Introduction to Computer Networks & Communications

Lecture 9-10: Routing
MAC/Physical Layer

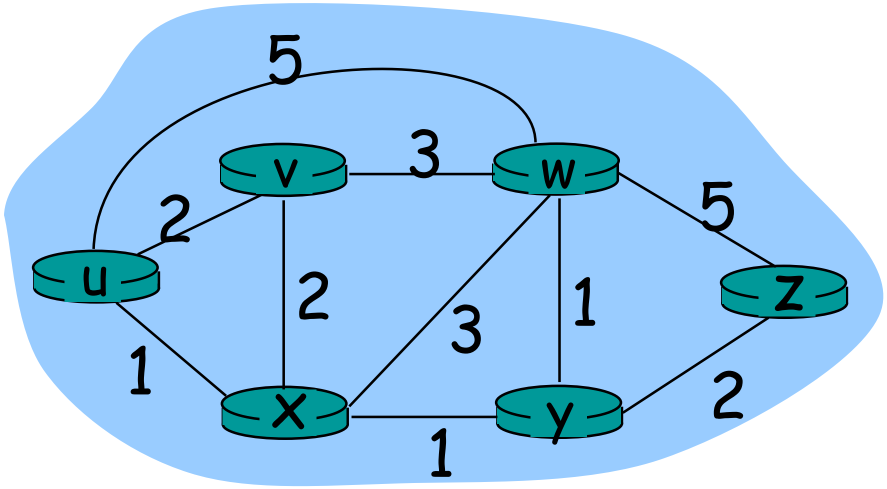
Dr. Amal ElNahas

Routing Algorithms Classification

- **Global**
  - Least-cost path is calculated using complete, global knowledge about the network
  - All routers have complete graph (topology, costs)
    - “Link state” algorithms (OSPF)

- **Decentralized**
  - Calculations are carried out in an iterative, distributed manner.
  - Router knows link costs to physically connected adjacent nodes
    - Run iterative algorithm to exchange information with adjacent nodes
    - “Distance vector” algorithms (RIP)

Graph Abstraction



- Graph: G = (N,E)
- N: set of nodes (routers) = { u, v, w, x, y, z }
- E: set of edges (links)= { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) } (Links are bi-directional)
- Each link is associated with a cost value (can represent delay, distance,..)
- Cost of a path (x1,x2 ,….,xn) = c(x1 ,x2) + c(x2 ,x3) +…..+ c(xn-1 ,xn)
- Routing algorithm objective: Find path with minimum cost between sender and receiver

Link State Algorithms

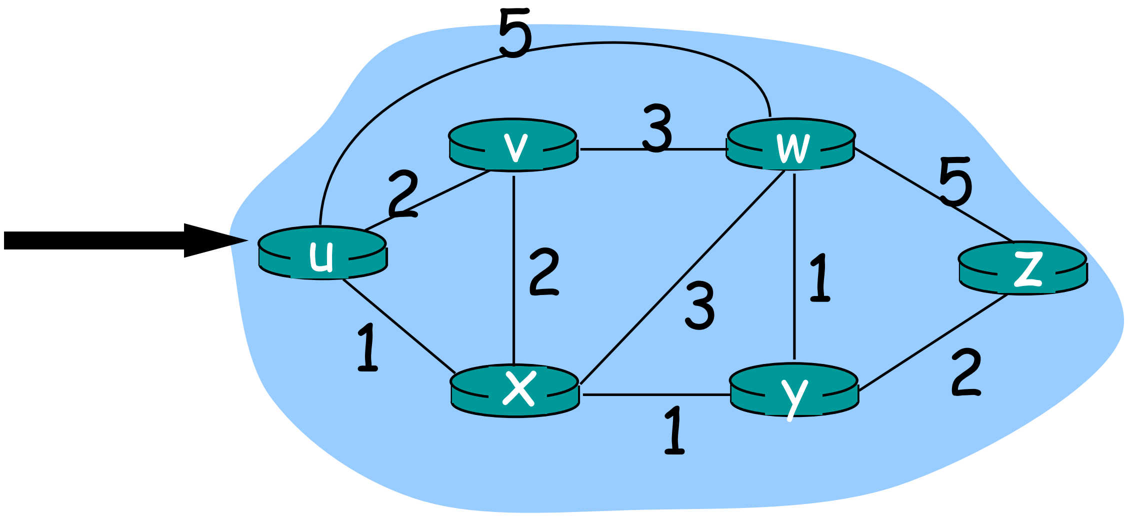
- All nodes have identical and complete view of the network (network topology and link costs). Done through link state flooding

- Each router runs the link state algorithm (Dijkstra's algorithm) and compute the set of least cost paths from itself to all other routers
  - Produces a forwarding table for that router
  - All routers compute consistent forwarding tables

- Each router broadcasts link's state periodically (every 30 min) or whenever a change happens

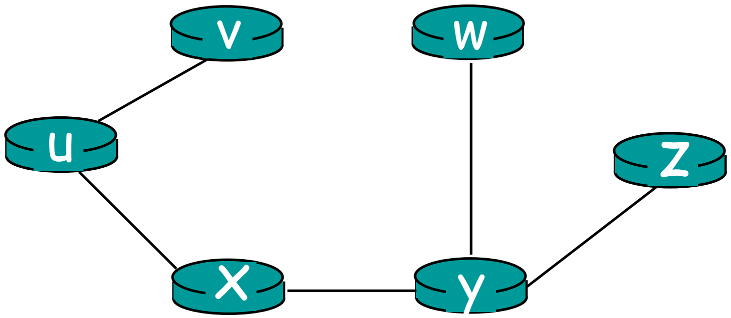
Dijkstra’s Routing Algorithm: Example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u		1,u	∞	∞
1	ux	2,u	5,u		2,x	∞
2	uxy	2,u	4,x			4,y
3	uxyv		3,y			4,y
4	uxyvw		3,y			4,y
5	uxyvwz					



Shortest Path Tree

- Shortest path tree from u
- Routing table for node u:



destination	Next hop	Cost
v	v	2
x	x	1
y	x	2
w	x	3
z	x	4

Dijkstra's Routing Algorithm

```
Initialization:
  N = {A}
    for all nodes v
      if v adjacent to A
          D(v) = c(A,v)
      else D(v) = ∞
      endif
Loop
  find w not in N such that D(w) is a minimum
  add w to N
  update D(v) for all v adjacent to w and not in N:
      D(v) = min( D(v), D(w) + c(w,v) )
   /* new cost to v is either old cost to v or known
      shortest path cost to w plus cost from w to v */
until all nodes in N
```

N : the set of nodes to which we have computed the minimum cost path

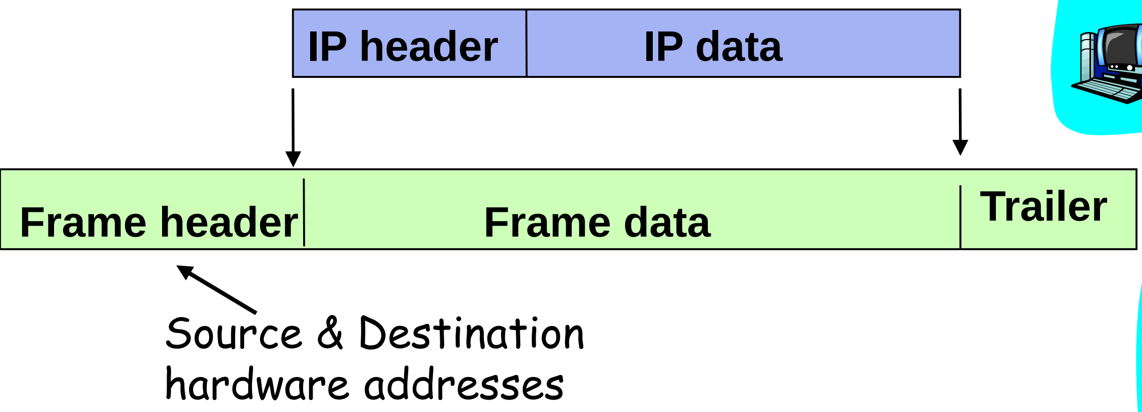
D(*x*): the current cost of path from source to *x*

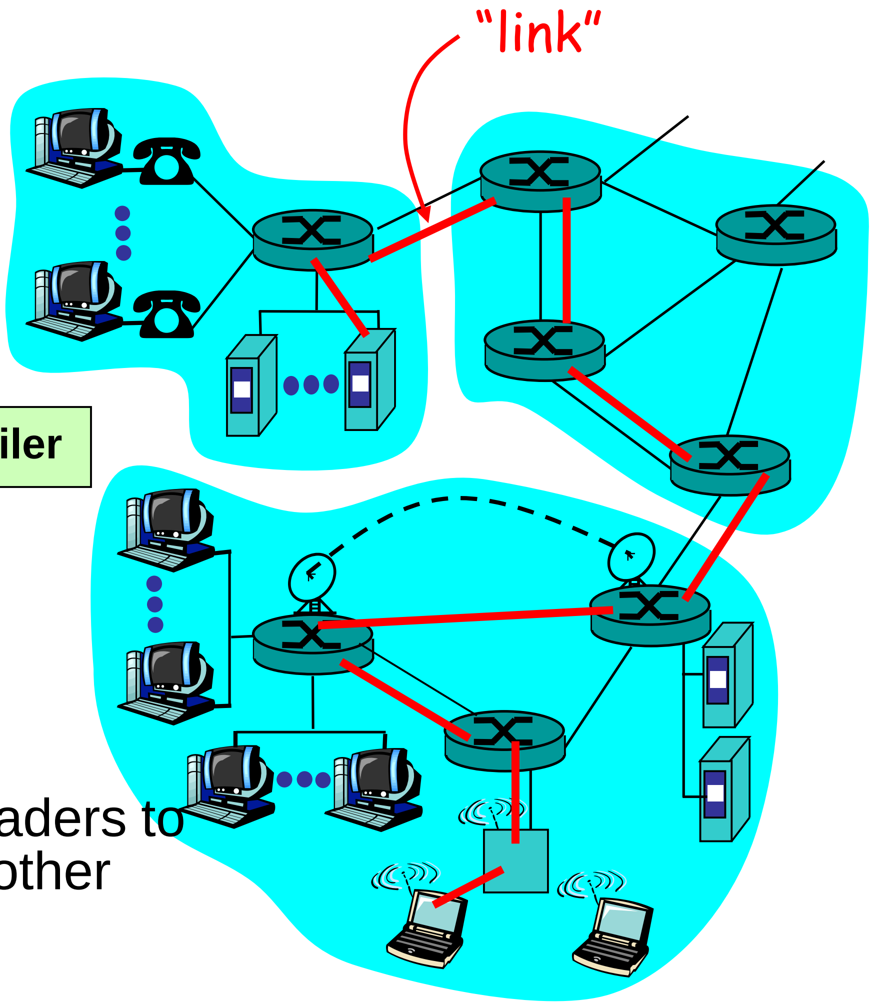
P(v): previous node along path from source to *v*

c(*n*,*m*): the cost of the link from *n* to *m* (= ∞ *if not adjacent)*

The Link Layer

- Data link layer:
  - Move datagrams from one node to adjacent node over a link

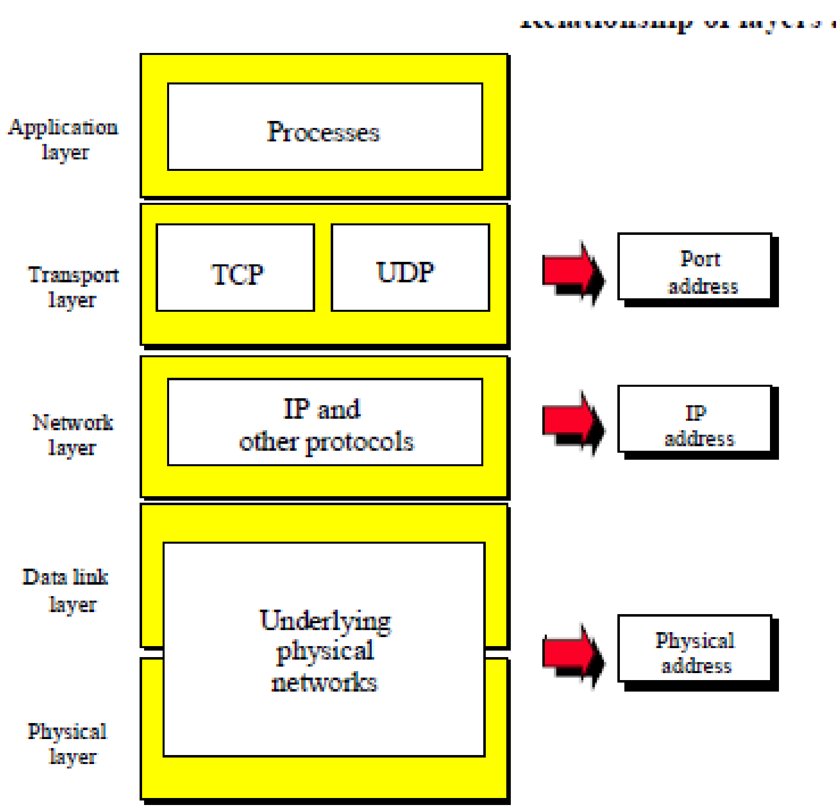




- hardware addresses used in frame headers to get datagram from one interface to another physically-connected interface (same network)

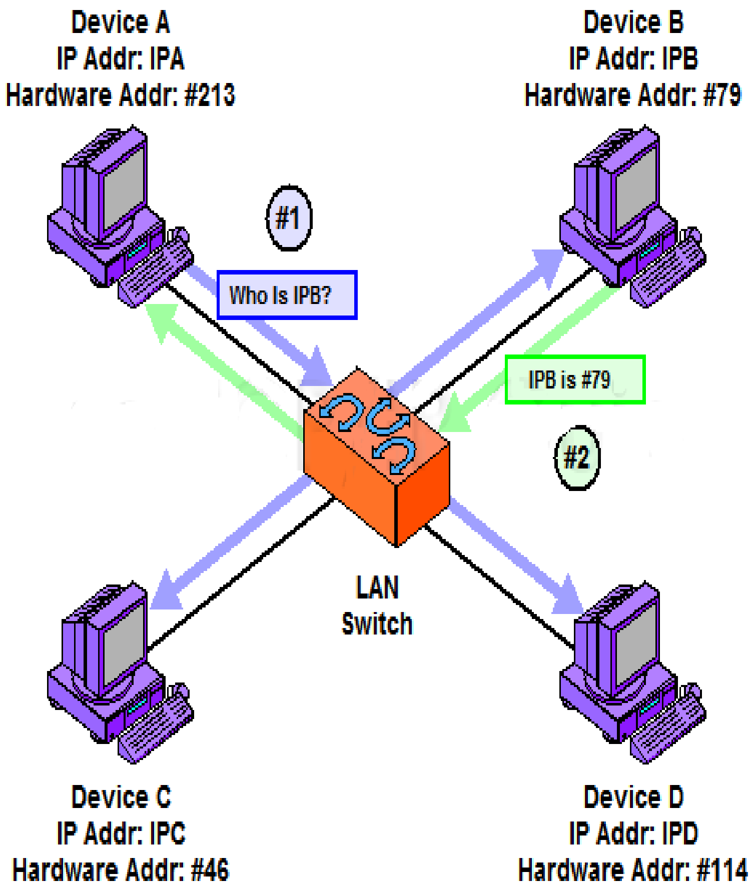
Translating IP addresses: Motivation

- Hardware only understands hardware addresses (must use hardware (physical) addresses to communicate over network)
- Applications only use Internet address
- Must map from IP address to hardware address for transmission (address resolution)
- Address resolution is performed at each step along the path



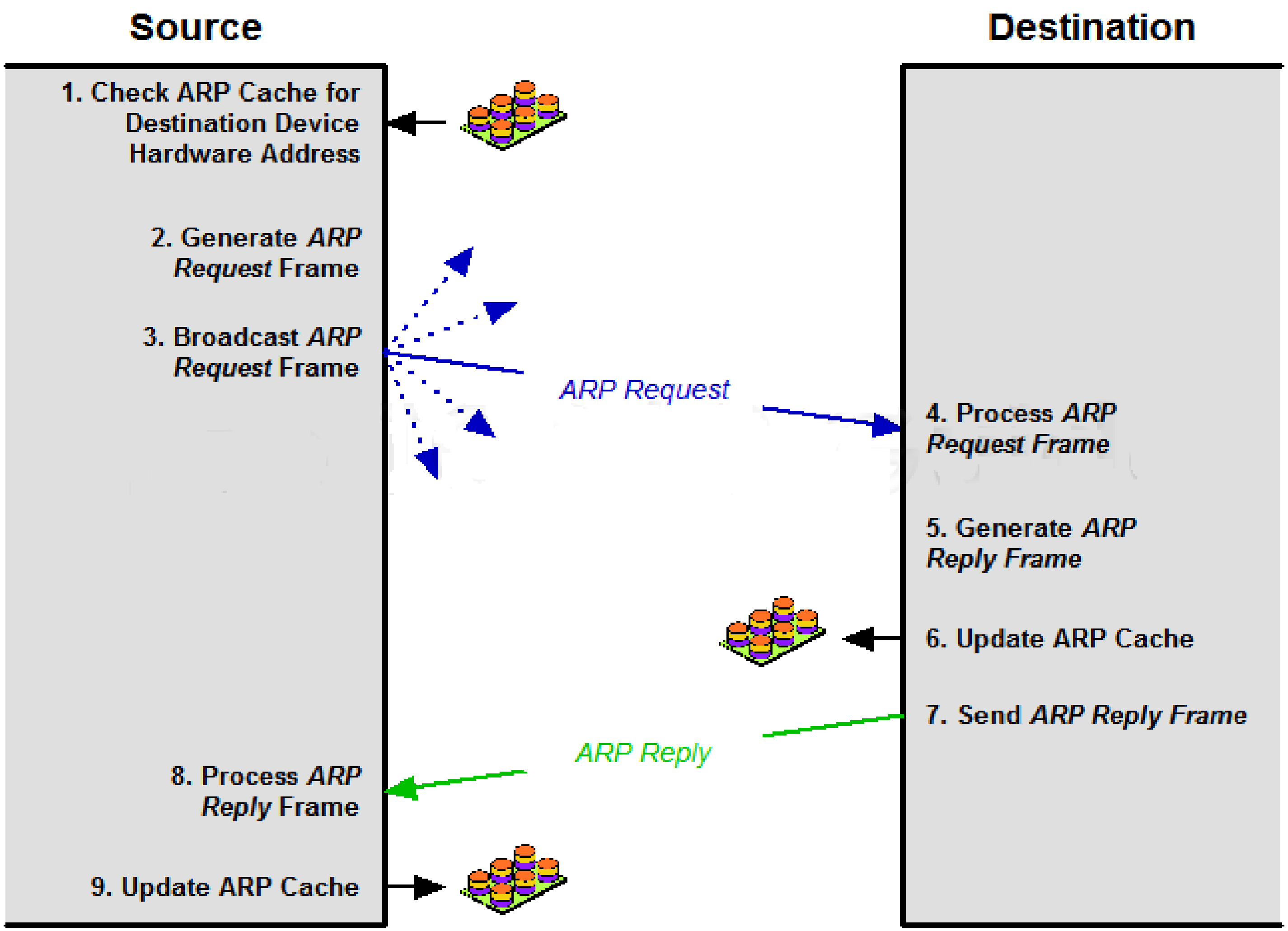
ARP: Address Resolution Protocol

- Querying host broadcasts a packet asking for a translation for some IP address
- Host with that IP address answers with ARP response containing the physical address
  - all hosts know their own addresses



ARP Caching

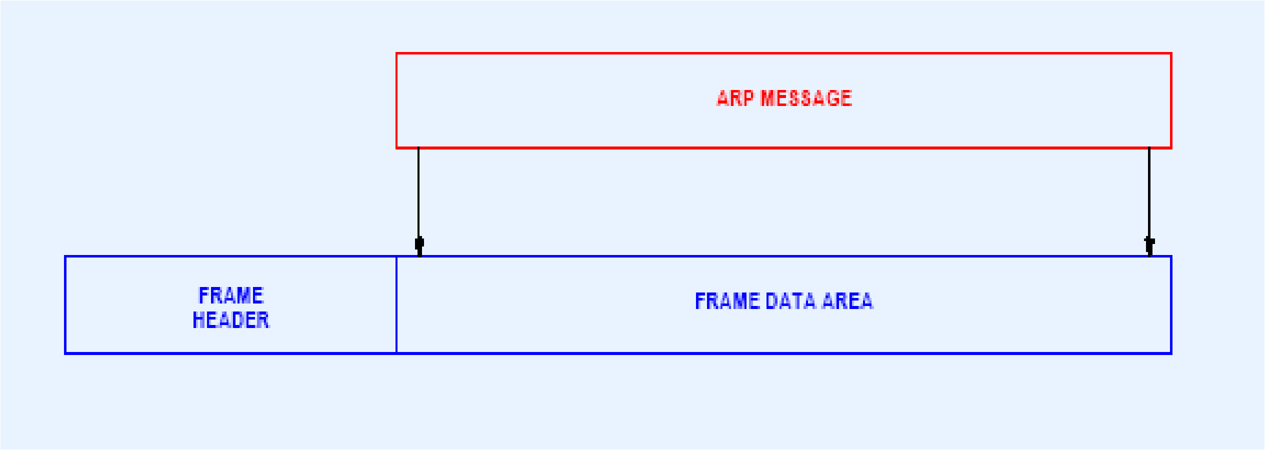
- Cannot afford to send ARP request for each packet
- Solution:
  - Maintain a table of bindings (ARP table is a cache)
  - Effect: Use ARP one time, place results in table, and then send many packets
  - To avoid stale bindings: Entries timeout and are removed (Typical timeout: 15-20 minutes)


Source
Destination
1. Check ARP Cache for Destination Device Hardware Address
2. Generate ARP Request Frame
3. Broadcast ARP Request Frame
ARP Request
4. Process ARP Request Frame
5. Generate ARP Reply Frame
6. Update ARP Cache
7. Send ARP Reply Frame
ARP Reply
8. Process ARP Reply Frame
9. Update ARP Cache

Algorithm Features

- If A ARPs B, B keeps A’s information
  - B will probably send a packet to A soon
- If A ARPs B, other machines do not keep A’s information
  - Avoids clogging ARP caches needlessly
- Conceptual purpose of ARP:
  - Isolates hardware address at low level
  - Allows application programs to use IP address

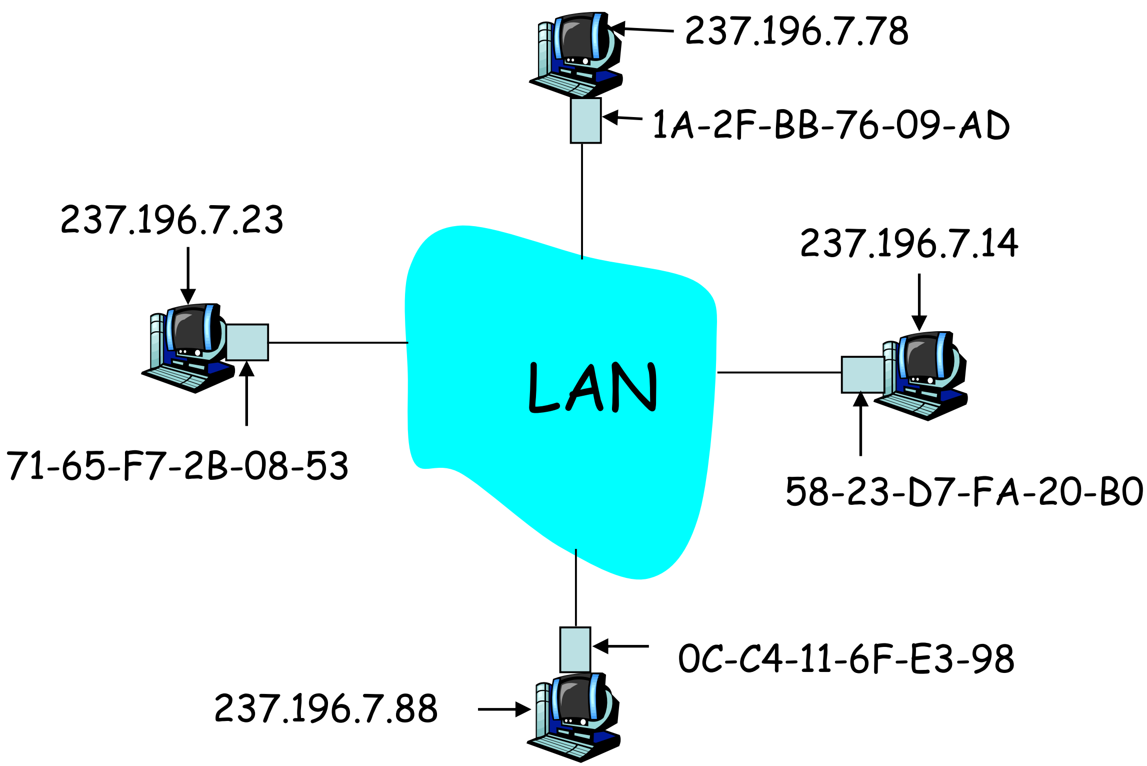
ARP Encapsulation



- ARP message placed in frame data area
- Data are padded with zeros if ARP message is shorter than the minimum ethernet frame

Hardware Addresses

- Each network card (adapter) has a permanent unique MAC address (a.k.a hardware address, physical address, LAN address) represented in hexadecimal notation and burned in the adapter ROM

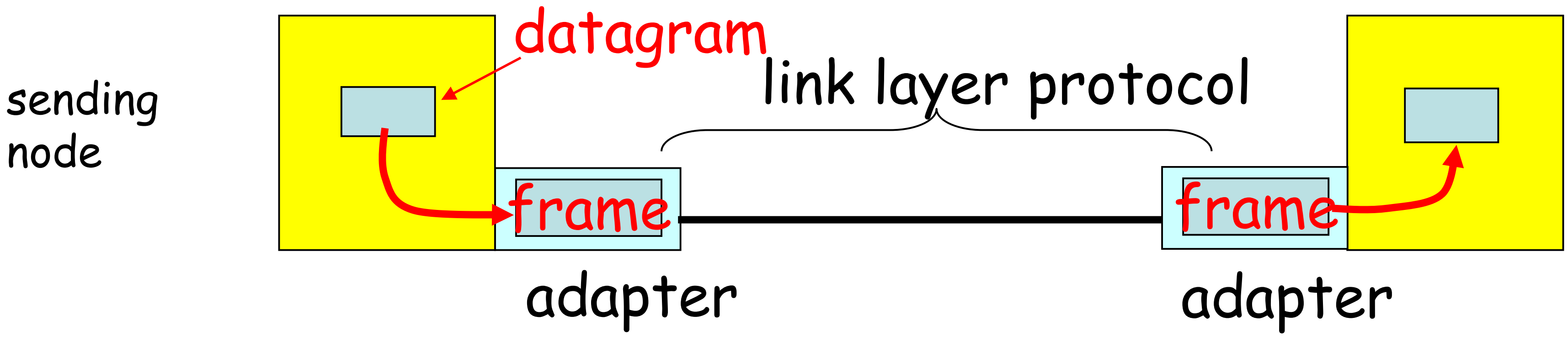


Broadcast address = FF-FF-FF-FF-FF-FF

= adapter

Ethernet Address

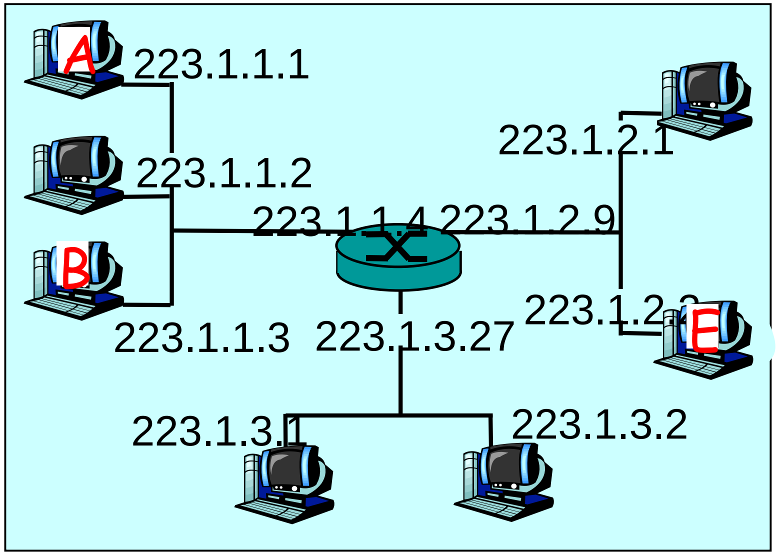
- Ethernet is the “dominant” LAN technology
  - Cheap
  - Simple
  - Kept up with speed race: 10 Mbps – 10 Gbps
- Ethernet address is 6 bytes long, the first 24 bits are fixed for a certain company.

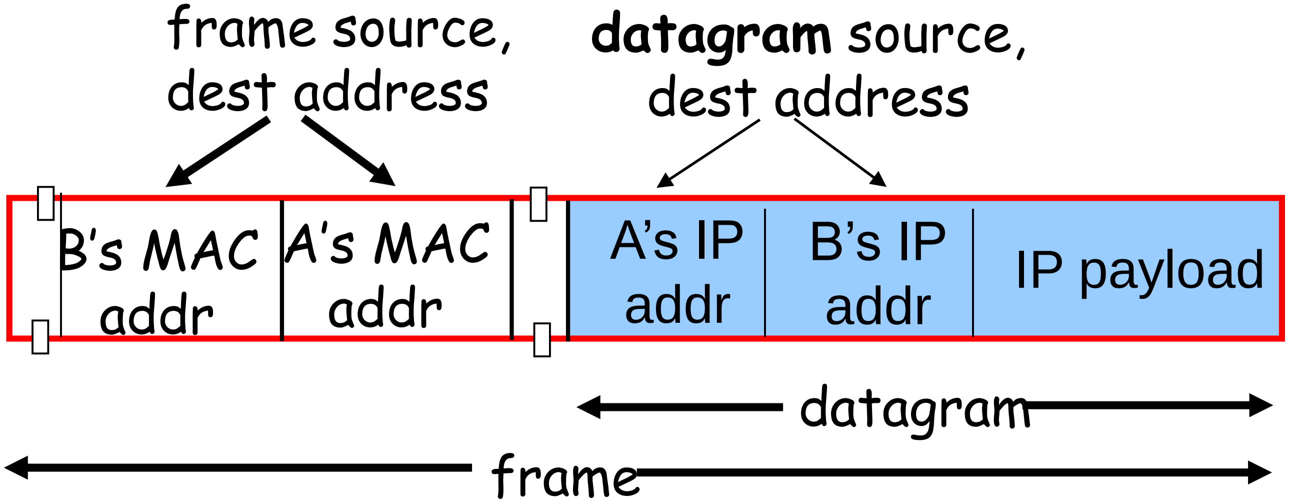


Back to Routing Decisions

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



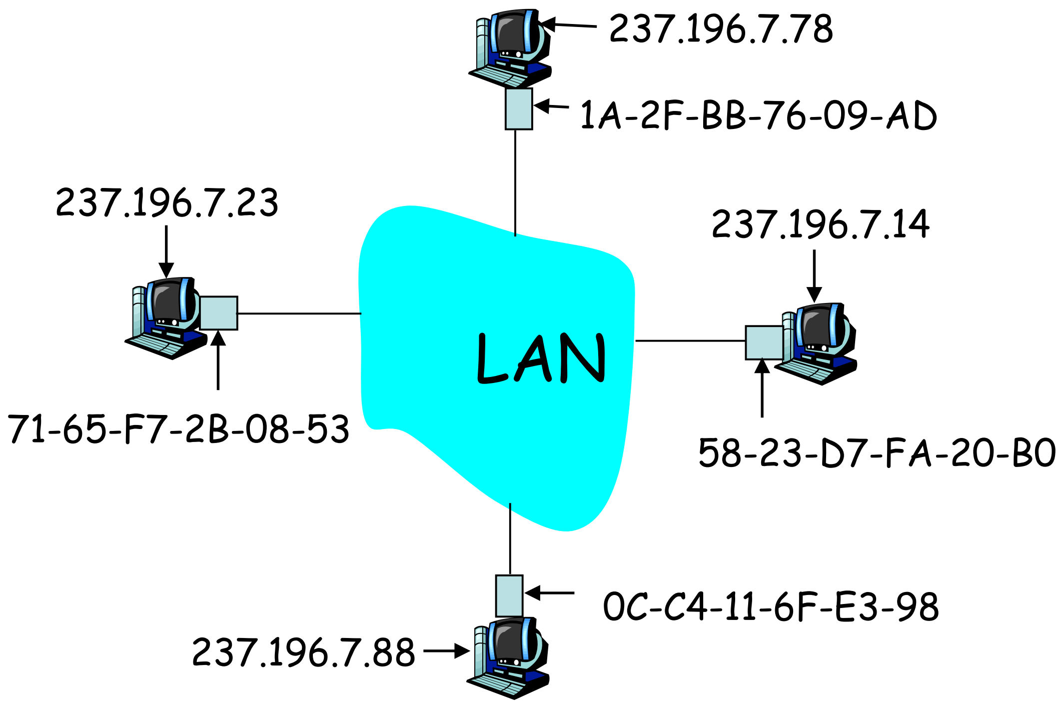


Back to Routing Decisions

- If adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
- Otherwise, adapter discards frame

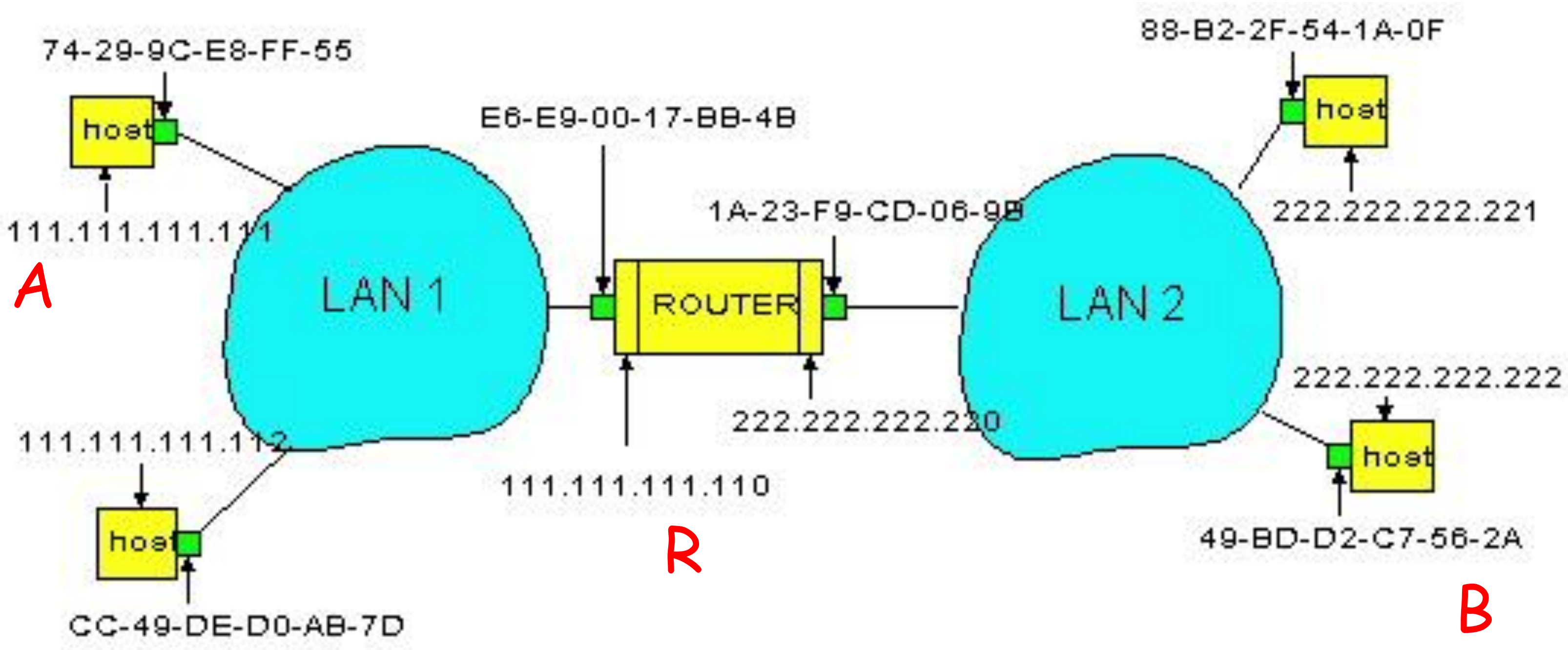
Routing to the Same LAN

- **A** broadcasts ARP query packet, containing **B**'s IP address
  - Dest 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)

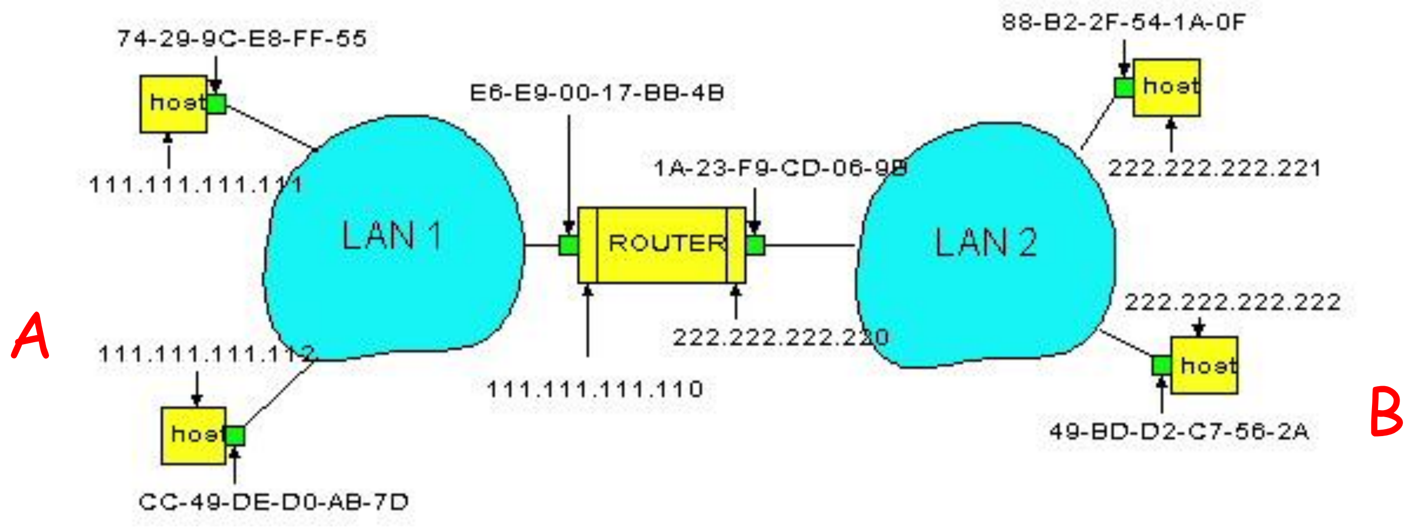


Routing to another LAN

send datagram from **A** to **B** via R (**A** knows **B** IP address)

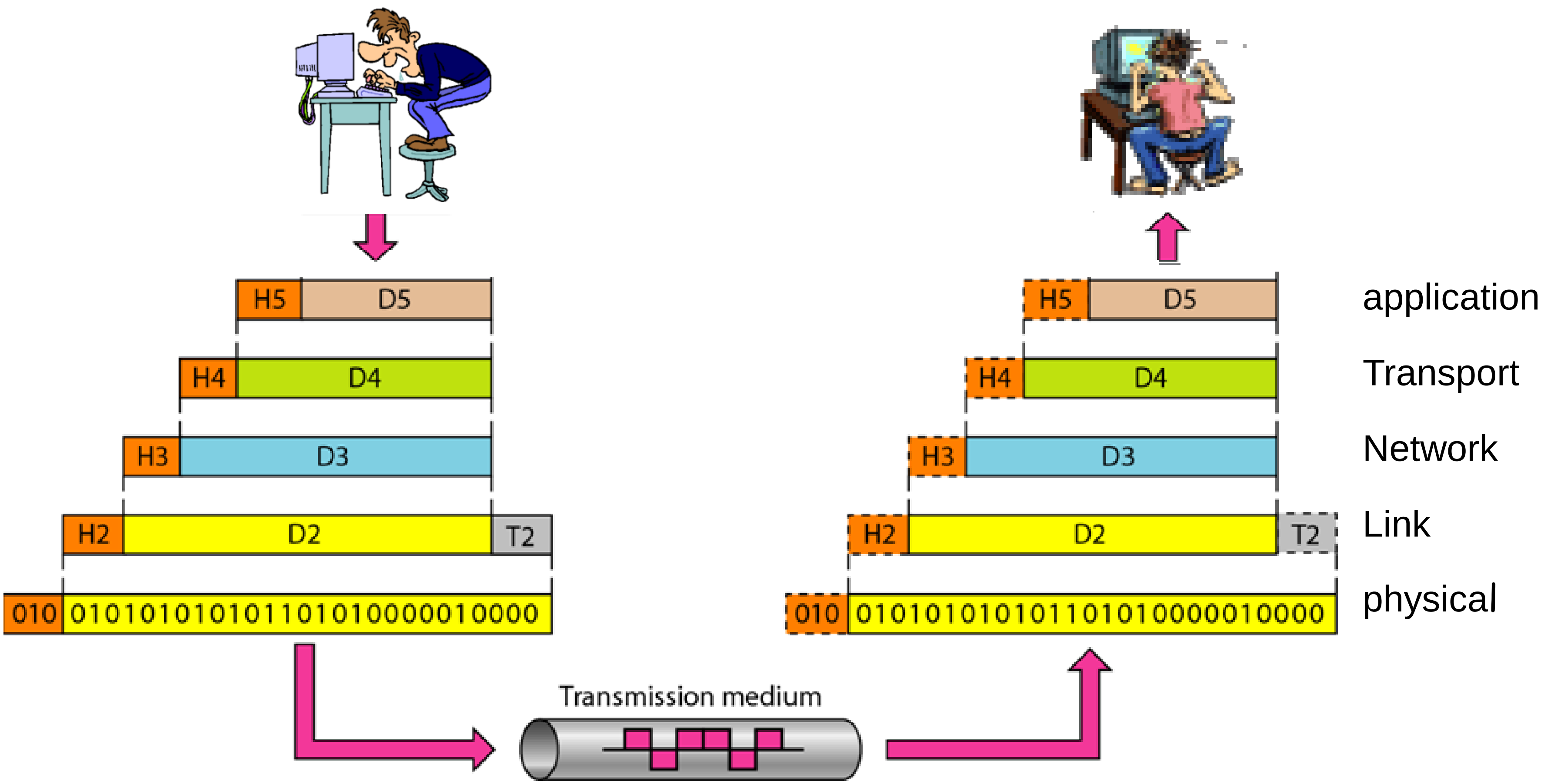


- Two ARP tables in router R, one for each IP network (LAN)
- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B



- 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 adapter sends frame
- R’s adapter receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B’s MAC address
- R creates frame containing A-to-B IP datagram sends to B

Networking: A Top-Down Approach



The physical layer is responsible for movements of individual bits from one hop (node) to the next.

To be transmitted, data must be transformed to electromagnetic signals.

Data can be analog or digital.
Analog data are continuous and take continuous values.
Digital data have discrete states and take discrete values.

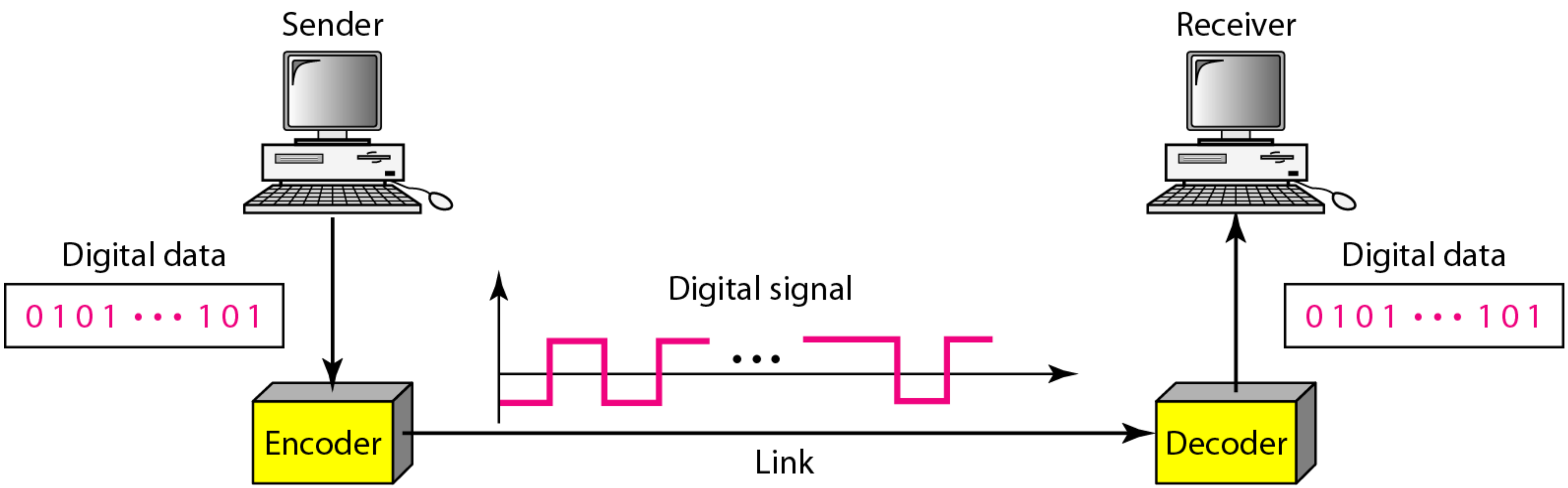
Digital Transmission

- How can we represent digital data by using digital signals ?

↓

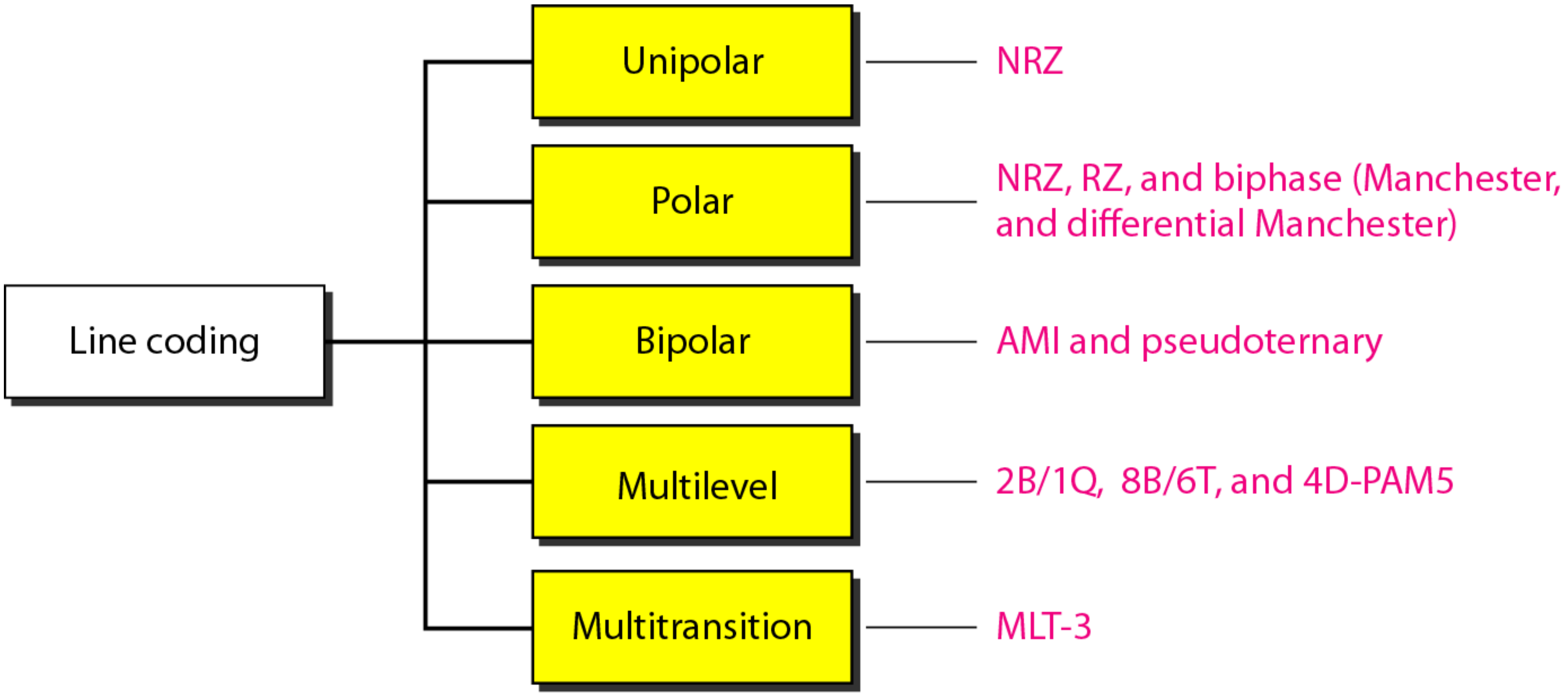
Line coding

Line Coding

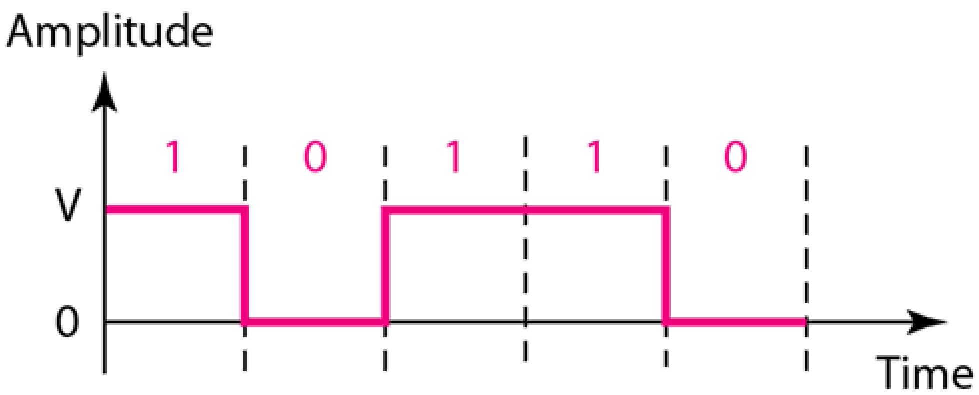


- **Line coding**
  - converting binary data to digital signal

Line coding schemes

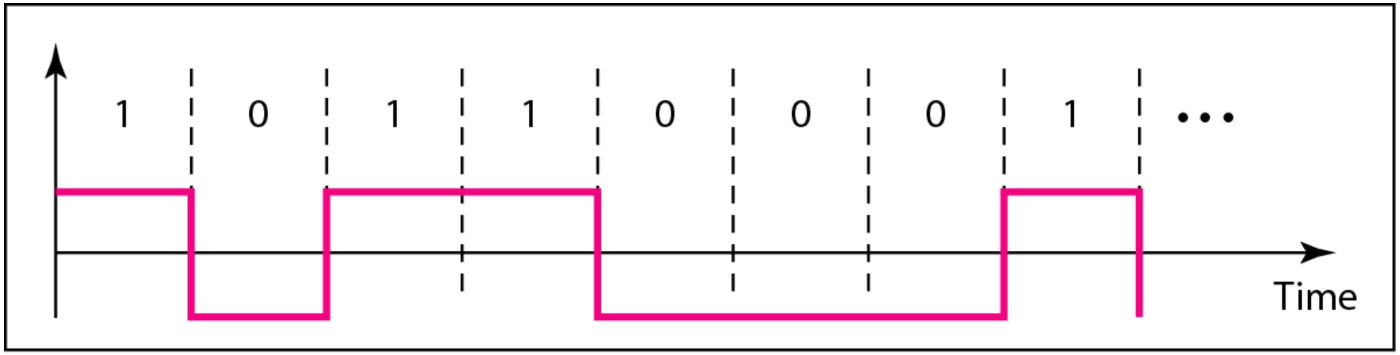


Unipolar scheme

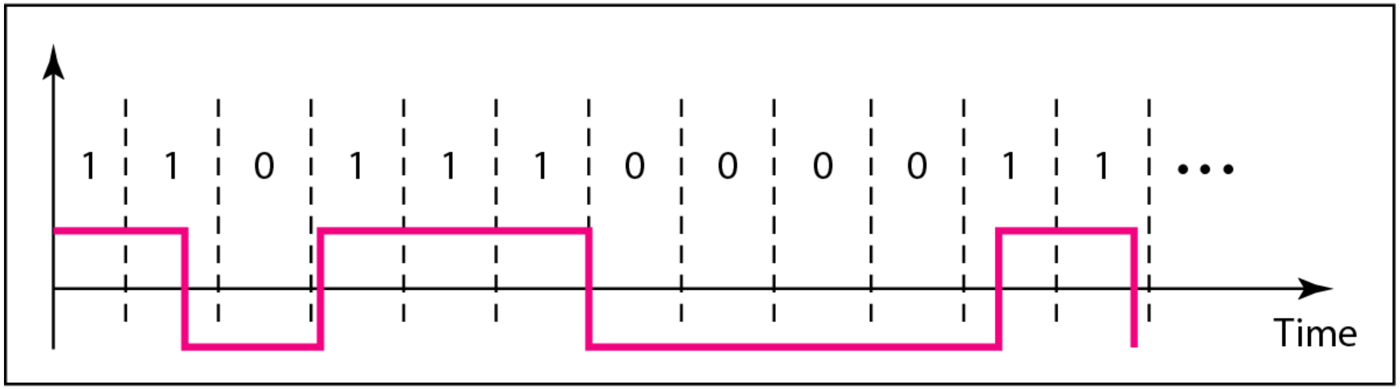


- Unipolar uses only **one polarity** (voltage level), assigned to one of the 2 binary states (usually to represent 1), **zero** voltage to represent the other state.
- Very simple and almost obsolete.
- **Problems:** lack of synchronization, DC component (average voltage level is nonzero)

Lack of synchronization



a. Sent



b. Received

DC Component

- Average voltage level is nonzero. Leading to undesirable effect:
  - When passing through some systems, signal is distorted causing errors in the output signal
  - Extra energy residing on the line and is useless

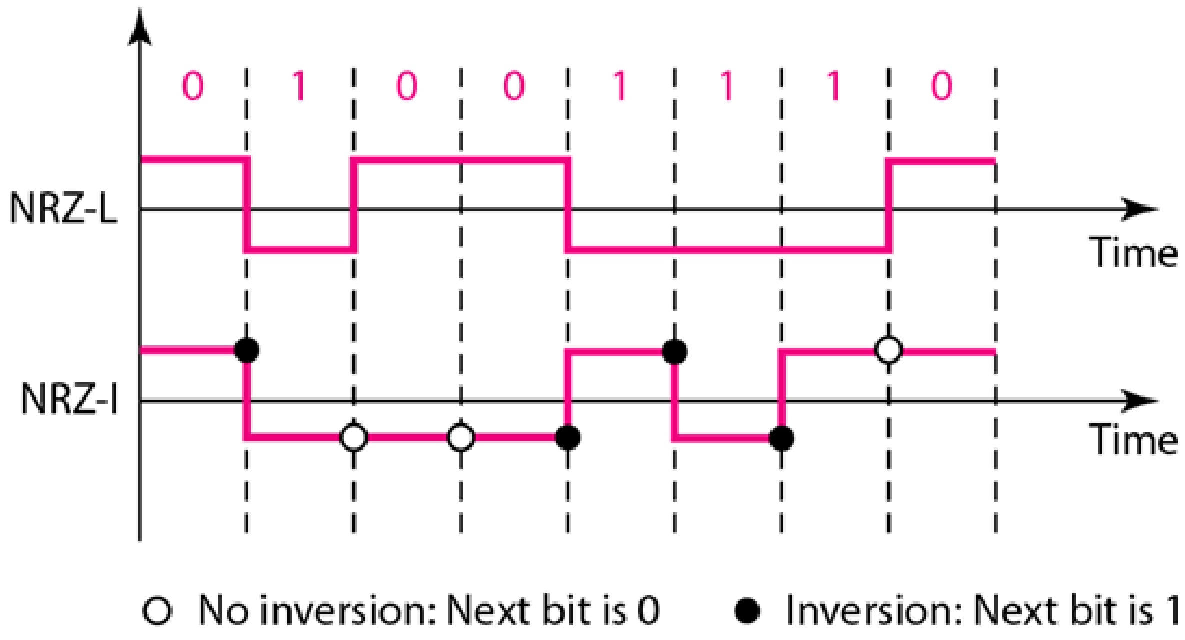
A positive voltage should be canceled by a negative voltage

Polar Scheme

- Uses **two** voltage levels, positive and negative.
- Average voltage level is reduced and DC component problem alleviated

- **Polar schemes:**
  - NRZ (Non-Return-to Zero)
  - RZ (Return-to-Aero)
  - Manchester
  - Differential Manchester

Polar NRZ-L and NRZ-I schemes

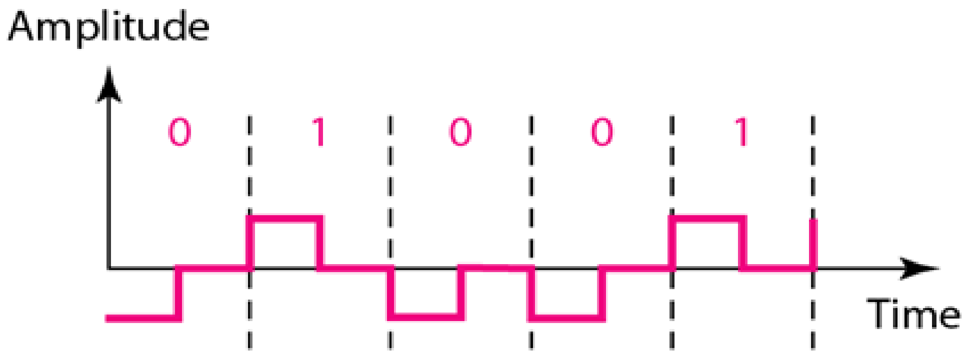


- Signal Level depends on the bit represented. +ve voltage for 0, -ve for 1
- Problems if long streams of 0’s or 1’s

- Inversion of the voltage level represents 1 (signal inverted each time 1 is encountered).
- No change in voltage level represents 0.

In NRZ-L the level of the voltage determines the value of the bit.
In NRZ-I the inversion or the lack of inversion determines the value of the bit.

Polar RZ scheme



- Uses **3** values: +ve (for 1), -ve (for 0) and 0.
- Signal changes half way through each bit interval
- 1 bit represented by +ve to 0
- 0 bit represented by –ve to 0
- Signal transition is used for synchronization

End of Course

Good Luck with the final