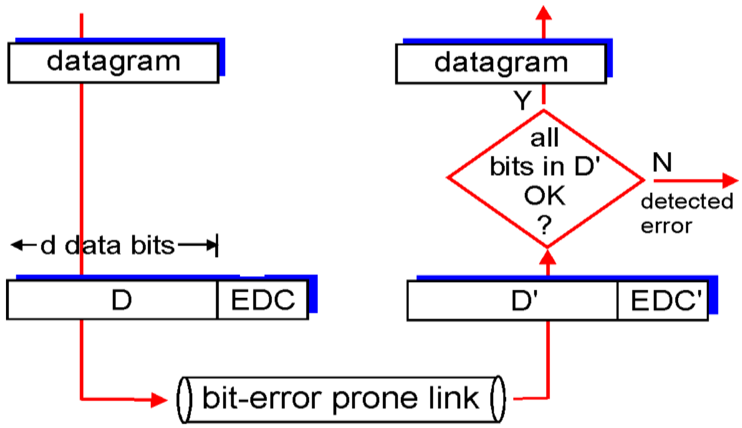
Error detection

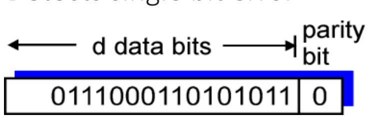
- EDC: Error Detection and Correction bits (redundancy)
- D: Data protected by error checking, may include header fields

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



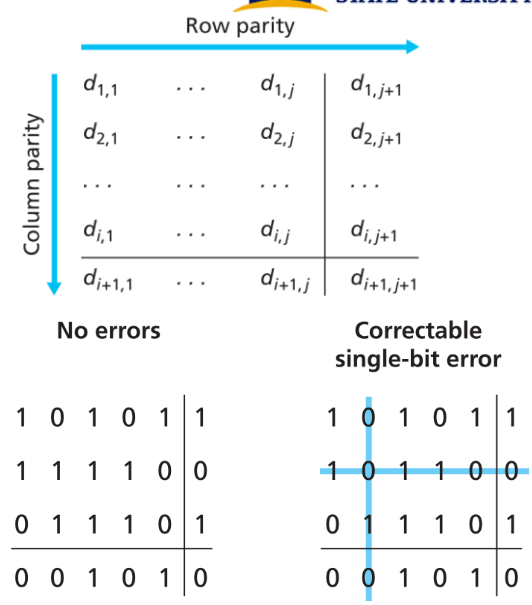
Parity checking

Single bit parity:
- Detects single bit error



What happens when errors occur in *bursts*? Say bits flip.

Two dimensional bit parity:
- Detects and **corrects** single bit error
- Detects two bit errors



Cyclic Redundancy Check (CRC)

MONTANA STATE UNIVERSITY

- Goal: Want a redundancy encoding that can tolerate **arbitrarily long** error bursts

- View data bits D as a binary number
- Agree on an an $r+1$ bit generator G, with the leftmost digit 1
- Compute r bits R such that:
  - $D \ll r \oplus \mathrm{R} = nG$ (nG – some multiple of G)
  - $D \ll r \oplus \mathrm{R} = [\mathrm{D}, \mathrm{R}]$ (operation creates the packet)

- How to compute R
  - $D \ll r \,\%\, G = R$
  - $D \ll r - R = D \ll r \oplus \mathrm{R} = nG$
    (⊕ equivalent to subtract mod 2 without carry)

- Destination uses known G to check if:
  - $(D \ll r \oplus \mathrm{R}) \,\%\, G == 0$ (if not, then error tx error)

d bits	*r* bits
D: Data bits to be sent	*R*: CRC bits

```
1011 XOR 0101 = 1110
1001 XOR 1101 = 0100
  1011 — 0101 = 1110
  1001 — 1101 = 0100
```

Can detect all burst errors up to r bits

$\boxed{G_{\text{CRC-32}} = 100000100110000010001110110110111}$

Widely used in practice (Ethernet, 802.11 WiFi)

Example

MONTANA STATE UNIVERSITY

D = 101110 G = 1011

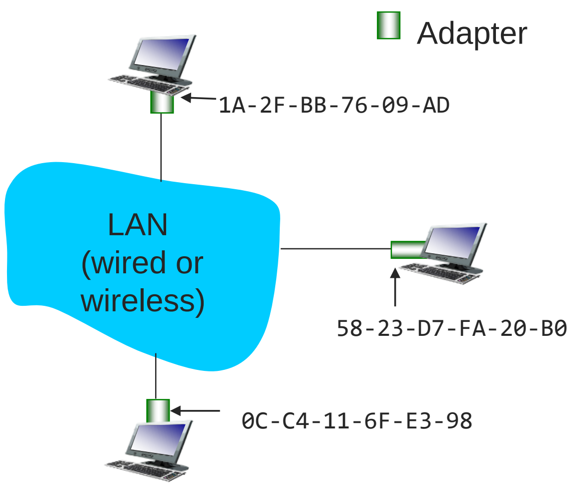
$R = D \ll r \,\%\, G$ = 101110 000 % 1011 = 101

$D \ll r - R$ = 101110 000 – 101 = 101101011

101101 011 % 1011 = 0

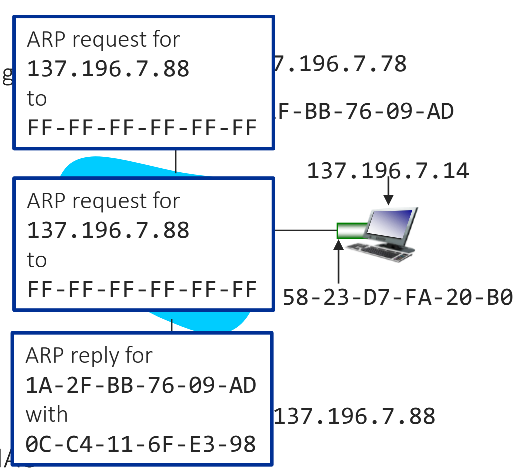
MAC Addresses

- MAC (or LAN or physical or Ethernet) address:
  - Addresses physical interfaces
  - Hardcoded 48 bit MAC address (from IEEE)
  - e.g.: `1A-2F-BB-76-09-AD`
- Analogy:
  - MAC address: Social Security Number
  - IP address: postal address
- Flat MAC address space
  - Plus: portability between LANs
  - Minus: switch tables not hierarchical (lower scalability)



Address Resolution Protocol (ARP)

- Question
  - How to determine interface's MAC address, knowing
    address?
- ARP table
  - Each IP node (host, router) on LAN has table
  - `< IP address; MAC address; TTL>`
- A wants to send datagram to B
  - B's MAC address not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - Broadcast address: `FF-FF-FF-FF-FF-FF`
  - All nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - Frame unicast to A's MAC address

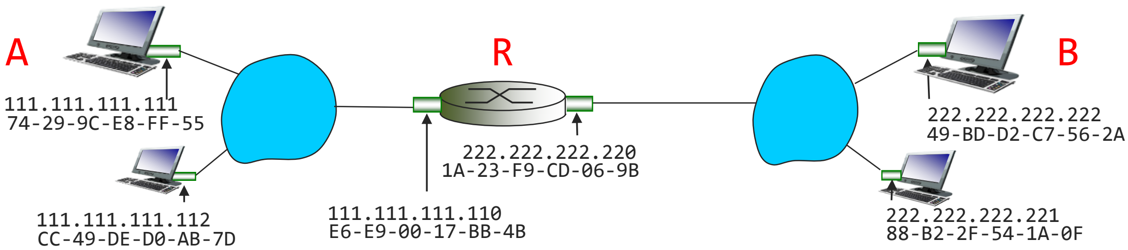


Plug-and-play (soft-state):
– self-configuring tables
– automatic entry expiry

Routing to another LAN

MONTANA STATE UNIVERSITY

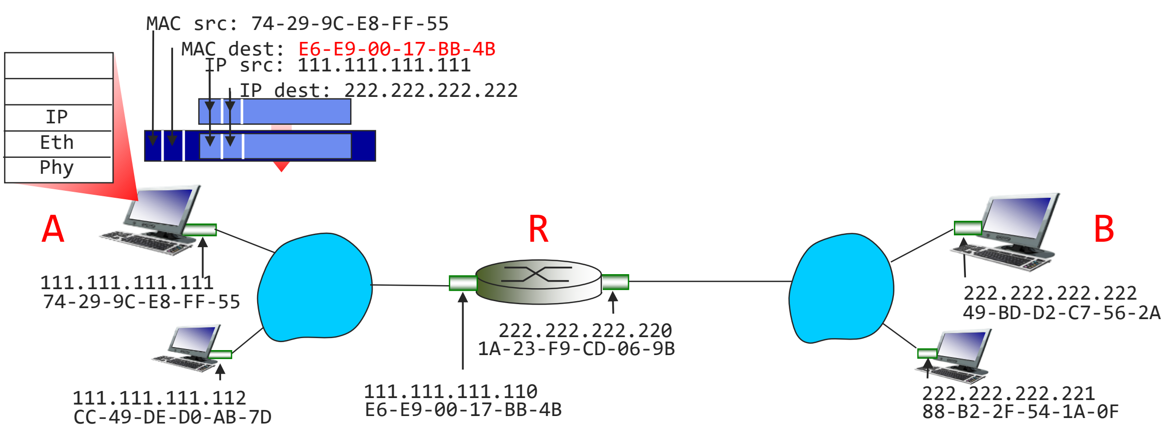
- Walkthrough: send datagram from A to B via R
  - Focus on addressing – at IP (datagram) and MAC layer (frame)
  - Assume A knows B's IP address. How? ← DNS
  - Assume A knows IP address of first hop router, R. How? ← DHCP
  - Assume A knows R's MAC address. How? ← ARP



Routing to another LAN

MONTANA STATE UNIVERSITY

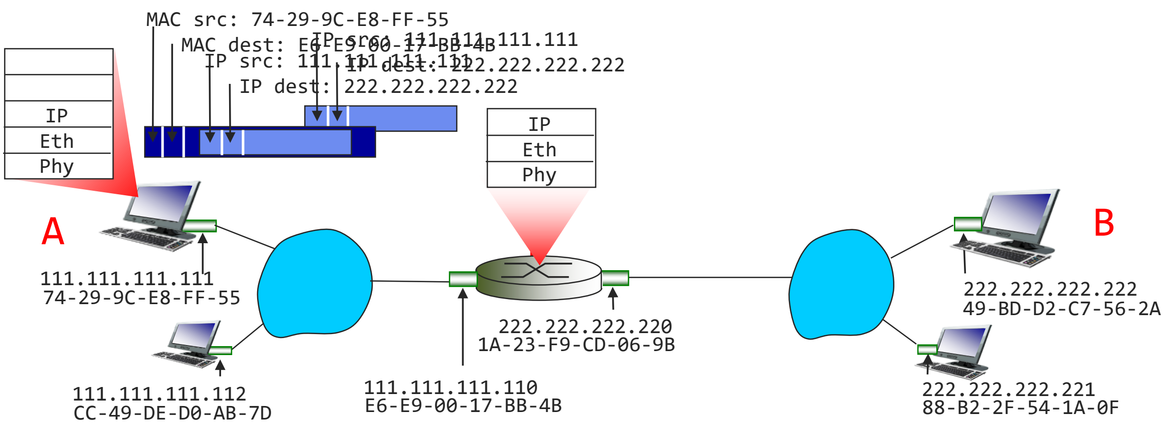
- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram



Routing to another LAN

MONTANA STATE UNIVERSITY

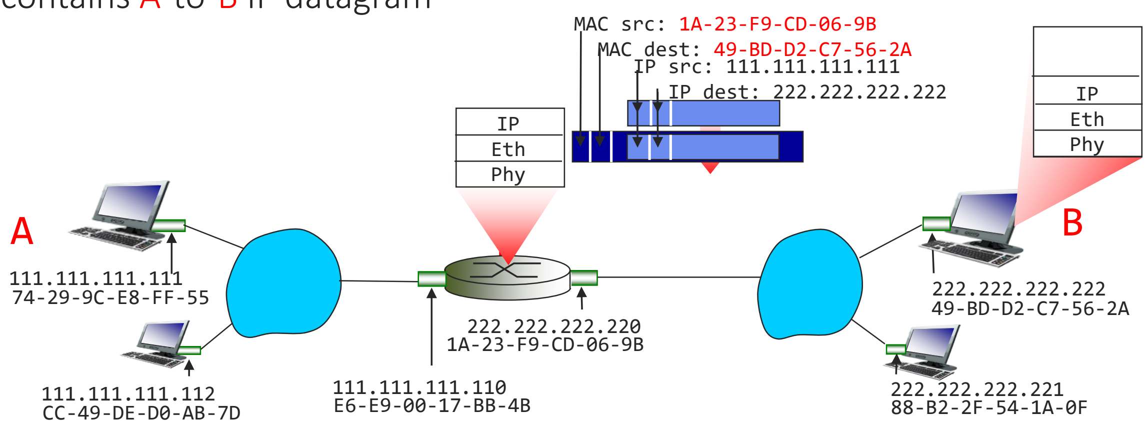
- Frame sent from A to R
- Frame received at R, datagram removed, passed up to IP



Routing to another LAN

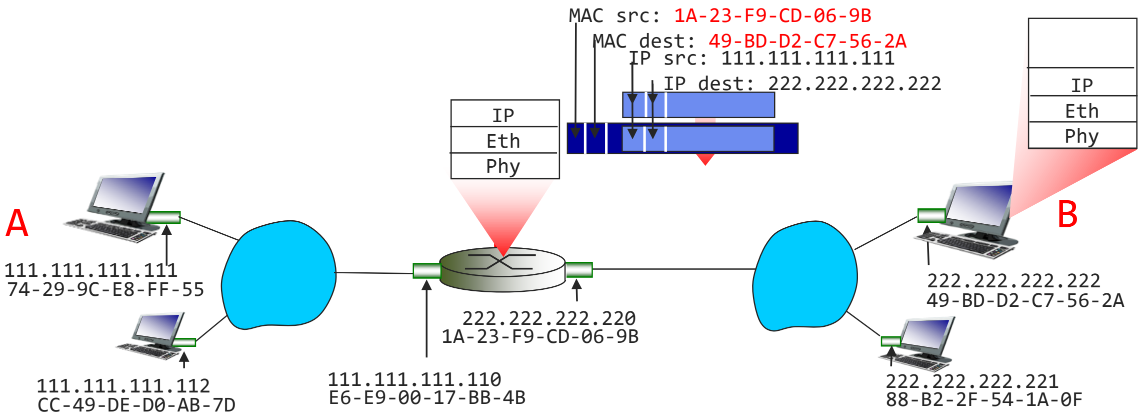
MONTANA STATE UNIVERSITY

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



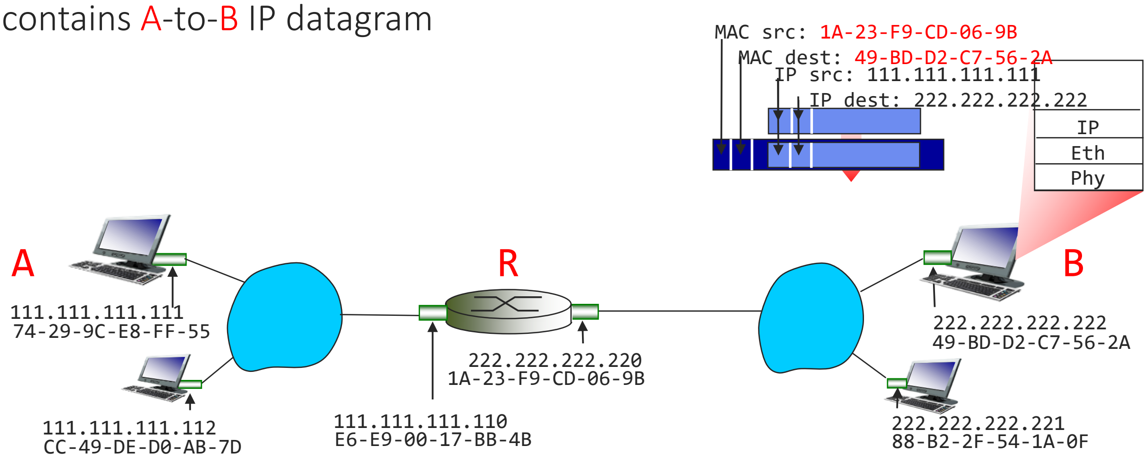
Routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Routing to another LAN

- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram



Exercise

Consider three LANs interconnected by two routers.

a) Assign IP addresses to all of the interfaces
b) Assign MAC addresses to all of the adapters
c) Consider sending an IP datagram from Host A to Host E. Enumerate all the steps assuming that the ARP table in the sending host is empty (and the other tables are up to date)