Computer Networking: A Top Down Approach Featuring the Internet,
Second Edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2002.

Data Link Layer

The Data Link Layer

Our goals:

- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control: *done!*
- instantiation and implementation of various link layer technologies

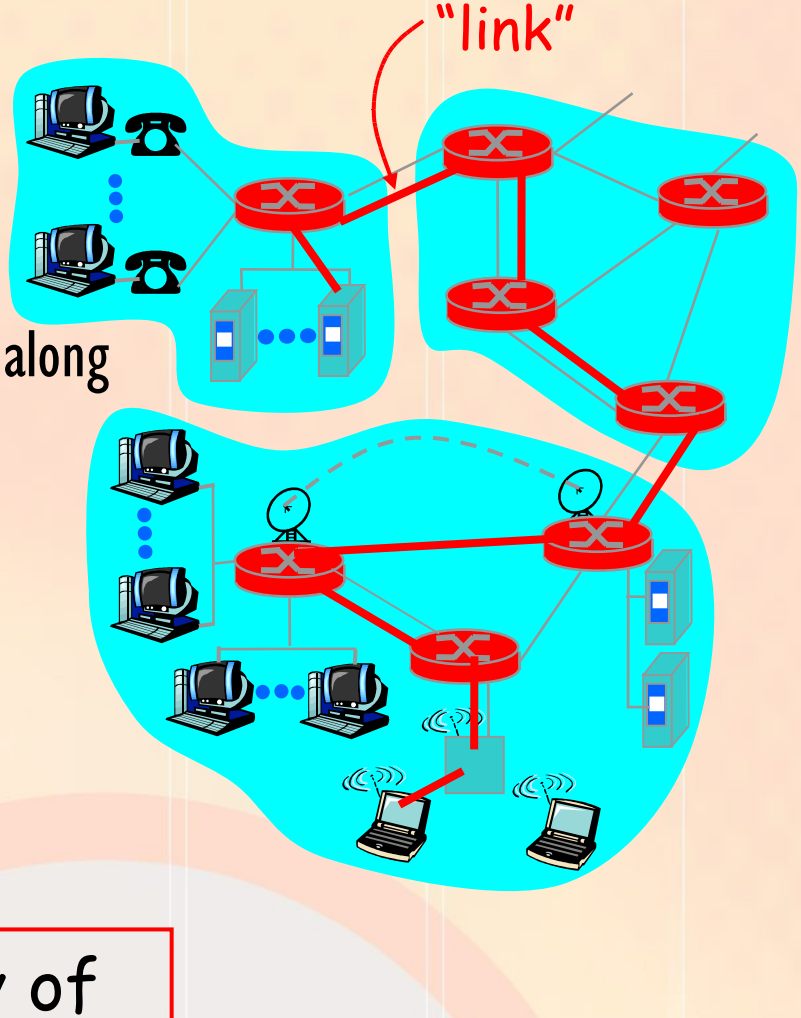
Outline

- **Introduction and services**
- Error detection and correction
- Multiple Access protocols
- LAN addresses and ARP
- Ethernet
- Hubs, bridges, and switches
- Wireless links and LANs
- PPP

Link Layer: Introduction

Terminology:

- hosts and routers are **nodes**
  (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- 2-PDU is a **frame,** encapsulating a datagram



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

Link layer: context

- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

transportation analogy

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm

Link Layer Services

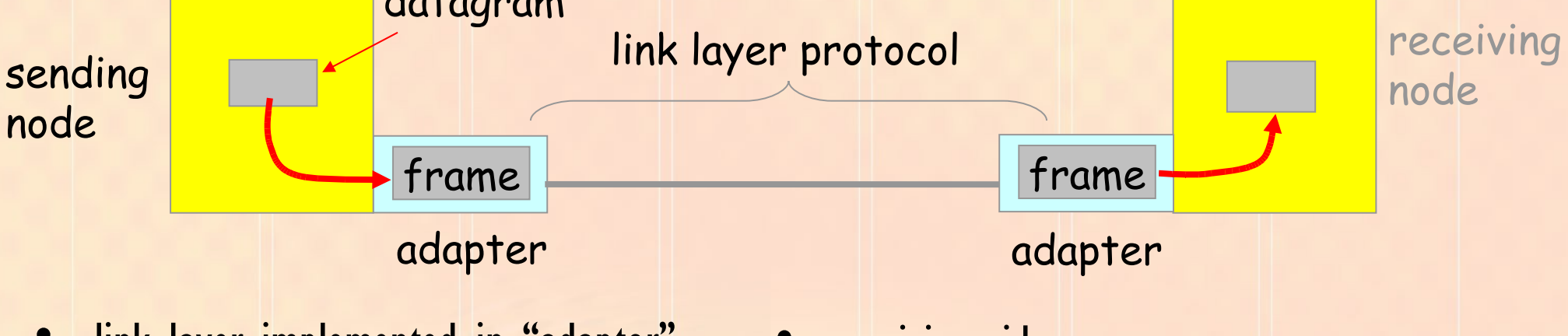
- **Framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - '**physical addresses**' used in frame headers to identify source and destination
    - different from IP address!
- **Reliable delivery between adjacent nodes**
  - seldom used on low bit error link (fibre, twisted pair)
  - wireless links: high error rates

    - Q: why both link-level and end-end reliability?

Link Layer Services

- **Flow Control:**
  - pacing between adjacent sending and receiving nodes
- *Error Detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *Error Correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *Half-duplex / Full-duplex*

Adaptors Communicating



- link layer implemented in "adaptor" (NIC)
  - Ethernet card, PCMCI card, 802.11 card
- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc
  - extracts datagram, passes to receiving node
- adapter is semi-autonomous
- link & physical layers

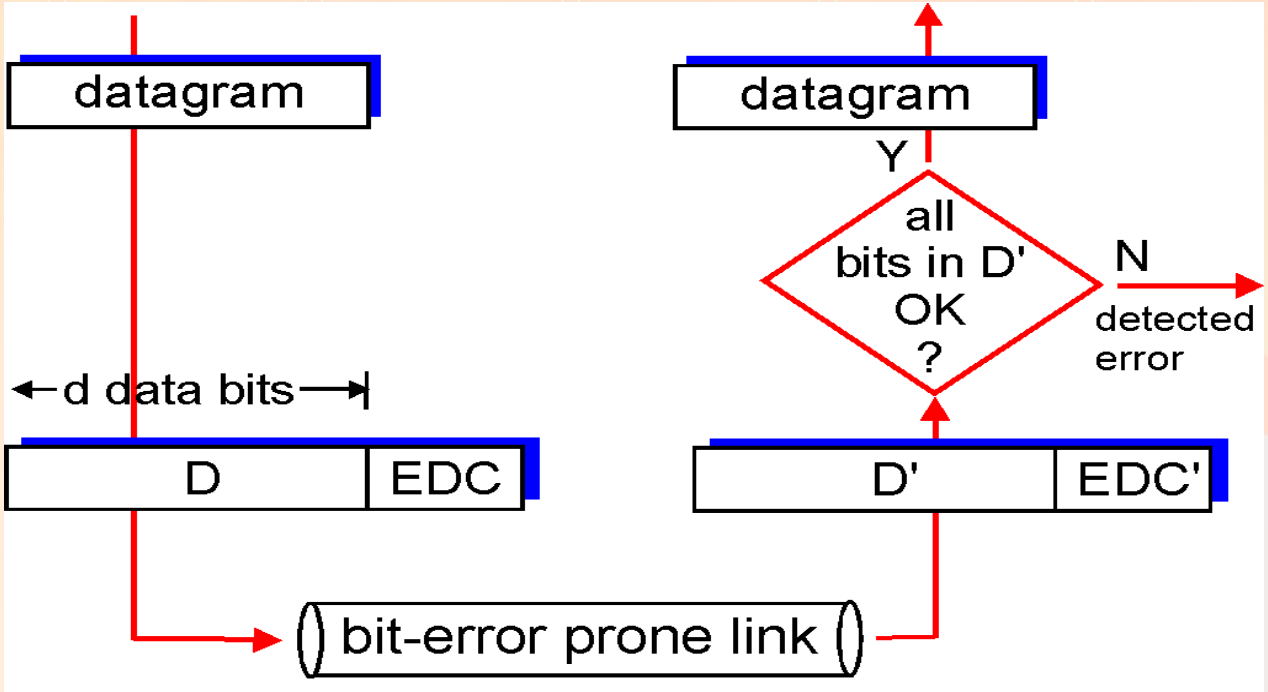
Outline

- Introduction and services
- **Error detection and correction**
- Multiple Access protocols
- LAN addresses and ARP
- Ethernet
- Hubs, bridges, and switches
- Wireless links and LANs
- PPP

Error Detection

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

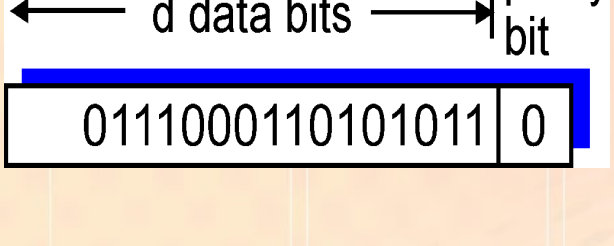
- **Error detection not 100% reliable!**
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Parity Checking

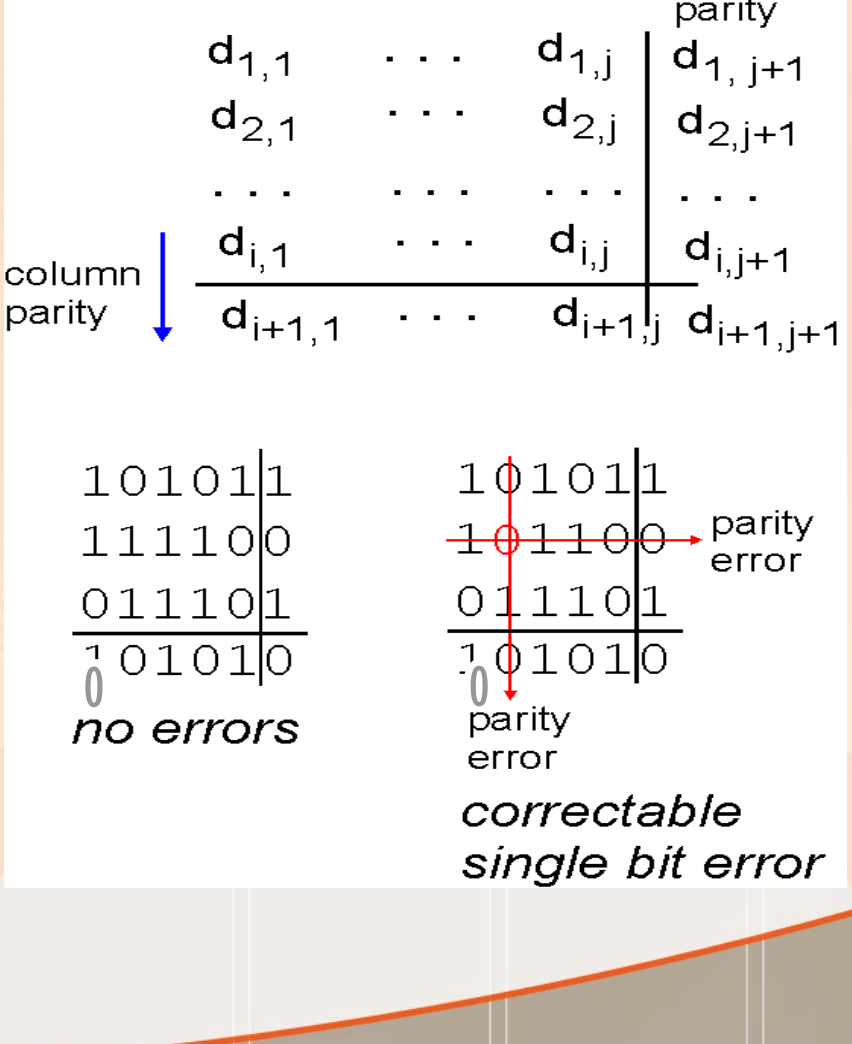
Single Bit Parity:

Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors



Internet checksum

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

Sender:

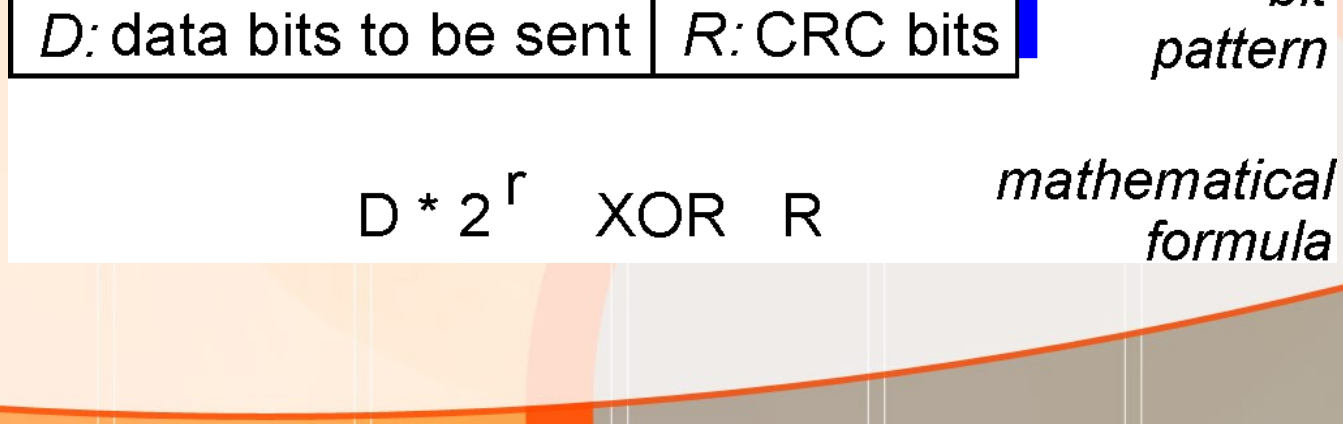
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

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 …**

Checksumming: Cyclic Redundancy Check

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: 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
- widely used in practice (ATM, HDCL)



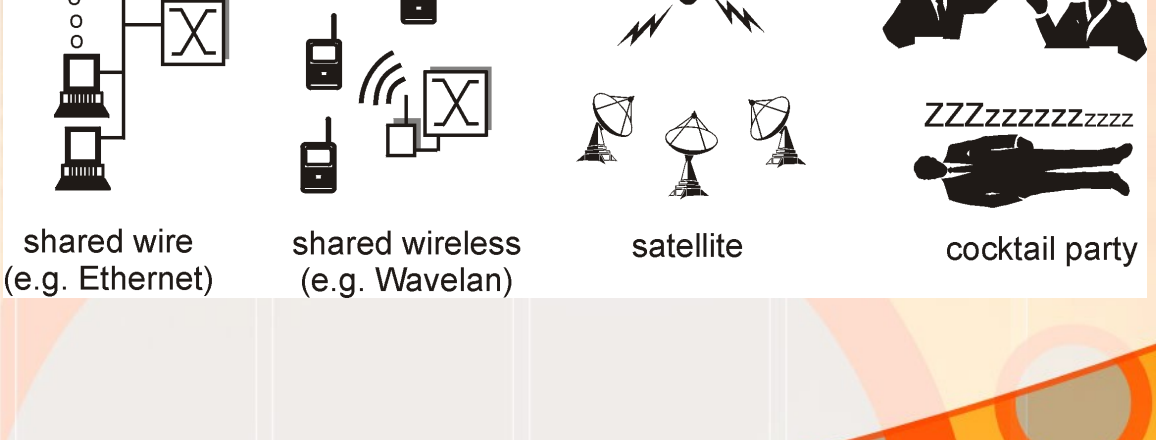
Outline

- Introduction and services
- Error detection and correction
- **Multiple Access protocols**
- LAN addresses and ARP
- Ethernet
- Hubs, bridges, and switches
- Wireless links and LANs
- PPP

Multiple Access Links and Protocols

Two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - 802.11x wireless LAN



Multiple Access protocols

- single shared broadcast channel
- *interference:* two or more simultaneous transmissions by nodes:
  - only one node can send successfully at a 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!
- what to look for in multiple access protocols:

Ideal Multiple Access Protocol

Broadcast channel of rate R bps

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: a taxonomy

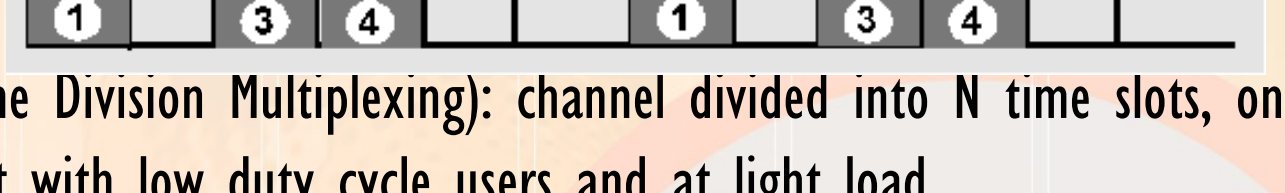
Three broad classes:

- 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"
  - tightly coordinate shared access to avoid collisions

Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

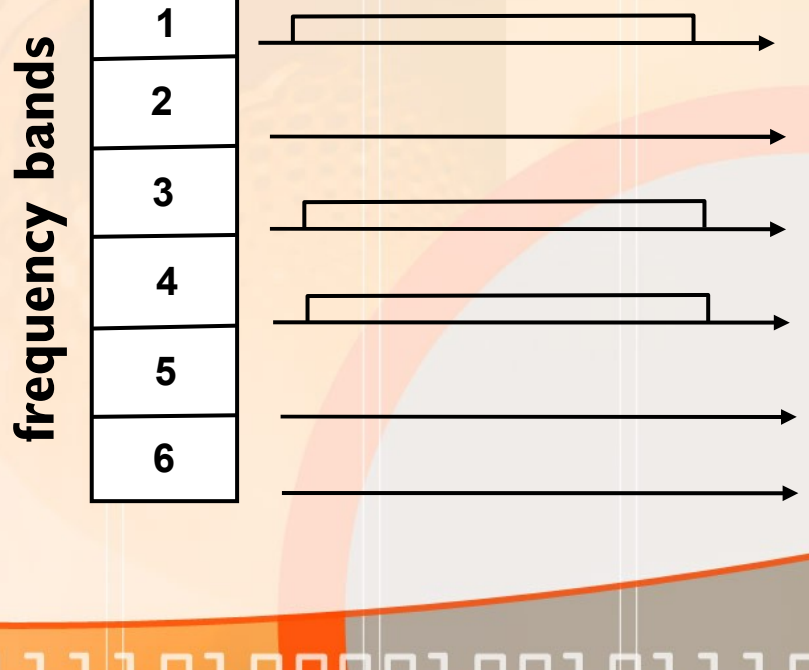


- TDM (Time Division Multiplexing): channel divided into N time slots, one per user; inefficient with low duty cycle users and at light load.

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 pkt, frequency bands 2,5,6 idle



Channel Partitioning (CDMA)

CDMA (Code Division Multiple Access)

- unique "code" assigned to each user; i.e., code set partitioning
- used mostly in wireless broadcast channels (cellular, satellite, etc)
- all users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data
- *encoded signal* = (original data) X (chipping sequence)
- *decoding:* inner-product of encoded signal and chipping sequence
- allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")

Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- When two or more transmitting nodes: *collision*
- Random Access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - CSMA, CSMA/CD, CSMA/CA

CSMA (Carrier Sense Multiple Access)

CSMA: **L**isten before transmit:

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

- Human analogy: don't interrupt others!

- Collisions *can* still occur though because of the propagation delay means two nodes may not hear each other's transmission

CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
- human analogy: the polite conversationalist

"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!

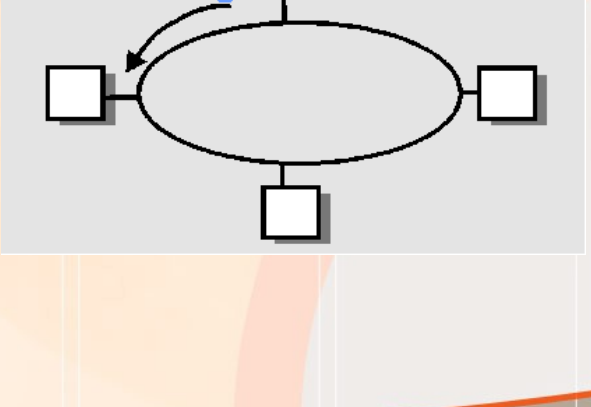
"Taking Turns" MAC protocols

Polling:

- master node "invites" slave nodes to transmit in turn
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Summary of MAC protocols

- What do you do with a shared media?
  - **Static Channel Partitioning**, by time, frequency or code
    - Time Division, Code Division, Frequency Division
  - **Dynamic Random partitioning**,
    - CSMA, CSMA/CD
    - carrier sensing: easy in some technologies (wire), hard in others (wireless)
    - CSMA/CD used in Ethernet
  - **Taking Turns**
    - polling from a central site, token passing

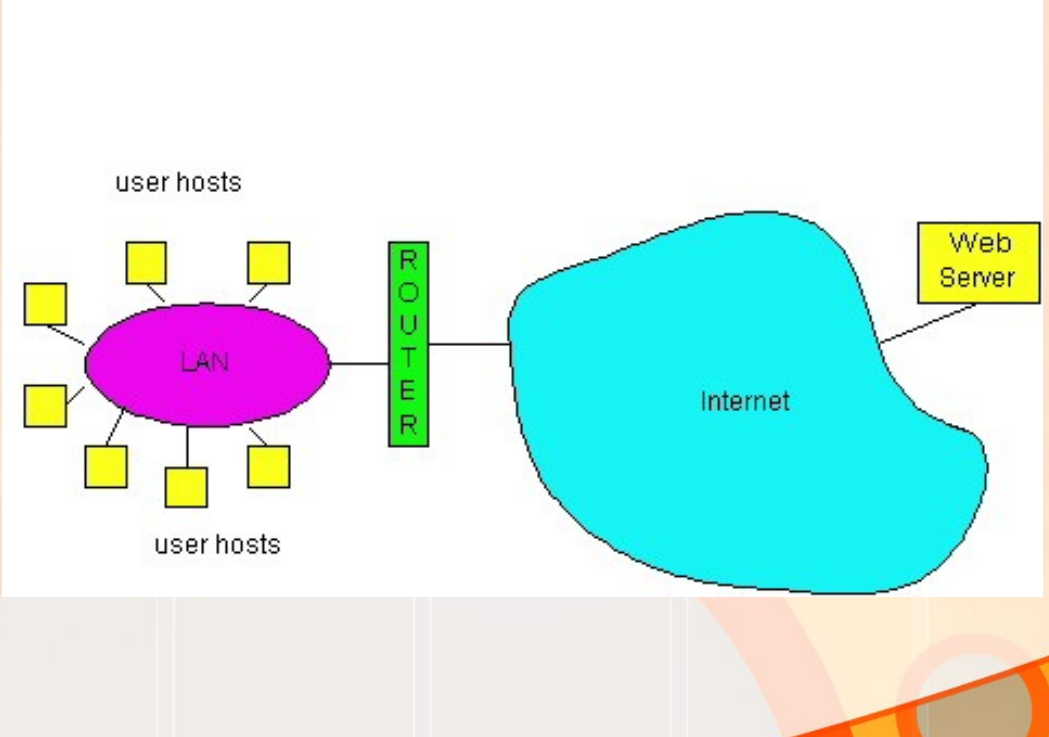
LAN technologies

Data link layer so far:

- services, error detection/correction, multiple access

Next: LAN technologies

- Addressing
- Ethernet
- hubs, bridges, switches
- 802.11x (*Wi-Fi*)
- PPP



LAN Addresses and ARP

32-bit IP address:

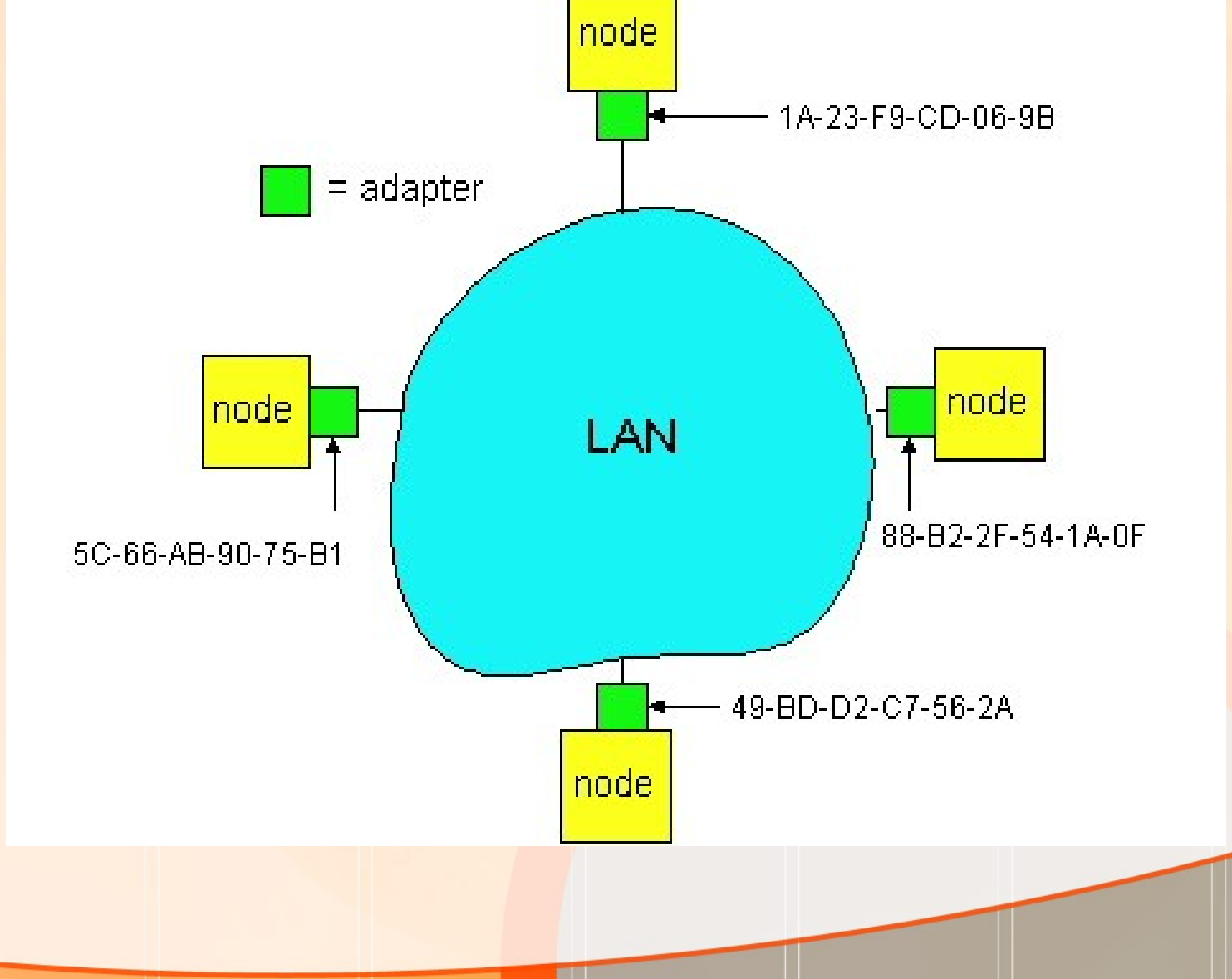
- *network-layer* address
- used to get datagram to destination IP network (recall IP network definition)

LAN (or MAC or physical or Ethernet) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs)
  burned in the adapter ROM

LAN Addresses and ARP

Each adapter on LAN has unique LAN address



LAN Address

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:

  (a) MAC address: like Social Security Number

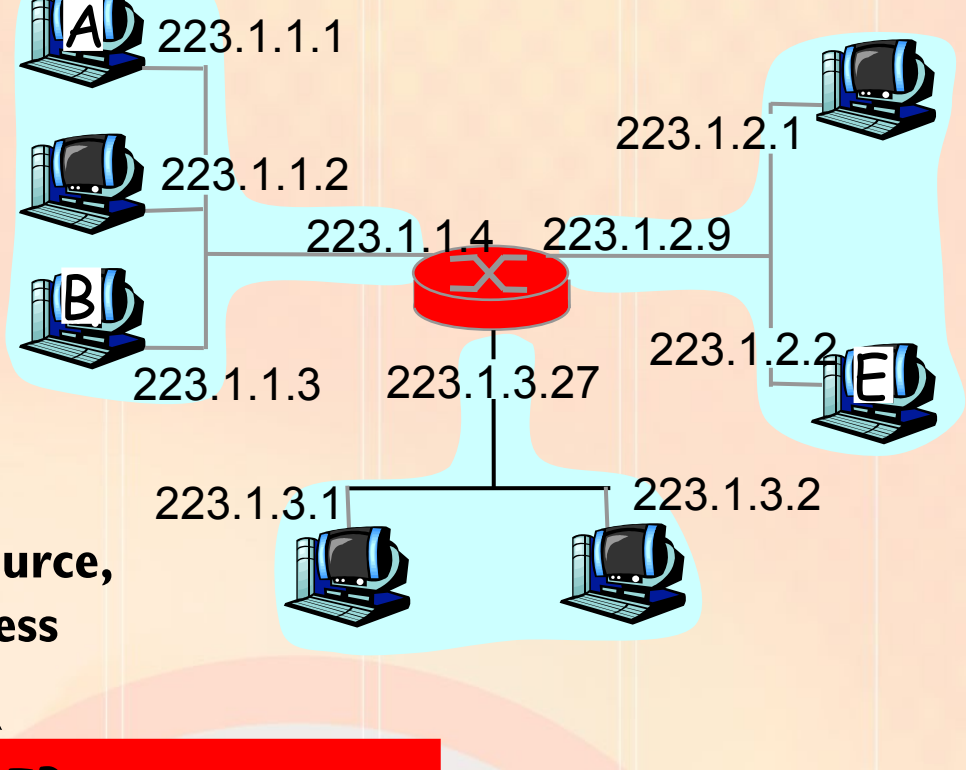
  (b) IP address: like postal address

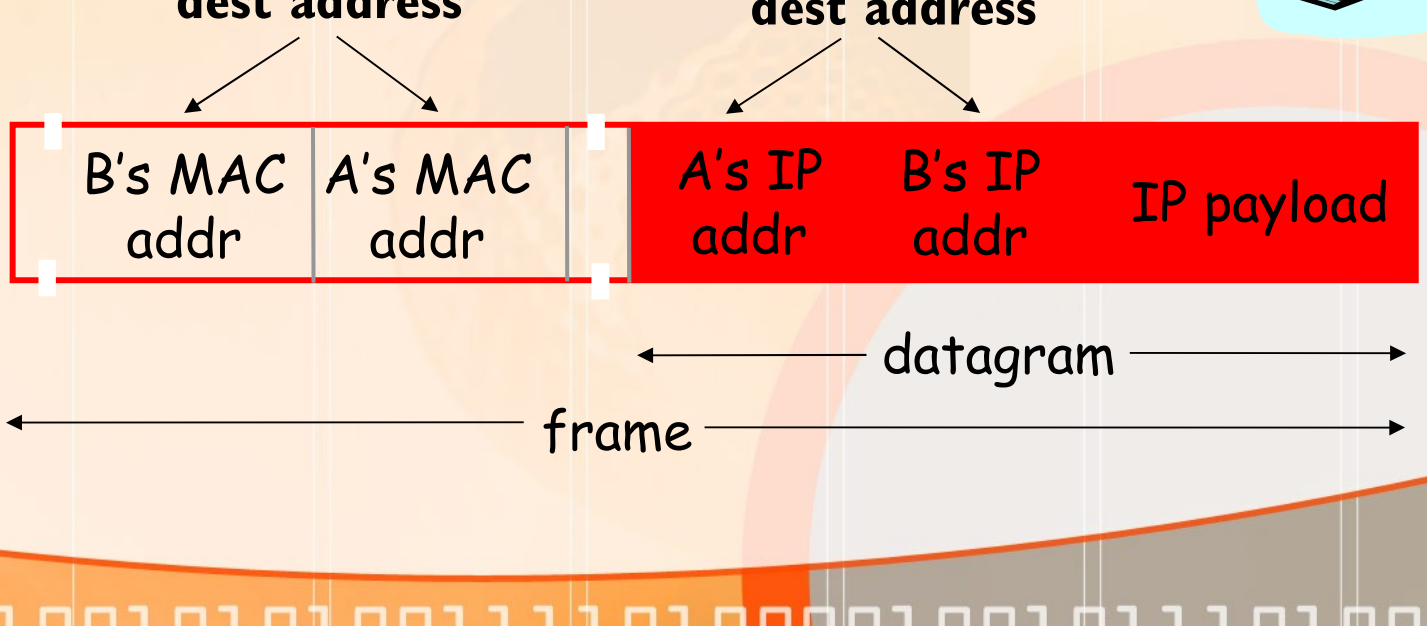
- MAC flat address => portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached

Recall earlier routing discussion

Starting at A, given IP datagram addressed to B:

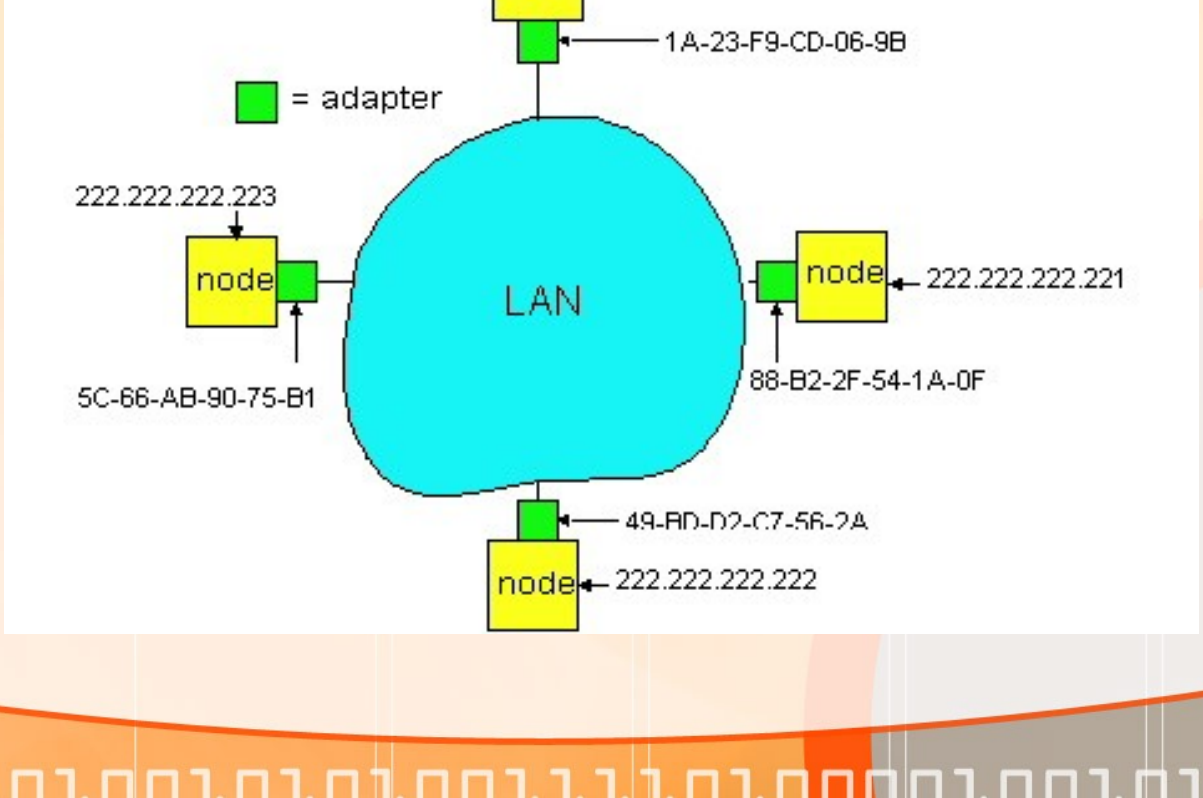
- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame





ARP: Address Resolution Protocol

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



- Each IP node (*host/router*) on LAN has ARP table
- ARP 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)

ARP protocol

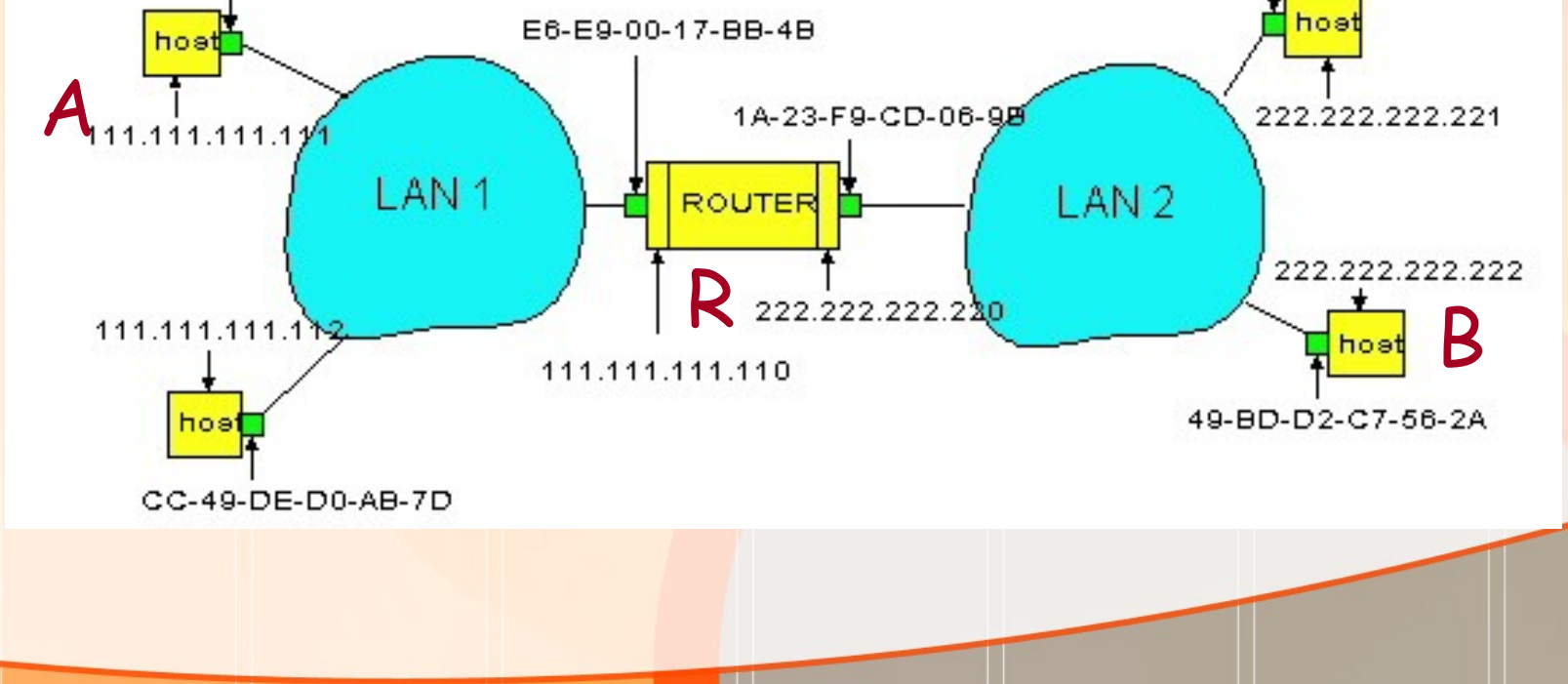
- A wants to send datagram to B, and A knows B's IP address.
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC 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

Routing to another LAN

send datagram from A to B via R;

Assume A know's B IP address

- Two ARP tables in router R, one for each IP network (LAN)
- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address **E6-E9-00-17-BB-4B**, etc



Routing to another LAN

- A creates datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's data link layer sends frame
- R's data link layer receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address
- R creates frame containing A-to-B IP datagram sends to B