Module M3

CPSC 317

October 3, 2022

Learning Goals

A. Link Layer

- Explain the purpose of the link layer, and the five type of services the link layer provides.
- Understand the basic types of mediums (point-to-point, broadcast) and what is meant by "access control"
- Explain why the link layer may use "error correction".
- Know three techniques for error detection: parity (1D and 2D), checksumming, CRC.
- Understand what errors can be detected and corrected in 2D parity checking.
- Advantages of CRC over parity checking and checksumming.
- Know the general structure of link layer "frames".
- Know the basic differences between a switch and a router.
- Understand link layer addressing, MAC addresses.

B. ARP

- What is the purpose of ARP
- Enumerate the steps to resolve an IP address on a LAN with ARP
- How is ARP implemented at Layer two.
- Describe the steps to send a datagram to one LAN to another LAN, assuming you need to resolve the IP addresses.
- Know how to perform a layer 2 broadcast.

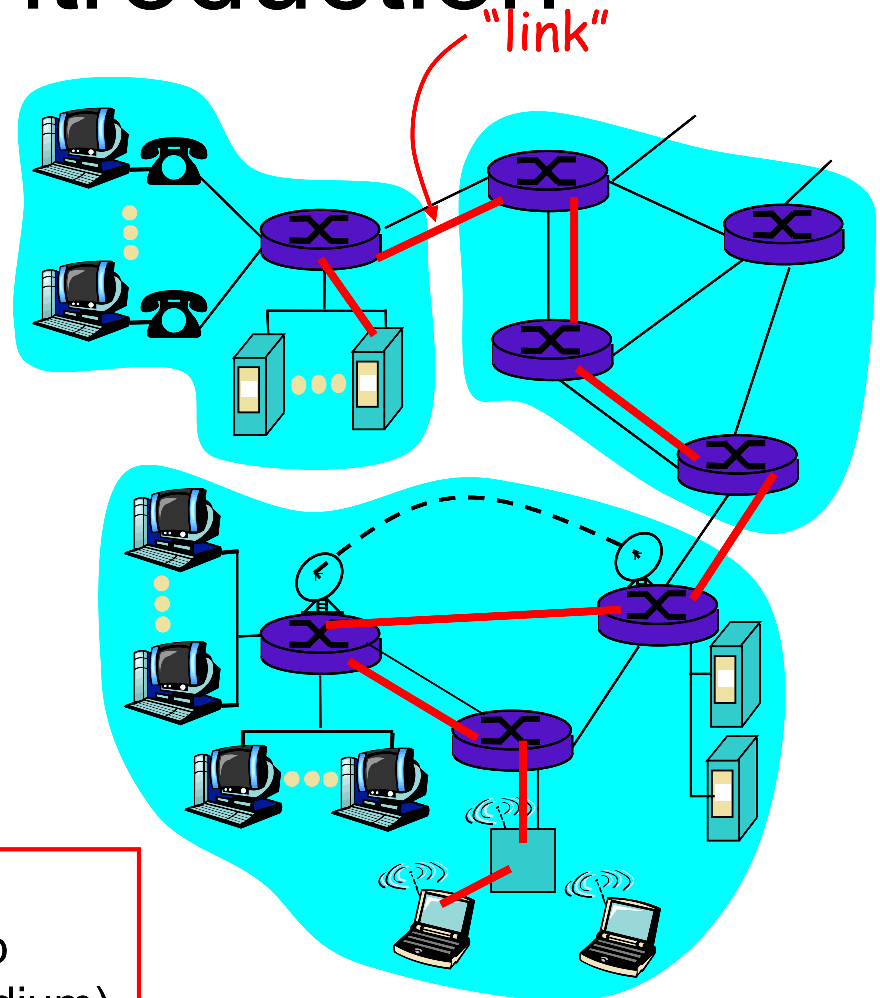
C. DHCP

- Describe the purpose of DHCP
- Trace the four steps to get an IP address for a particular interface on a LAN
- Know the configuration of DHCP on a LAN and the configuration information issued to support IP.
- Know the difference between a static and dynamic address, and the notion of leasing an IP.
- Understand the reason broadcast is needed.

Link Layer: Introduction

Some terminology:

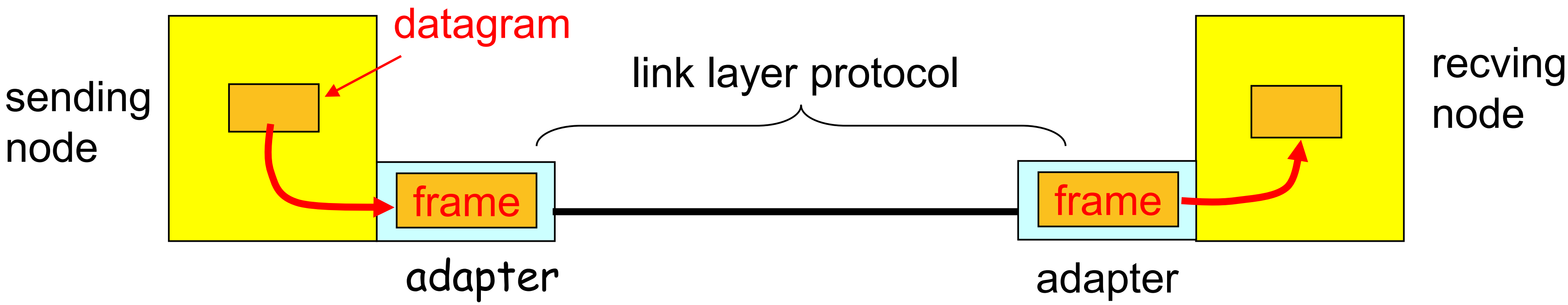
- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram



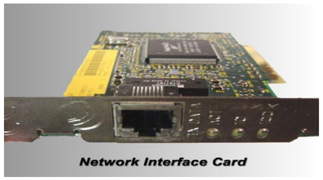
data-link layer has responsibility of transferring datagram from one node to adjacent node over a link (physical medium)

PHYSICAL PIECES

Adaptors Communicating



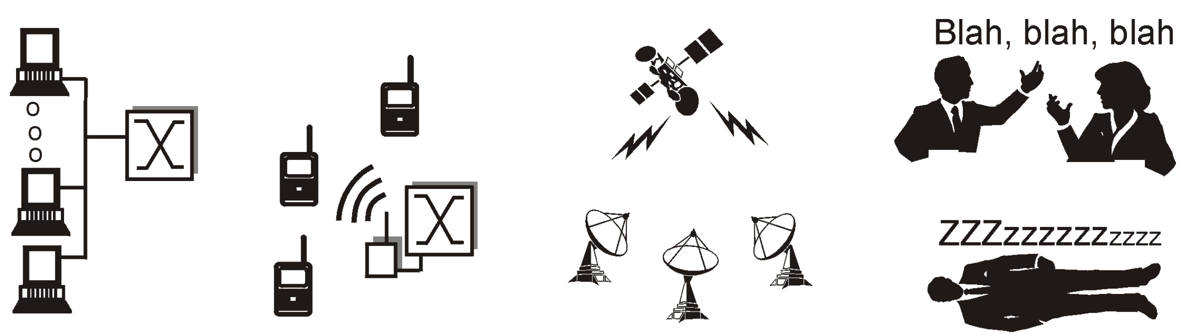
- link layer implemented in "adaptor" (aka NIC)
  - Ethernet card, PCMCI card, 802.11 card
- sending side
- receiving side
- adapter is semi-autonomous
- link & physical layers



Multiple Access Links and Protocols

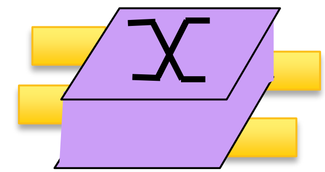
Two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between two routers
- broadcast (shared wire or medium)
  - traditional Ethernet
  - upstream HFC
  - 802.11 wireless LAN



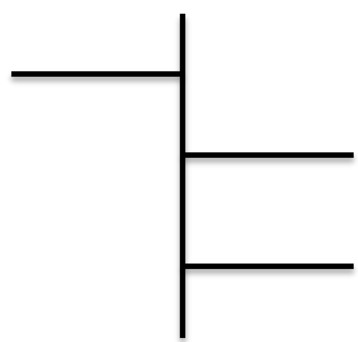
shared wire (e.g. Ethernet) | shared wireless (e.g. Wavelan) | satellite | cocktail party

Switches (Layer 2, device)

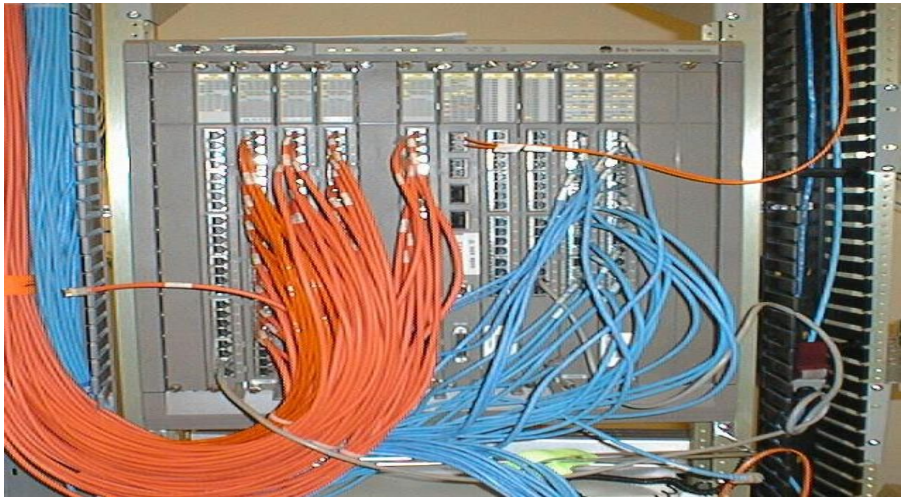


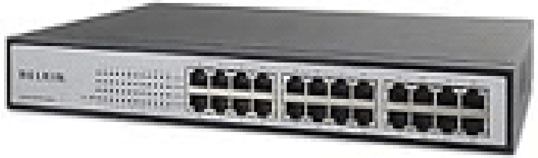
Box or Square

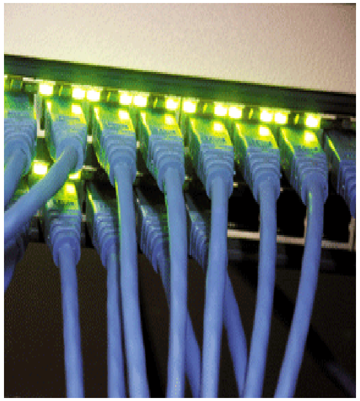
Single Broadcast Domain (link-layer or physical layer)



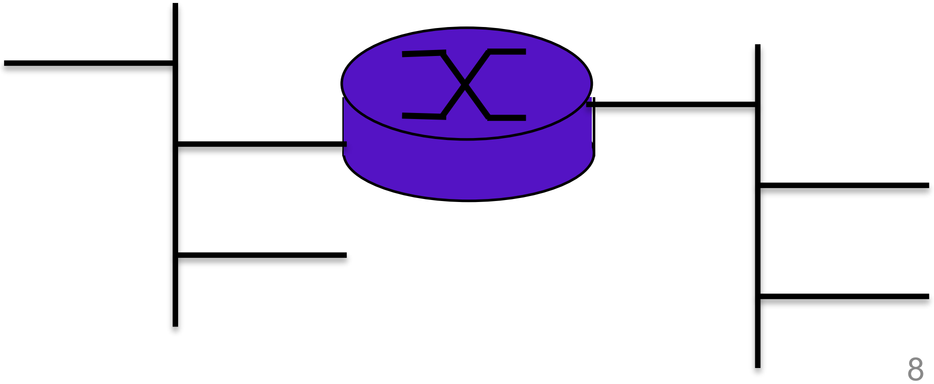
Network Router





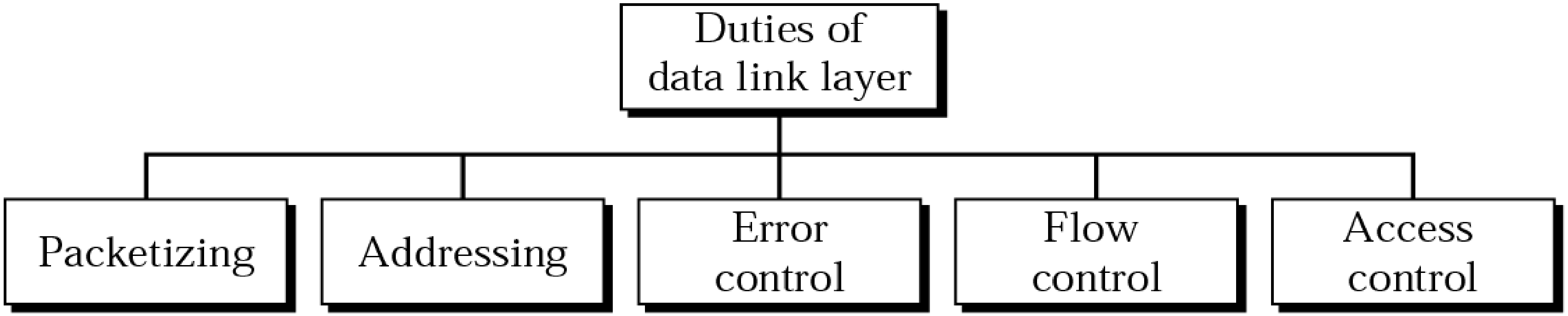


Separate Broadcast Domains



LINK LEVEL SERVICES

Link Layer Services



- Framing (or packetizing):
  - encapsulate datagram into frame, adding header, trailer
  - channel access, if shared medium
- Link access (media access control)
  - MAC protocol
  - Half duplex, both ends of the link can transmit, but not at same time
  - Full duplex, data can be transmitted in both directions at the same time.
  - Point-to-point links versus broadcast media
- *Addressing:* (media access control addresses)
  - "MAC" addresses used in frame headers to identify source, destination
    - **different from IP address!**
  - Broadcast medium (to whom, from whom)

Link Layer Services (more)

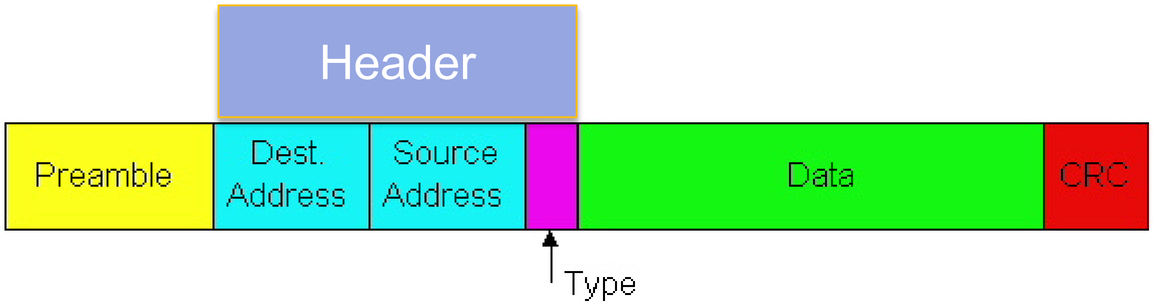
- *Flow Control:*
  - pacing between adjacent sending and receiving nodes
- More reliable delivery between adjacent nodes
  - seldom used on low bit error link (fiber, some twisted pair)
  - wireless links: high error rates
    - Q: why both link-level and end-to-end reliability?
- Error Detection:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - receiver signals sender for retransmission; drops frame
- Error Correction:
  - receiver detects *and corrects* bit error(s) without resorting to retransmission (Error correction/detection called ECD).

ADDRESS RESOLUTION PROTOCOL (ARP)

PURPOSE: The Address Resolution Protocol (ARP) is a is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. (Wikipedia)

Frame on the wire (802.xxx)

Ethernet



Preamble	8 bytes	Synchronization of the receivers Bit sequence that initiates the frame
Destination address (MAC)	6 bytes	Hardware address of the destination network adapter
Source address (MAC)	6 bytes	Hardware address of the source network adapter
Type	2 bytes	Ethernet II: labeling of layer 3 protocols
CRC	4 bytes	error detection

Media Access Control Address

MAC address:

(a) 48 bits --- 6 bytes

(b) Given in hexadecimal

(c) Locally administered address

88-B2-2F-54-1A-0F

MAC Address

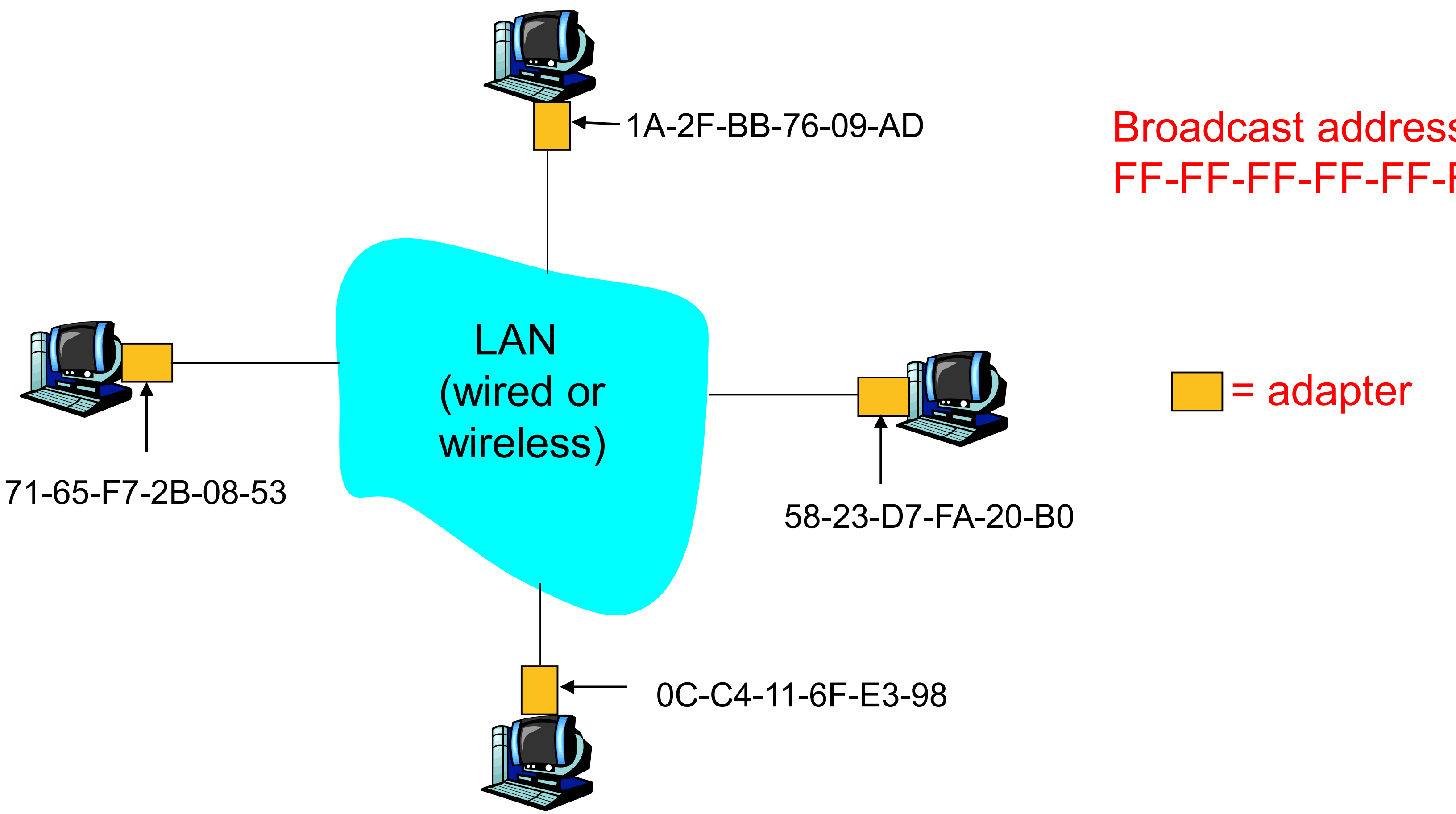
- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - (a) MAC address: like Social Insurance number never changes, only has local meaning
  - (b) IP address: like postal address
- MAC flat address ➜ portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP subnet to which node is attached

MAC versus IP

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- 48-bit MAC (or LAN or physical or Ethernet) address:
  - used to get datagram from one interface to another physically-connected interface (same network)
  - MAC address (for most LANs) burned in the adapter ROM

LAN Addresses and ARP

Each adapter on LAN has unique LAN address



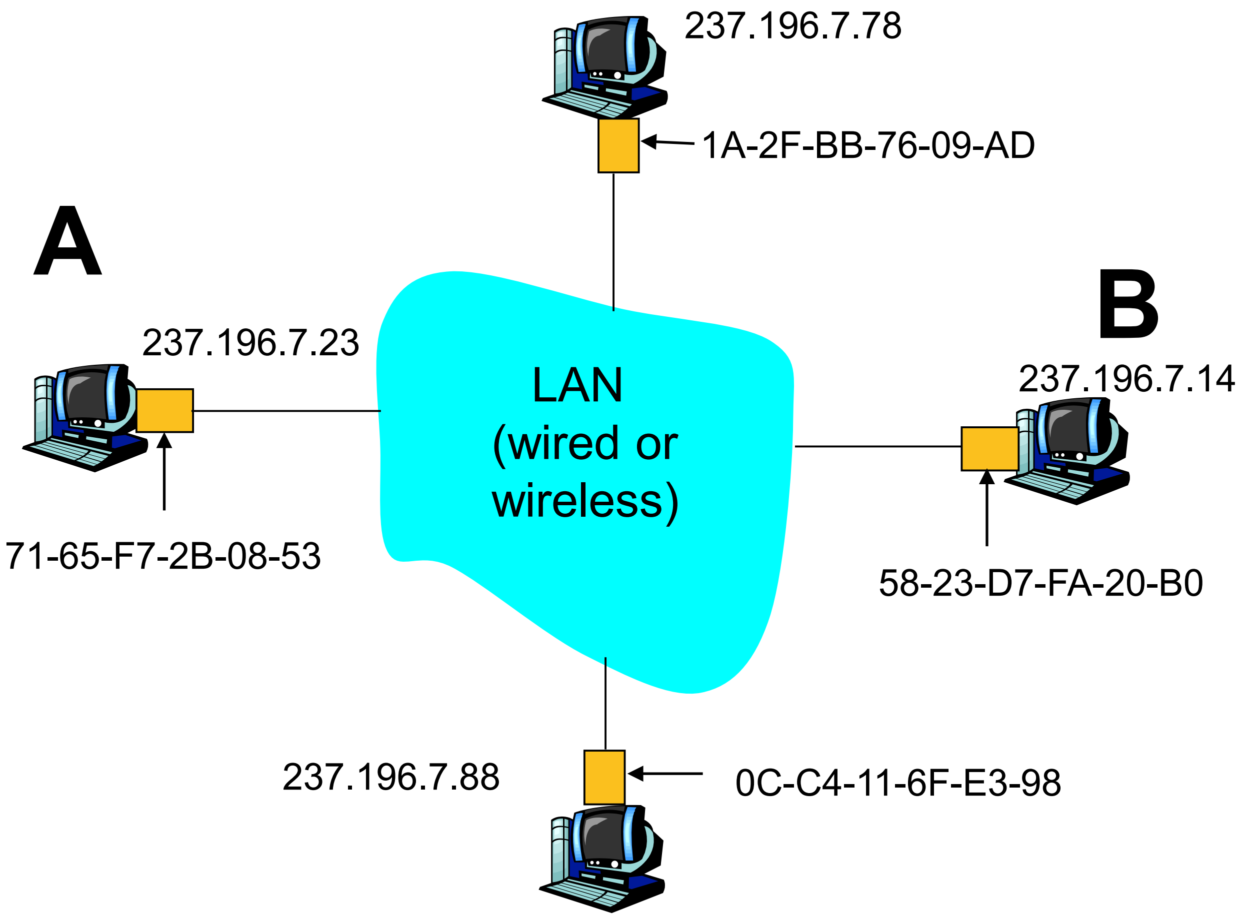
Broadcast address = FF-FF-FF-FF-FF-FF

Address Resolution Protocol

Hardware type (16 bits)	
Protocol type (16 bits)	
Length of the hardware address	Length of protocol address
Operator (16 bits)	
Hardware address of the sender	
IP address of the sender	
Hardware address of the receiver	
IP address of the receiver	

What do we do with it?

How to determine MAC address of B knowing B's IP address?



Broadcast address = FF-FF-FF-FF-FF-FF

= adapter

ARP Summary

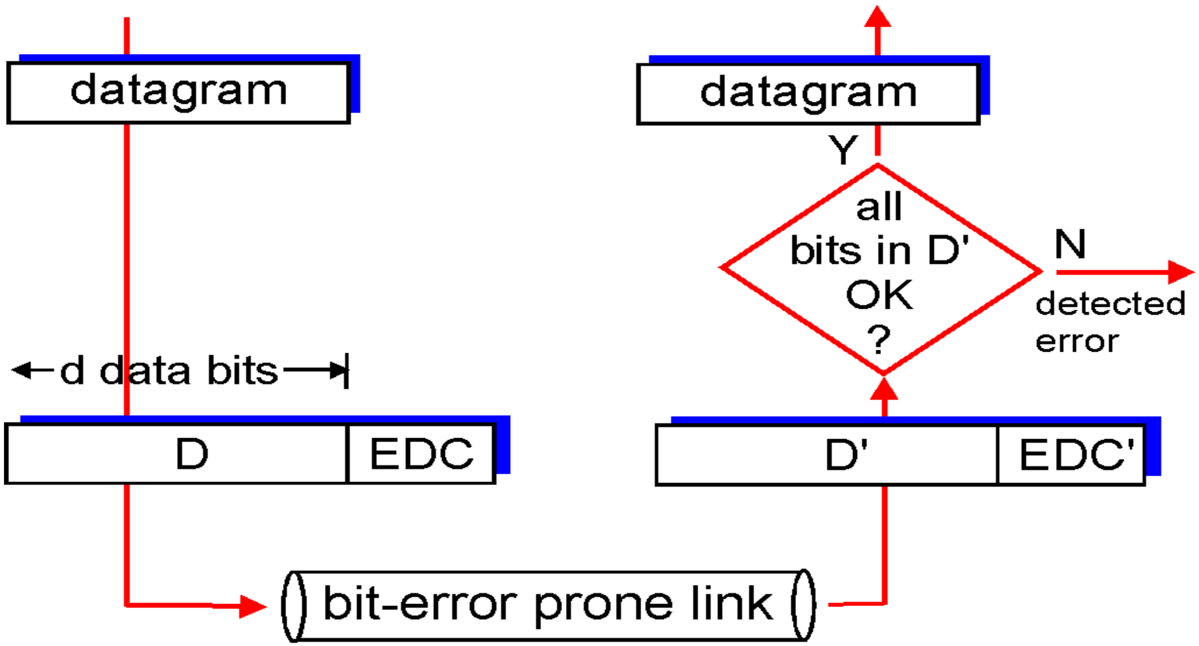
- A wants to send datagram to B, and B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Destination MAC address = FF-FF-FF-FF-FF-FF
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

ERROR DETECTION CORRECTION

Error Detection

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

- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



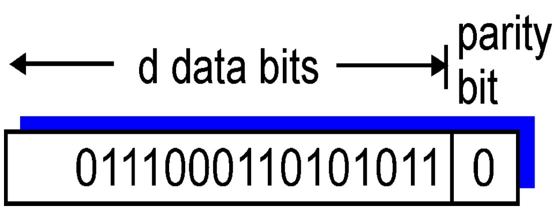
Single Bit Parity

1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 parity ?

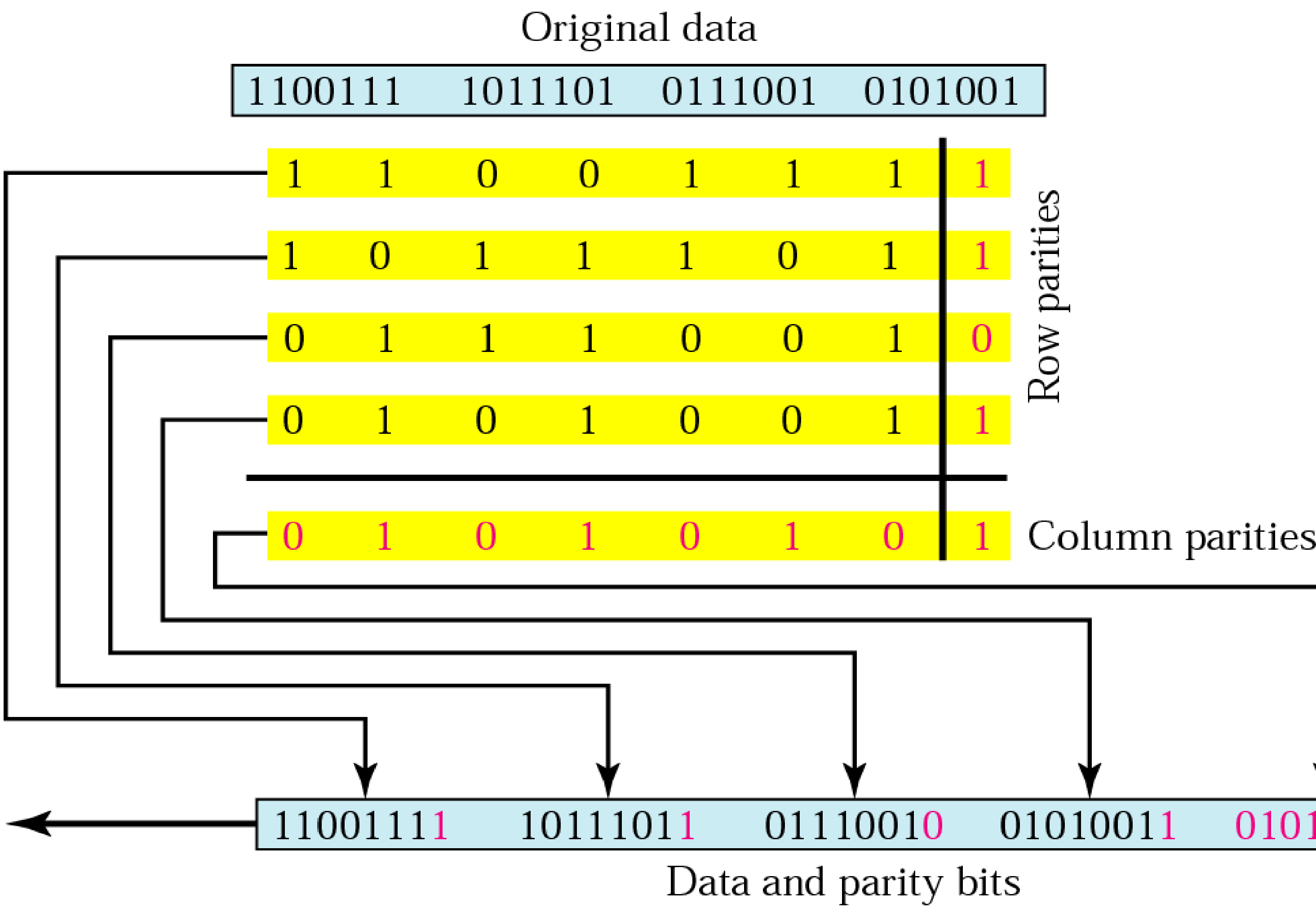
1

Even parity: add on a parity bit to make the parity even

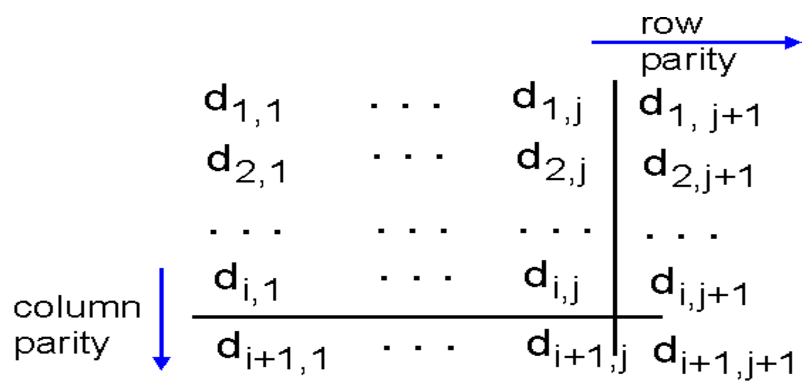
Detect single bit errors



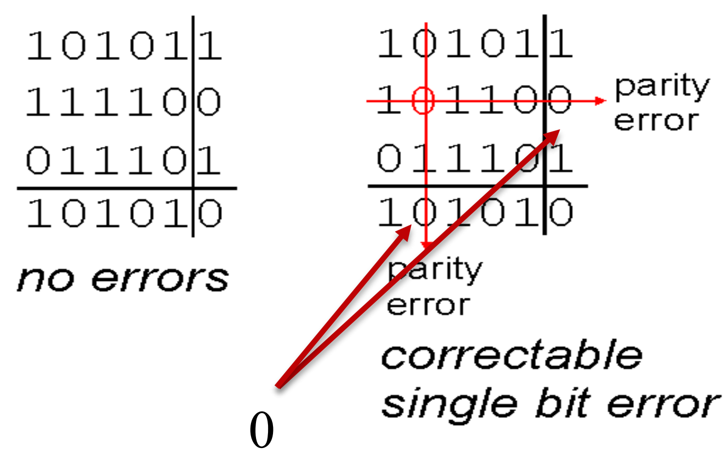
Two Dimensional Even Parity



Two Dimensional Parity Checking

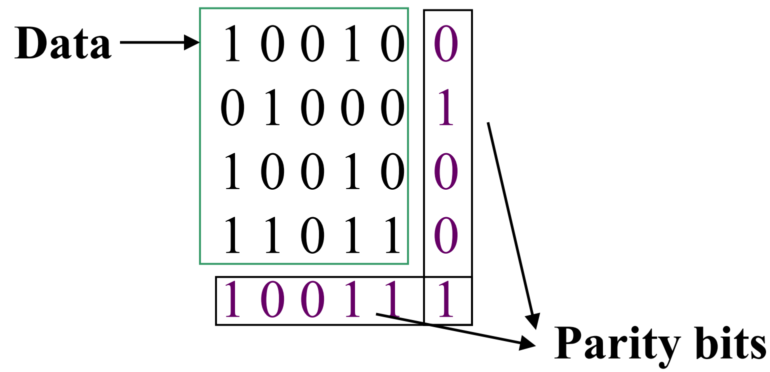


Detect *and correct* single bit errors



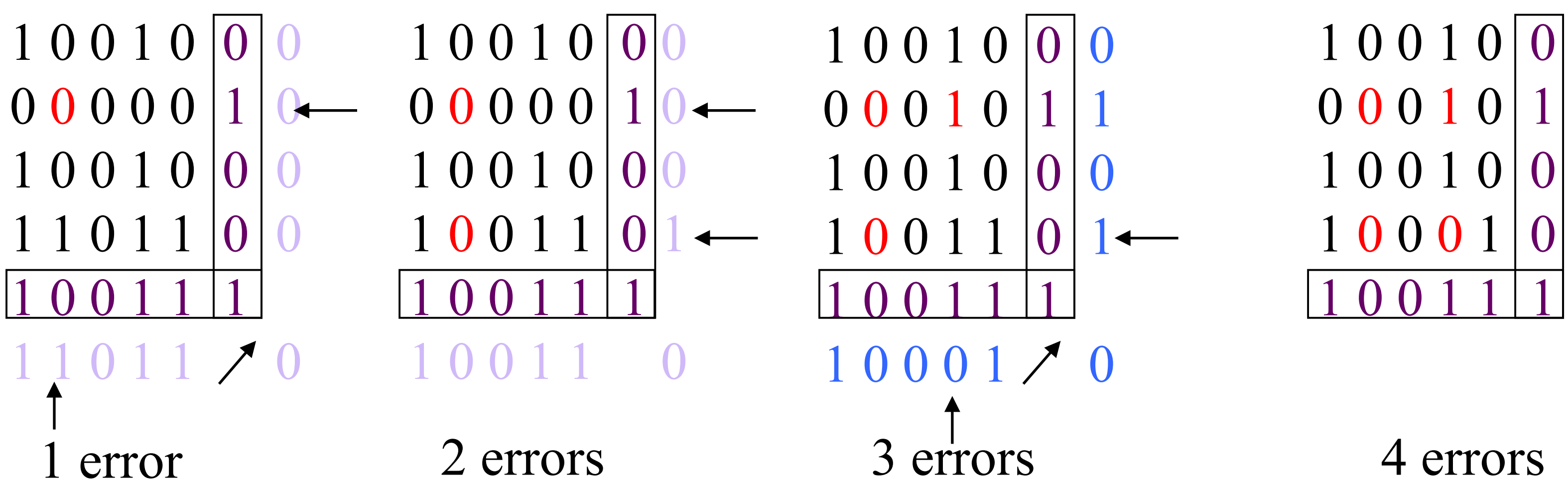
What is Detectable – Correctable?

1. Data blocks are organized into table
2. Last column: check bits for rows
3. Last row: check bits for columns
4. Can **detect** and **correct single** bit error



Can detect one, two, three errors,
But NOT all four errors.

Red bits are errors



Internet checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer *only*)

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

Receiver:

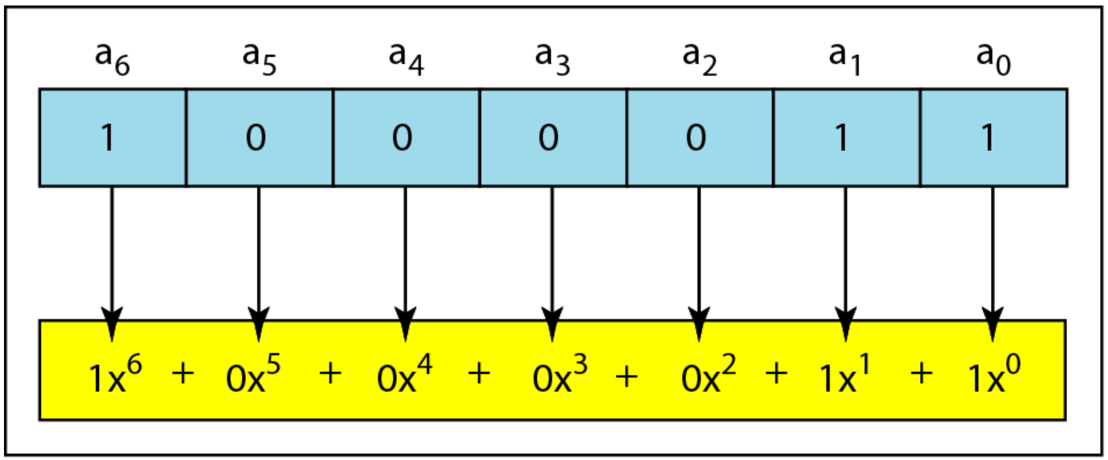
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?* More later ….

Checksum example

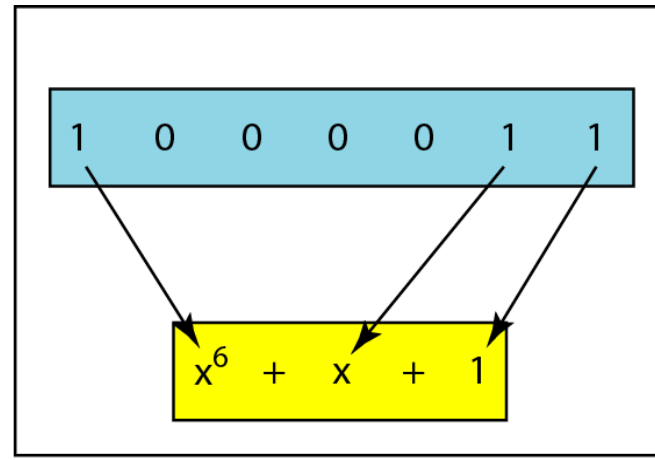
1	1	0	1	0
0	1	0	0	1
1	0	1	1	0
1	0	0	1	1

CYCLIC REDUNDANCY CHECK (CRC)

CRC -- Mod 2 Polynomials

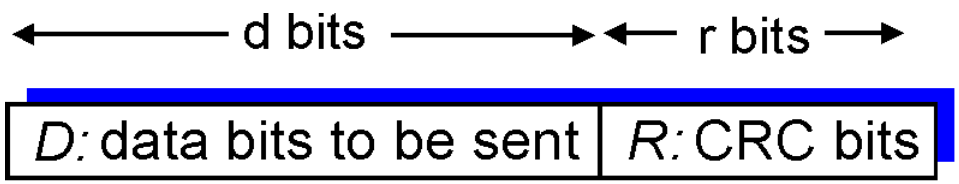


a. Binary pattern and polynomial



b. Short form

CRC



bit pattern

$D * 2^r$ XOR R

mathematical formula

Generator:

Remainder:

CRC Sending

$$1*x^3 + 0*x^2 + 0*x^1 + 1*x^0$$

CRC Sending

$$1\ 0\ 0\ 1 \,\overline{\big)\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ {\color{red}0\ 0\ 0}}$$

CRC Sending

1 1 1 1 1 1 1 0 0 0

1 0 0 1

Cyclic Redundancy Check

- Choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 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, and burst errors greater that r+1 with probability 1-0.5^r
- Not good for authentication? Why?
- widely used in practice (ATM – Asynchronous Transfer Mode, HDCL – High Level Data Link Control)
- Easy to do in hardware

CRC Example

Want:

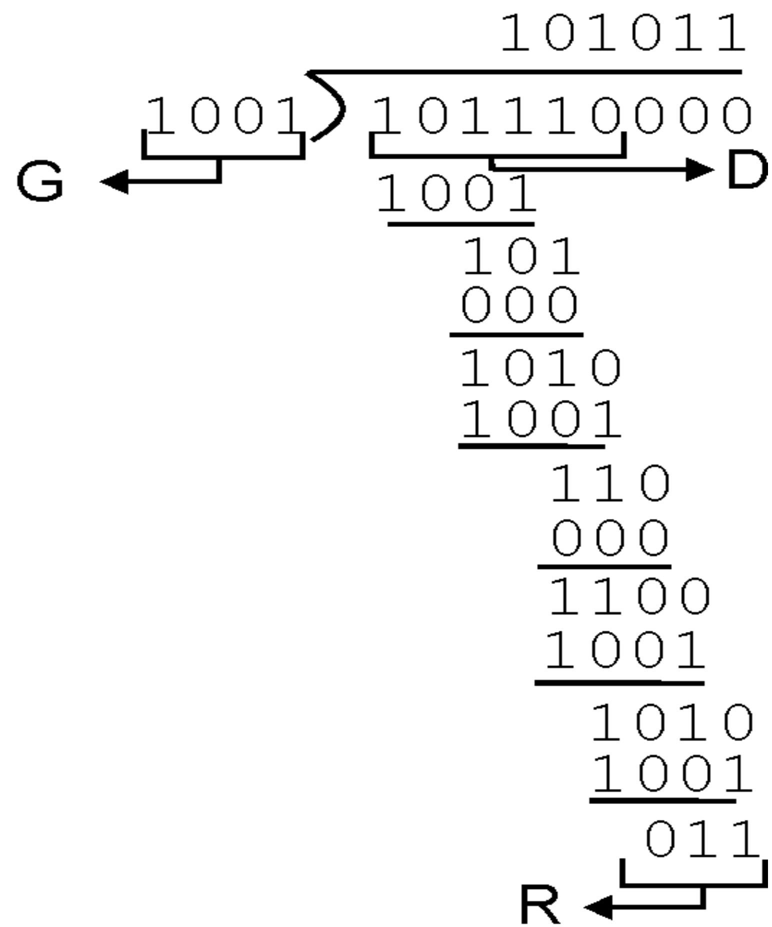
$D \cdot 2^r \text{ XOR } R = nG$

equivalently:

$D \cdot 2^r = nG \text{ XOR } R$

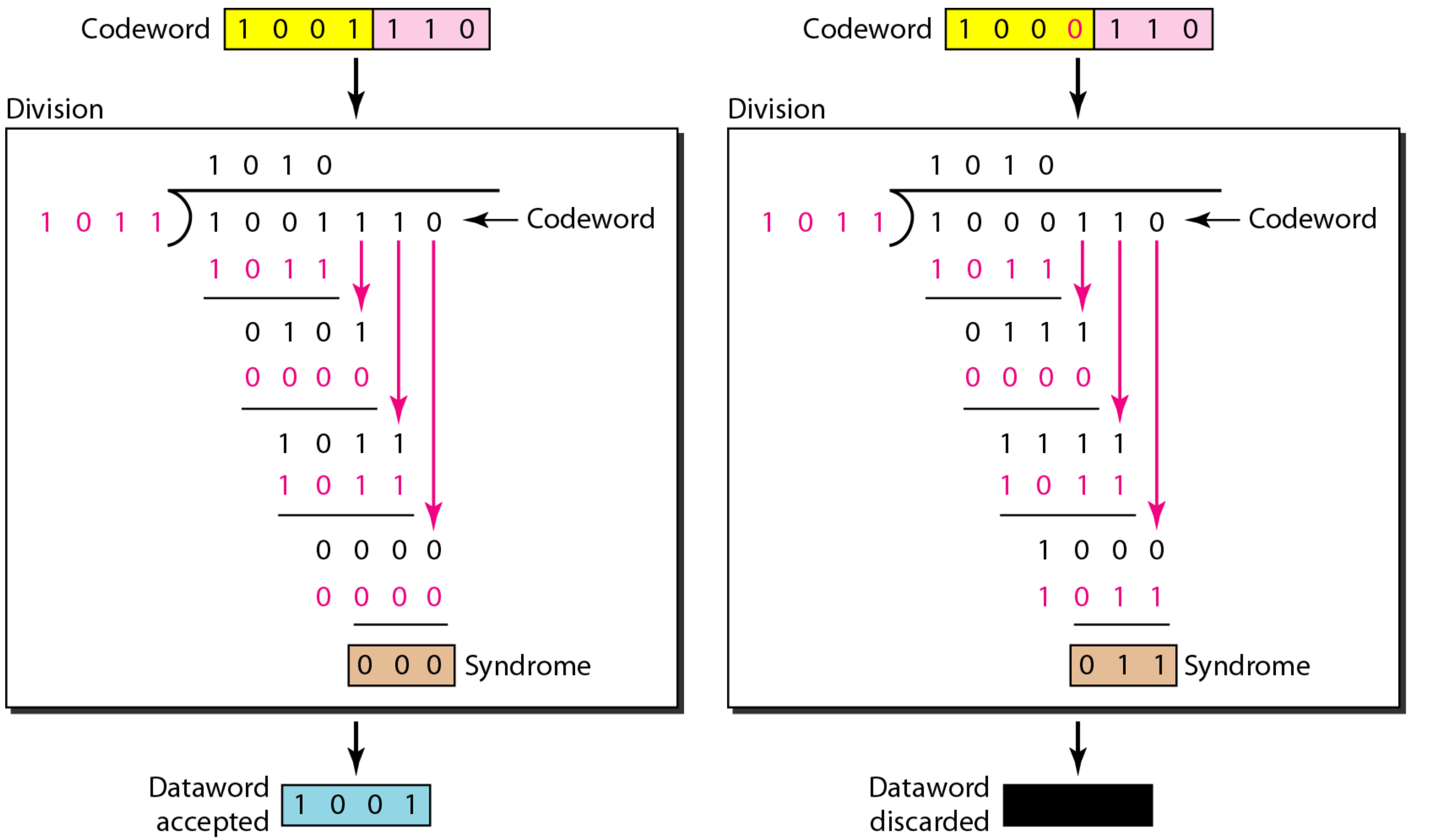
equivalently:

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



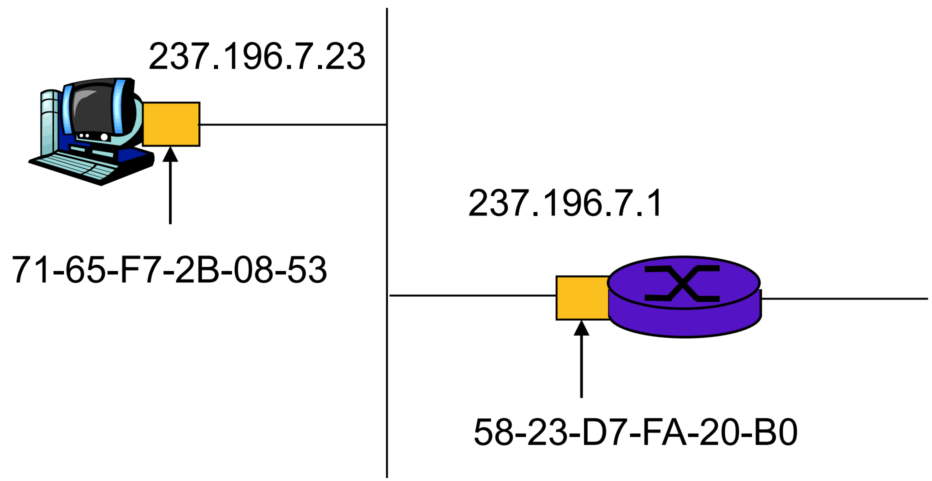
Sender Side -- detection

The seven bits received by the sender. Two examples, correct and incorrect.



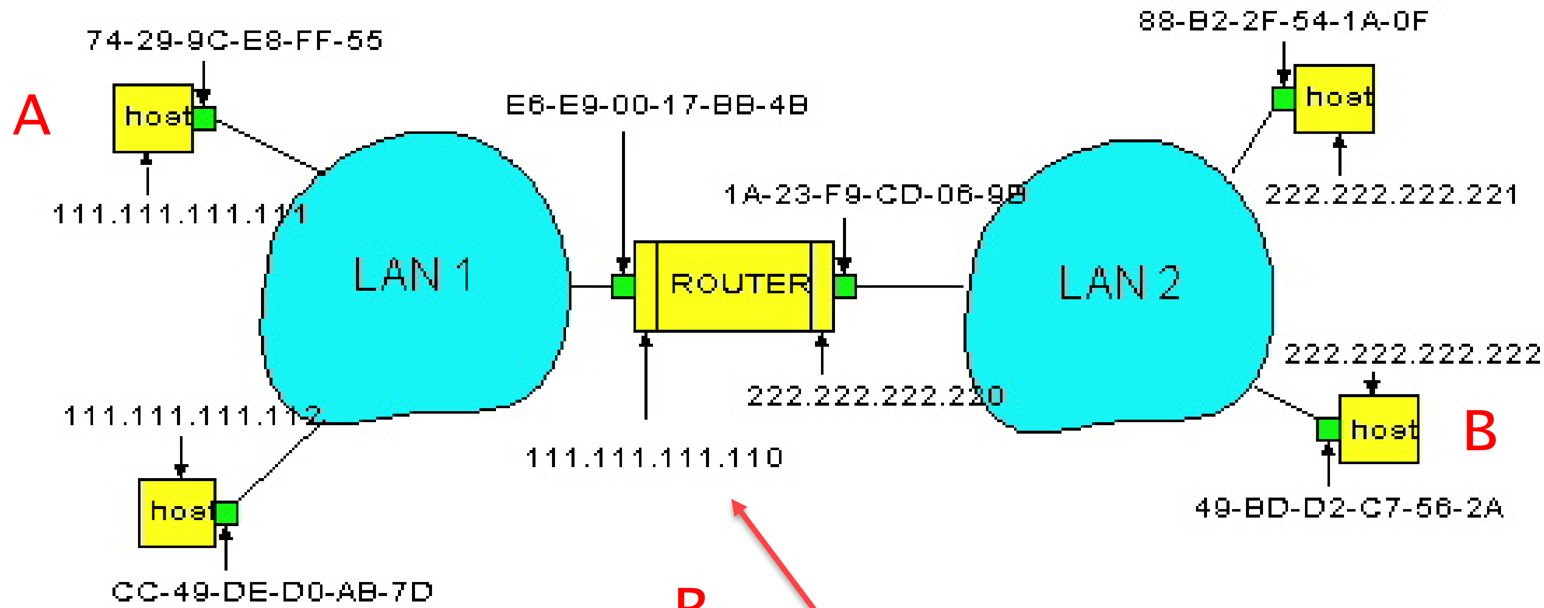
LAN to LAN

Default Router



Adapter (network interface card) is configured with the IP address of the "default router". The adapter learns the MAC address of the router using ARP. This is the first hop for all traffic leaving the local network.

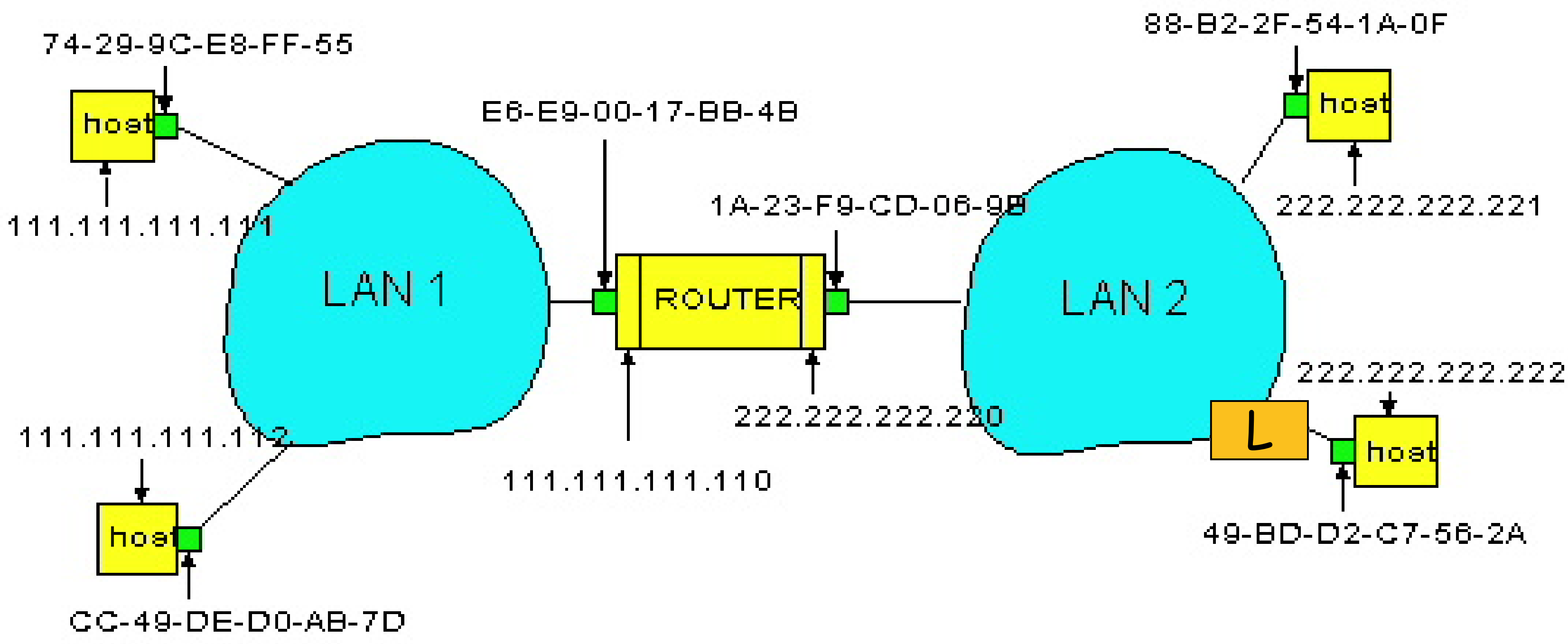
Example: sending msg A→B (Kurose, Ross Example)



Let's get a better router address 111.111.111.97

Network prefix is missing --- it should be there --- we will just need to make a reasonable guess at them.

What if B replies to A



Fun Facts about ARP

- ARP is stateless, always read a response even if it didn't make a request
- ARP is not authenticated, anyone can ARP
- Can you spoof ARP?
- ARP is in a single "broadcast domain"
- Reverse-ARP used to be used to get an IP address (obsolete) --- DHCP

DYNAMIC HOST CONFIGURATION PROTOCOL

DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

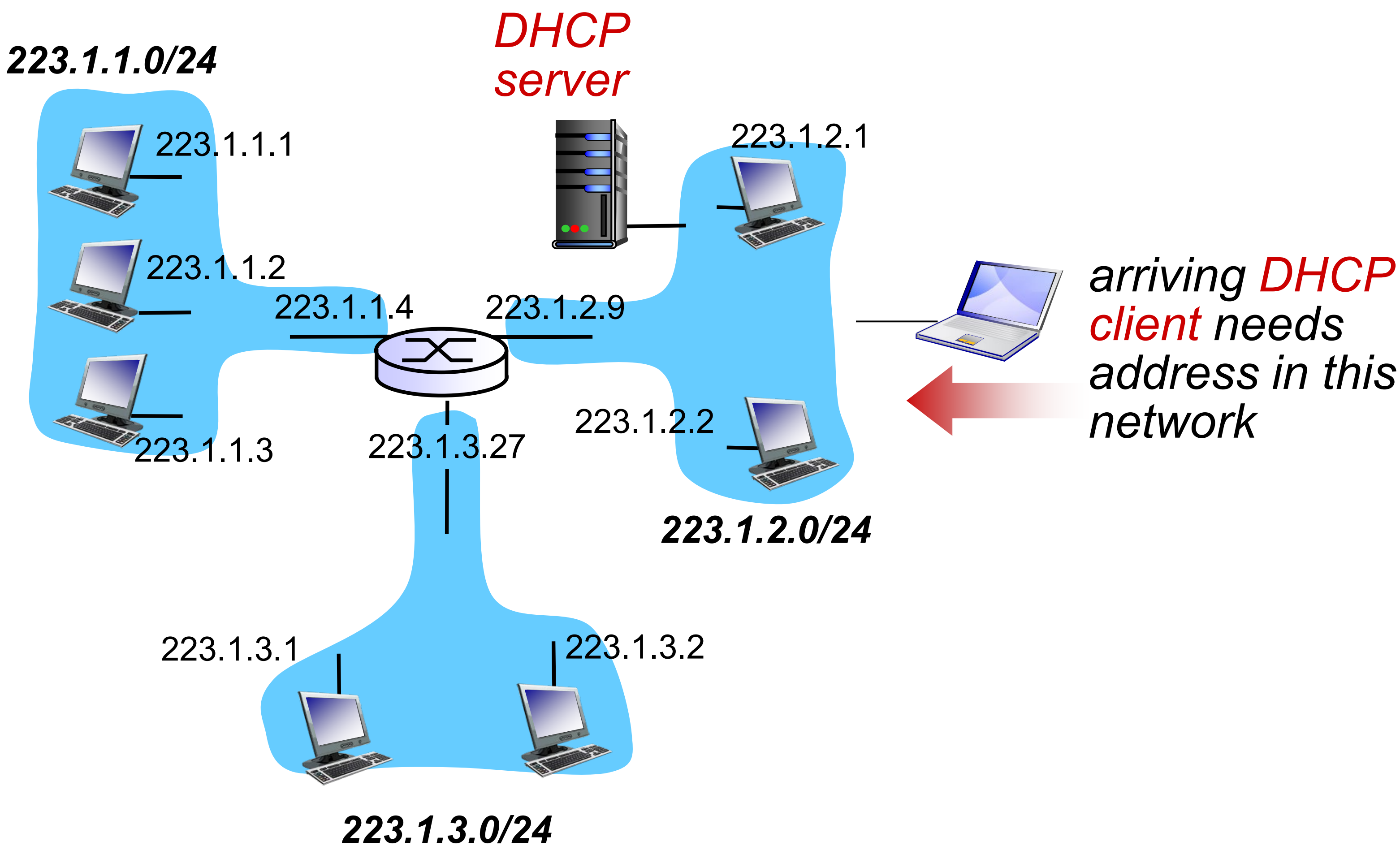
DHCP overview:

- DHCP is built on a client-server model
- DHCP supports several modes of IP-Address allocation

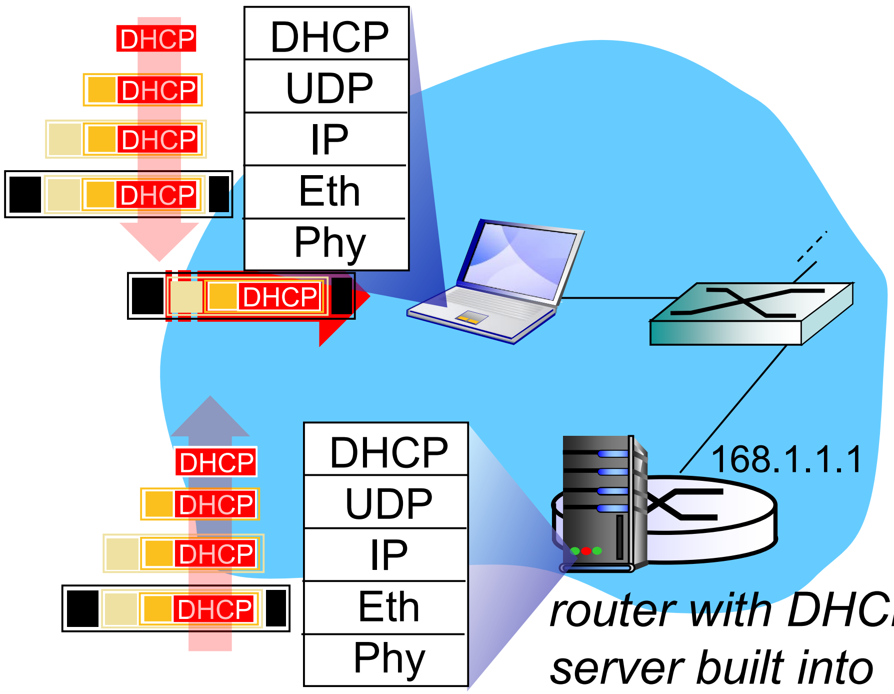
DHCP basic messages:

- host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



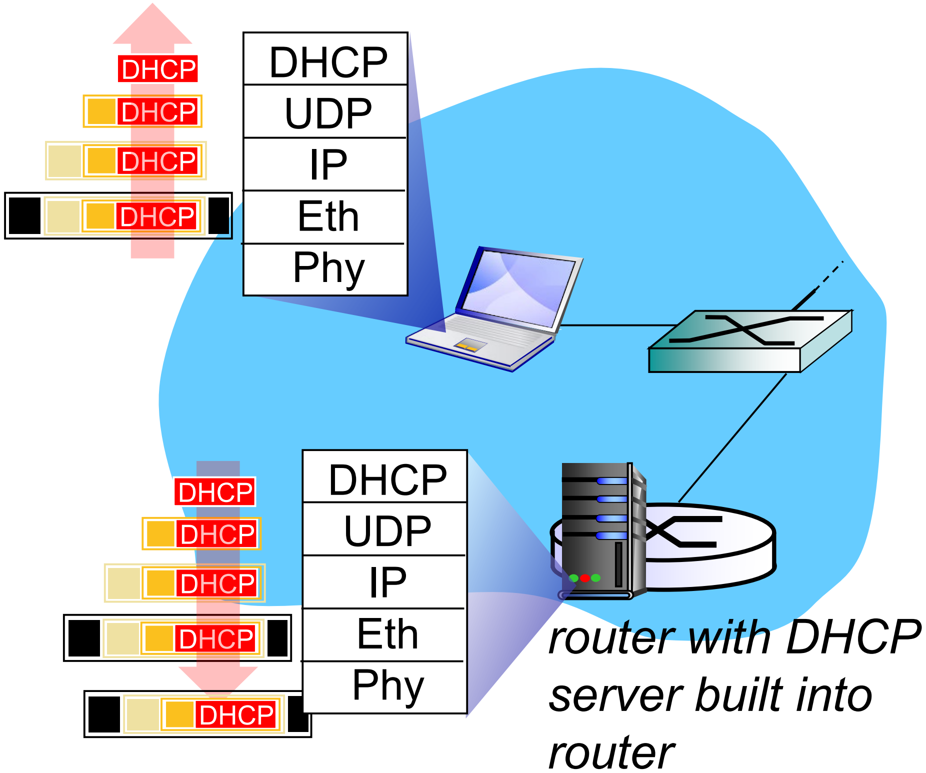
DHCP packet to server



router with DHCP server built into router

- connecting laptop needs its IP address, address of first-hop router, network prefix, address of DNS server
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet de-muxed to IP de-muxed, UDP de-muxed to DHCP

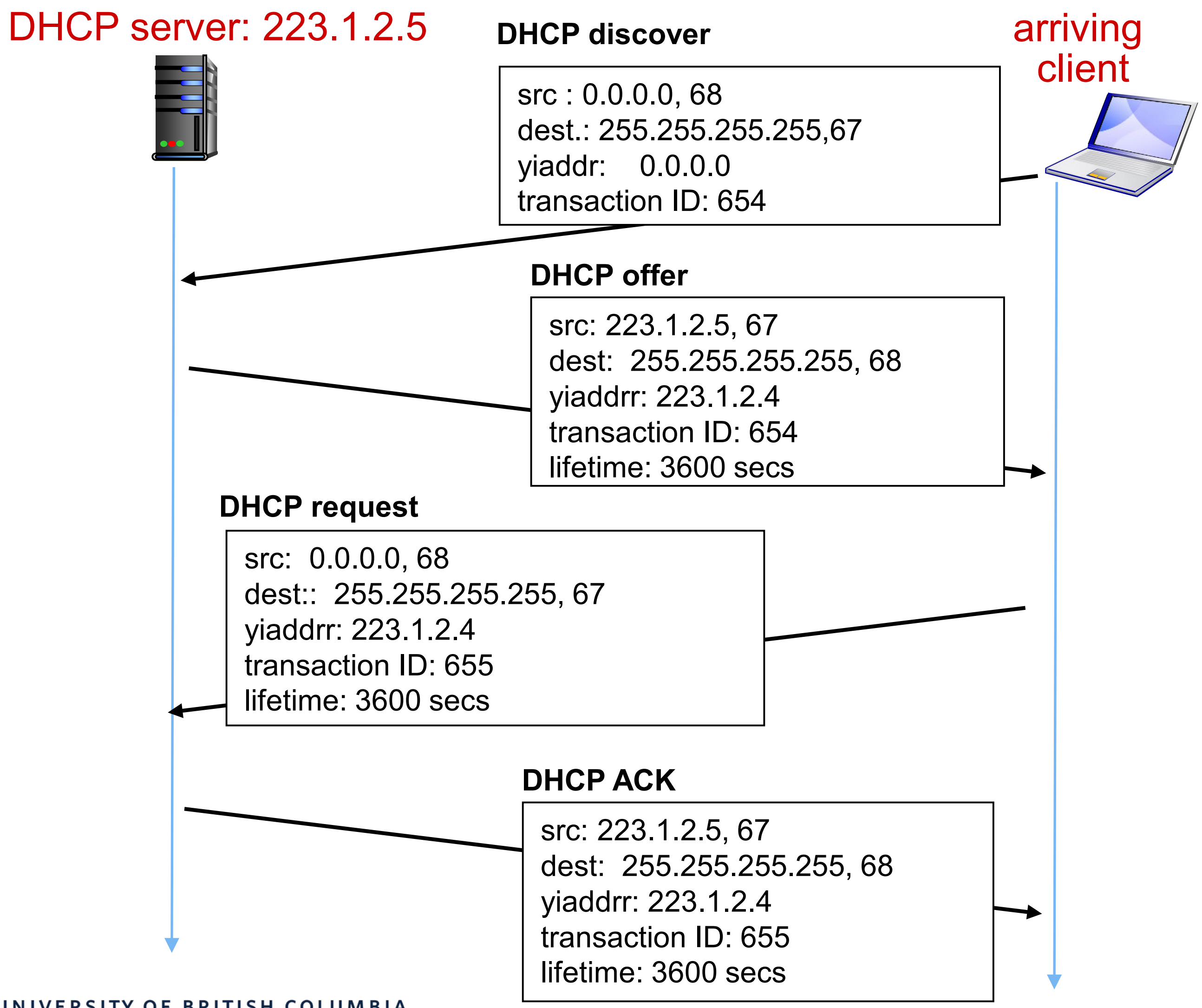
DHCP message from server



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

DHCP client-server scenario



DHCP: more than IP addresses

DHCP can return more than just allocated IP address on network:

- address of first-hop router for client
- name and IP address of DNS server
- network mask (indicating network versus host portion of address, network prefix)

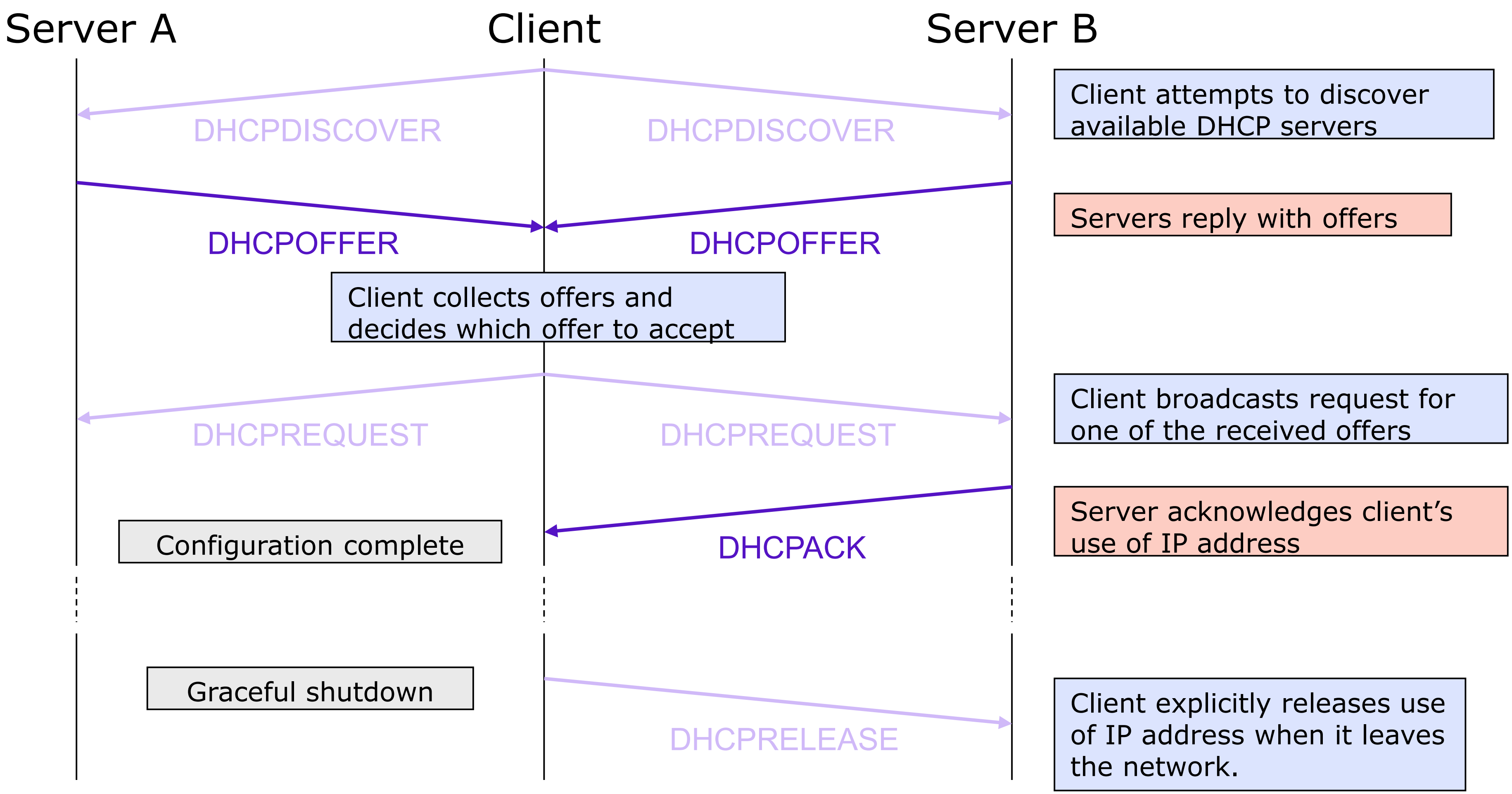
Why

- Why use leases?

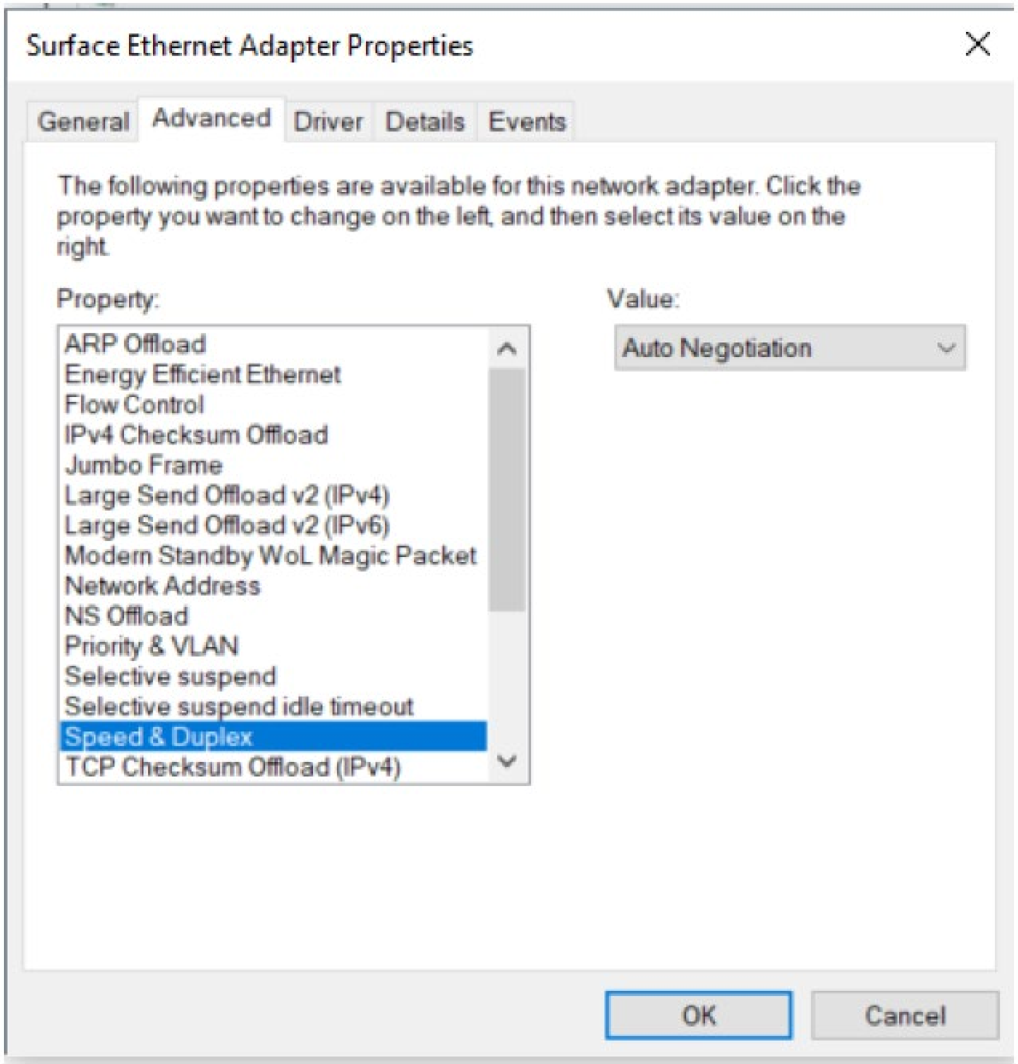
- What if two machines request an IP at the same time?

- What if there are more than one servers?

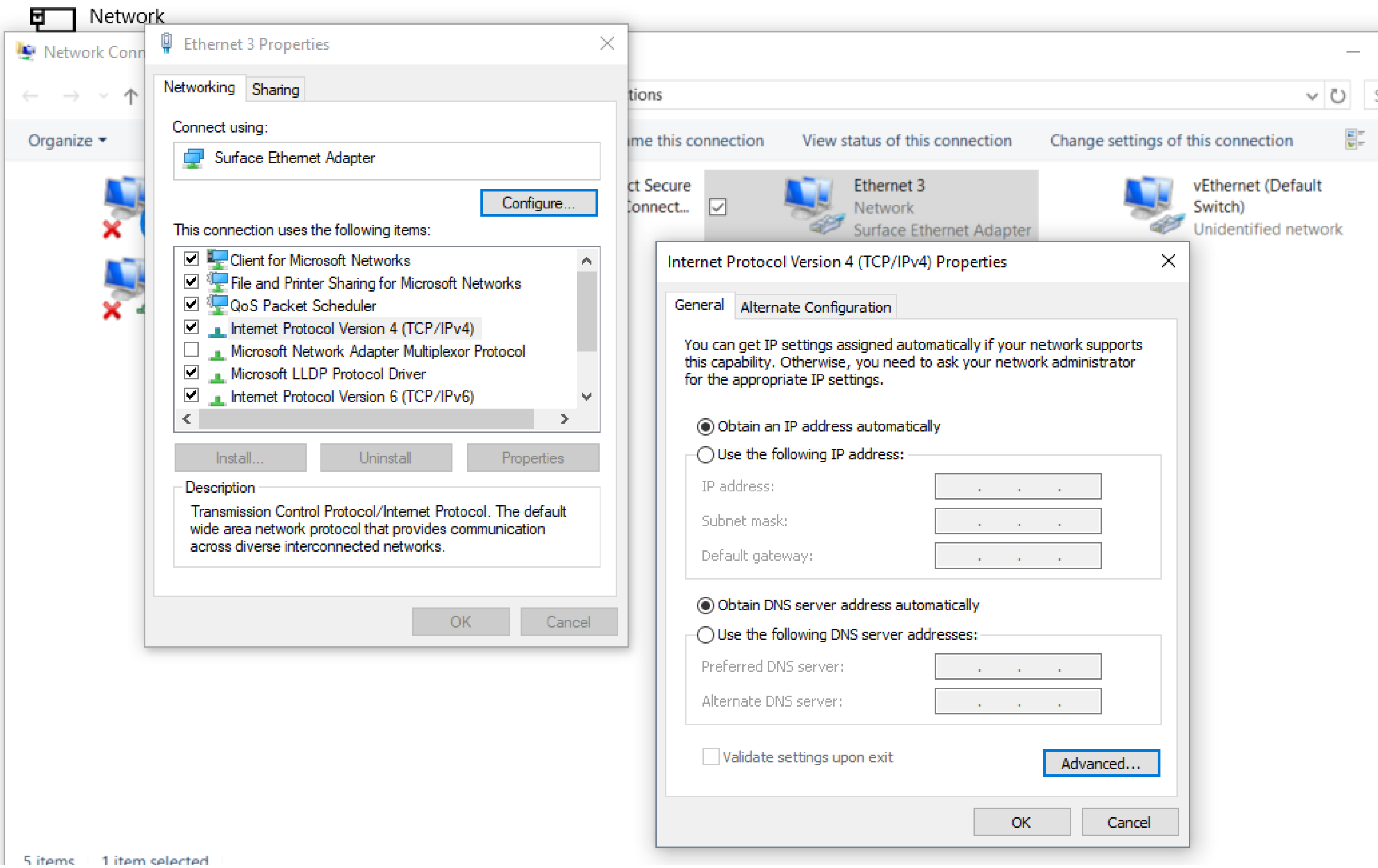
DHCP Conversation (two servers)



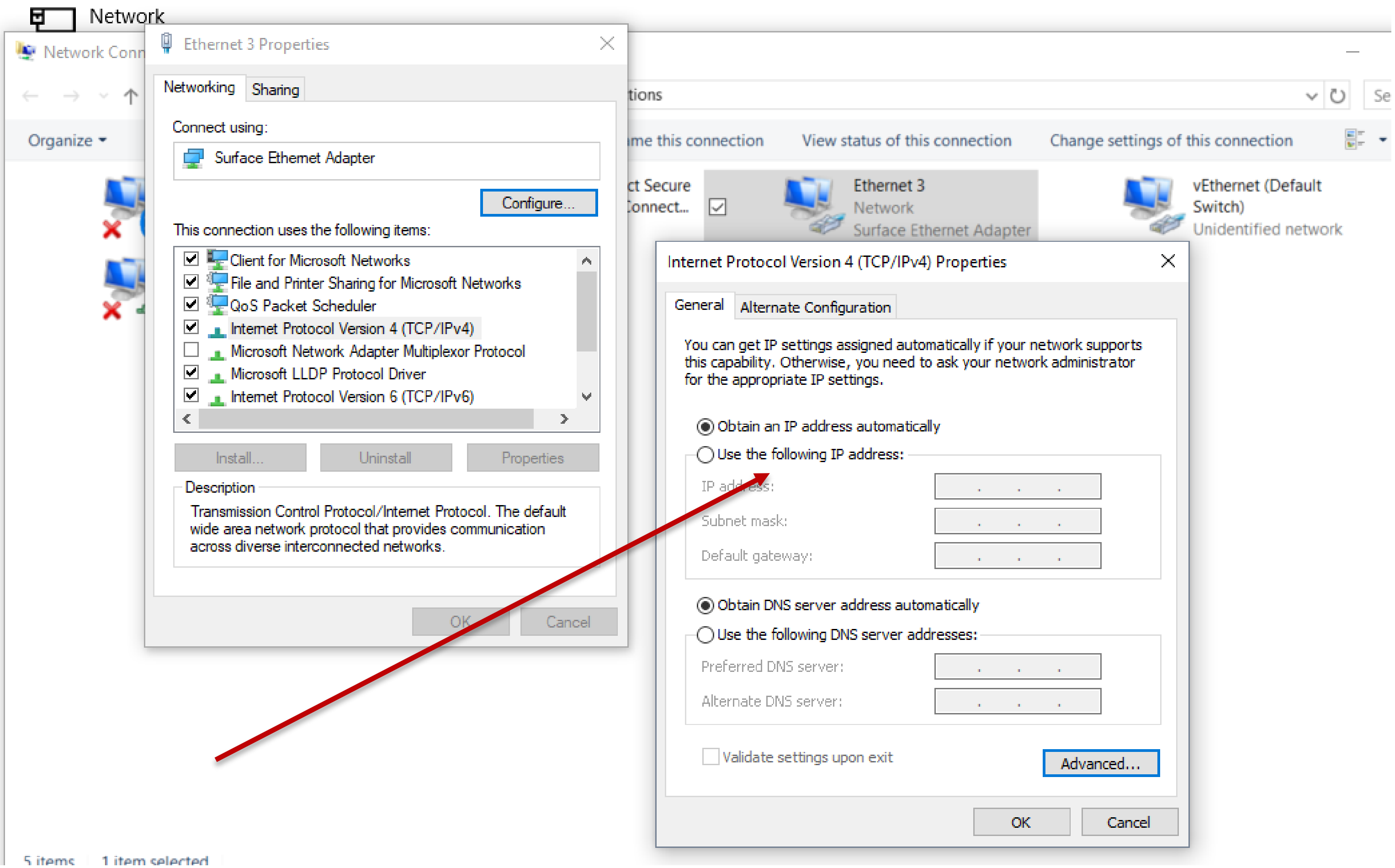
Configuring an Adapter



Configuring the Host



Configuring the Host



Plug and Play