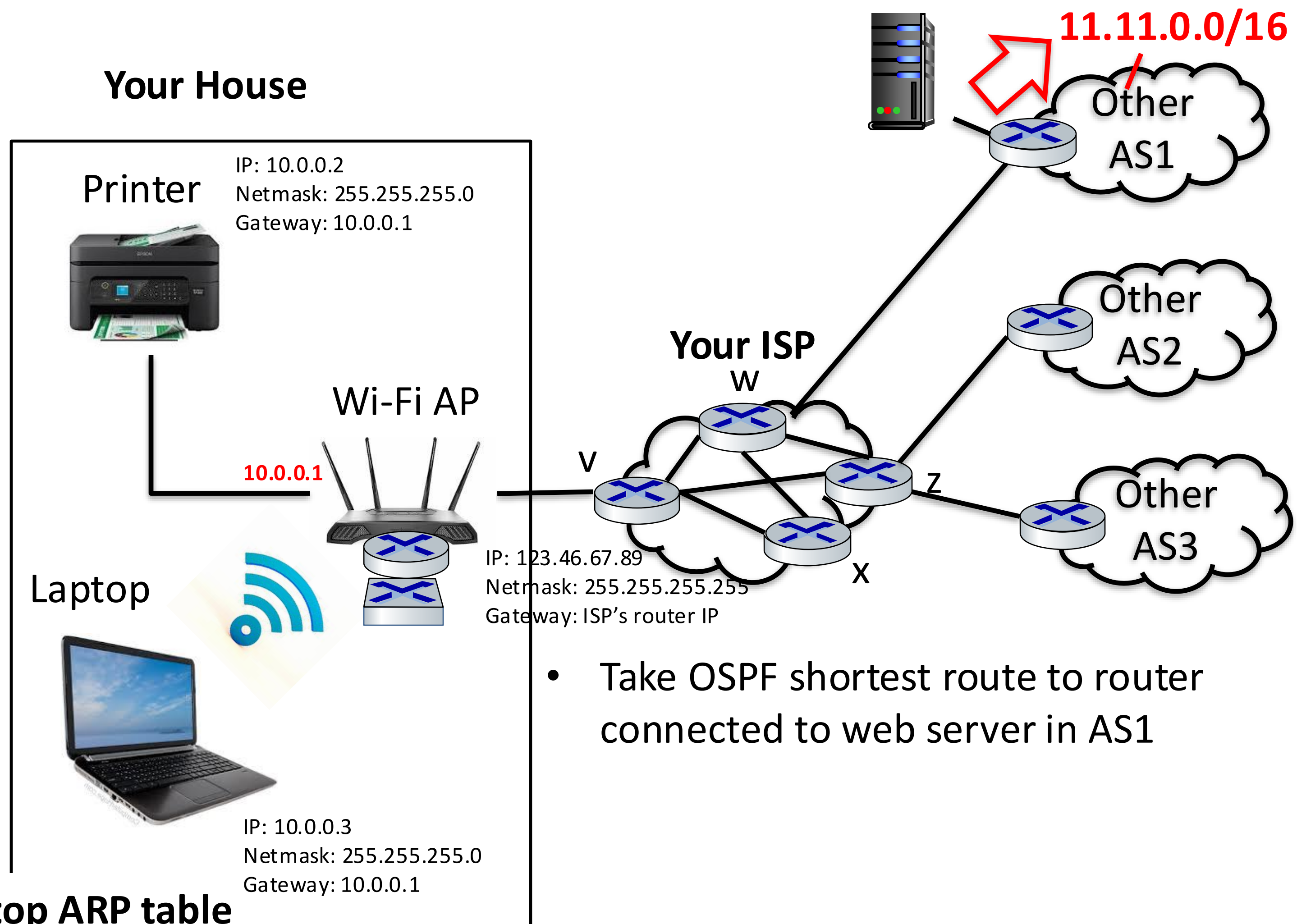
Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



- Take OSPF shortest route to router connected to web server in AS1

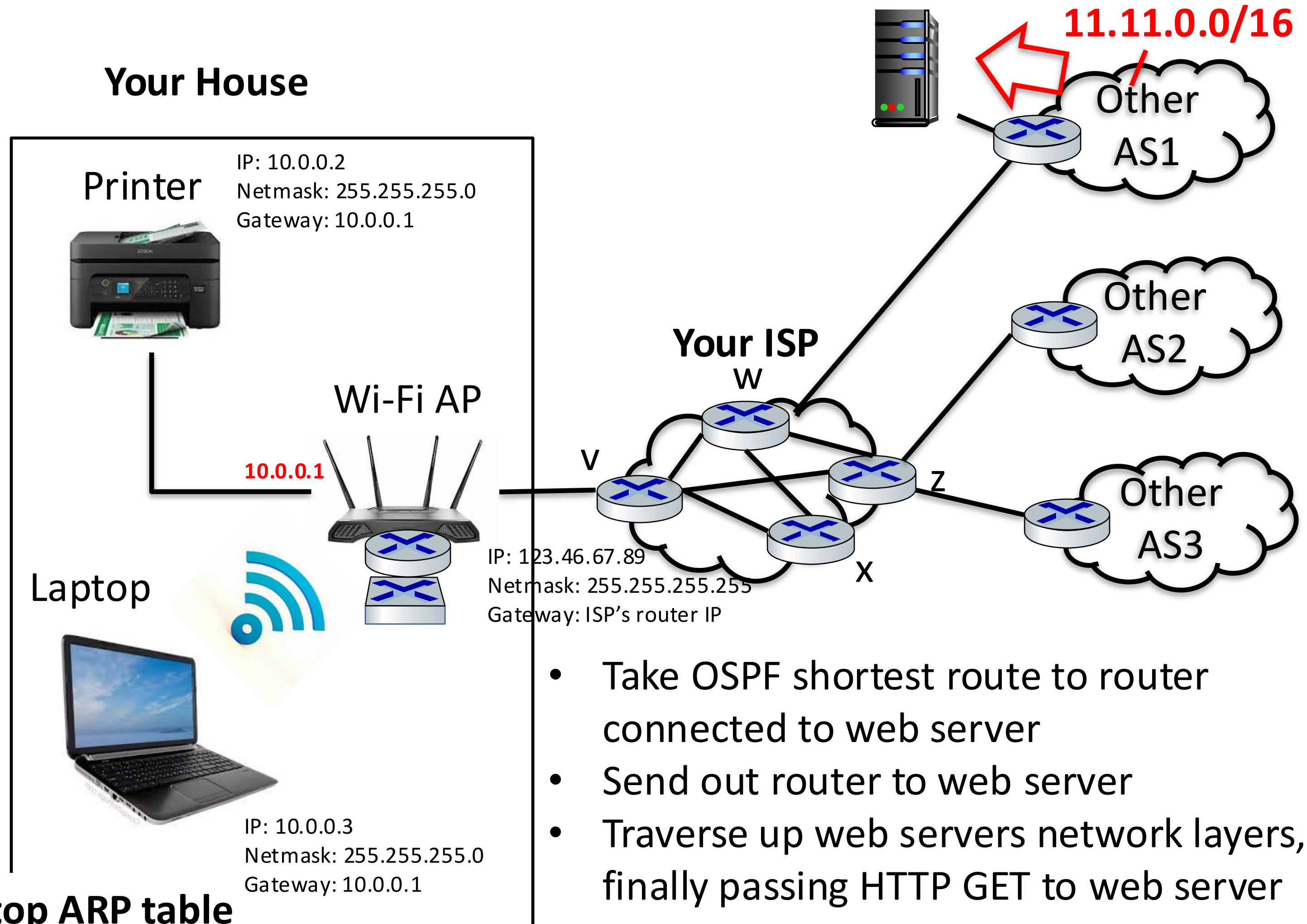
Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



Laptop ARP table

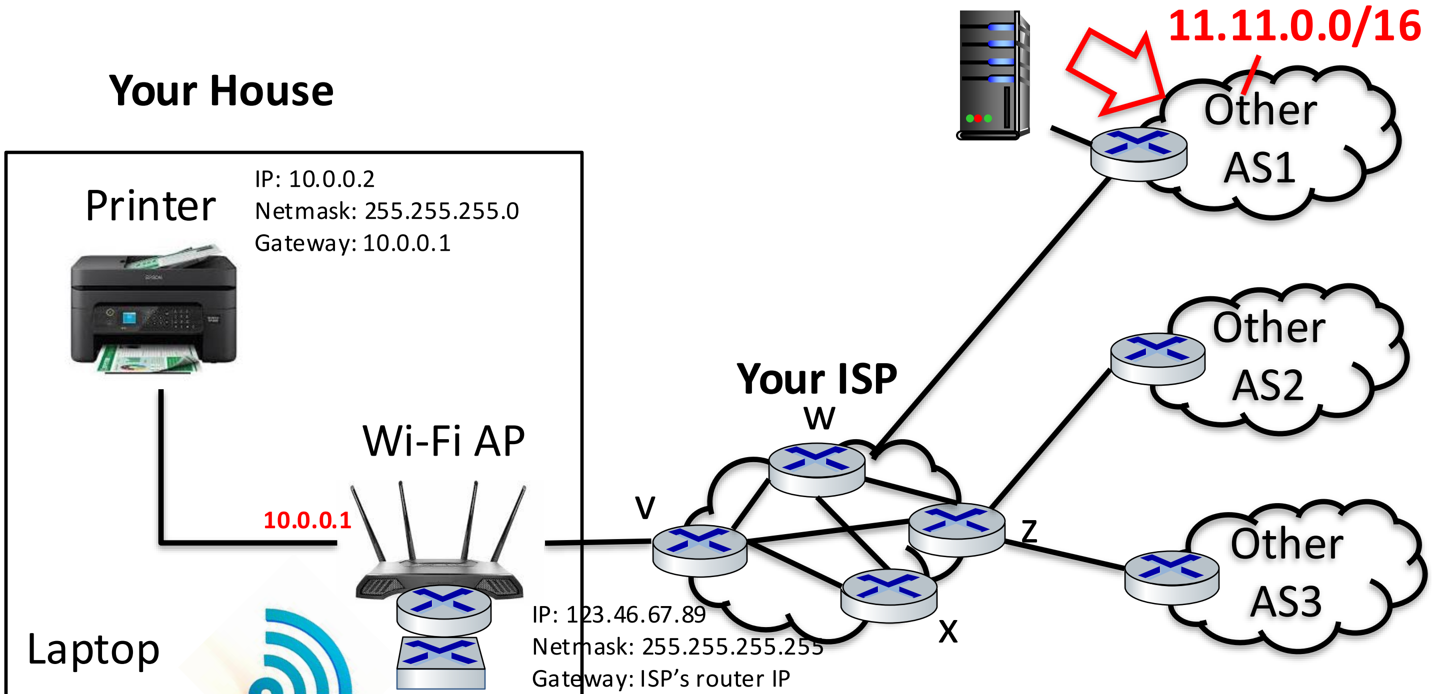
IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

- Take OSPF shortest route to router connected to web server
- Send out router to web server
- Traverse up web servers network layers, finally passing HTTP GET to web server

Laptop makes request for web page from server at 11.11.53.27

Networking Concepts:

- Ethernet/Wi-Fi
- ISP
- IP address
- AS
- Router
- Forwarding
- Routing
- OSPF
- Forwarding table
- BGP (eBGP/iBGP)
- DHCP
- Broadcast
- NAT
- ARP
- LAN
- Switch
- MAC address
- MAC table
- ARP table
- Netmask
- HTTP/TCP
- DNS



Laptop ARP table

IP	MAC	TTL
10.0.0.2	Printer MAC	450
10.0.0.1	Router's MAC	500

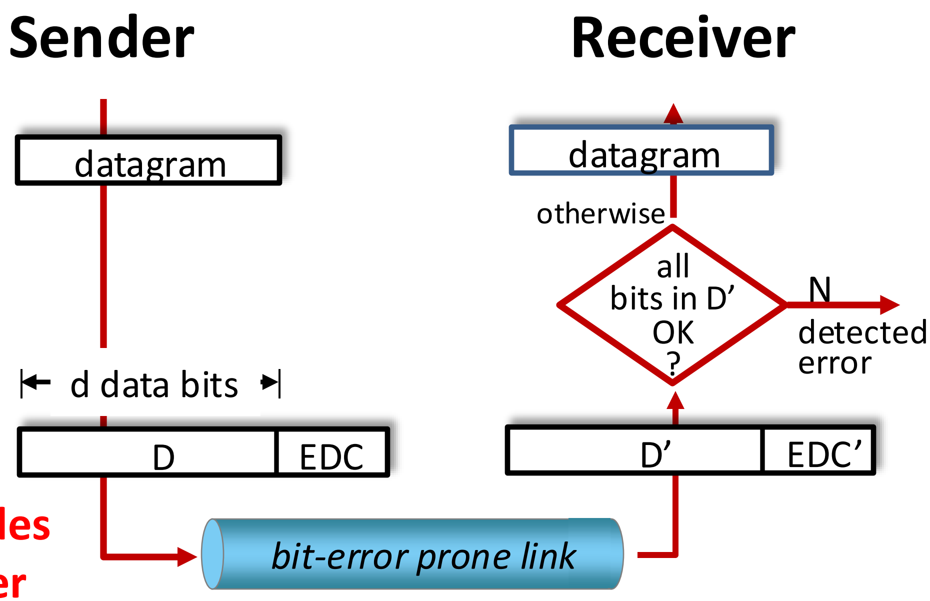
- Take OSPF shortest route to router connected to web server
- Send out router to web server
- Traverse web servers networks, finally passing HTTP GET to web server
- Reply reverses process, reverse src and dst IP

Agenda

1. Ethernet
2. Putting it all together
3. Error detection/correction
4. Channel sharing

Error detection and correction can help with noisy links

EDC: Error Detection and Correction bits (e.g., redundancy)
D: Data protected by error checking, may include header fields



D includes link layer headers

Augmented with error detection and correction bits

D' may differ from D (same for EDC' and EDC) after going through link

Error detection not 100% reliable!

- Protocol may miss some errors, but rarely
- Larger EDC field yields better detection and correction

What happens if error is passed up the network stack?
Likely caught at higher layer
Low probability error gets to App

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Three techniques for error detection and correction: 1) Parity checking

Single bit parity:

- Detect single bit errors

0111000110101011	1

← *d* data bits → | parity bit

9 data 1's + 1 parity bit
10 total bits
Even parity, set bit =1
Odd parity, set bit = 0

Even/odd parity: set parity bit so there is an even/odd number of 1's

At receiver:

- Compute parity of *d* received bits
- Compare with received parity bit – if different then error detected (more precisely odd number of bits in error)

Even number of bit errors would be undetected!

Can detect *and* correct errors (without retransmission!)

- Two-dimensional parity: detect *and correct* single bit errors

D bits divided into i rows and j columns

row parity →

column parity ↓

$d_{1,1}$	. . .	$d_{1,j}$	$d_{1,j+1}$
$d_{2,1}$	. . .	$d_{2,j}$	$d_{2,j+1}$
. . .	. . .	. . .	. . .
$d_{i,1}$	. . .	$d_{i,j}$	$d_{i,j+1}$
$d_{i+1,1}$	. . .	$d_{i+1,j}$	$d_{i+1,j+1}$

Parity bit for each row and column

no errors: (even parity)

1 0 1 0 1	1
1 1 1 1 0	0
0 1 1 1 0	1
0 0 1 0 1	0

detected and correctable single-bit error:

1 0 1 0 1	1
1 0 1 1 0	0
0 1 1 1 0	1
0 0 1 0 1	0

parity error

parity error

Can detect *and* fix single bit errors
Can detect but not fix two-bit errors
Resend data with errors immediately

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Three techniques for error detection and correction: 2) Internet checksum

Goal: detect errors (*i.e.,* flipped bits) in transmitted segment

Sender:

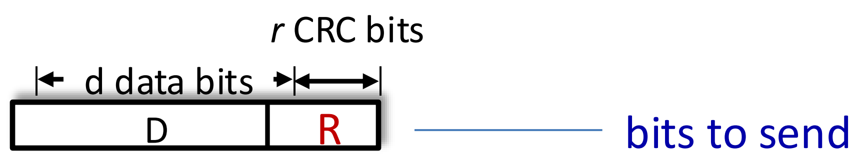
- Treat contents of Layer 3 segment (including header fields and IP addresses) as sequence of 16-bit integers
- Checksum: addition (one's complement sum) of segment content
- Checksum value put into IP and UDP/TCP checksum field

Receiver:

- Compute checksum of received segment
- Check if computed checksum equals checksum field value:
  - Not equal - error detected
  - Equal - no error detected. *But maybe errors nonetheless?*

Three techniques for error detection and correction: 3) Cyclic Redundancy Check

- More powerful error-detection coding
- D: data bits (given, think of these as a binary number)
- G: bit pattern (generator), of *r+1* bits (given, specified in CRC standard)



<D,R> = D∗2^r XOR R — formula for these bits

Sender: Compute *r* CRC bits, R, such that <D,R> *exactly* divisible by G (mod 2)

- Receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
- Can detect all burst errors less than r+1 bits
- Widely used in practice (Ethernet, 802.11 Wi-Fi)

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Cyclic Redundancy Check (CRC): example

Multiplying a number by 2^r shifts left by r bits

Sender wants to compute R such that:

$$D \cdot 2^r \ XOR \ R = nG$$

... or equivalently (XOR R both sides):

$$D \cdot 2^r = nG \ XOR \ R$$

... which says:

if we divide $D \cdot 2^r$ by G, we want remainder R to satisfy:

$$R = \textit{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$

algorithm for computing R

```
G
                1 0 1 0 1 1
1 0 0 1 ) 1 0 1 1 1 0 0 0 0
          1 0 0 1
              1 0 1
              0 0 0
              1 0 1 0
              1 0 0 1
                  1 1 0
                  0 0 0
                  1 1 0 0
                  1 0 0 1
                    1 0 1 0
                    1 0 0 1
                      0 1 1
                      R
```

$D * 2^r$ (here, r=3)

Can detect r+1 bit errors

International standards have created G for 8-, 12-, 16-, and 32-bit generators

G is know by all ahead of time

G_{32} = 100000100110000010001110110110111

Agenda

1. Ethernet
2. Putting it all together
3. Error detection/correction
4. Channel sharing

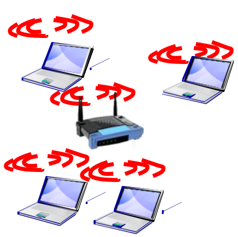
Multiple access links and protocols

Two types of “links”:

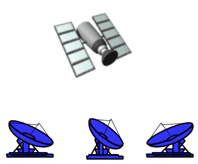
- Point-to-point
  - Point-to-point link between Ethernet switch and host
  - PPP for dial-up access
- Broadcast (shared wire or medium)
  - Old-school Ethernet
  - Upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G, satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G



shared radio: WiFi



shared radio: satellite

humans at a cocktail party (shared air, acoustical)

Multiple Access Protocols allow multiple hosts to share a medium

- Single shared broadcast channel
- Two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

Multiple Access Protocol

- Distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- Communication about channel sharing must use channel itself!
  - Assume no out-of-band channel for coordination

An <u>ideal</u> multiple access protocol

Given: multiple access channel (MAC) of rate *R* bps

Desired properties:

1. When one node wants to transmit, it can send at rate *R*
2. When *M* nodes want to transmit, each can send at average rate *R/M*
3. Fully decentralized:
   - No special node to coordinate transmissions
   - No synchronization of clocks, slots
4. Simple

MAC protocols: taxonomy

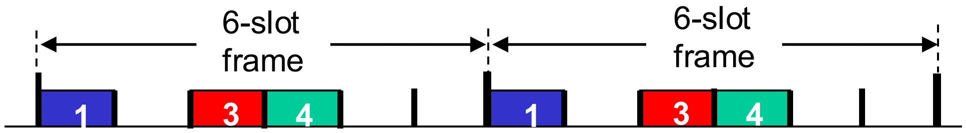
Three broad classes:

- Channel partitioning
  - Divide channel into smaller “pieces” (time slots, frequency, code)
  - Allocate piece to node for exclusive use
- Random access
  - Channel not divided, allow collisions
  - “Recover” from collisions
- “Taking turns”
  - Nodes take turns, but nodes with more to send can take longer turns

Channel partitioning MAC protocols: TDMA

TDMA: Time Division Multiple Access

- Access to channel in “rounds”
- Each station gets fixed length slot (length = packet transmission time) in each round
- Unused slots go idle
- Example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle

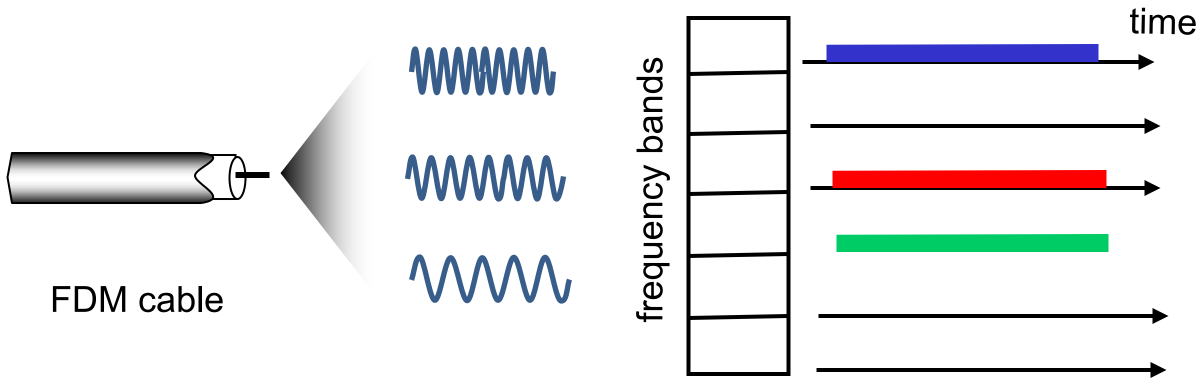


Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Channel partitioning MAC protocols: FDMA

FDMA: Frequency Division Multiple Access

- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands go idle
- Example: 6-station LAN, 1,3,4 have packet to send, frequency bands 2,5,6 idle

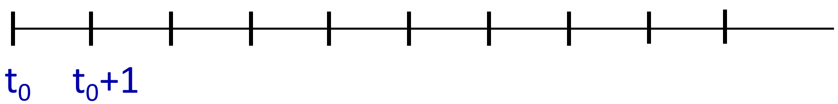


Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Random access protocols

- When node has packet to send
  - Transmit at full channel data rate R
  - No *a priori* coordination among nodes
- Two or more transmitting nodes: “collision”
- Random access protocol specifies:
  - How to detect collisions
  - How to recover from collisions (e.g., via delayed retransmissions)
- Examples of random-access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



Assumptions:

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only slot beginning
- Nodes are time synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

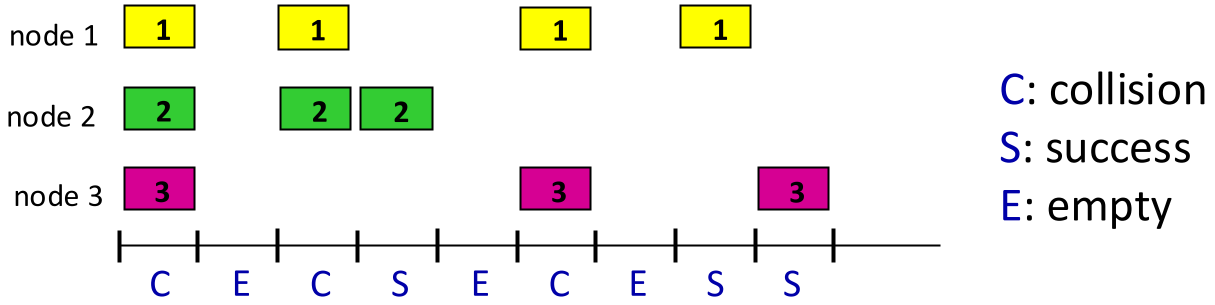
Operation:

- When node obtains fresh frame, transmits in next slot
  - *If no collision:* node can send new frame in next slot
  - *If collision:* node retransmits frame in each subsequent slot with probability *p* until success

randomization – *why*?

Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

Slotted ALOHA



Pros:

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons:

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

Slotted ALOHA: efficiency

Efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- *Suppose: N* nodes with many frames to send, each transmits in slot with probability *p*
  - prob that given node has success in a slot = $p(1-p)^{N-1}$
  - prob that *any* node has a success = $Np(1-p)^{N-1}$
  - max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
  - for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

Max efficiency = 1/e = .37

- *At best:* channel used for useful transmissions 37% of time!

CSMA (carrier sense multiple access)

Simple CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy: defer transmission

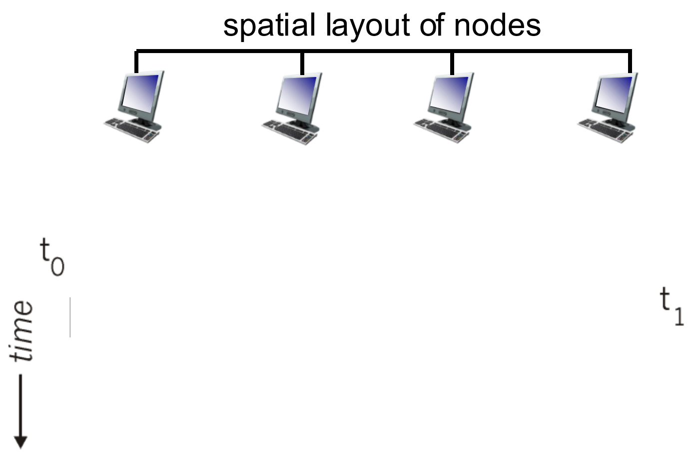
- Human analogy: <u>don't interrupt others!</u>

CSMA/CD: CSMA with *collision detection*

- Collisions *detected* within short time
- Colliding transmissions aborted, reducing channel wastage
- Collision detection easy in wired, difficult with wireless

- Human analogy: <u>the polite conversationalist</u>

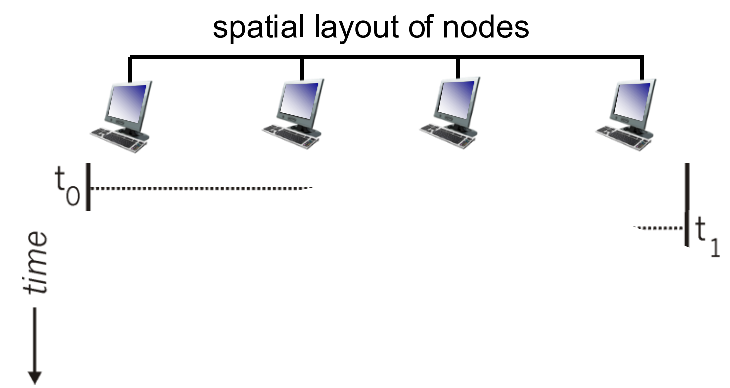
CSMA: collisions



- Collisions can *still* occur with carrier sensing:
  - Propagation delay means two nodes may not hear each other's just-started transmission
- Collision: entire packet transmission time wasted
  - Distance & propagation delay play role in in determining collision probability

CSMA/CD:

- CSMA/CD reduces the amount of time wasted in collisions
  - Transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. Ethernet receives datagram from network layer, creates frame
2. If Ethernet senses channel:
   If idle: start frame transmission.
   If busy: wait until channel idle, then transmit
3. If entire frame transmitted without collision - done!
4. If another transmission detected while sending: abort, send jam signal
5. After aborting, enter *binary (exponential) backoff:*
   - After *m*th collision, chooses *K* at random from $\{0,1,2, \ldots, 2^m-1\}$. Ethernet waits $K \cdot 512$ bit times, returns to Step 2
   - More collisions: longer backoff interval (busy channel!)

"Taking turns" MAC protocols

Channel partitioning MAC protocols:

- Share channel *efficiently* and *fairly* at high load
- Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- Efficient at low load: single node can fully utilize channel
- High load: collision overhead

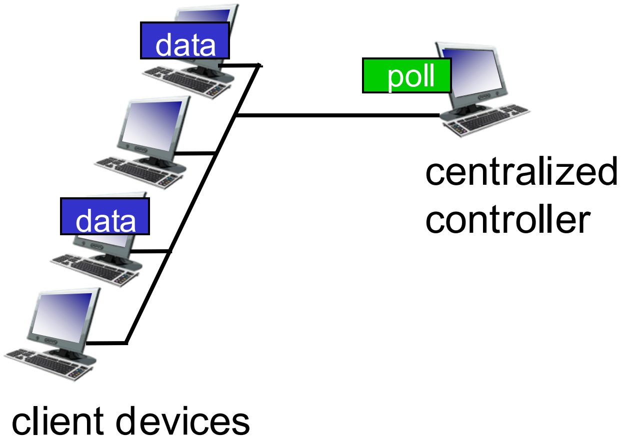
"Taking turns" protocols

- Look for best of both worlds!

"Taking turns" MAC protocols

Polling:

- Centralized controller "invites" other nodes to transmit in turn
- Typically used with "dumb" devices
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)
- Bluetooth uses polling

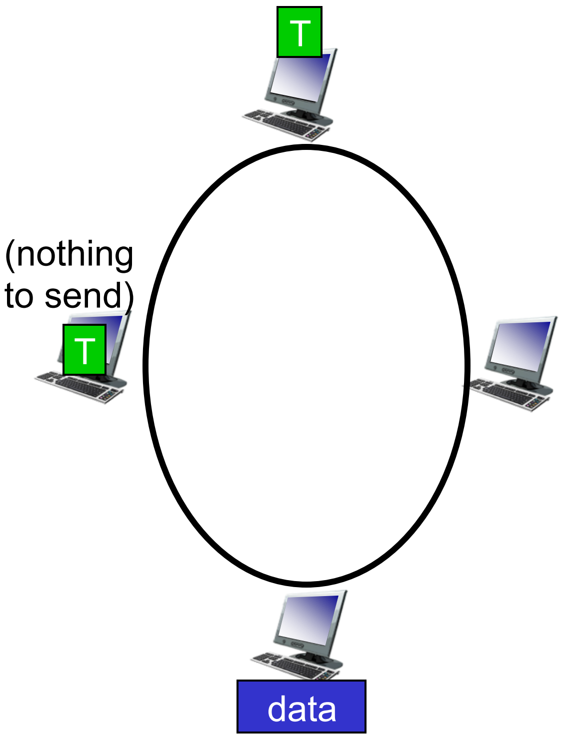


Adapted from Kurose and Ross: Computer Networking: A Top-Down Approach

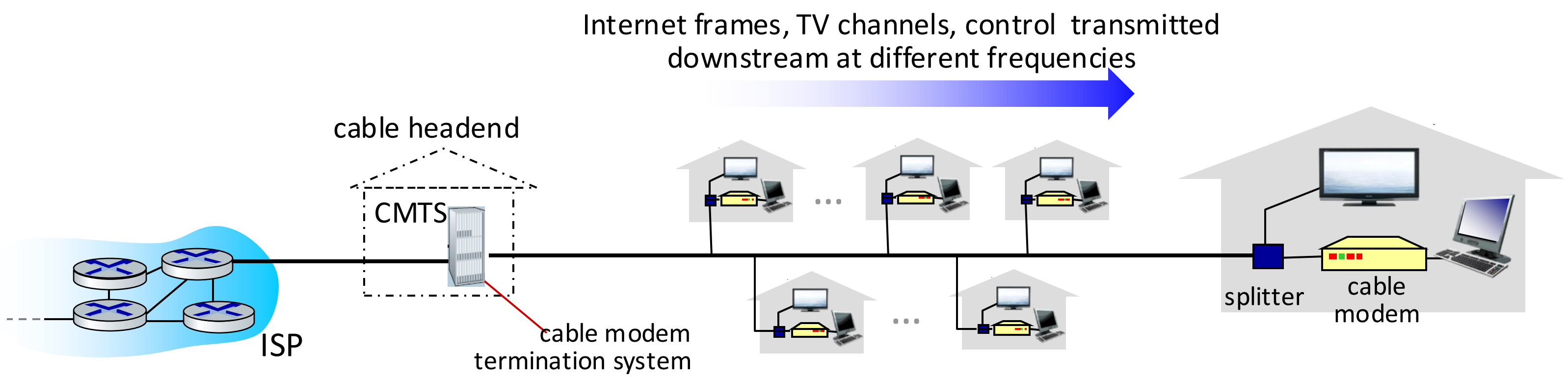
“Taking turns” MAC protocols

Token passing:

- Control *token* message explicitly passed from one node to next, sequentially
  - Transmit while holding token
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)

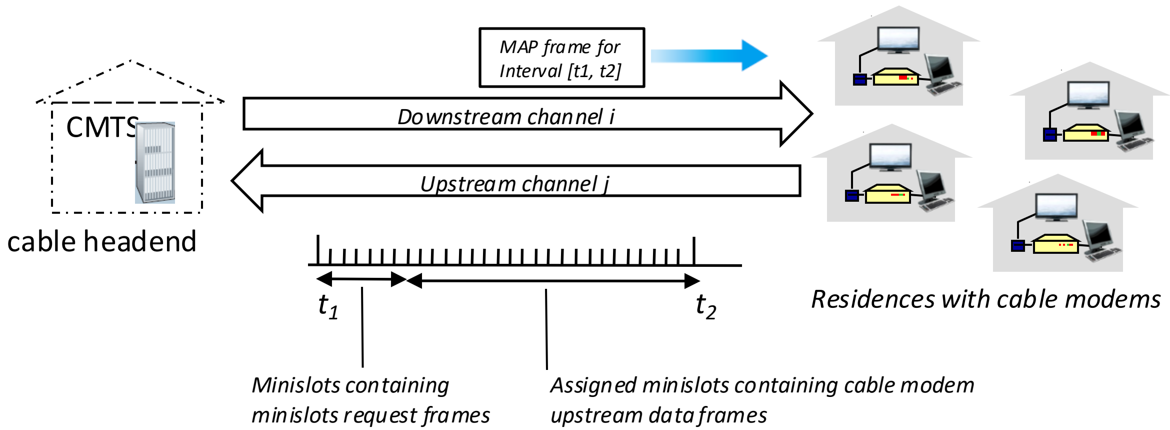


Cable access network: FDM, TDM *and* random access!



- Multiple downstream (broadcast) FDM channels: up to 1.6 Gbps/channel
  - Single CMTS transmits into channels
- Multiple upstream channels (up to 1 Gbps/channel)
  - Multiple access: all users contend (random access) for certain upstream channel time slots; others assigned TDM

Cable access network:



DOCSIS: data over cable service interface specification

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - Downstream MAP frame: assigns upstream slots
  - Request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Summary of MAC protocols

- Channel partitioning, by time, frequency or code
  - Time Division, Frequency Division
- Random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- Taking turns
  - Polling from central site, token passing
  - Bluetooth, FDDI, token ring