CSMC 417

Computer Networks

Prof. Ashok K Agrawala

PAN Technologies and Standards

- IEEE has assigned the number 802.15 to PAN standards
- Several task groups and industry consortia have been formed for each of the key PAN technologies

Standard	Purpose
802.15.1a	Bluetooth technology (1 Mbps; 2.4 GHz)
802.15.2	Coexistence among PANs (noninterference)
802.15.3	High rate PAN (55 Mbps; 2.4 GHz)
802.15.3a	Ultra Wideband (UWB) high rate PAN (110 Mbps; 2.4 GHz)
802.15.4	Zigbee technology – low data rate PAN for remote control
802.15.4a	Alternative low data rate PAN that uses low power

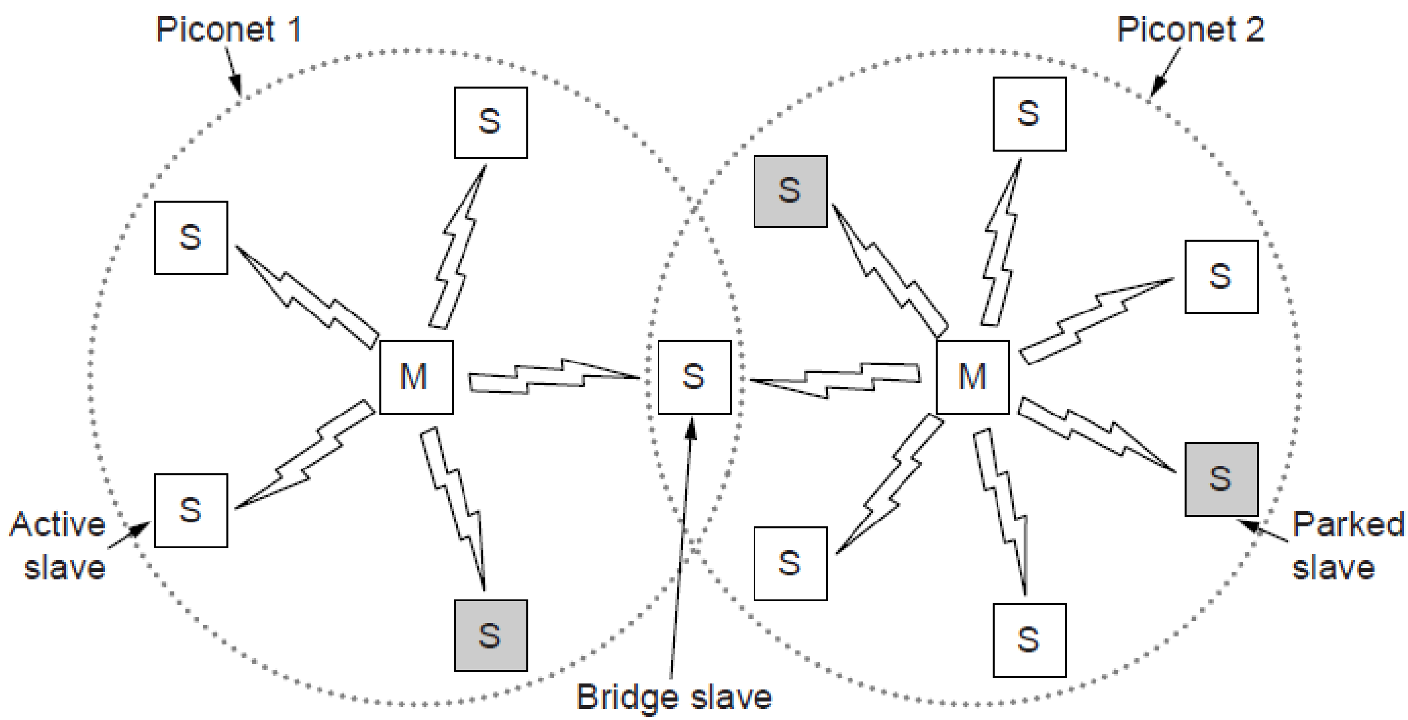
PAN Technologies and Standards

- Bluetooth
  - The IEEE 802.15.1a standard evolved after vendors created Bluetooth technology as a short-distance wireless connection technology
- The characteristics of Bluetooth technology are:
  - Wireless replacement for cables (e.g., headphones or mouse)
  - Uses 2.4 GHz frequency band
  - Short distance (up to 5 meters, with variations that extend the range to 10 or 50 meters)
  - Device is master or slave
  - Master grants permission to slave
  - Data rate is up to 721 Kbps

Bluetooth Architecture

Piconet master is connected to slave wireless devices

- Slaves may be asleep (parked) to save power



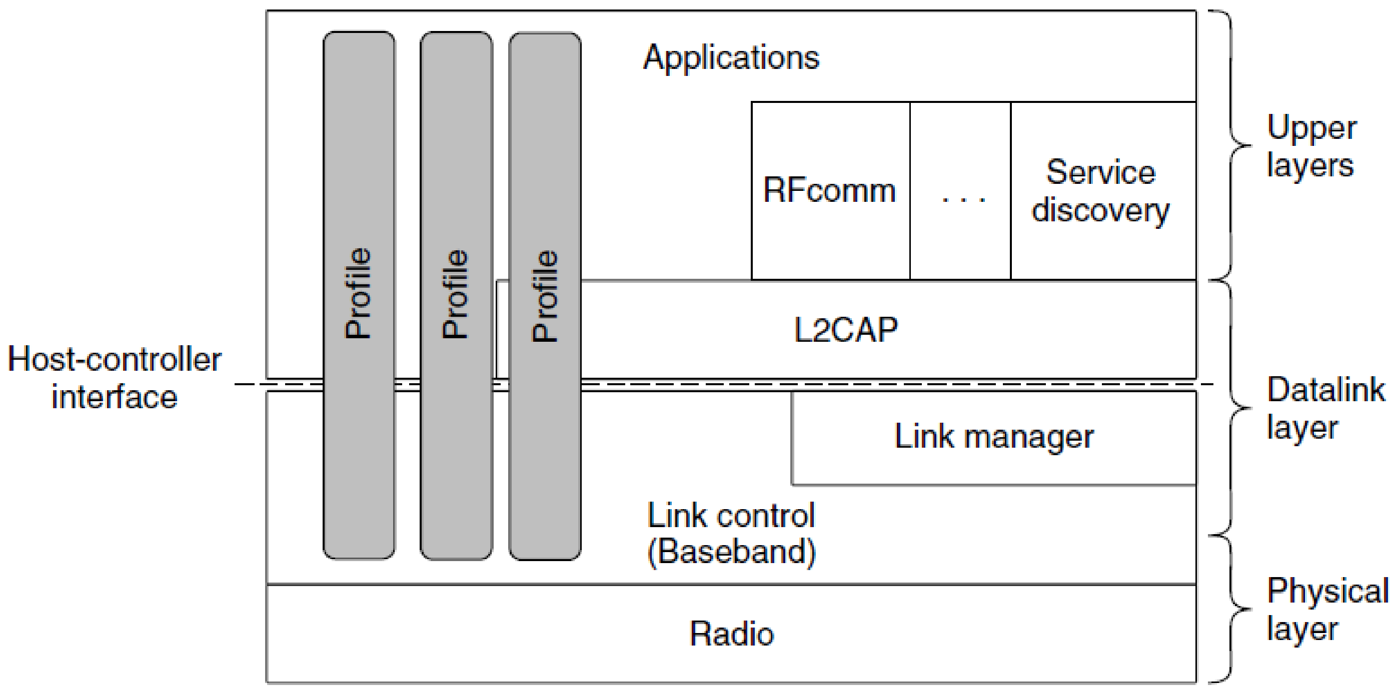
Bluetooth Applications

Name	Description
Generic access	Procedures for link management
Service discovery	Protocol for discovering offered services
Serial port	Replacement for a serial port cable
Generic object exchange	Defines client-server relationship for object movement
LAN access	Protocol between a mobile computer and a fixed LAN
Dial-up networking	Allows a notebook computer to call via a mobile phone
Fax	Allows a mobile fax machine to talk to a mobile phone
Cordless telephony	Connects a handset and its local base station
Intercom	Digital walkie-talkie
Headset	Intended for hands-free voice communication
Object push	Provides a way to exchange simple objects
File transfer	Provides a more general file transfer facility
Synchronization	Permits a PDA to synchronize with another computer

Bluetooth Applications / Protocol Stack

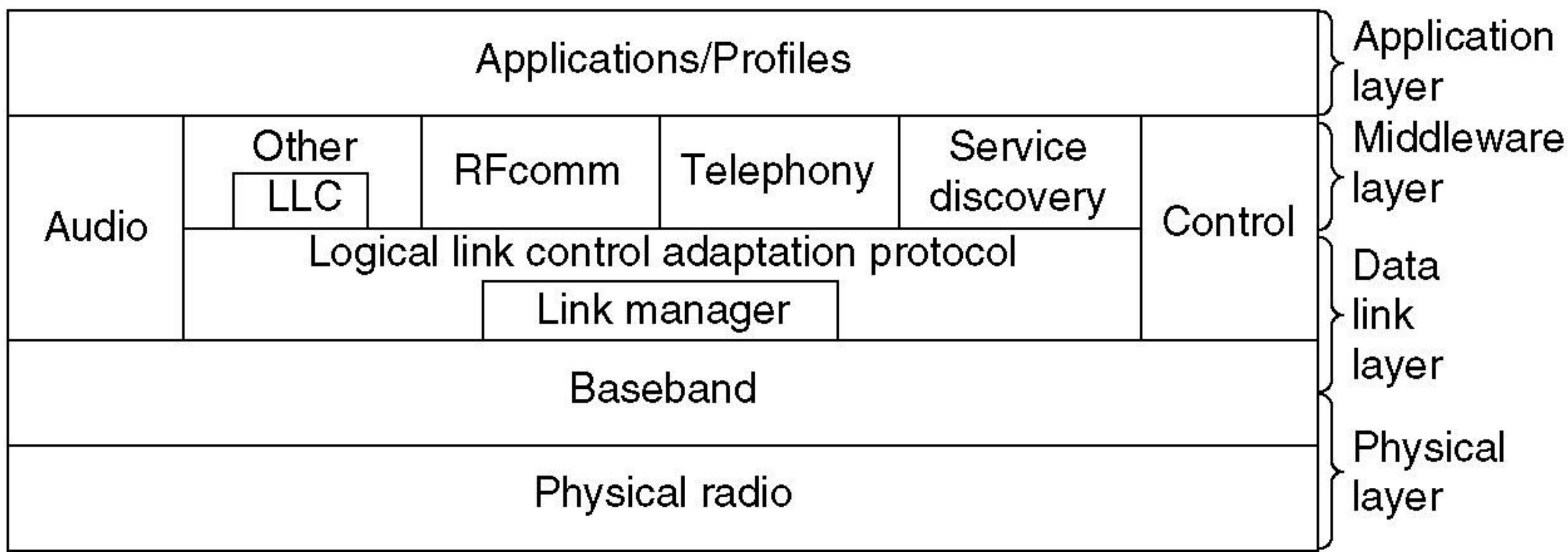
Profiles give the set of protocols for a given application

- 25 profiles, including headset, intercom, streaming audio, remote control, personal area network, …



The Bluetooth Protocol Stack

The 802.15 version of the Bluetooth protocol architecture.



Bluetooth Radio / Link Layers

Radio layer

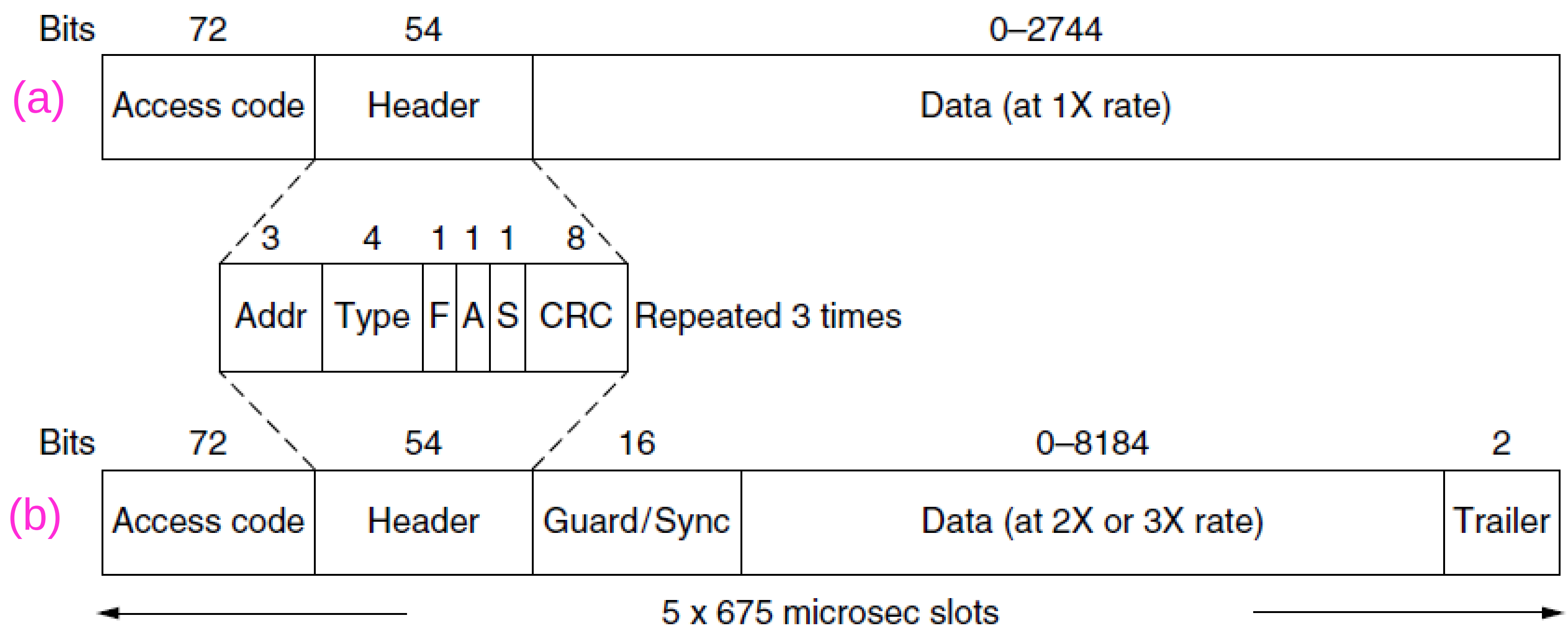
- Uses adaptive frequency hopping in 2.4 GHz band

Link layer

- TDM with timeslots for master and slaves
- Synchronous CO for periodic slots in each direction
- Asynchronous CL for packet-switched data
- Links undergo pairing (user confirms passkey/PIN) to authorize them before use

Bluetooth Frames

Time is slotted; enhanced data rates send faster but for the same time; addresses are only 3 bits for 8 devices



(a) Basic rate data frame (b) Enhanced rate data frame

PAN Technologies and Standards

- Ultra Wideband (UWB)
  - The idea behind UWB communication is that spreading data across many frequencies
    - requires less power to reach the same distance
- The key characteristics of UWB are:
  - Uses wide spectrum of frequencies
  - Consumes very low power
  - Short distance (2 to 10 meters)
  - Signal permeates obstacles such as walls
  - Data rate of 110 at 10 meters, and up to 500 Mbps at 2 meters
  - IEEE unable to resolve disputes and form a single standard

PAN Technologies and Standards

- Zigbee
  - The Zigbee standard (802.15.4) arose from a desire to standardize wireless remote control technology
    - especially for industrial equipment
  - Because remote control units only send short command
    - high data rates are not required
- The chief characteristics of Zigbee are:
  - Wireless standard for remote control, not data
  - Target is industry as well as home automation
  - Three frequency bands used (868 MHz, 915 MHz, and 2.4 GHz)
  - Data rate of 20, 40, or 250 Kbps, depending on frequency band
  - Low power consumption
  - Three levels of security being defined

Other Short-Distance Communication Technologies

- Two other wireless technologies provide communication over short distances, but they are not listed under PANs
  - InfraRED technologies provide control and low-speed data communications
  - RFID technologies are used with sensors

Other Short-Distance Communication Technologies

- InfraRED
  - InfraRED technology is often used in remote controls
    - and may be used as a cable replacement (e.g., for a wireless mouse)
  - The Infrared Data Association (IrDA) has produced a set of standards that are widely accepted
- The chief characteristics of the IrDA technology are:
  - Family of standards for various speeds and purposes
  - Practical systems have range of one to several meters
  - Directional transmission with a cone covering 30
  - Data rates between 2.4 Kbps (control) and 16 Mbps (data)
  - Generally low power consumption with very-low power versions
  - Signal may reflect from surfaces
    - but cannot penetrate solid objects

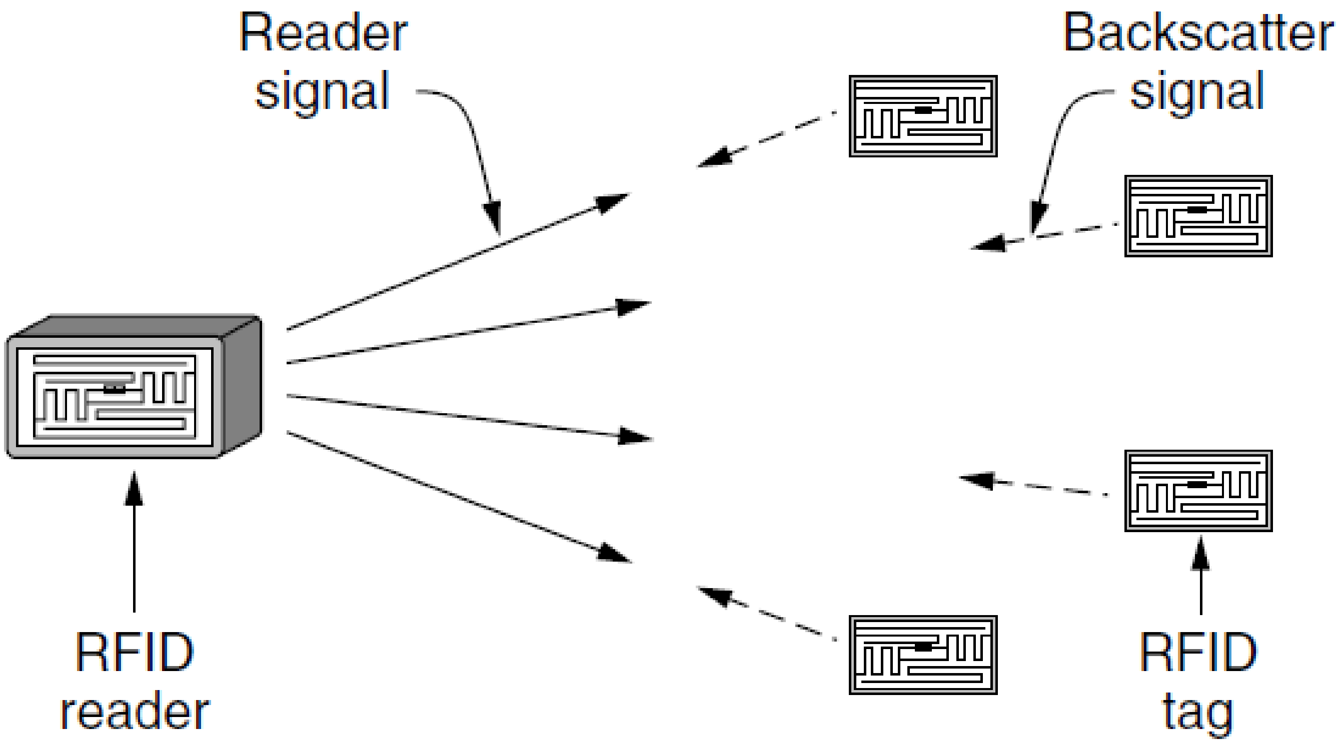
Other Short-Distance Communication Technologies

- Radio Frequency Identification (RFID)
  - RFID technology uses an interesting form of wireless communication to create a mechanism
  - A small tag contains identification information
    - that a receiver can “pull” from the tag
- Some features of RFID:
  - Over 140 RFID standards exist for a variety of applications
  - Passive RFIDs draw power from the signal sent by the reader
  - Active RFIDs contain a battery
    - which may last up to 10 years
  - Limited distance
    - although active RFIDs extend farther than passive
  - Can use frequencies from less than 100 MHz to 868-954 MHz
  - Used for
    - inventory control, sensors, passports, and other applications

RFID

- EPC Gen 2 architecture
- EPC Gen 2 physical layer
- EPC Gen 2 tag identification layer
- Tag identification message formats

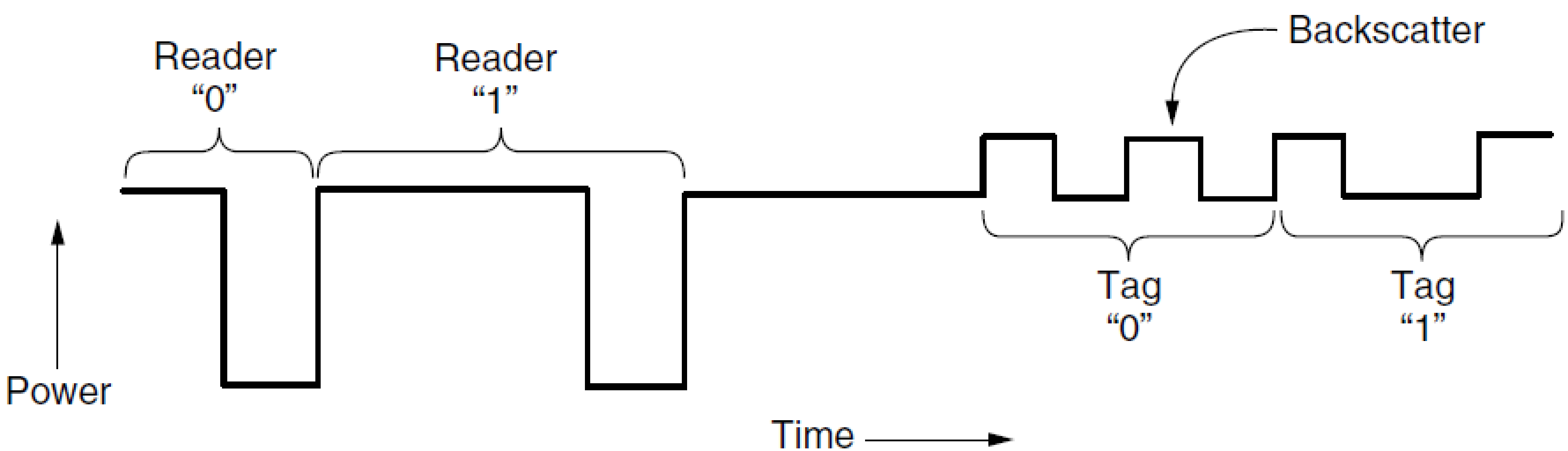
EPC Gen 2 Architecture



RFID architecture.

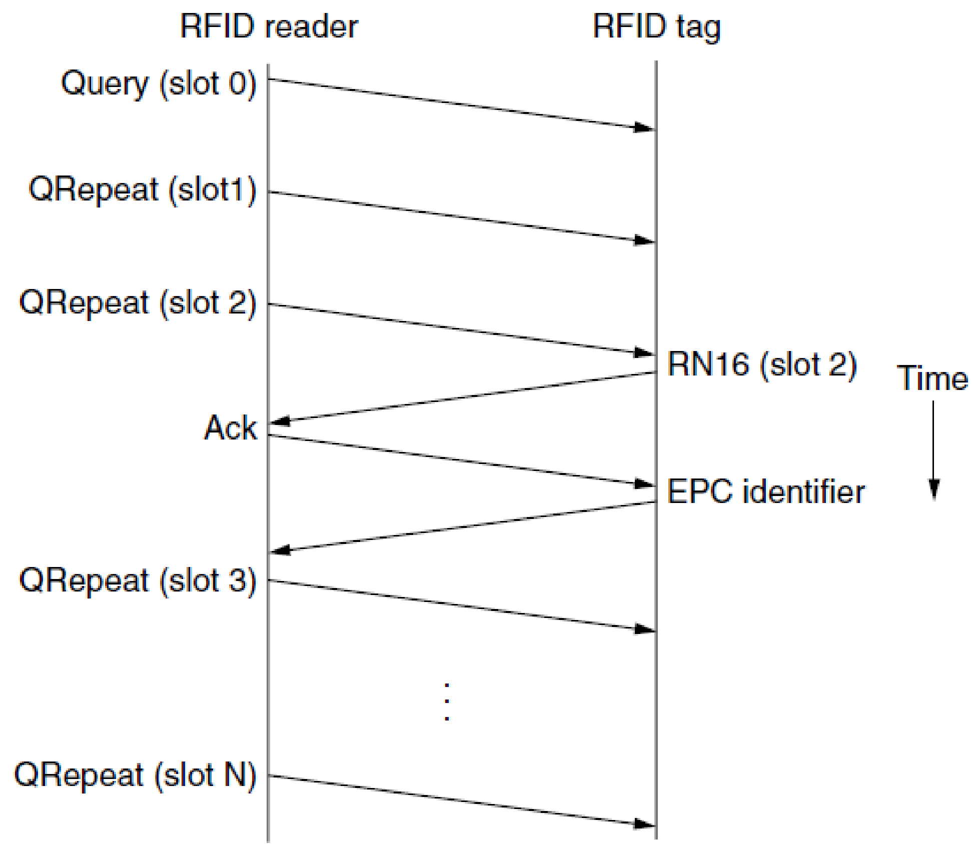
Gen 2 Physical Layer

- Reader uses duration of on period to send 0/1
- Tag backscatters reader signal in pulses to send 0/1



Gen 2 Tag Identification Layer

- Reader sends query and sets slot structure
- Tags reply (RN16) in a random slot; may collide
- Reader asks one tag for its identifier (ACK)
- Process continues until no tags are left



Gen 2 Frames

- Reader frames vary depending on type (Command)
  - Query shown below, has parameters and error detection
- Tag responses are simply data
  - Reader sets timing and knows the expected format

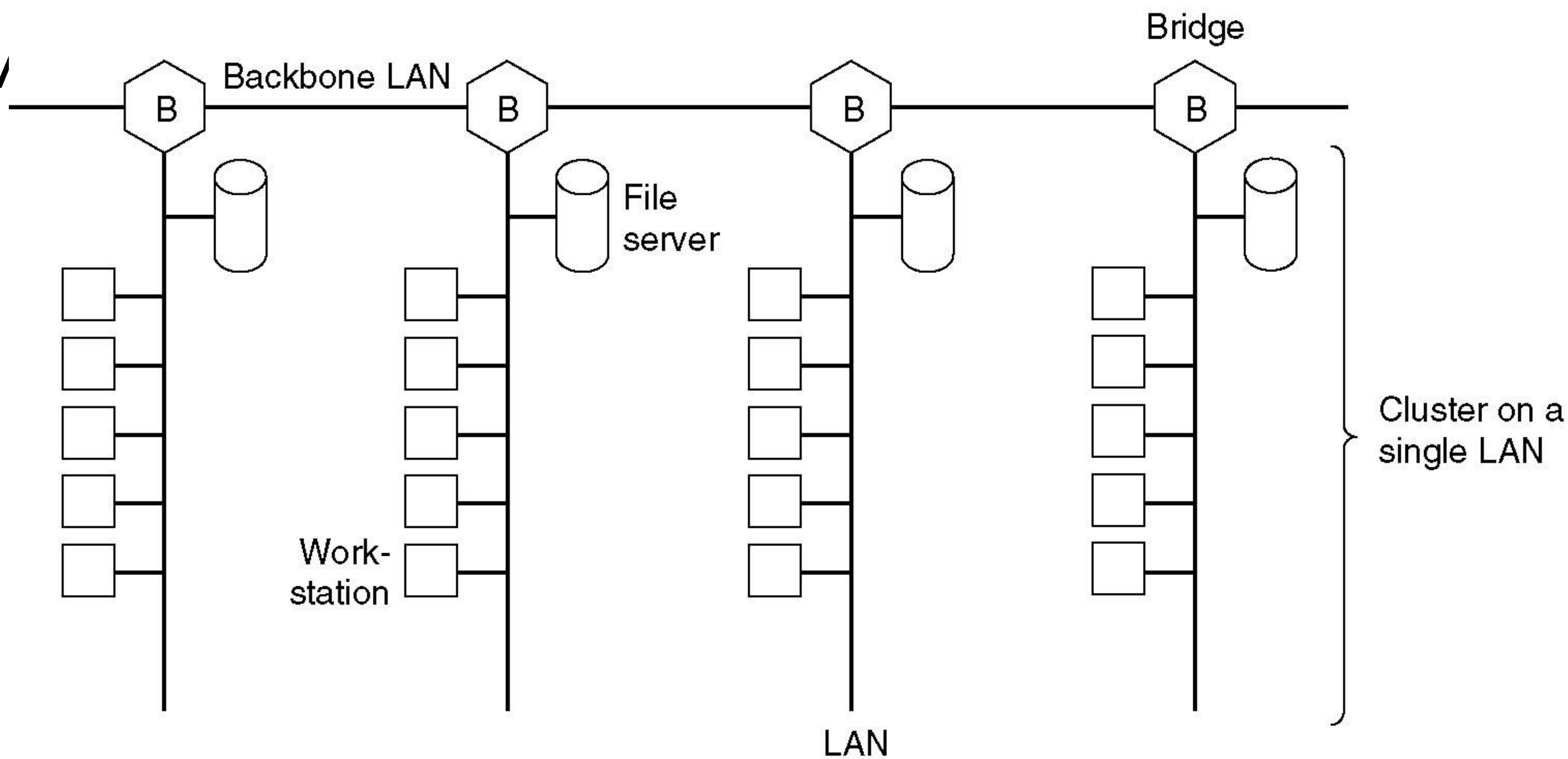
Bits	4	1	2	1	2	2	1	4	5
	Command 1000	DR	M	TR	Sel	Session	Target	Q	CRC
		Physical parameters			Tag selection				

Query message

Data Link Layer Switching

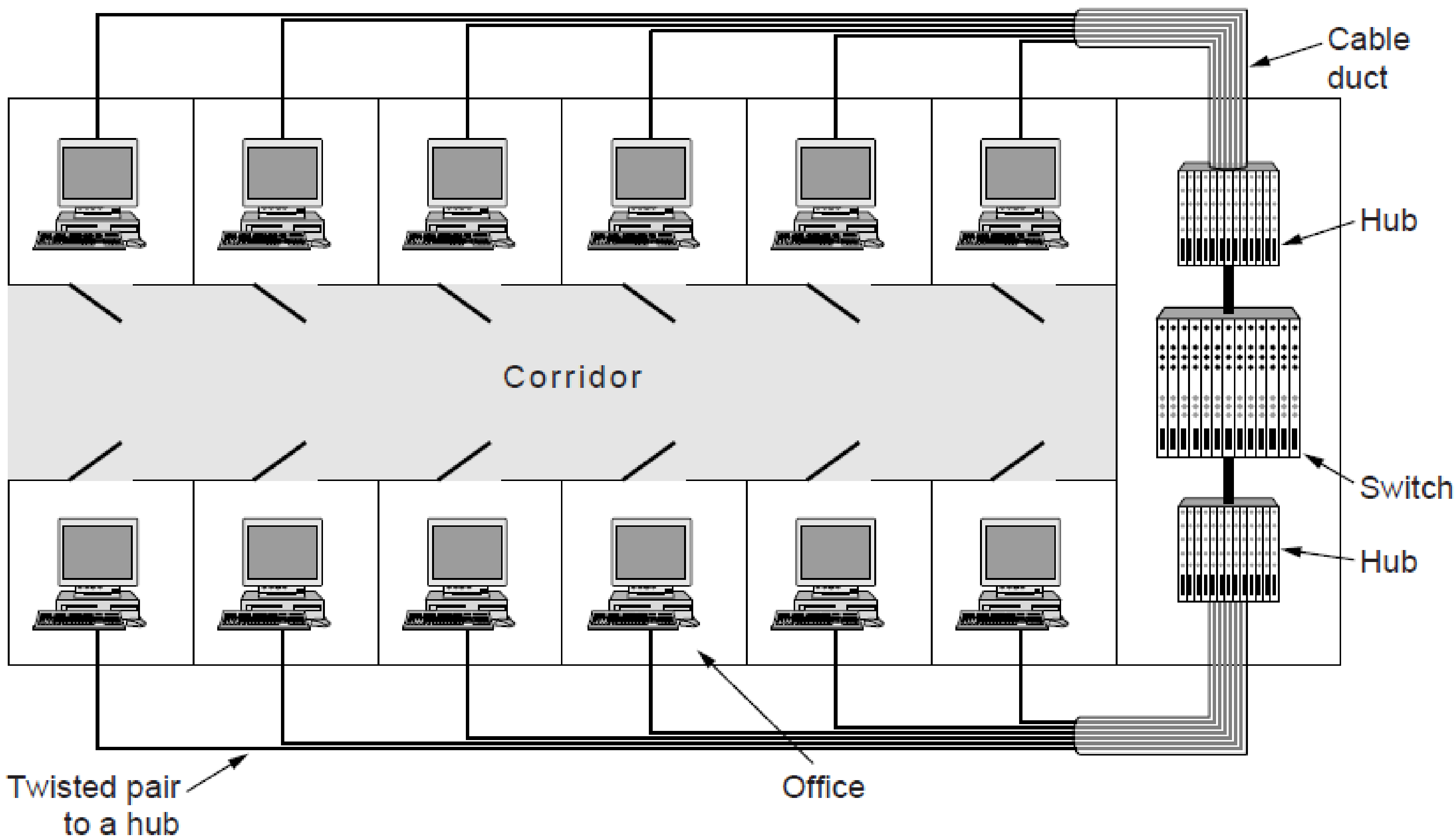
- Bridges from 802.x to 802.y
- Local Internetworking
- Spanning Tree Bridges
- Remote Bridges
- Repeaters, Hubs, Bridges, Switches, Routers, Gateways
- Virtual LANs

Data Link Layer Switching



Uses of Bridges

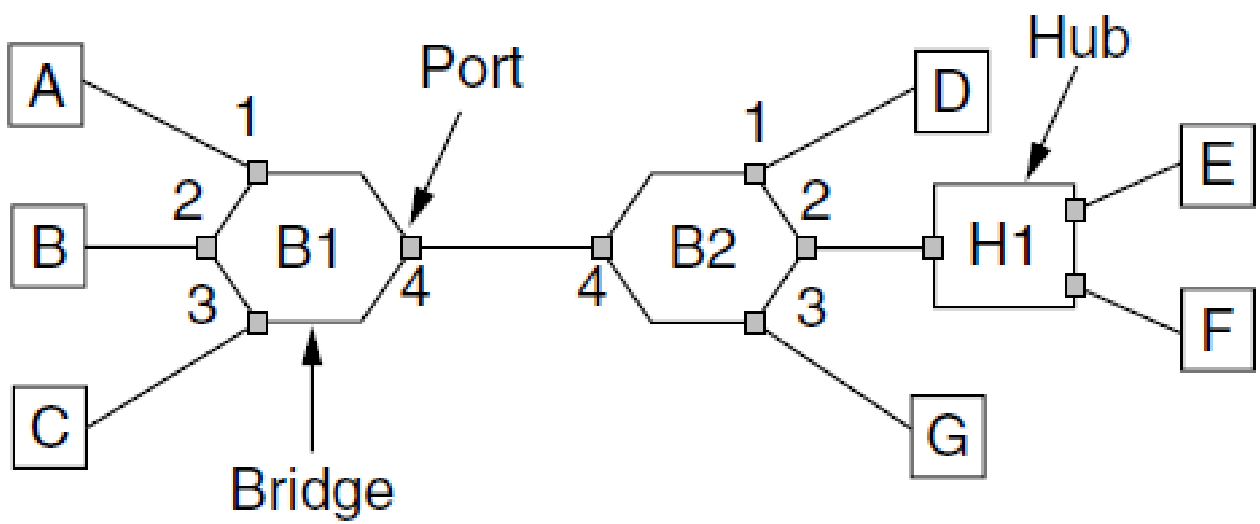
- Common setup is a building with centralized wiring
  - Bridges (switches) are placed in or near wiring closets



Learning Bridges

A bridge operates as a switched LAN (not a hub)

- Computers, bridges, and hubs connect to its ports



Learning Bridges

Backward learning algorithm picks the output port:

- Associates source address on frame with input port
- Frame with destination address sent to learned port
- Unlearned destinations are sent to all other ports

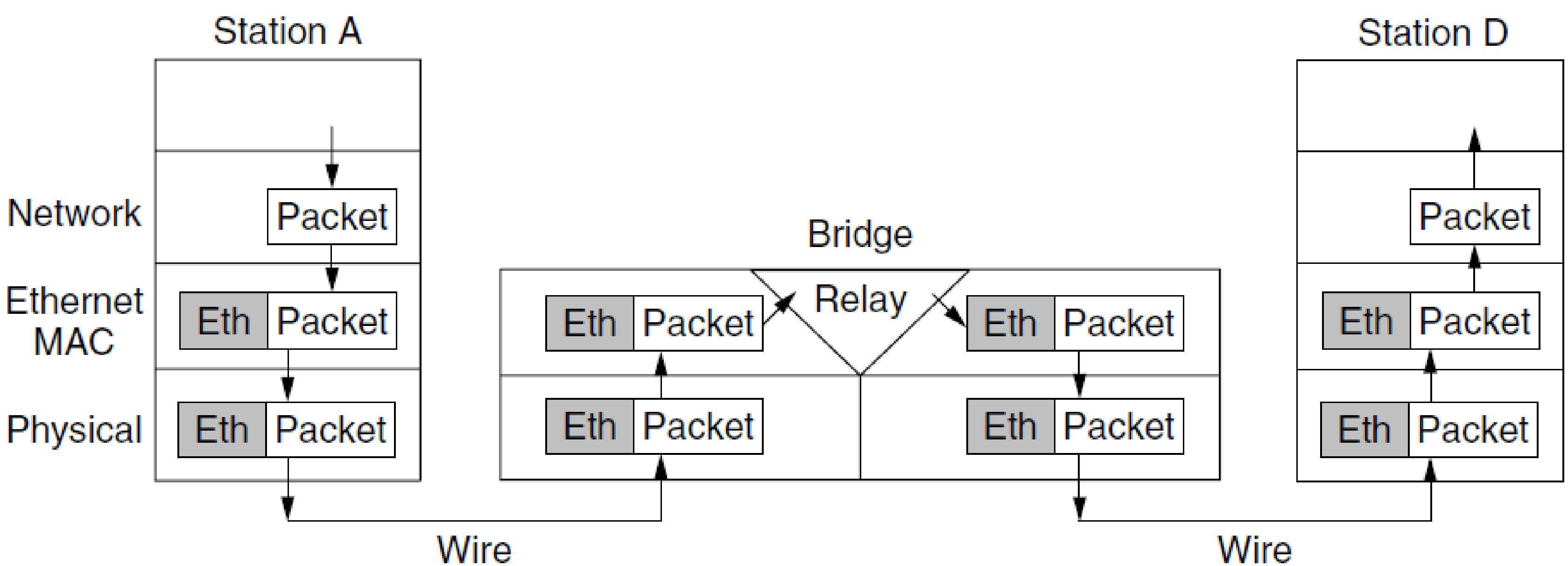
Needs no configuration

- Forget unused addresses to allow changes
- Bandwidth efficient for two-way traffic

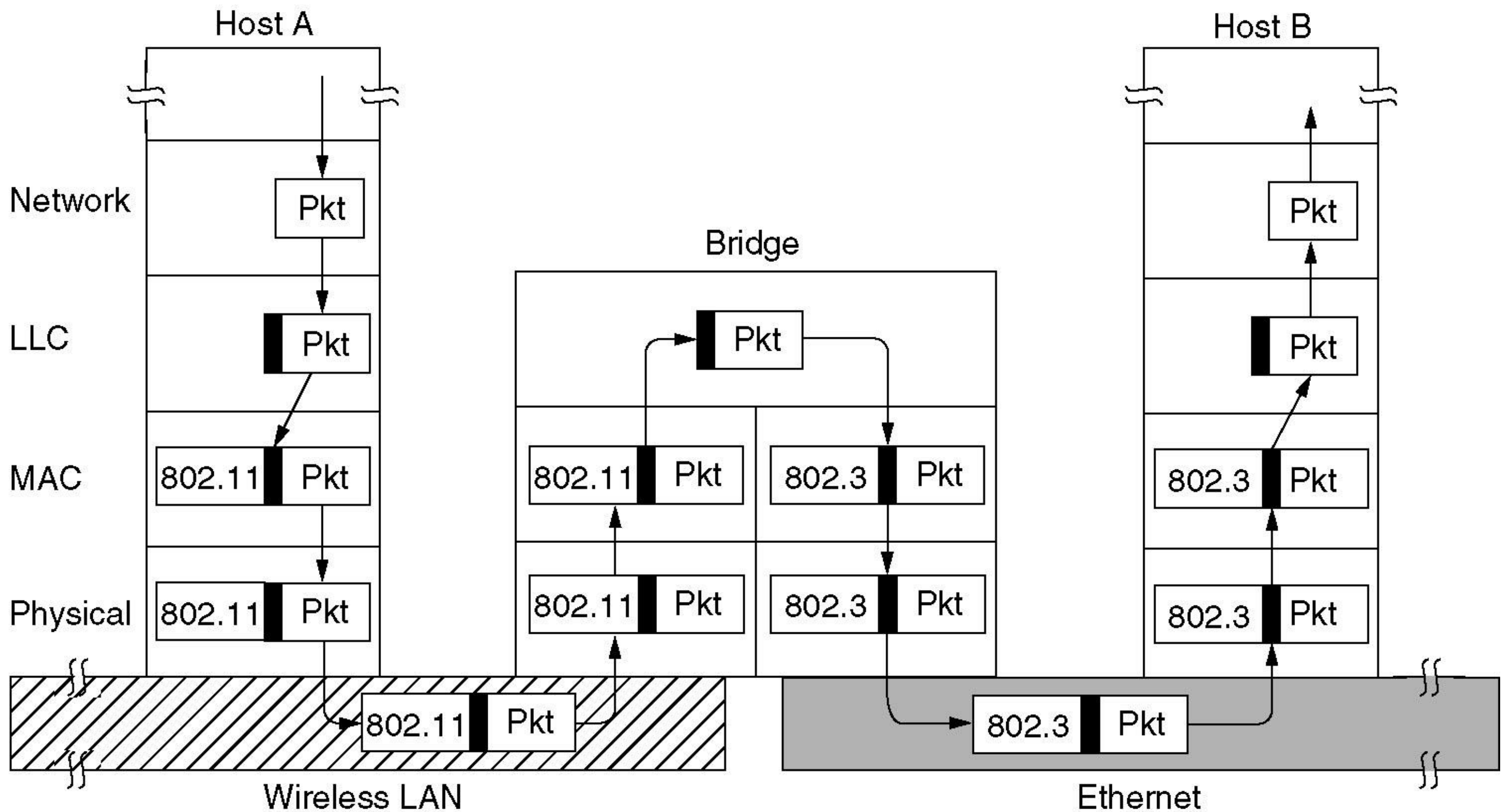
Learning Bridges

Bridges extend the Link layer:

- Use but don't remove Ethernet header/addresses
- Do not inspect Network header

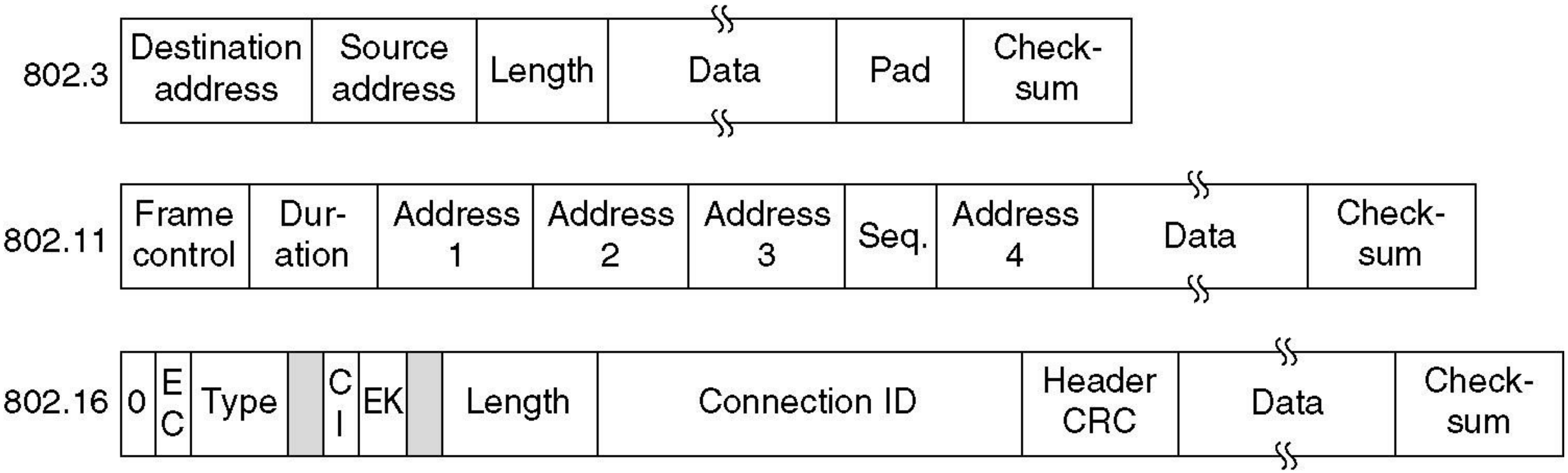


Bridges from 802.x to 802.y



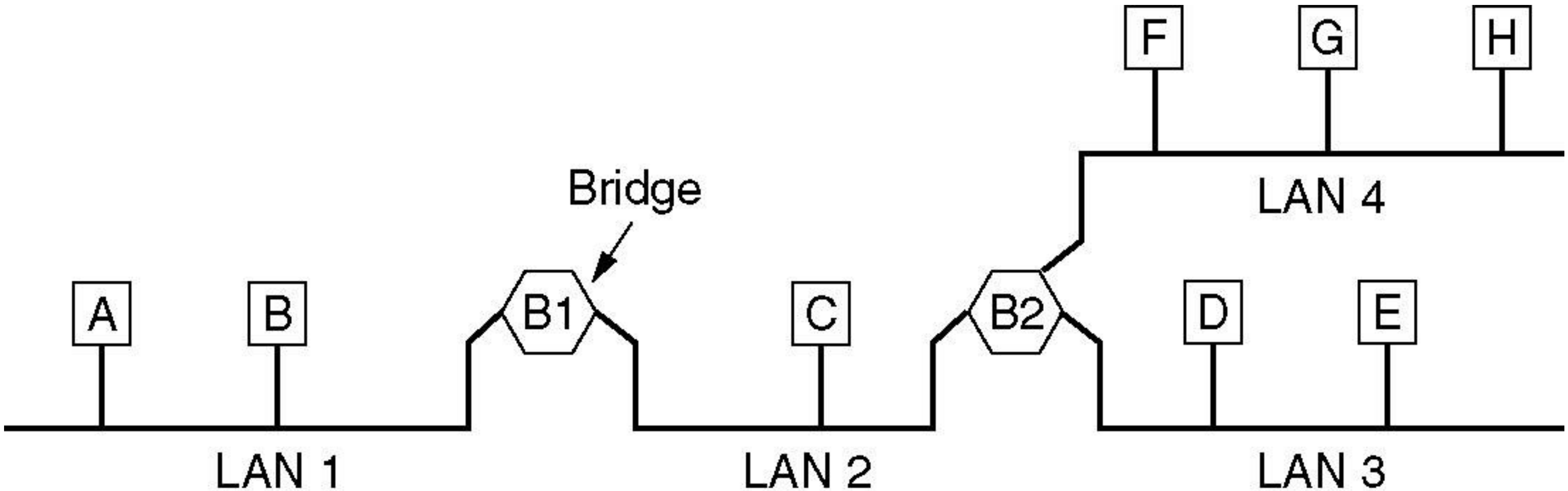
Bridges from 802.x to 802.y (2)

The IEEE 802 frame formats. The drawing is not to scale.

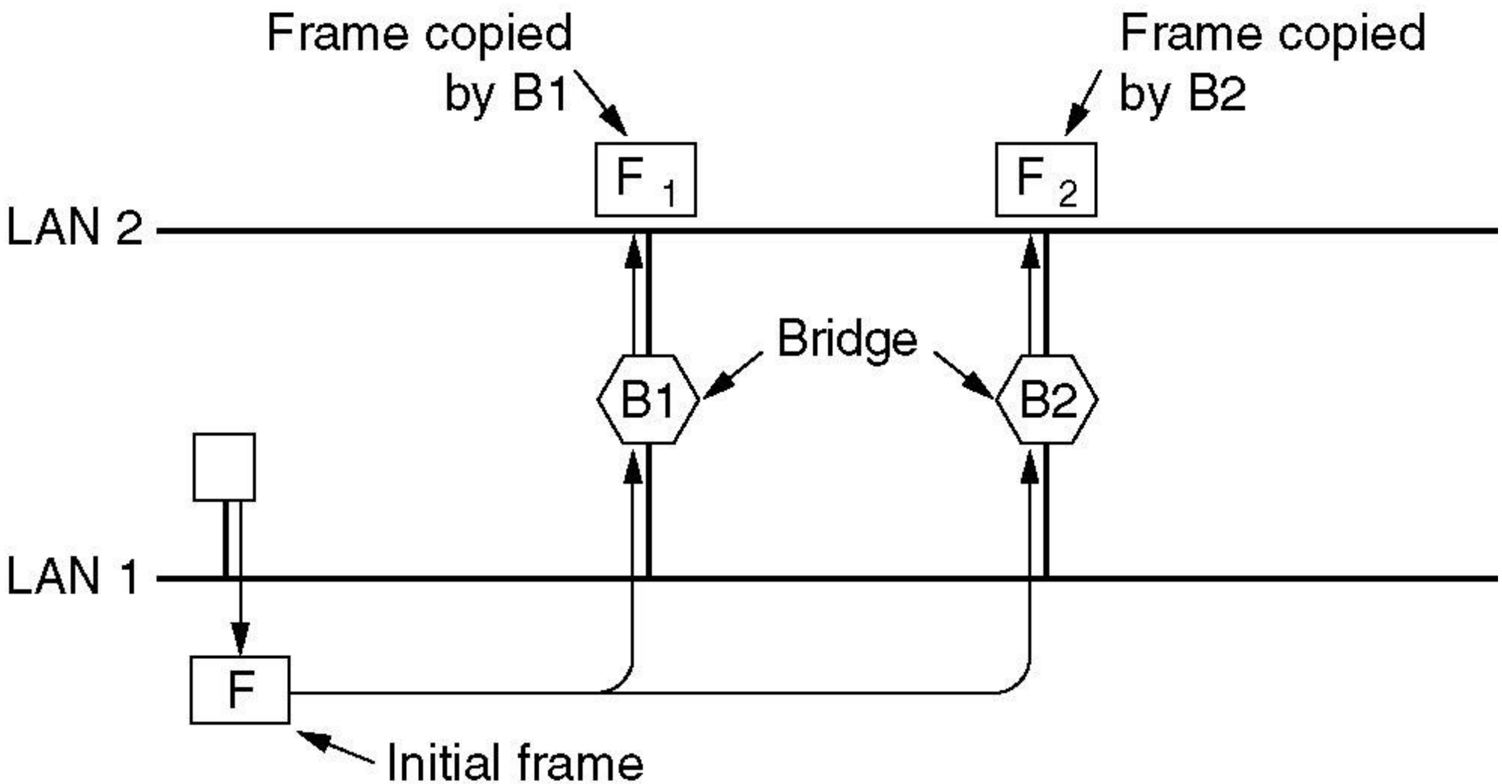


Local Internetworking

A configuration with four LANs and two bridges.



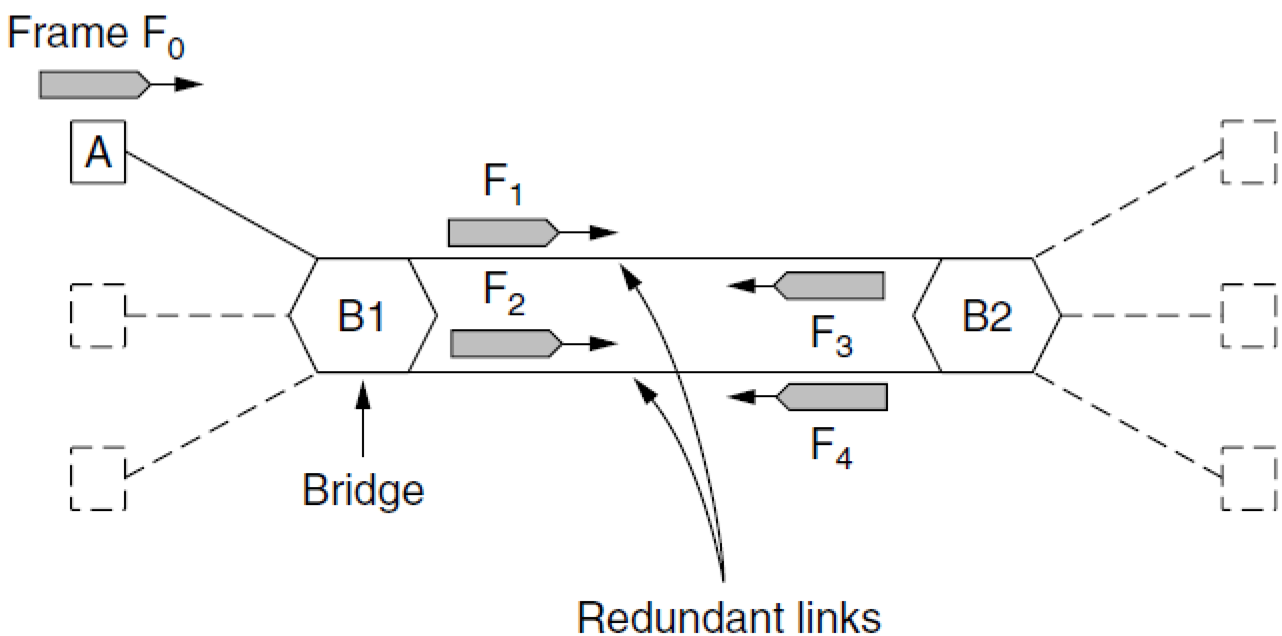
Spanning Tree Bridges



Spanning Tree (1) – Problem

Bridge topologies with loops and only backward learning will cause frames to circulate for ever

- Need spanning tree support to solve problem



Spanning Tree (2) – Algorithm

- Subset of forwarding ports for data is use to avoid loops
- Selected with the spanning tree distributed algorithm by Perlman

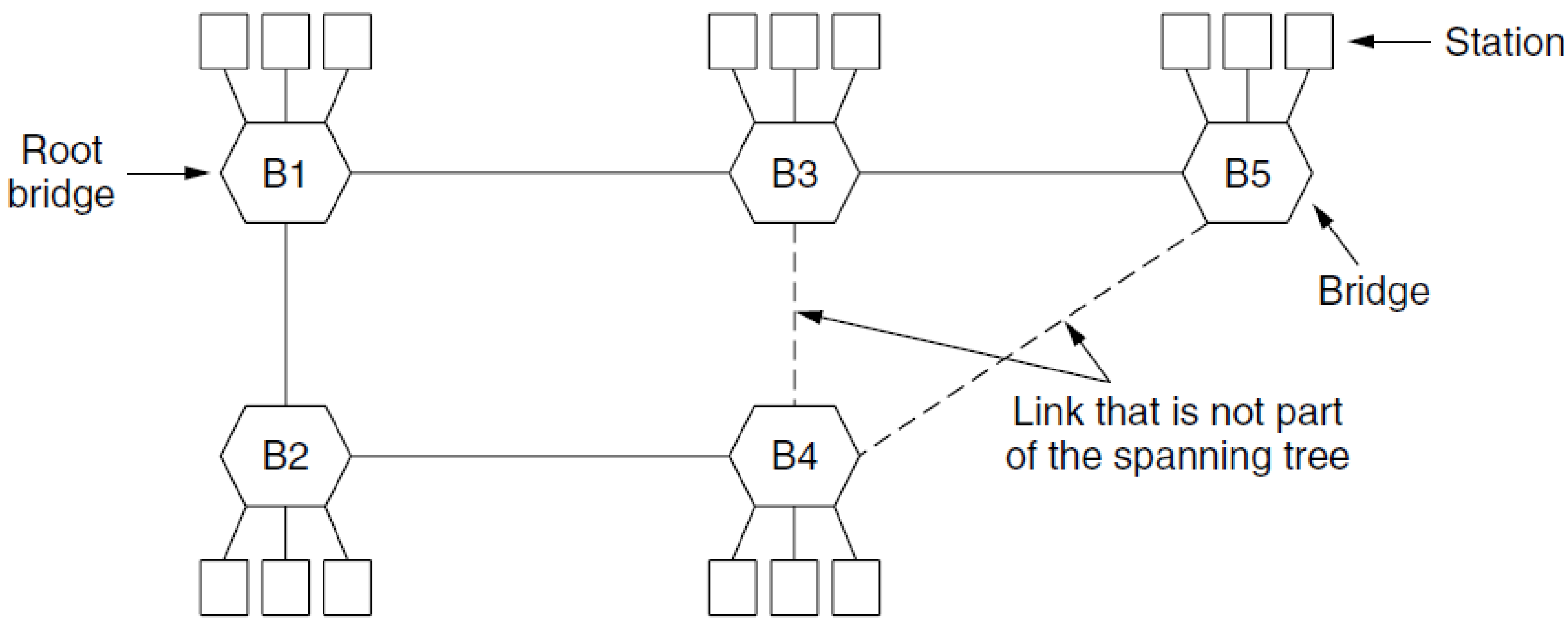
I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree.

– Radia Perlman, 1985.

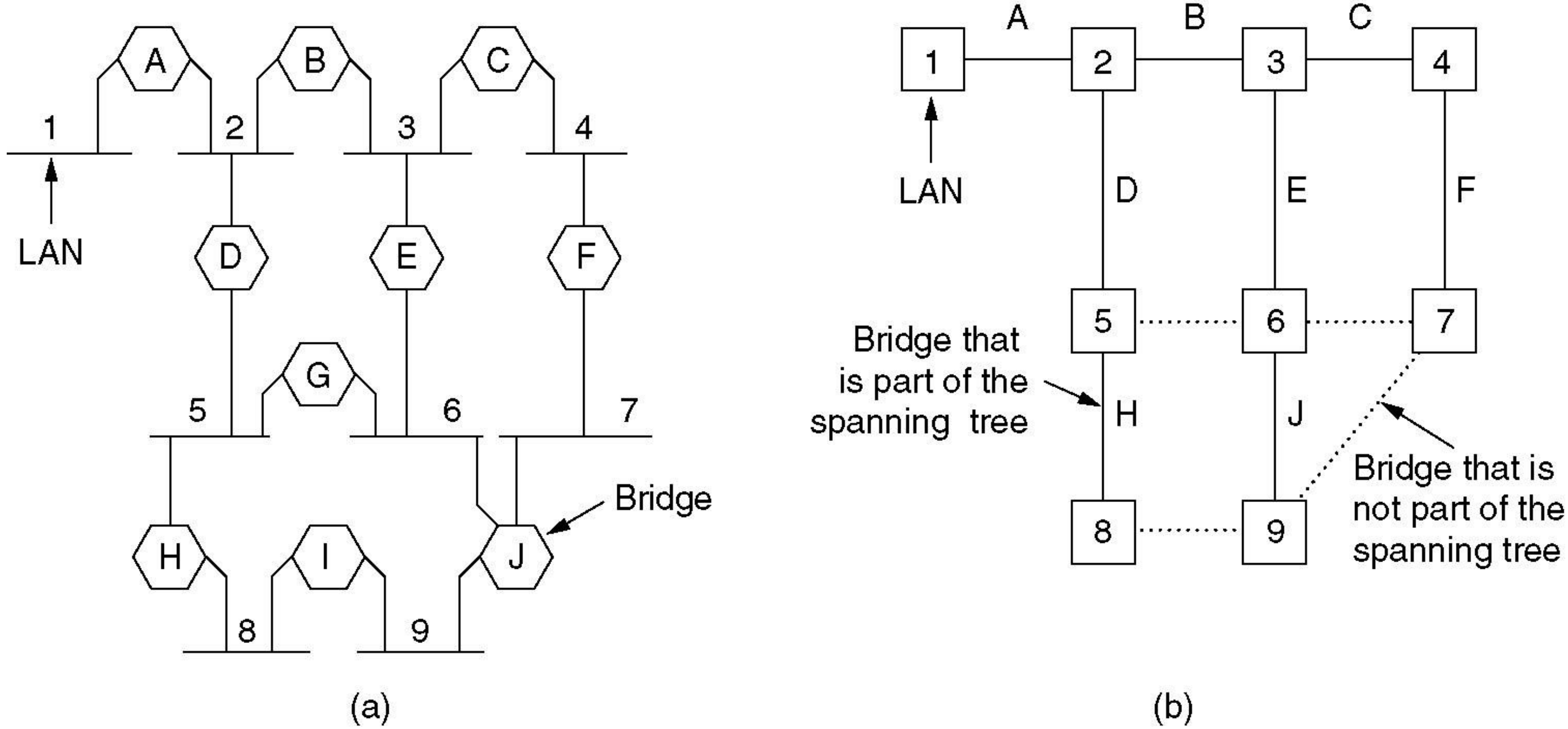
Spanning Tree (3) – Example

After the algorithm runs:

- B1 is the root, two dashed links are turned off
- B4 uses link to B2 (lower than B3 also at distance 1)
- B5 uses B3 (distance 1 versus B4 at distance 2)



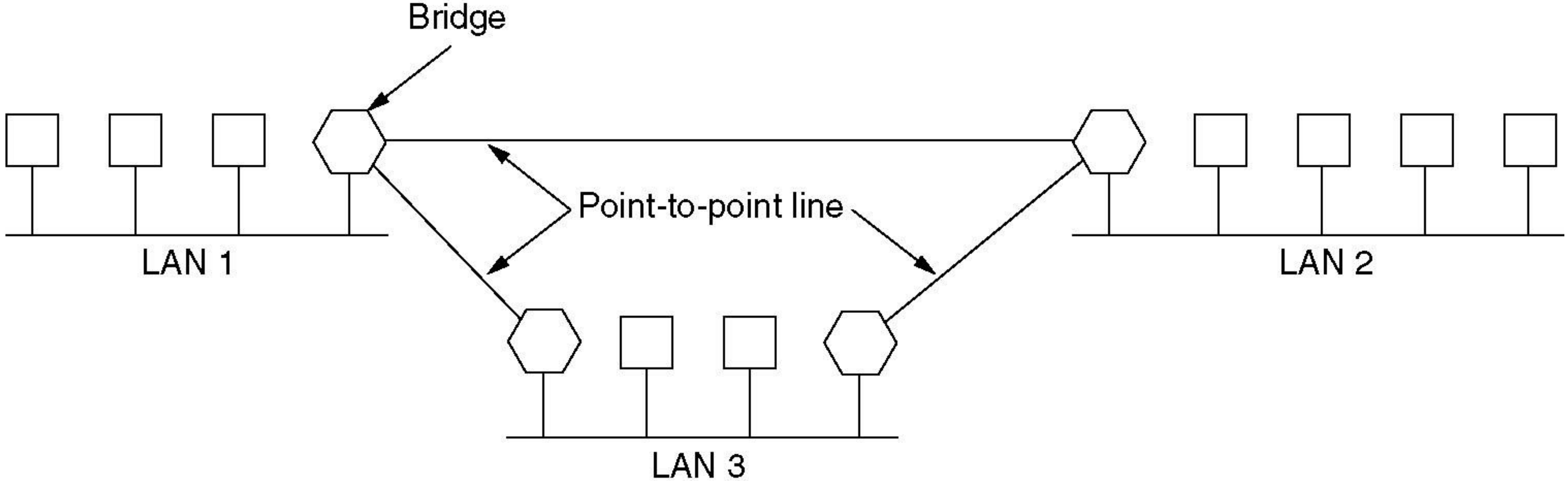
Spanning Tree Bridges (2)



(a) Interconnected LANs. (b) A spanning tree covering the LANs. The dotted lines are not part of the spanning tree.

Remote Bridges

Remote bridges can be used to interconnect distant



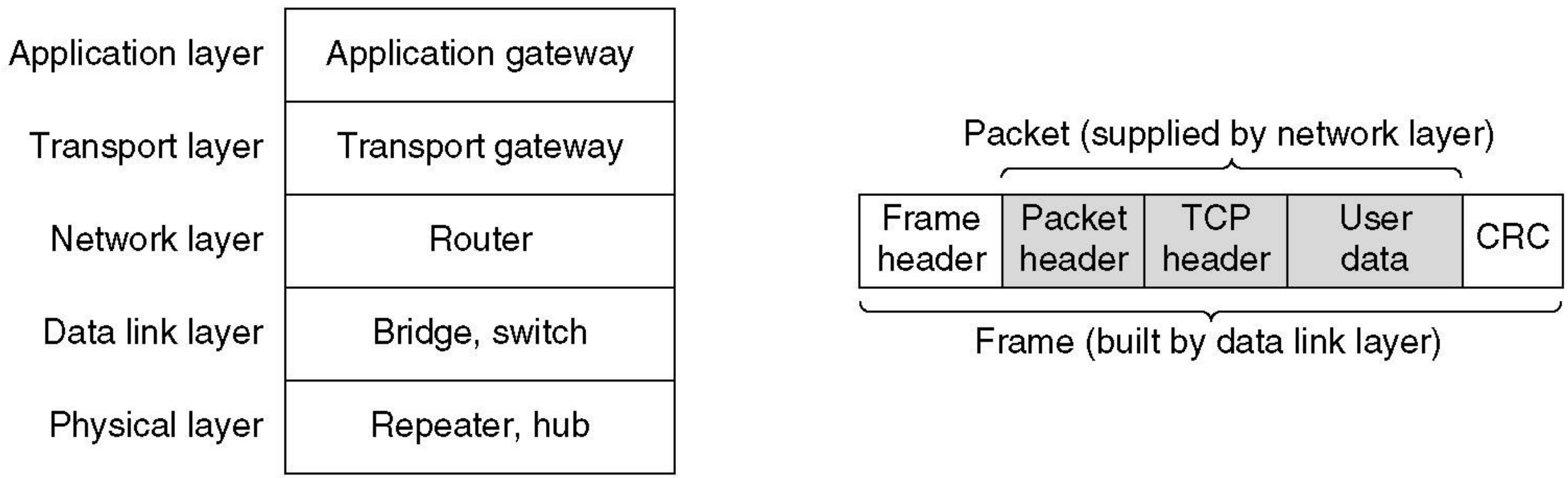
Repeaters, Hubs, Bridges, Switches, Routers, and Gateways

Devices are named according to the layer they process

- A bridge or LAN switch operates in the Link layer

Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

Repeaters, Hubs, Bridges, Switches, Routers and Gateways



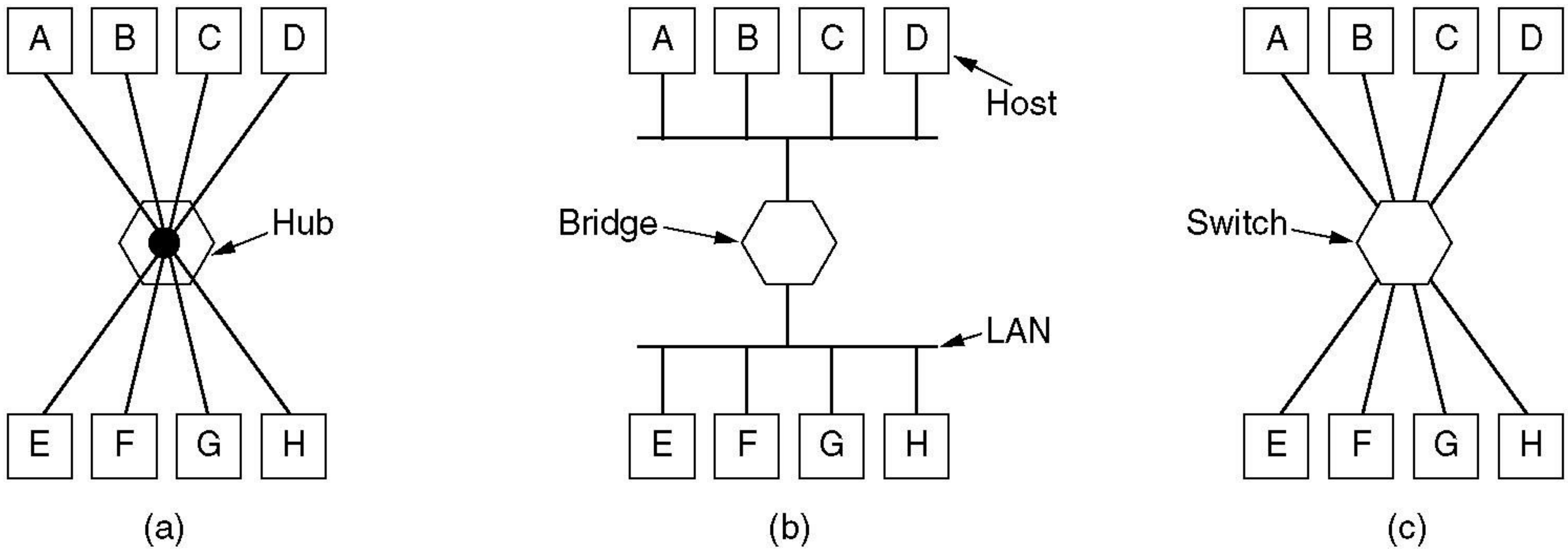
(a) (b)

(a) Which device is in which layer.

(b) Frames, packets, and headers.

Repeaters, Hubs, Bridges, Switches, Routers and Gateways (2)

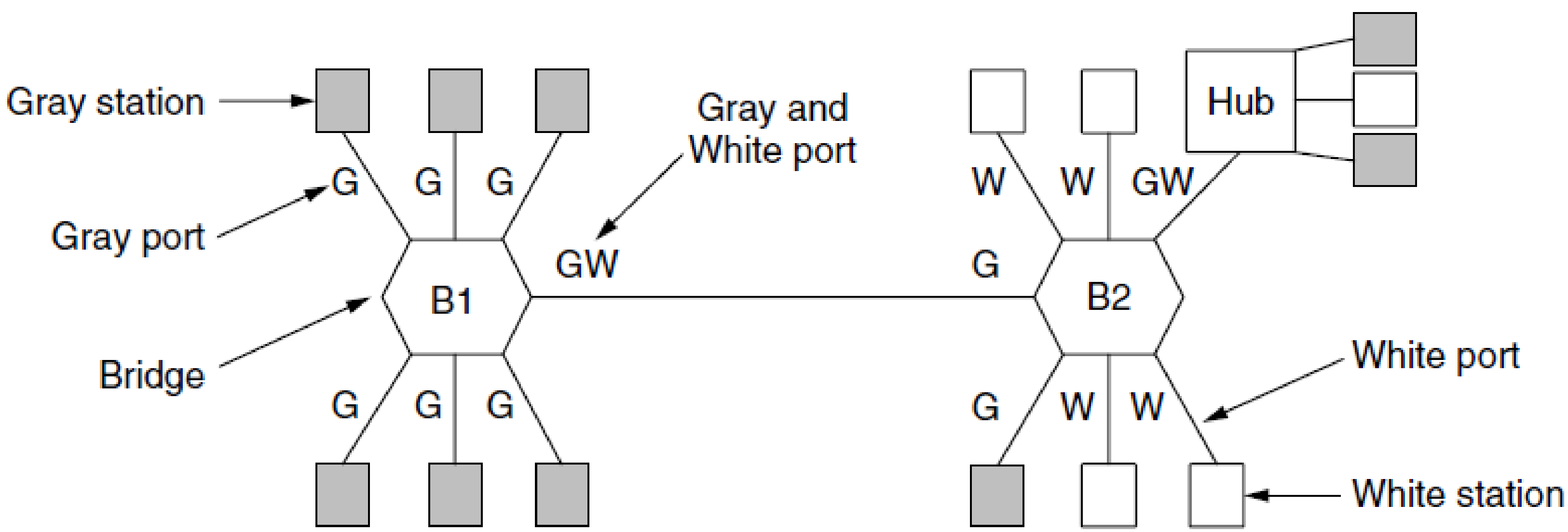
(a) A hub. (b) A bridge. (c) a switch.



Virtual LANs

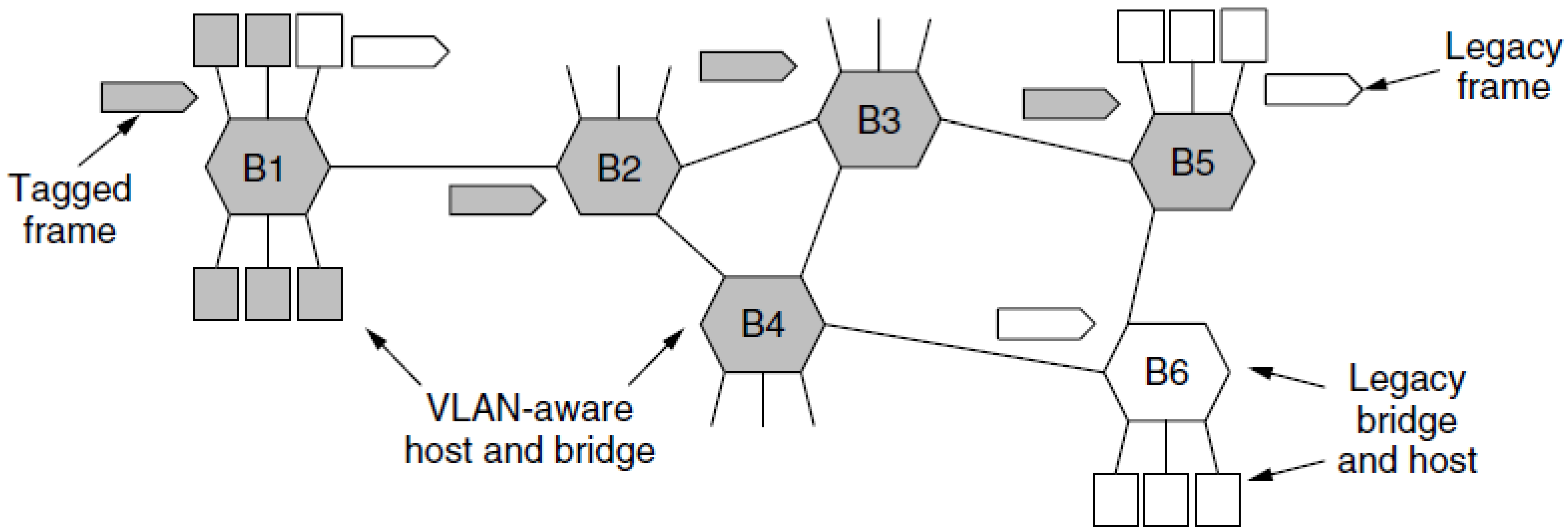
VLANs (Virtual LANs) splits one physical LAN into multiple logical LANs to ease management tasks

- Ports are “colored” according to their VLAN



Virtual LANs– IEEE 802.1Q

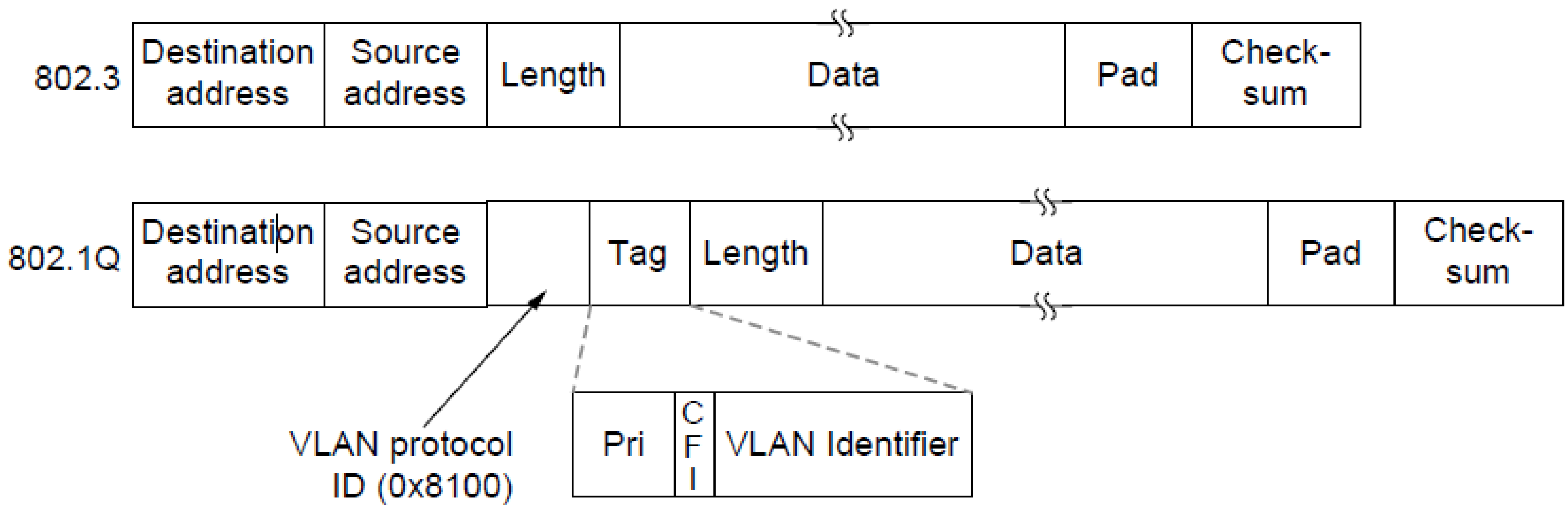
- Bridges need to be aware of VLANs to support them
  - In 802.1Q, frames are tagged with their “color”



Virtual LANs (3) – IEEE 802.1Q

802.1Q frames carry a color tag (VLAN identifier)

- Length/Type value is 0x8100 for VLAN protocol

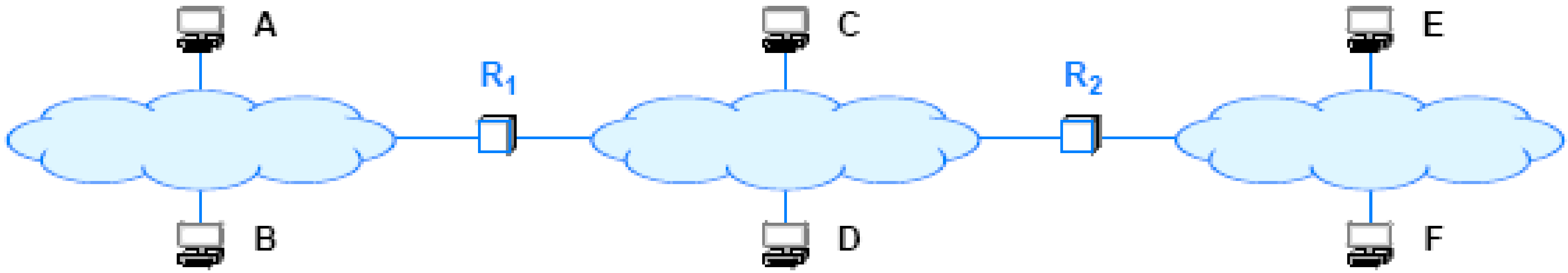


Address Resolution

- A crucial step of the forwarding process requires a translation:
  - forwarding uses IP addresses
  - a frame transmitted must contain the MAC address of the next hop
  - IP must translate the next-hop IP address to a MAC address
- The principle is:
  - IP addresses are abstractions
    - provided by protocol software
  - Network does not know how to locate a computer from its IP address
    - the next-hop address must be translated to an equivalent MAC address
- Translation from a computer's IP address to an equivalent hardware address is known as address resolution
  - And an IP address is said to be resolved to the correct MAC address
- Address resolution is local to a network

Address Resolution

- One computer can resolve the address of another computer only if both computers attach to the same physical network
  - A computer never resolves the address of a computer on a remote network
  - Address resolution is always restricted to a single network.
- For example, consider the simple internet



An example internet of three networks and computers connected to each.

The Address Resolution Protocol (ARP)

- What algorithm does software use to translate?
  - The answer depends on the protocol and hardware addressing
    - here we are only concerned with the resolution of IP
- Most hardware has adopted the 48-bit Ethernet Address
- In Ethernet: Address Resolution Protocol (ARP)
- Consider
  - Suppose B needs to resolve the IP address of C
  - B broadcasts a request that says:

"I'm looking for the MAC address of a computer that has IP address C"

  - The broadcast only travels across one network
  - An ARP request message reaches all computers on a network
  - When C receives a copy of the request along other hosts
    - Only C sends a directed reply back to B that says:

"I'm the computer with IP address C, and my MAC address is M"

The Address Resolution Protocol (ARP)

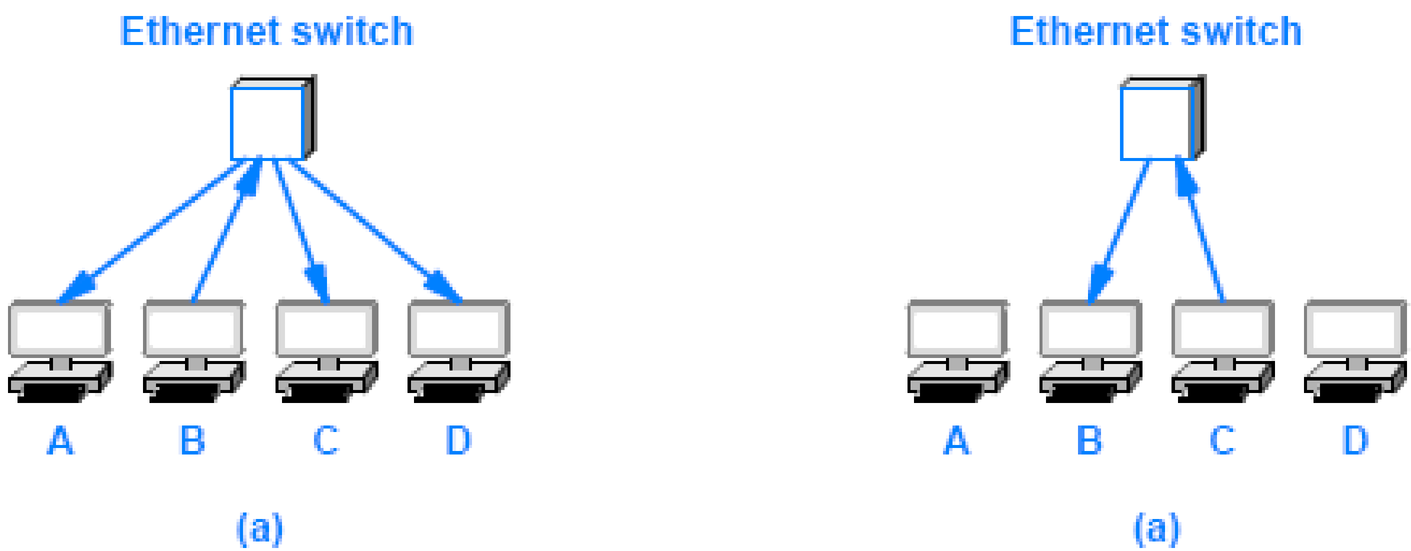


Illustration of the ARP message exchange when computer B resolves the address of computer C.

ARP Message Format

- Rather than restricting ARP to IP and Ethernet
  - The standard describes a general form for ARP messages
  - It specifies how the format is adapted for each type of protocol
- Choosing a fixed size for a hardware address is not suitable
  - New network technologies might be invented that have addresses larger than the size chosen
  - The designers included a fixed-size field at the beginning of an ARP message to specify the size of the hardware addresses being used
- For example, when ARP is used with an Ethernet
  - the hardware address length is set to 6 octets
    - because an Ethernet address is 48 bits long

ARP Message Format

- To increase the generality of ARP
  - the designers also included an address length field
- ARP protocol can be used to bind an arbitrary high-level address to an arbitrary hardware address
- In practice, the generality of ARP is seldom used
  - most implementations of ARP are used to bind IP addresses to Ethernet addresses
- Figure illustrates the format of an ARP message
  - when the protocol is used with an IP version 4 address (4 octets) and Ethernet hardware address (6 octets)
  - each line of the figure corresponds to 32 bits of an ARP message

ARP Message Format

0 8 16 24 31

<table>
<tr><td colspan="2">HARDWARE ADDRESS TYPE</td><td colspan="2">PROTOCOL ADDRESS TYPE</td></tr>
<tr><td>HADDR LEN</td><td>PADDR LEN</td><td colspan="2">OPERATION</td></tr>
<tr><td colspan="4">SENDER HADDR (first 4 octets)</td></tr>
<tr><td colspan="2">SENDER HADDR (last 2 octets)</td><td colspan="2">SENDER PADDR (first 2 octets)</td></tr>
<tr><td colspan="2">SENDER PADDR (last 2 octets)</td><td colspan="2">TARGET HADDR (first 2 octets)</td></tr>
<tr><td colspan="4">TARGET HADDR (last 4 octets)</td></tr>
<tr><td colspan="4">TARGET PADDR (all 4 octets)</td></tr>
</table>

The format for an ARP message when binding an IPv4 address to an Ethernet address.

ARP Message Format

- HARDWARE ADDRESS TYPE
  - 16-bit field that specifies the type of hardware address being used
  - the value is 1 for Ethernet
- PROTOCOL ADDRESS TYPE
  - 16-bit field that specifies the type of protocol address being used
  - the value is 0x0800 for IPv4
- HADDR LEN
  - 8-bit integer that specifies the size of a hardware address in bytes
- PADDR LEN
  - 8-bit integer that specifies the size of a protocol address in bytes
- OPERATION
  - 16-bit field that specifies whether the message
    - request (the field contains 1) or
    - response (the field contains 2)

ARP Message Format

- SENDER HADDR
  - HADDR LEN bytes for the sender's hardware address
- SENDER PADDR
  - PADDR LEN bytes for the sender's protocol address
- TARGET HADDR
  - HADDR LEN bytes for the target's hardware address
- TARGET PADDR
  - PADDR LEN bytes for the target's protocol address

ARP Message Format

- An ARP message contains fields for two address bindings
  - one binding to the sender
  - other to the intended recipient, ARP calls it target
- When a request is sent
  - the sender does not know the target's hardware address (that is the *information being requested*)
    - therefore, field TARGET HADDR in an ARP request can be filled with zeroes (0s) because the contents are not used
- In a response
  - the target binding refers to the initial computer that sent the request
  - Thus, the target address pair in a response serves no purpose
    - the inclusion of the target fields has survived from an early version of the protocol

ARP Encapsulation

- When it travels across a physical network
  - an ARP message is encapsulated in a hardware frame
- An ARP message is treated as data being transported
  - the network does not parse the ARP message or interpret fields

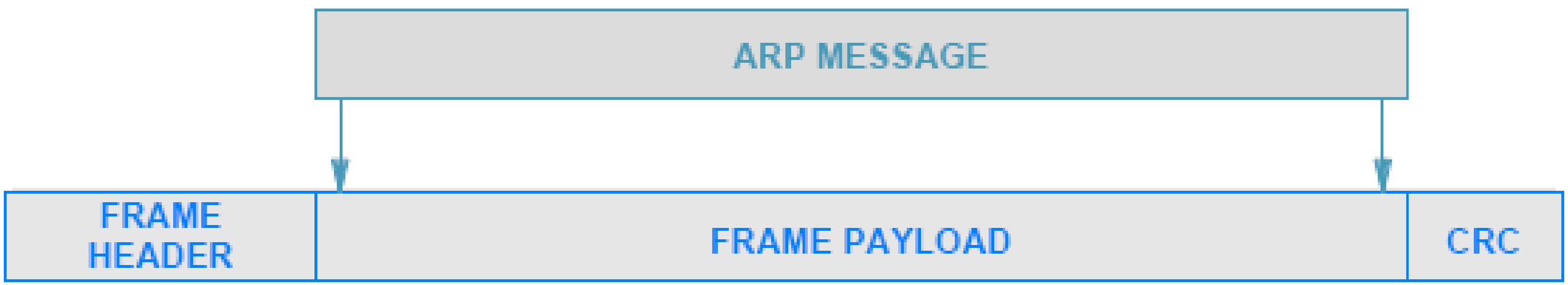


Illustration of ARP encapsulation in an Ethernet frame.

ARP Encapsulation

- The type field in the frame header specifies that the frame contains an ARP message
- A sender must assign the appropriate value to the type field
  - before transmitting the frame
- And a receiver must examine the type field
  - in each incoming frame
- Ethernet uses type field 0x806 to denote an ARP message
- The same value is used for both ARP requests/ responses
  - Frame type does not distinguish between types of ARP messages
  - A receiver must examine the OPERATION field in the message
    - to determine whether an incoming message is a request or a response

ARP Caching and Message Processing

- Sending an ARP request for each datagram is inefficient
  - Three (3) frames traverse the network for each datagram (an ARP request, ARP response, and the data datagram itself)
- Most communications involve a sequence of packets
  - a sender is likely to repeat the exchange many times
- To reduce network traffic
  - ARP software extracts and saves the information from a response
    - so it can be used for subsequent packets
  - The software does not keep the information indefinitely
    - Instead, ARP maintains a small table of bindings in memory
- ARP manages the table as a cache
  - an entry is replaced when a response arrives
  - the oldest entry is removed whenever the table runs out of space or after an entry has not been updated for a long period of time
  - ARP starts by searching the cache when it needs to bind an address

ARP Caching and Message Processing

- If the binding is present in the cache
  - ARP uses the binding without transmitting a request
- If the binding is not present in the cache
  - ARP broadcasts a request
  - waits for a response
  - updates the cache
  - and then proceeds to use the binding
- The cache is only updated when an ARP message arrives (either a request or a response)

ARP Caching and Message Processing

```
Given:
    An incoming ARP message (either a request or a response)
Perform:
    Process the message and update the ARP cache
Method:
    Extract the sender's IP address, I, and MAC address, M
    If ( address I is already in the ARP cache ) {
        Replace the MAC address in the cache with M
    }
    if ( message is a request and target is "me" ) {
        Add an entry to the ARP cache for the sender
            provided no entry exists;
        Generate and send a response;
    }
```

The steps ARP takes when processing an incoming message.

ARP Caching and Message Processing

- For optimization, it is necessary to know two facts:
  - Most computer communication involves two-way traffic
    - if a message from A to B, probability is high that a reply will be from B back to A
  - Each address binding requires memory
    - a computer cannot store an arbitrary number of address bindings
- The first fact explains why extracting the sender's address binding optimizes ARP performance

The Conceptual Address Boundary

- ARP provides an important conceptual boundary between MAC addresses and IP addresses:
  - ARP hides the details of hardware addressing
  - It allows higher layers of software to use IP addresses
- There is an important conceptual boundary imposed between the network interface layer and all higher layers
- illustrates the addressing boundary

The Conceptual Address Boundary

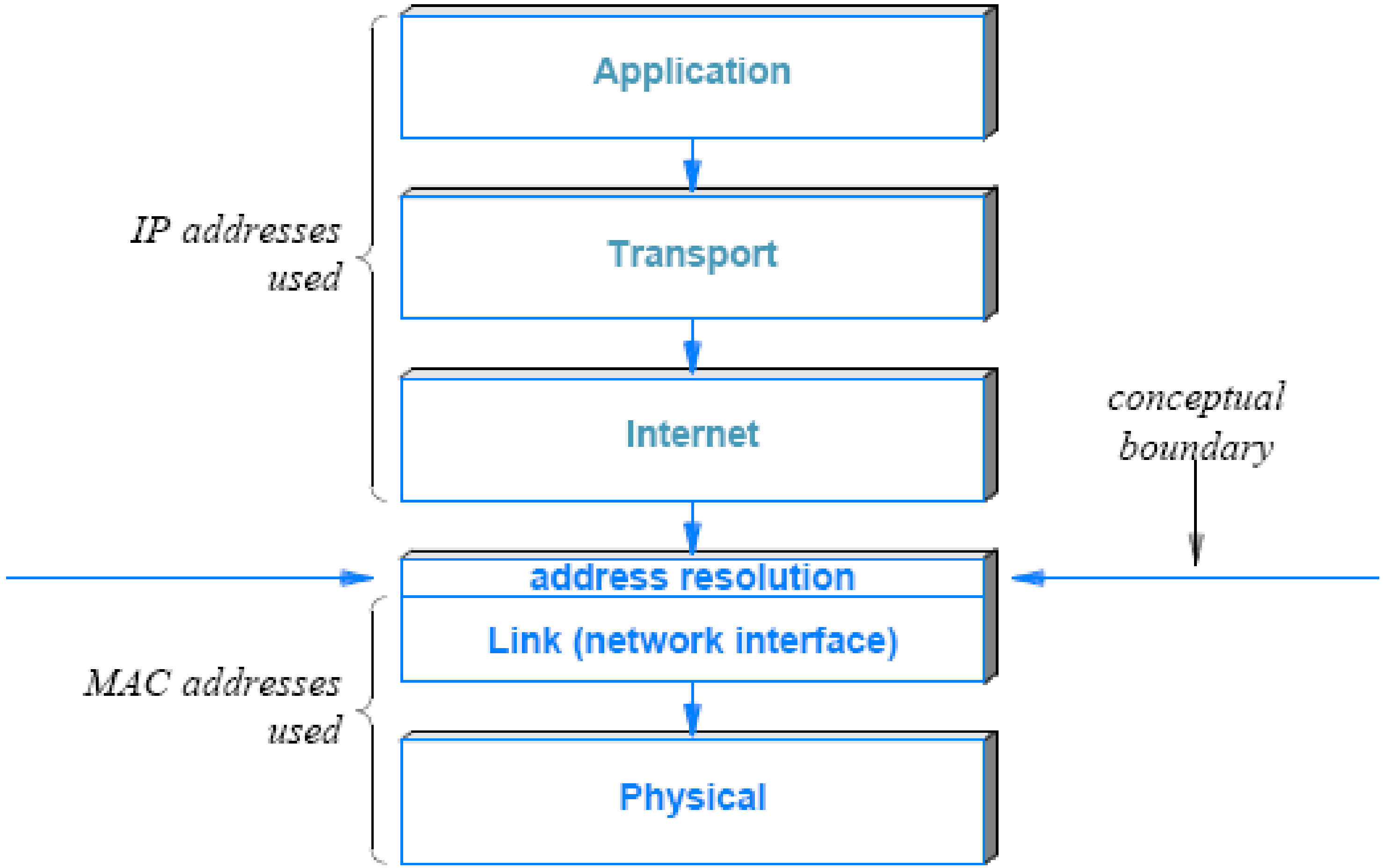
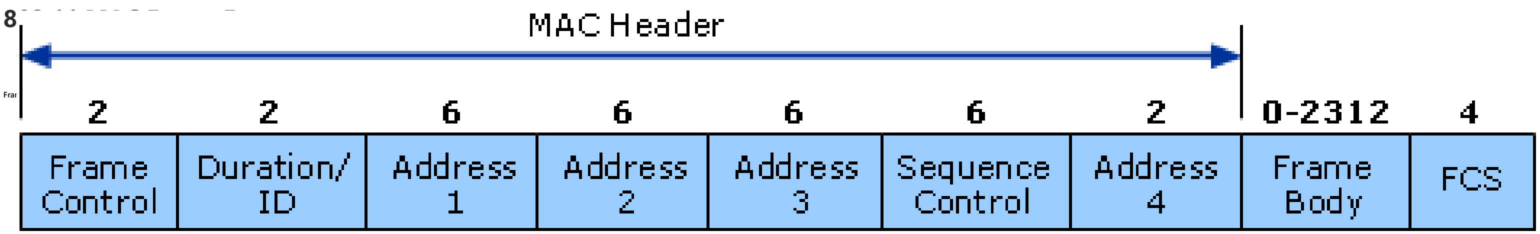


Illustration of the boundary between the use of IP addresses and MAC addresses.


MAC Header
2
2
6
6
6
6
2
0-2312
4
Frame Control
Duration/ ID
Address 1
Address 2
Address 3
Sequence Control
Address 4
Frame Body
FCS

Frame Control Field

2 bits	2	4	1	1	1	1	1	1	1	1
Protocol Version	Type	Subtype	To DS	From DS	More Fragments	Retry	Power Mgt.	More data	WEP	Order

•**Protocol Version** provides the current version of the 802.11 protocol used. Receiving STAs use this value to determine if the version of the protocol of the received frame is supported.

•**Type and Subtype** determines the function of the frame. There are three different frame type fields: control, data, and management. There are multiple subtype fields for each frame type . Each subtype determines the specific function to perform for its associated frame type.

To DS and From DS indicates whether the frame is going to or exiting from the DS (distributed system), and is only used in data type frames of STAs associated with an AP.

More Fragments indicates whether more fragments of the frame, either data or management type, are to follow.

Retry indicates whether or not the frame, for either data or management frame types, is being retransmitted.

Power Management indicates whether the sending STA is in active mode or power-save mode.

More Data indicates to a STA in power-save mode that the AP has more frames to send. It is also used for APs to indicate that additional broadcast/multicast frames are to follow.

WEP indicates whether or not encryption and authentication are used in the frame. It can be set for all data frames and management frames, which have the subtype set to authentication.

Order indicates that all received data frames must be processed in order.

Address Fields

- **BSS Identifier (BSSID).** BSSID uniquely identifies each BSS. When the frame is from an STA in an infrastructure BSS, the BSSID is the MAC address of the AP. When the frame is from a STA in an IBSS, the BSSID is the randomly generated, locally administered MAC address of the STA that initiated the IBSS.

- **Destination Address (DA).** DA indicates the MAC address of the final destination to receive the frame.

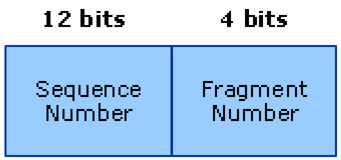
- **Source Address (SA).** SA indicates the MAC address of the original source that initially created and transmitted the frame.

- **Receiver Address (RA).** RA indicates the MAC address of the next immediate STA on the wireless medium to receive the frame.

- **Transmitter Address (TA).** TA indicates the MAC address of the STA that transmitted the frame onto the wireless medium.

Sequence Control

- The Sequence Control field contains two subfields, the Fragment Number field and the Sequence Number field, as shown in the following figure.



- **Sequence Number** indicates the sequence number of each frame. The sequence number is the same for each frame sent for a fragmented frame; otherwise, the number is incremented by one until reaching 4095, when it then begins at zero again.

- **Fragment Number** indicates the number of each frame sent of a fragmented frame. The initial value is set to 0 and then incremented by one for each subsequent frame sent of the fragmented frame.