

Lecture 5

Multiple access data link

References:

- CN: 4.1–4.4, 4.7.1–4.7.5, 5.6.3
- RFCs: 826 (ARP), 903 (RARP), 951, 1542, (BOOTP), 2131, 2132 (DHCP).

Network(0234B)

Network(0234B)-5.1

Shared broadcast media

Suppose we want to build a LAN containing 100 computers.

- We can achieve it using **point-to-point** links. Clearly it is **infeasible** to build a **link between each of the 4950 pairs**.
- So we must arrange computers into a tree (or something similar), and the “internal nodes” must **route** traffic between computers.
- An alternative: use a **single** media, and connect **all** computers to it. We won’t need routing, but bandwidth must be **shared** among computers.
- If we simply use the air (i.e., wireless) to do the communication, that is indeed the only option.
- But if the media provides adequate bandwidth, there can (could?) still be reason to use it as a broadcast media: **the system is less costly**.
But for Ethernet it became too fast to be feasible. We’ll get back to this.

Network(0234B)-5.2

Why conventional methods don’t work...

- How to let the 100 computers share the bandwidth? At a first glance, **FDM, TDM or CDMA** might be a good solution.
Each computer has a channel for communication, thus a part of the bandwidth.
- But indeed it is **bad choice**: computers don’t talk like people. Most of the time there is nothing to send, and once it has something to send, it is usually something large—**bursty traffic**.
- FDM, TDM and CDMA are all based on **reserving a channel for each computer**, in our case we will downgrade our broadcast media 100 times, **even when none of the computers except one is talking!**
- What we want: a protocol which **if the channel is idle, a computer starting to use it can use most of the bandwidth**.
And when, say, 5 computers are talking, bandwidth is shared between only those 5 computers, not all the 100 computers.

Network(0234B)-5.3

Pure Aloha

- The first such system is **Aloha**: a Hawaiian radio system.
- The idea: when one has a frame to send, send it down the broadcast media **as if nobody else is using the channel**.
- If another computer is talking, this will **screw up the channel**. We just **wait for a random while** and try again.
So frames must be short, otherwise it gets too likely to be screwed up.
- In a sense, different computers **contend** for a slot for transmission.
- The frame contains **hardware addresses to identify the sender and the receiver**. The receiver thus know to get the frame.
- **Others discard the frame** after knowing it is for someone else.
- This seems quite chaotic. But in fact, under the correct situations this yields very good performance.

Network(0234B)-5.4

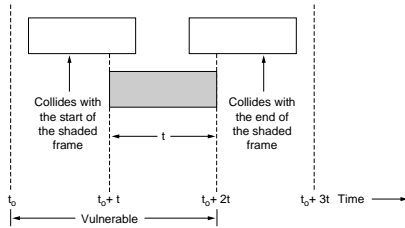
Assumptions

1. There are some **independent** computers, or **stations**, that are connected by a **single broadcast media**.
2. If two frames are transmitted simultaneously, **both will be garbled**. All channels know that there is a **collision**, including the sender.
Contrast this assumption to the CDMA assumption.
3. We say time is **continuous** if **transmission can start at any time**. If all computers share a common “clock”, and a frame can be sent at regular intervals, then we say time is **slotted**.
4. We say the channel support **carrier sense** if stations can detect that someone is talking and as a result doesn’t start talking until the conversation ends. Otherwise we say we have **no carrier sense**.
Primarily in wireless situations, as we will see.

Network(0234B)-5.5

Vulnerable period

- Pure Aloha has no carrier sense, and operate in continuous time. These are exactly the parameters that it performs the worst.
- Let's assume all frames are of the same length, t .
- A frame starting at t_0 will be corrupted if another sent beginning at a time between $t_0 - t$ and $t_0 + t$: vulnerable period.



Network(0234B)-5.6

Performance of pure Aloha

- Once we know what is the condition for a frame to get transmitted correctly, now we want to know the channel utilization.
- It is tempting to start with the average rate of frame arrival, but there is a complication: **garbled frames will be retransmitted**, so real rate of frame generation depends on the rate of collisions.
- Let's start from the middle: assume that rate of frames transmission (including both new and retransmit) is G frames per time t .
- Assume frame transmission times to be **completely independent**, the probability of no other frames in an arbitrary period of length τ is $e^{-G\tau/t}$. We call the distribution a "Poisson Distribution of rate G/t ".
- We need a clean period of $2t$, so probability of success is e^{-2G} .

Network(0234B)-5.7

What this means

- The **rate of successful** transmission is $S = Ge^{-2G}$ per time t .
- Ideally, every t unit of time we can transmit 1 frame, so $S = 1$. In fact, no value of G allow $S = 1$, so this never happen.
- The function is maximized when $G = 0.5$, when $S = 1/2e = 0.184$.
- What if the system generates more than 0.184 frames per time t ?
- Answer: the system **collapses**. Since S is smaller than the arrival rate, some will stay in the system in form of retransmissions. So G increases, and reducing S further, which in turn causes G to increase further. The whole channel becomes unusable. We call this a positive feedback loop.
- Conclusion: **channel utilization is at most 18.4%**, above which Aloha cannot operate.
Question: on average, how many times each frame is retransmitted?

Network(0234B)-5.8

Improvements: slotted time

- The key parameter control the formula of S is the **length of vulnerable period**. The longer it is (in relation to t), the less is the efficiency.
- One way to shorten the vulnerable period is to use **slotted** time.
- Now we don't have to worry about another station becoming active at some time **after** we start transmission, because then that station will start at the next slot.
- This **reduces the vulnerable period by half**, so the formula becomes $S = Ge^{-G}$. Following the same calculations we will get a maximum channel utilization of 38.4%.
- Still, this is far from 100% utilization. Is there any alternative?

Network(0234B)-5.9

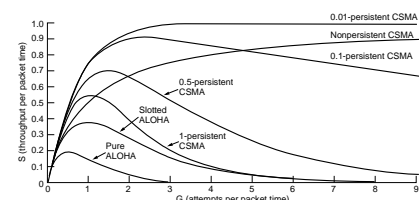
CSMA

- A completely different strategy: use carrier sense, if available.
- We allow any channel to **send at any time**, but right before transmission it **senses the channel**.
- If a channel is being used, it will **wait** until the end of transmission before sending. We call this protocol **Carrier Sense Multiple Access (CSMA)**.
- Assuming that the **delay between a channel starts sending and the other stations sense it is negligible**, it is impossible for a collision, ...
- except that if **two channels wait** at the same time, both will wake up at the same time, tries transmitting, and collide.
Under our assumption the probability that two channels starts at the same time without waiting is negligible.
- They will then use up another period t to discover the collision.

Network(0234B)-5.10

Persistence

- Possible improvement: not always **retransmit immediately** after a collision **only with probability of p** . **Otherwise wait for random amount of time** and try again. We call this p -persistent CSMA.
- Accordingly, the protocol in the last page is **1-persistent CSMA**. If $p = 0$, we always wait for a random amount of time, and the protocol is called **non-persistent CSMA**.
- Smaller p means larger delay, but better utilization. Mathematically:



Network(0234B)-5.11

Collision detection

- Another dimension that we can help is **what to do if a collision occurs**. So far the transmission continues to the end before an error is known. Typically, we know of an error when the checksum doesn't match.
- What if we can test for collisions? We can **go to physical layer** and require it to **receive** data from the channel as data is sent, and **test whether the timing and value of values sent** are the same. Manchester encoding allow this to be done much easier, which is the real reason of the generosity of Ethernet.
- If that is feasible, we can **quickly know a collision occurs**, and hence **stop before** the frame is completely sent: **Collision Detection**.
- We call this CSMA/CD (the "CD" stands for Collision Detection).
- How efficient is it? Under our assumption of negligible delay, the collision no longer waste us any time! But what is the **effect of delay**?

Network(0234B)-5.12

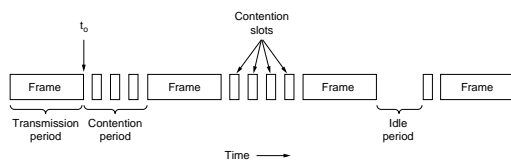
Delay and minimum length

- With (significant) delay, we need to rethink whether the sender really can **detect collisions** that occurs at the receiver.
- Suppose both A and B sends a short frame completely. The frames are considered garbled (someone at the middle will hear noise), but **if the message is shorter than the delay**, both A and B doesn't know it.
- If the frames are long enough, this won't happen. But how long?
- The worst case: A sends a message to B, involving a delay of τ . Just before it arrives, B sends a message to A, involving a delay of τ again.
- So it takes at most 2τ time to learn a collision. No frame should be shorter than this.
- Minimum frame length **depends on network diameter**. For 10Mbps Ethernet, the diameter is at most 2500m plus response for at most 2 repeaters. This is measured to be around $50 \mu\text{s}$, i.e., 500 bits. In practice, it is set to 512 bits or 64 bytes = 18 byte header + 46 bytes data.

Network(0234B)-5.13

Contention slots

- With this in mind, CSMA/CD communication can be visualized:



- Each contention slot is $51.2 \mu\text{s}$ long. Carrier Sense last for that long, after which the channel is considered **seized** by the station.
- It is **possible for the receiver to give an acknowledgement**: reserve the first slot after completion to the receiver.
- Ethernet doesn't do that. Receiver is free to drop frames, senders must add flow control measures in upper layers. Perhaps the designers think that Ethernet is too reliable for this to pay off.

Network(0234B)-5.14

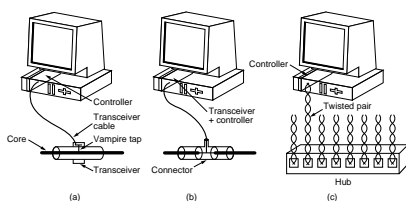
Ethernet: binary exponential backoff

- Now comes the last piece of puzzle: when to start transfer after seeing a transfer ends? (i.e., what is the "random time"?)
- If the random number is too large, we wait a long time, so response time suffers. If the random number is too small, we always get collision.
- Solution of Ethernet: wait for a **random number (from 0 to n) of contention slot** before trying. If someone seize the channel, repeat the whole process. If a collision occurs, try again (with a larger n).
- What is n ? Immediately after a non-colliding transfer in the channel, it is 1. **Everytime a collision occurs, the number is doubled**, until an upper bound of 1024. After 16 tries, the transfer is declared failed.
- Advantage: if few channels contend, a slot is found in a few tries. If many channels contend for a slot, soon we will be trying random numbers which are large.

Network(0234B)-5.15

Ethernet wiring

From early days to now...



- At beginning, the analog to digital logic is built near the wire.
- Later, it is found to be less costly to build this in the network card.
- At the end, the management of long inflexible wire is replaced by a simpler UTP, and a Hub make it looks like a broadcast media. By repeating everything it receive from one computer to everyone else.

Network(0234B)-5.16

Switched Ethernet

- As more and more computers are connected, **10Mbps bandwidth is not sufficient**.
- One optimization: at some strategic points, employ a special hub, called **switch** (or switching hub), that knows about the Ethernet headers.
- It **learns hardware addresses** of computers as frames comes across. In such a way, packets **needs only to be sent to that computer**, or the segment of the network containing it.
- The network is broken up into segments, each connecting to a switch.
- Within a segment, hub is used, so only 10Mbps is available. But each segment is of a smaller size, so the "many user" problem is solved.
- A switch looks much more like a **router** than a hub: it will **keep some buffers**, so even when there are multiple frame to the same segment, it needs not be treated as a collision.

Network(0234B)-5.17

Fast Ethernet

- 10Mbps soon becomes too slow. Fiber technologies (FDDI, Fibre Channel) are difficult to use, expensive, and very few want them. 100Mbps Ethernet is thus designed as an upgrade of 10Mbps Ethernet.
- Most things are not changed, except that it only allow UTP connections from hosts to hubs or switches, not coax. Hubs and switches must be used.
- Each connection consists of **2 UTPs** (i.e., 4 wires). One is for traffic from computer to Hub, the other for the reverse direction.
A hub connects the two UTPs together anyway, while a switch won't.
- Manchester encoding is not used. **100Mbps Ethernet uses 4B/5B encoding**: transfer 4 bits of data in 5 bits, which leaves some code for framing. (The clock is thus 125MHz instead of 200MHz.)
This is possible because of better synchronization technologies. 4B-5B guarantees that there is at most 5 consecutive 0's or 1's, so it is not really that hard.

Network(0234B)-5.18

Limitations and Gigabit Ethernet

- 10Mbps Ethernet **assumes a small diameter** for collisions detection.
- This fails **when network speed increases**. In particular, going from 10Mbps to 1Gbps, we must reduce the diameter 100 times accordingly, so it becomes 25m and no repeater.
- 25m is too short for most LANs. The answer: forget about contention.
- Hubs are completely replaced by **switches**. The network becomes **point-to-point**.
Maximum length is now determined by signal rather than protocol issues.
- Then we can say goodbye to CSMA on the wire. But... we still have radio channels which needs CSMA.

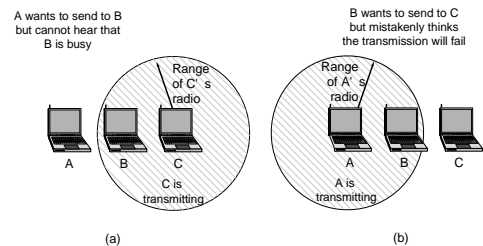
Network(0234B)-5.19

Wireless media

Wireless complications

Wireless has a range problem: signal strength falls as square of distance, so it fade out sharply. Each station can be listened by those in range.

This gives rise to a. Hidden station problem; b. Exposed station problem.



But there is a mode in Wireless LAN cards which uses CSMA anyway.

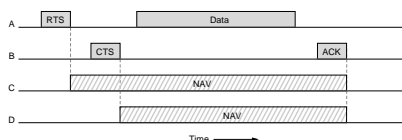
Network(0234B)-5.20

Network(0234B)-5.21

Virtual channel sensing (Collision Avoidance)

Noisy channel

- What is the real problem? Whether a channel is busy depends on the range of the **receiver**, not the **sender**!
If B is about to receive from A, people around B should shut up.
- Solution: **Virtual channel sensing**—A tells B that it wants to talk to it, and **how long it wants to talk**. ("Request to Send", RTS)
- **B** then sends a frame ("Clear to Send", CTS) **telling everybody except A to shut up**. The ones who shut up depends on B's range.



C is around the sender A, D is around the receiver B. Both remain silent. C remains silent anyway to increase S/N ratio for B.

Network(0234B)-5.22

Network(0234B)-5.23

- Note that in the picture of last slide, we have an ACK (acknowledgement), which is included in the NAV ("Network Allocation Vector") period ("shut up").
- Why in Ethernet we don't have ACK while in Wireless we want one? It's because **radio channels are much noisier than wired network**.
- Most real implementation allows a single frame to be **fragmented** to multiple ones, since a shorter frame is more likely to be in a "calm" period.
- The best size of fragment is determined by the current noise level.

PCF mode

- The IEEE Wireless standard (802.11) uses a protocol called **CSMA/CA** (Common Sense Multiple Access with Collision Avoidance).
- The protocol has two modes, one **Distributed Coordination Function** (DCF), which allows either 1-persistent CSMA or collision avoidance ("MACAW") to be used, with binary exponential backoff.
- There is another mode, **Point Coordination Function** (PCF). It requires a **central base station**, which will poll the computers asking them **whether they want to send something**.
- If one has something to send, it waits until the base station poll it, and at that time send the frame. There is **no contention**.
Again, things are simpler if everyone talk to the same person.
- How to tell the base station that you're there initially?
- Periodically, **base stations will send a beacon frame** telling what are the physical layer parameters, and invite parties to join.

Network(0234B)-5.24

Other services of 802.11

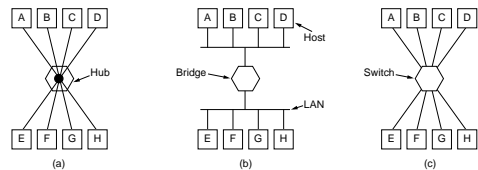
There are 2 more noteworthy features of 802.11...

- Mobility:** In PCF mode, a single base station talks with many mobile users. But they are "mobile", i.e., they move.
- Once it get far from one base station and get near to another, there need some way for a **hand-over** to occur. (Otherwise the user must manually stop the current connection, restart a new one, get a new IP address, restart all sessions, etc.)
A region served by a base station is called a **cell**.
- Security:** Wireless links are much easier to tap than Ethernet. So it allows the use of **WEP** (Wired Equivalent Privacy) to **encrypt** the message.
You shouldn't expect too much from the name anyway, since there is no privacy in "wired" network either. More about it later.

Network(0234B)-5.25

Repeating in broadcast media

Repeaters/Hubs vs. Bridges/Switches



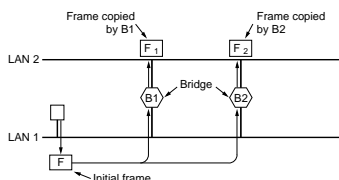
- Repeaters and hubs works on voltages. They connect things up, possibly amplifying the voltage.
- Bridges and switches works on frames. They use **hardware addresses**, to forward frames to the correct recipient.
If it connects LAN segments we call it bridges, while if it connects hosts we call it switches.
- Routers work on packets which use a different addressing scheme. They allow different data link technologies to interoperate.

Network(0234B)-5.26

Network(0234B)-5.27

Redundancy

- Sometimes we want **multiple bridges** to connect two LANs, so that if **one fail the other can take over**.
- Recall that bridges tries to learn addresses and forward packets to the other end if the address belongs there. So multiple bridges cause duplicated packets:

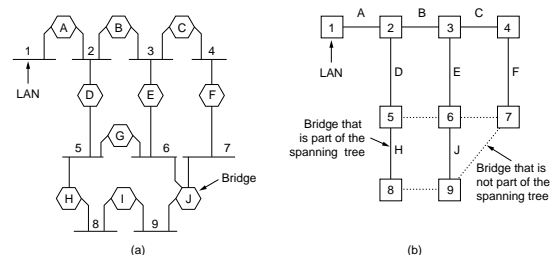


- Solution: **shut down** some of the links unless a bridge fails.

Network(0234B)-5.28

Illustration

In essence, we want to remove some edge to remove all cycles. So we end up with a spanning tree.



Network(0234B)-5.29

Distributed spanning tree

- But which bridge to shut down? How it is awoken? The process is called **distributed spanning tree computation**.
- At the beginning, every bridge **broadcast its serial number**, which is unique among all switches in the world.
- The smallest one will be chosen as the root of tree. It broadcast a message so that **every bridge knows which is the shortest way towards the bridge**.
- Once this is done, a **spanning tree** is constructed. Bridges that does not correspond to tree edges will not forward frames.
- This algorithm is repeated after some minutes so that when the topology changes (e.g., one bridge down), the spanning tree is recomputed.

Network(0234B)-5.30

Broadcasting in broadcast media

Local broadcasting and multicasting

- So far we concentrate on performing **unicast**, i.e., communication involving single sender and receiver.
- What if we want **broadcast** (sending the same frame to everybody) or **multicast** (sending to many people)?
- **Exactly the same mechanism** can be used. There is only one complication: **how the receiver knows it has to process** the frame.
- Answer: **reserve some hardware addresses** for broadcasting and multicasting. Upper layers configure the hardware to listen if they want.
- E.g., for Ethernet, all addresses with bit-7 on is for multicast, and the address with all 1-bits is for broadcast.
- Upper layers (e.g., IP) should provide addresses (e.g., IP address) that **maps to these address** if broadcast and multicast is needed.

Network(0234B)-5.32

Network layer configuration

- In last chapter, we see that when PPP starts, the network layer is configured using a link-level protocol **NCP**.
- This is also needed for broadcast media, unless a computer **hard-codes** its network layer parameters (IP address, etc) in its disk.
- **Broadcasting** provides the mechanism. When a computer boots, it can broadcast a frame: "**anyone know my IP?**". Some server can then reply the information in a frame to the booting host to start its network layer.
- The **RARP** (Reverse Address Resolution Protocol) allows exactly that.
- The problem: it requires a server to be configured **per LAN**, which must be configured to **translate hardware address to IP addresses**.
Broadcasts are not forwarded by routers, to avoid choking the whole internetwork with packets.

Network(0234B)-5.31

Network(0234B)-5.33

BOOTP and DHCP

- The **Bootstrap Protocol** (BOOTP) is made to sidestep the administration problem. It is later refined to **DHCP** (Dynamic Host Configuration Protocol).
- The idea quite funny: ask the newly booted hosts to **broadcast a network packet, before** the network layer is configured!
- This is possible since the broadcast address is **fixed**.
- The benefit: the administrator can setup a **relay server** to forward the request to a host in another LAN, so configurations are centralized.
- Many administrators want to support computers without configuration. To allow this, DHCP allows **dynamic allocation** of IP addresses.
- This adds the concept of a **lease**, a maximum amount of time for which the IP address remains valid. The host must **renew its lease** before the IP address expires to prevent it to be allocated to another host.

Network(0234B)-5.34

Resolving local network packets: ARP

- Network layer in general **won't use hardware address**, since they are clumsy to administrator, and provide no information for routing.
- E.g., In Internet, IP address is used. Suppose a computer want to send a packet and get convinced that it is a **local address**...
- There's a problem: **no router** will help the routing (since it is local), but it **might not know** the hardware address of that IP address!
- Answer: **Broadcast** a frame saying essentially "**anyone has this IP address?**". **ARP** (Address Resolution Protocol) allows that.
- Every computer receive the ARP request and tells the OS. Exactly one will **reply** its hardware address using a **broadcast**.
- Every hosts in the network will see the reply, and record it in the OS ARP table to avoid making ARP request later.
The entry expires in a few minutes, to cope up with changing topology.

Network(0234B)-5.35