Computer Networks

Chapter 04

prof. dr ir Maarten van Steen

Vrije Universiteit Amsterdam
Faculty of Science
Dept. Mathematics and Computer Science
Room R4.20. Tel: (020) 444 7784

steen@cs.vu.nl

Contents

01 Introduction
02 Physical Layer
03 Data Link Layer
04 MAC Sublayer
05 Network Layer
06 Transport Layer
07 Application Layer
08 Network Security

Medium Access Sublayer (1/2)

So far: We have discussed the Data Link Layer's functionality and some protocols related to point-to-point communication.

A large class of networks is built on top of **broadcast channels**: a number of stations that share the same "wire." If one station sends, all the others get to hear it.

What's the problem: if you're sharing a channel, then two stations may decide to start frame transmission at the same time ⇒ **frame collision**, which means rubbish on the wire.

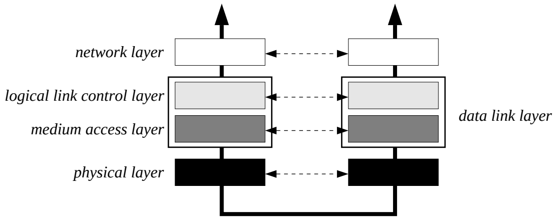
Solution: Allocate the channel to one of the competing stations.

Problem: You'll have to use that same channel to figure out the competition and the allocation.

Medium Access Layer (2/2)

Three strategies for channel allocation:

- Exercise **no control** at all: simply let a station try to use the channel, and do something when a collision happens. Applied in **contention systems**.
- Employ a **round-robin** technique: each station in turn is allowed to use the channel. Applied in **token-based** systems – the station that has the token may use the channel.
- Let a station place a **reservation** for the channel. Used in **slotted** systems. The problem is how to make a reservation.



Contention systems: ALOHA (1/2)

Principle: if you want to send a frame, just do it. If a collision occurs, finish your current transmission and retry later.

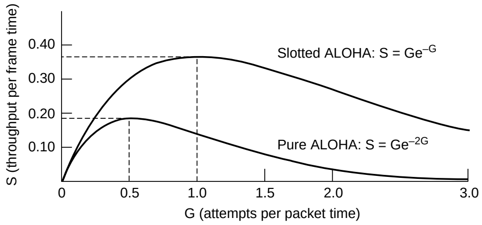
Note: it's pretty hard to do any worse than this, but efficiency is not really that bad:

- Let S be the average number of new frames submitted during a *frame time* T_{frame} (time needed to transmit a full frame). Poisson distributed.
- Let G be the number of old and new frame submissions during a frame time ($G \geq S$). Also Poisson distributed:
  $\mathcal{P}[k \text{ frames submitted}] = \frac{G^k \cdot e^{-G}}{k!}$
- Let P_0 be the probability that frame does not suffer from collision $\Rightarrow S = G \cdot P_0$.

Contention systems: ALOHA (2/2)

- When a frame is being sent, it shouldn't bump into its predecessor, and shouldn't be bumped into by a successor. The predecessor requires T_{frame} time, and our frame as well.
- Probability that a frame will not be damaged is $P_0 = e^{-2G}$
- $S = G \cdot e^{-2G}$

We can improve a bit by removing some randomness. in **Slotted ALOHA** frame transmission can only start at fixed times:



CSMA Protocols

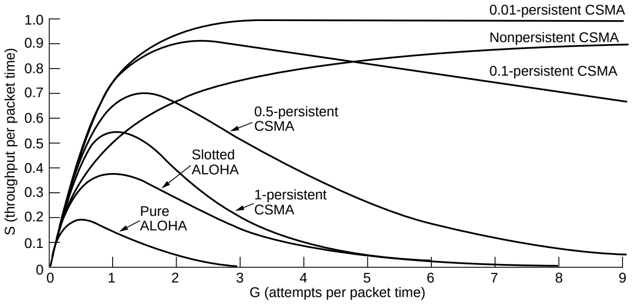
Carrier Sense Multiple Access Protocols do better than ALOHA: you monitor the channel before and/or during transmission.

1-persistent: Listen whether the channel is free before transmitting. If busy, wait until it becomes free and then immediately start your transmission.

Nonpersistent: Less greedy – when the channel is busy, wait a random period of time before trying again. If you wait too long, the channel utilization drops.

p-Persistent: Used with slotted systems. If you find the channel idle during the current slot, you transmit with probability p, and defer until next slot with probability $1-p$. $p=1$ is not really good, $p=0$ makes you *really* polite.

Protocol Comparison

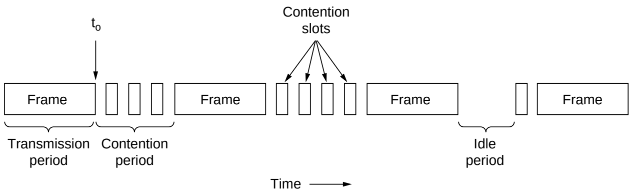


Question: What are we actually displaying here? Should the conclusion be that p-persistent protocols are really good with $p \approx 0$?

CSMA w/Collision Detection (1/2)

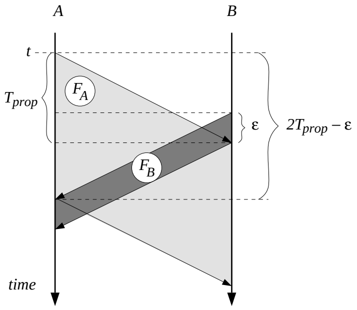
Improvement: sense the channel, but immediately stop transmission when you detect a collision. **Ethernet** works like this.

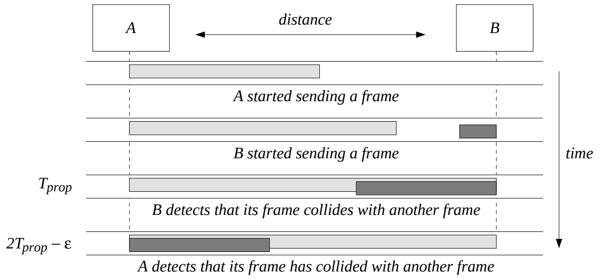
1. Listen to see whether the channel is free. Transmission is delayed until the channel is no longer used.
2. During transmission, keep listening in order to detect a collision. If a collision occurs, transmission immediately stops.
3. If a collision occured, wait a random period of time, and proceed with the first step again.



CSMA w/Collision Detection (2/2)

Problem: how big should the contention period be?

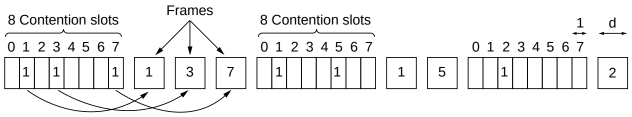




A
distance
B
A started sending a frame
B started sending a frame
time
T_{prop}
B detects that its frame collides with another frame
$2T_{prop} - \varepsilon$
A detects that its frame has collided with another frame

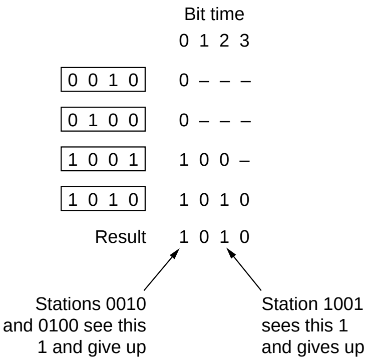
Collision Free Protocols (1/2)

Bit-map: the contention period contains N slots. If station k wants to transmit a frame, it transmits a 1 during the k^{th} slot. The highest-numbered station goes first.



Collision Free Protocols (1/2)

Binary Countdown: In the contention period a total of $\log_2 N$ bits can be transmitted. Each station transmits its number (bit by bit), and stops as soon as it detects a higher-numbered contender:



	Bit time 0 1 2 3
0 0 1 0	0 – – –
0 1 0 0	0 – – –
1 0 0 1	1 0 0 –
1 0 1 0	1 0 1 0
Result	1 0 1 0

Stations 0010 and 0100 see this 1 and give up

Station 1001 sees this 1 and gives up

Limited-Contention Protocols (1/2)

Principle: Contention systems are good when there's not much going on – a station can immediately transmit a frame. We do some repairing when things go wrong.

Collision-free systems are good when there's generally a lot of traffic – a station first has to get the channel explicitly before frame transmission. We do a lot of work avoiding collisions.

What we really want is the contention strategy during light loads, and collision-free strategy during rush hours.

Limited-Contention Protocols (2/2)

A solution: Dynamically regulate the number of competing stations during a contention period. If there's a collision during the k^{th} slot, divide the contenders into two groups.

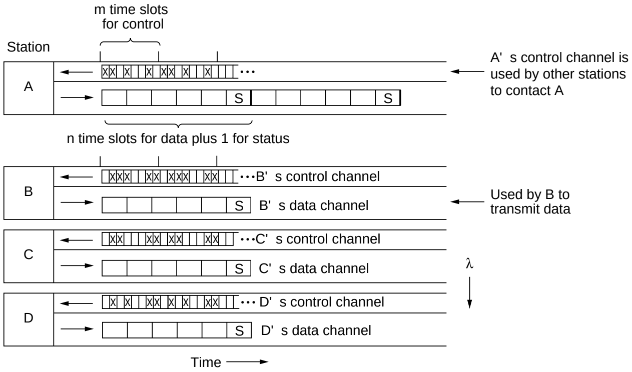
The first group gets to try it again during the next slot $(k+1)$. If no collisions occur then, the second group gets a try during the slot after that $(k+2)$. Otherwise, the first group is split up again.

Note: if there's not much traffic, the first station will be immediately allowed to transmit a frame. With a lot of traffic, the strategy reduces to the bit-map protocol.

Wavelength Division (1/2)

Principle: if you have a lot of bandwidth, just divide the channel into sub-channels, and dynamically allocate the sub-channels. Used in fiber optics.

- Each station gets two channels: one control channel for handling incoming requests, one for the actual data transfer.
- Each channel repeatedly carries a fixed series of slots. The data channel contains a status slot carrying info on free slots in its control channel.



Wavelength Division (2/2)

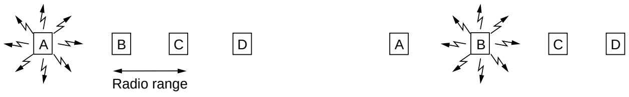
- In order to contact a station A, you first read its status slot from the data channel to see which control slot is unused. You then put a transmission request in a free control slot.
- If the transmission request is accepted, you can send data on your own data channel that will be picked up by A. You put the data in specific slot, and tell A which slot that is.

Note: there's still a lot of competition. If two senders say that there's data in slot #4 for station A, one of the slots will not be read – A can only read from one channel at a time.

Wireless Networks

Basic idea: Often, there are a number of **base stations** connected through guided media. A base station can communicate with a mobile computer. The mobile computers use radio/infrared signals for communication.

Problem: There can be subtle interference:

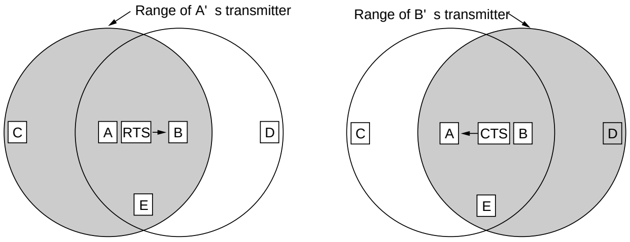


(a) (b)

Issue (a): How can C be prevented from trying to transmit something to B? In that case it will ruin any receipt by B (hidden station problem).

Issue (b): How can we tell C that it is allowed to transmit to D, because this will not interfere with the communication from B to A? (exposed station problem).

Multiple Access with Collision Avoidance



(a) (b)

- A first sends a Request To Send (RTS).
- B answers with a Clear To Send (CTS).
- C hears only RTS and can freely transmit, knowing it will not interfere with A's transmission.
- D hears only the CTS and keeps still for otherwise it would interfere with B's reception.

802.3 (or Ethernet) (1/2)

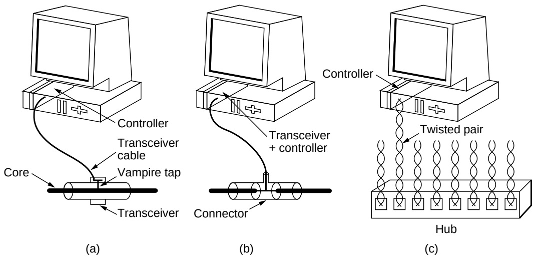
Ethernet stands for a near implementation of the **IEEE 802.3** protocol. It is CSMA/CD based (sense the channel, wait until idle, and backoff when you detect a collision).

Name	Cable	Max. dist.	Nodes/Seg.
10Base5	Thick coax	500 m	100
10Base2	Thin coax	200 m	30
10Base-T	Twisted pair	100 m	1024
10Base-F	Fiber optics	2000 m	1024

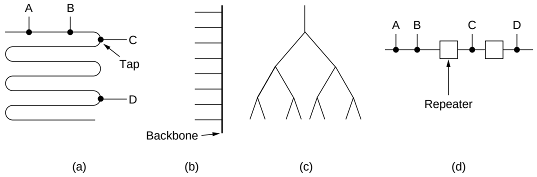
Note: 10Base-T is popular as it can make use of telephone lines, and is easy to maintain when it comes to cable breaks (cables go to **hubs**).

802.2 (or Ethernet) (2/2)

Note: Different Ethernets use different connection and cabling schemes: (a) 10Base5, (b) 10Base2, (c) 10Base-T.



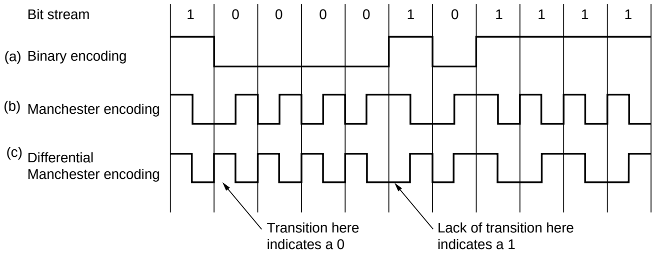
In 10Base-F we can apply different schemes (linear, backbone, tree). Segmented networks with repeaters are used to build large local internetworks.



Manchester Encoding

Problem: We can't just send straight binary codes across the wire, because stations can't distinguish a 0 from an idle line.

Solution: use an encoding scheme in which a voltage transition occurs during every bit time (Manchester Encoding):


Bit stream
1 0 0 0 0 1 0 1 1 1 1
(a) Binary encoding
(b) Manchester encoding
(c) Differential Manchester encoding
Transition here indicates a 0
Lack of transition here indicates a 1

802.3 Frame Layout

Bytes	8		6	6	2	0-1500	0-46	4
(a)	Preamble		Destination address	Source address	Type	Data	Pad	Check-sum
(b)	Preamble	SoF	Destination address	Source address	Length	Data	Pad	Check-sum

Preamble: Seven times 10101010 is used to synchronize the receiver's clock with that of the sender.

Start: Just a delimiter to tell that the real info is now coming.

Address: Generally 48-bit fields. Leftmost bit indicates ordinary or group addresses. Second bit indicates global or local address.

Length: Ranges from 0-1500. Frames should always be at least 64 bytes. See Tanenbaum for the details.

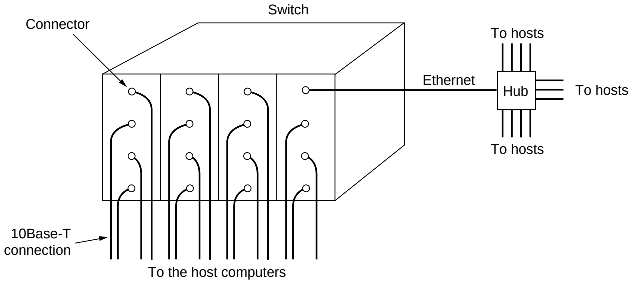
Pad: you got it...

Checksum: Calculated over the data field. CRC-based.

Switched 802.3 LANs

Problem: As more stations are added, traffic will go up, and so will the possibility of collisions ⇒ the network will saturate.

Solution: Divide the network into separate sub-LANs and connect them through a high-speed switch:



Question: Are we really improving things here?

Fast Ethernet (1/2)

Problem: Ethernet by itself was too slow, and new alternatives like FDDI were just too expensive (they were okay for backbones, but not for basic LAN segments).

Solution: Upgrade existing base of LANs (i.e. Ethernets) in such a way that the interfaces remain the same, but the capacity goes up ⇒ 100 Mbps Ethernet.

Data formats, interfaces, and protocols are all the same.

That means that we can only drop the bit time from 100 nsec to 10 nsec. Just telling everyone to shorten their wires won't really do.

The solution is to use only hubs (10Base-T) and combine it with switching technology. (10Base-T can make use of telephone lines, and is easy to maintain when it comes to cable breaks.)

Fast Ethernet (2/2)

Name	Cable	Length	Pros
100Base-T4	Twisted pair	100 m	Cat 3 UTP
100Base-TX	Twisted pair	100 m	full duplex Cat 5 UTP
100Base-F	Fiber optics	2000 m	full duplex, long runs

To illustrate: T4 uses 4 twisted pairs, one to the hub, one from the hub, and two switchable to current direction (half-duplex).

Rather than binary signalling, we use ternary signalling $\Rightarrow$ three twisted pairs in the right direction gives 3^3 possible signals, which can encode strings of 4 bits.

With a signalling speed of 25 Mhz, we then get 100 Mbps across the category 3 UTP wires. Not really bad...

Note: If we then subsequently add frame-buffering capabilities to the hub, combine that with a high-speed backplane, we can have stations transmit at the same time. The buffering at the hub can avoid a lot of collisions.

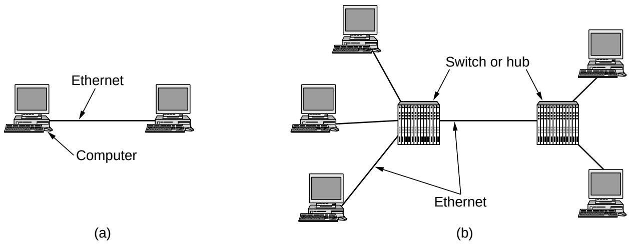
Gigabit Ethernet (1/2)

Because Fast Ethernet may not always be fast enough, we simply move the decimal point one position and get Gigabit Ethernet: transfer Ethernet frames at 1000 Mbps.

Name	Cable	Max. range
1000BASE-SX	Fiber	550 m
1000BASE-LX	Fiber	5000 m
1000BASE-CX	Copper	25 m
1000BASE-T	Twisted pair	100 m

Note: Gigabit Ethernet works only in point-to-point mode, supporting either full duplex (normal case) or half duplex transmissions. Full duplex requires a switch; half duplex can be used with hubs.

Gigabit Ethernet (2/2)



Note: With a switch, there is no need for the CSMA/CD part! Maximal cable length is determined by signal strength.

Note: With a hub, the maximal cable length is restricted to 25 m (it used to be 2500 for 10 Mbit Ethernet). To facilitate longer cables, tricks are used:

- **Carrier extension**: let the hardware extend a frame to 512 bytes.
- **Frame bursting**: let the hardware put several frames into a 512-byte frame.

Wireless LANs

Observation: Wireless LANs need to apply special techniques to achieve high bandwidth. In brief:

Infrared: Applicable for 1-2 Mbps. Not very popular, also because sunlight degrades performance.

Frequency hopping: Use 79 channels, each 1 MHz wide in an unregistered band (i.e., free to be used by anyone). In effect, frames are sent at different frequencies each time. Low bandwidth, but good resistance against security attacks and interference from other devices.

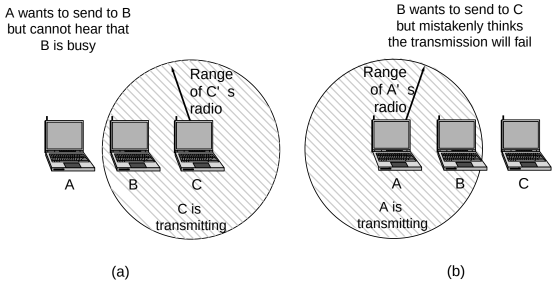
Direct sequencing: Similar to CDMA, restricted to 1-2 Mbps.

Orthogonal FDM: Akin to ADSL: apply FDM across multiple channels (48 for data, 4 for control). Can reach 54 Mbps. We’re talking serious business.

High rate direct sequencing: Consider it enhanced CDMA to get to 11 Mbps.

802.11: Channel Allocation

Problem: How do we solve the hidden/exposed station problem: one way or the other, stations should not be allowed to continuously interfere with each other's transmissions.



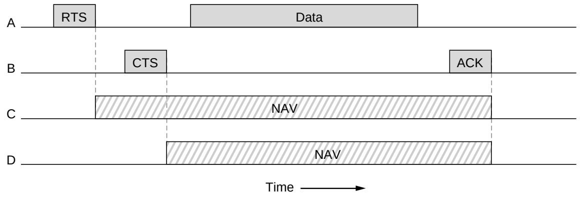
Distributed coordination: let the stations figure it out by using a **collision avoidance** protocol (CSMA/CA).

Point coordination: There's a central base station that controls who goes first.

CSMA/CA

Ethernet-like: Sense the channel and send only if it's free. Don't sense the channel during transmission: if a collision occurred, wait a random time and try again later.

MACAW: Send RTS/CTS packets to see where you are and whether you should defer transmission to avoid interference with a transmission between two other stations:



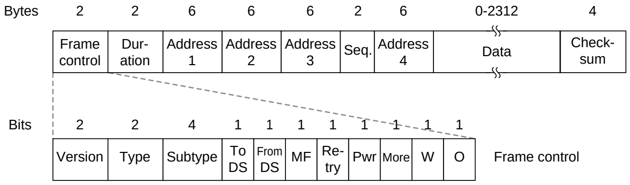
NAV: It's a virtual channel that a station assigns to itself telling it to shut up.

Point Coordination

Essence: Let a single base station control who gets to send a frame.

Approach: Send a **beacon frame** once every 10 or 100 ms. This frame carries information on frequencies and such, and invites stations to sign up for transmission.

802.11: Frame Structure (1/3)



Bytes	2	2	6	6	6	2	6	0-2312	4
	Frame control	Dur-ation	Address 1	Address 2	Address 3	Seq.	Address 4	Data	Check-sum

Bits	2	2	4	1	1	1	1	1	1	1	1	
	Version	Type	Subtype	To DS	From DS	MF	Re-try	Pwr	More	W	O	Frame control

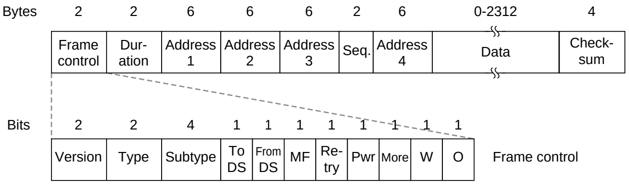
Type: Data, control, or management frame

Subtype: Are we dealing with RTS, CTS, an ACK, etc.

DS: Is the frame entering/leaving the current cell?

MF: Frames are allowed to be fragmented to increase reliability. This bit tells whether more fragments are on their way.

802.11: Frame Structure (2/3)



Bytes	2	2	6	6	6	2	6	0-2312	4
	Frame control	Duration	Address 1	Address 2	Address 3	Seq.	Address 4	Data	Checksum

Bits	2	2	4	1	1	1	1	1	1	1	1	
	Version	Type	Subtype	To DS	From DS	MF	Retry	Pwr	More	W	O	Frame control

Retry: Is this a retransmission?

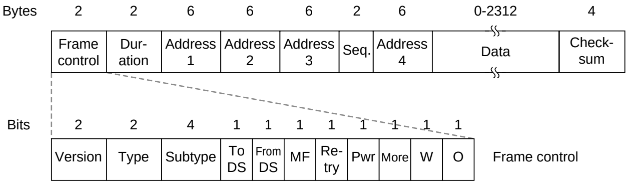
Power: Used by a base station to activate/passivate a station (important in view of power saving)

More: Additional frames can be expected.

W: Data is encrypted using the Wired Equivalence Privacy algorithm.

O: Stick to ordered delivery of frames.

802.11: Frame Structure (3/3)



Bytes	2	2	6	6	6	2	6	0-2312	4
	Frame control	Duration	Address 1	Address 2	Address 3	Seq.	Address 4	Data	Checksum

Bits	2	2	4	1	1	1	1	1	1	1	1	
	Version	Type	Subtype	To DS	From DS	MF	Retry	Pwr	More	W	O	Frame control

Duration: Tells how long the transmission of this frame will take, allowing other stations to set their NAV accordingly.

Addresses: Source/destination *in* a cell; and those of stations *outside* the cell when dealing with intercell traffic.

Sequence: Sequence number of this frame. 4 bits are used to identify a fragment of a frame.

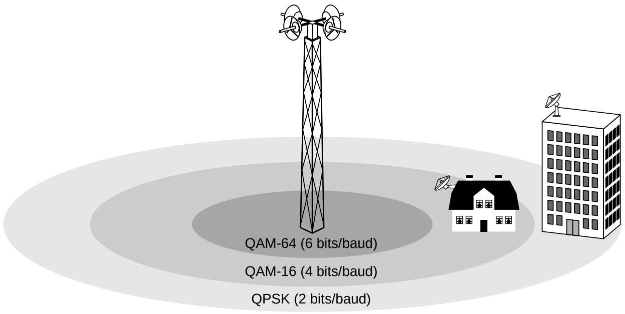
Broadband Wireless

Goal: Use wireless connection between buildings (e.g., avoiding the use of the local loop).

Observation: 802.11 is great for indoor networking, but is not that good for wireless communication between buildings:

- Buildings do not move, so much of the mobility stuff from 802.11 is not needed
- Several computers should be able to make use of the same "connection" (i.e., it should be broadband). 802.11 is intended to support one transmission at a time.
- Broadband connections can be supported by powerful radios (money is less of a problem), making power management less of an issue.
- We may need to cross longer distances, up to several kilometers.

802.16: The Basics (1/2)

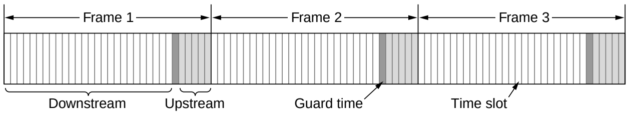


Note: At the physical layer, three different modulation techniques are used, each leading to different transmission rates. Typical rates are: 150 Mbps (short range), 100 Mbps (medium range), and 50 Mbps (long range).

802.16: The Basics (2/2)

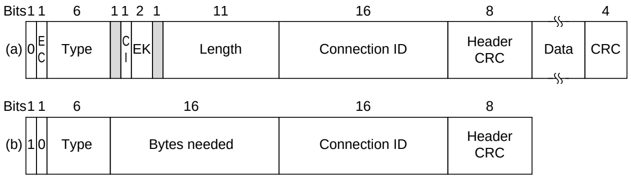
Observation: We need a flexible way to allocate bandwidth for downstream and upstream data. It can be expected that, just as in ADSL, an asymmetric approach works best.

Solution: Let the base station send out frames containing time slots:



Guard time: time used for stations to switch transmission direction. Number of upstream and downstream slots can be changed dynamically.

802.16: Frame Structure



Bits	1	1	6	1	1	2	1	11	16	8		4
(a)	0	EC	Type		CI	EK		Length	Connection ID	Header CRC	Data	CRC

Bits	1	1	6	16	16	8
(b)	1	0	Type	Bytes needed	Connection ID	Header CRC

Some observations:

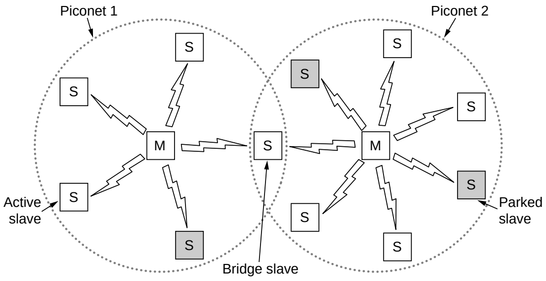
- 802.16 offers connection-oriented services, very unlike other MAC protocols
- Checksumming the data is optional: the phsyical layer uses error correction techniques and there are no facilities for retransmissions.

Bluetooth

Essence: Bluetooth is to allow very different (portable and fixed) devices located in each other's proximity to exchange information:

- Let very different portable devices (PDA, cellular phone, notebook) set up connections
- Replace many of the existing cables (headset, keyboard, mouse, printer)
- Provide better wireless connection (handsfree solutions)
- Provide wireless access to Internet entry points
- Relatively high bandwidth: 1 Mbit/second

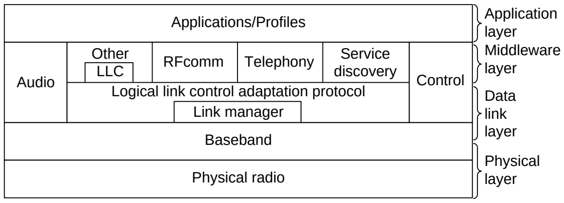
Bluetooth Architecture



Piconet: Group of devices with one **master** and multiple **slaves**. There can as much as 7 active slaves, but a total of 255 parked ones (i.e., in a power-saving state).

Scatternet: An interconnected collection of piconets.

Bluetooth Protocol Stack (1/3)

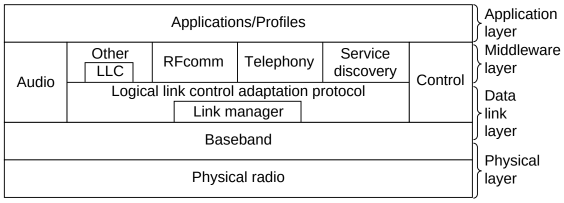


Radio: Operates in unlicensed band around 2.4 GHz using **frequency hopping**:

- Take data signal and modulate it with a carrier signal that changes frequency in hops.
- Hops for Bluetooth: fixed at $2402+k$ MHz, $k=0,1,\ldots,78$.
- Good to minimize interference from other devices (microwave ovens!)

Baseband: Core of the data link layer. Determines timing, framing, packets, and flow control. Provides synchronous and asynchronous data communication.

Bluetooth Protocol Stack (2/3)

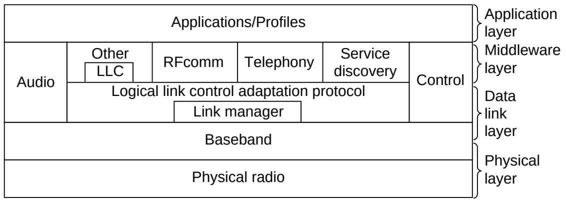


Link manager: Manages connections, power management

Logical link control: Multiplexing of higher-level protocols, segmentation and reassembly of large packets, device discovery

Audio: Handles streaming for voice-related applications

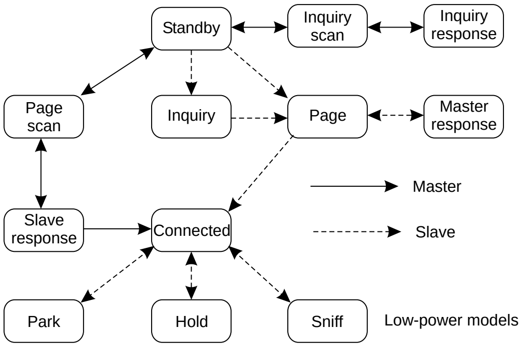
Bluetooth Protocol Stack (3/3)



RFCOMM: Emulate serial cable based on GSM protocol

Application: WAP and IrDA interoperability and all the fun stuff

Connection Establishment



Sniff: Listen for messages at fixed time intervals

Hold: No support for asynchronous data transfer; only synchronous data (i.e., voice). Allows the master to perhaps attend other piconet.

Park: Give up everything until master wakes you up.

DLL Switching (1/2)

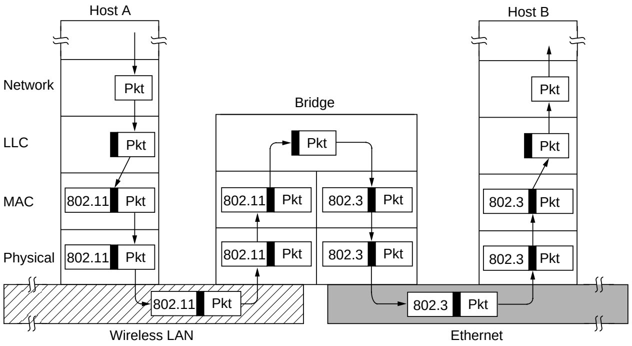
Basic idea: we want to interconnect a number of LANs, rather than having one big one. Two LANs are connected through a **bridge**.

- You want to let existing LANs (in departments, buildings, etc.) as they are. On the other hand, you do want to connect them.
- When an organization is spread over several buildings, it may be cheaper to have a different interconnect (e.g., infrared) than coax cacble. You may also have no choice.
- Splitting things up (rather than just tying things together) may be good for load balancing. There can be a file server per LAN.

DLL Switching (2/2)

- Physical distance sometimes precludes building one big LAN. For example, UTP 100 Mbps Ethernet can handle cables only up to 100 m. Not very much.
- The reliability can be improved: if one part goes down, the other LAN segments may still operate.
- For security reasons, bridges can check frames, and refuse to forward those that seem suspicious.

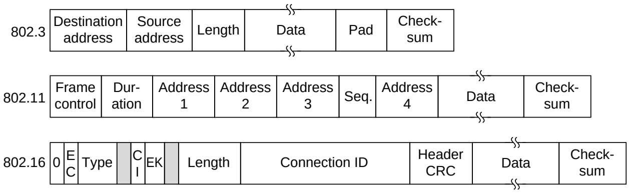
Bridges: Basics



- A packet is passed to the data link layer (LLC part)
- It is then passed to the MAC layer (specific access strategy)
- A bridge **converts** the stuff above the MAC layer, in the LLC layer

Bridging Problems

Problem: Really great – committees for 802.x invented different frame formats:



802.3	Destination address	Source address	Length	Data	Pad	Check-sum

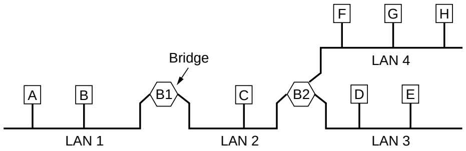
802.11	Frame control	Dur-ation	Address 1	Address 2	Address 3	Seq.	Address 4	Data	Check-sum

802.16	0	EC	Type	▨	CI	EK	▨	Length	Connection ID	Header CRC	Data	Check-sum

- We also have the problem of different date rates. If a higher speed LAN starts pumping frames on a lower speed one, we've got a problem.
- What should we do if the target LAN can't accept the frame length of the source LAN? Splitting a frame into pieces is often out of the question! (**WHY?**)

Transparent Bridges

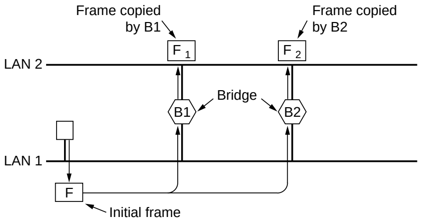
Main issue: Can we develop a bridge that interconnects LANs in a completely transparent way, i.e. seems to turn it into one big LAN?



- An incoming frame is simply forwarded to all other LAN segments connected to the bridge.
- Because an incoming frame contains the source address, a bridge can gradually get to know through which interface it can reach a host.
- Use a timeout mechanism to flush all knowledge a bridge has ⇒ it will gradually build up a fresh view again

Parallel Transparent Bridges (1/2)

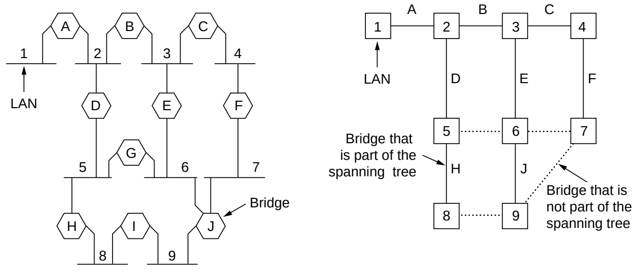
Problem: Sometimes LANs are connected by multiple bridges in such a way that we no longer have a tree, but a graph containing cycles ⇒ we can't just forward frames anymore.



Question: why would we use multiple bridges in such a way that we don't have a tree anymore?

Parallel Transparent Bridges (2/2)

Solution: let the bridges construct a **spanning tree** on their own.

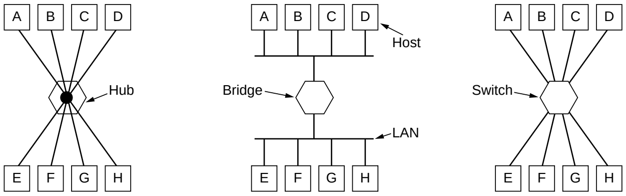

A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
Bridge
H
I
J
8
9
A
B
C
1
2
3
4
LAN
D
E
F
Bridge that is part of the spanning tree
5
6
7
H
J
Bridge that is not part of the spanning tree
8
9

(a) (b)

- Each bridge broadcasts its ID across the attached LAN segments. The lowest numbered bridge becomes root for that segment.
- A root bridge for a segment knows it can never be the root for the tree, if it finds out there's a bridge with a lower number.
- Bridges advertise their distance to the "real" root ⇒ that's how we build a spanning tree.

From Repeater to Switch

Observation: There's a lot of confusion when it comes to placing "connectors" in reference models:



(a) (b) (c)

Repeater: Amplifies incoming signal

Hub: Takes an incoming frame and passes it to all other ports

Bridge: Connects two or more LANs

Switch: Connects several computers (and routes frames between them)

From Router to Gateway

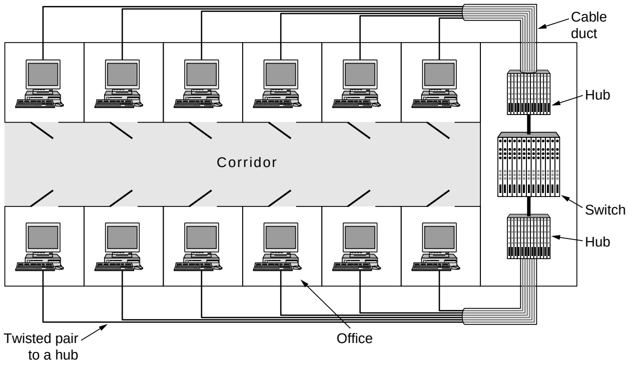
Routers: Placed in “classical” networks, and forwards packets to other routers

Transport gateways: Connects two networks at the transport layer: go from a TCP connection to an ATM transport connection.

Application gateway: Connects two different application protocols, such as sending SMS messages to a Web server, or connecting an X.400 mail system to an Internet-based mail system.

Virtual LANs (1/2)

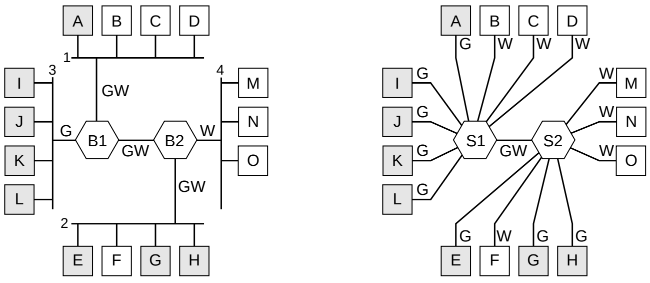
Observation: Many LANs are organized according to a physical division of workstations, possibly having multiple LANs connected by bridges for sake of better management (and security):



Problem: The physical organization may not correspond at all with what would seem **logically** the best organization (e.g., based on membership of a department).

Virtual LANs (2/2)

Solution: Adjust the bridges and switches such that an incoming frames is forwarded only to those outgoing ports that connect hosts (or LAN segments) belonging to the same logical group as the sending host:



(a) (b)

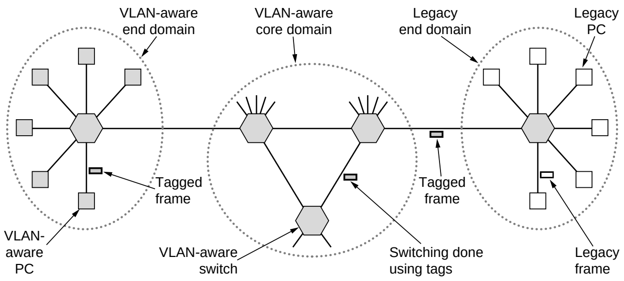
Identifying a VLAN (1/2)

Observation: It's not really important that you know to which group the sending host belongs, as long as the frame it is sending has the logical group encoded in it ⇒ add a tag to frame headers.

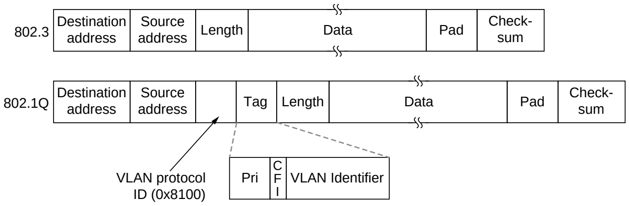
Problem: You don't want existing networks (Ethernet!) to break down. As a consequence, you need to apply a trick so that non-VLAN aware frames can be adjusted to one in which the VLAN is identified by a tag.

Basic idea: Let bridges and switches make the adjustment: add the tag when a frame comes in for the first time; remove it when it is sent to a non-VLAN aware host.

Identifying a VLAN (2/2)



802.1Q: VLAN Ethernet



VLAN protocol ID: value 0x8100, by which it is automatically interpreted as a type (correct according to the 802.3 standard)

VLAN ID: This is what it is all about

CFI: Bit indicating that the payload is an 802.5 frame that is being tunneled (token ring protocol)

Pri: So, finally, we can introduce QoS into Ethernet networks.