49202 Communication Protocols

The Data Link Layer

Daniel R. Franklin

Faculty of Engineering & IT

May 3, 2023

Role of the data link layer

- The data link layer is responsible for delivery of data within a **single** network.
- It is the protocol layer which is responsible for adding structure to the physical layer so that information can be transferred from one host to another
- Key functions include:
  - **Framing** - organising a payload plus metadata into a frame structure which is suitable for transmission over some physical medium
  - **Addressing** - using a unique identifier so that a frame can be sent to a specific destination (and providing a means to discover the address of the destination)
  - **Error detection and/or correction/recovery** - providing one or more mechanisms for data integrity - at a minimum, this would include a checksum, and in some cases local retransmission capabilities.
  - If the physical medium is *shared*, **collision detection (and maybe avoidance)** might be needed
- Importantly - Layer 2 protocols are generally closely linked to an associated Layer 1 protocol

Framing

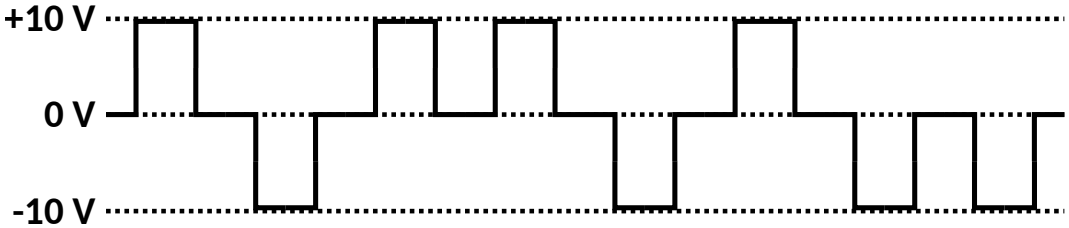
- **Frames** are the name used for packets at the data link layer
- A typical frame structure includes
  1. A header, which comprises:
     - A **preamble** - a fixed sequence of ones and zeros that a receiver can use for proper timing recovery and synchronisation
     - **Destination and source addresses** - these are known as **medium access control** or MAC addresses at the data link layer
     - Payload **data type** (e.g. IP datagram, ARP request etc.)
     - (possibly) **Data length** (although there are other ways this can be determined)
     - (possibly) several **flags** to indicate (for example) different packet types (e.g. in layer 2 protocols such as WiFi which include acknowledgements, RTS/CTS etc.)
     - (possibly) a **virtual LAN** (VLAN) tag
  2. The **payload**
  3. A trailer containing some form of **checksum** known as a **frame check sequence** for verifying frame integrity
  4. (possibly) an **End of Frame** indicator

But first: what about the physical layer?

- Design of Layer 2 is strongly dependent on the characteristics of the physical medium. So we should briefly discuss what the physical layer actually offers:
- Its job is to transmit an arbitrary stream of bits from one place to another
- But bits (binary digits) are just numbers, and numbers are an abstract concept - **information**. They are not a physical **thing**. How do you transmit an idea?
- We must represent the binary information using **symbols** to represent the bits, which are encoded in the physical properties of the medium.
- For example, using an electrical circuit, we can use the following symbols:
  - A zero is represented by a voltage of -10 V held constant for a period of 1 unit
  - A one is represented by a voltage of +10 V held constant for a period of 1 unit
  - Symbols must have a gap between them of one unit

Simple physical layer example - pulse coded modulation

- Using this simple representation, we can unambiguously transmit a binary sequence of arbitrary length
- We only need to **synchronise** to ensure that we measure the voltage of the waveform at the right moment in time. We can do this by waiting half a time unit following the start of either a positive or negative pulse.
- Using our encoding scheme, 10110100 could be represented with the following waveform:



Different physical layer media

- Real electrical signal encoding schemes may be much more complex - for example, Manchester coding
- Non-electrical signalling schemes also exist; We can also use
  - **Radio (or microwave) signals** (e.g. WiFi, Bluetooth, satellite)
  - **Optical signals** (either on optical fibres or in free space - such as inter-satellite communications in Starlink)
  - **Acoustic signals** (e.g. wideband ultrasound, for underwater signalling)
- For these transmission mechanisms, a **baseband** signal (such as in the preceding example) is used to modulate a carrier signal (e.g. the amplitude or wavelength of a beam of light)
- In each case, however, **information** is represented as a variation in some **physical property**

Relationship with Layer 2

- So, Layer 1 provides the means to transfer **bits** from source to destination; but how do we interpret these bits?
- A receiver may receive random signals that *resemble* actual data, due to noise or interference. How do we distinguish a real signal from random garbage?
- **Frame structure** must be added on top of the bit stream so that it can be interpreted properly by the receiver
- In most protocols, we don't know **when** a frame is due to arrive - so our first task is **synchronisation**
- This is achieved with the **preamble** - the first part of the frame structure

Layer 2 preamble

- For example, in Ethernet, the preamble is a series of 7 identical bytes containing the sequence `10101010`
- This is followed by a single byte called the **start frame delimiter** (SFD) containing the sequence `10101011` - note the last bit is a 1 instead of a 0
- A receiver does not necessarily need to detect the entire preamble - it just has to detect the one-zero-one-zero pattern. Why?
  - It may take some time for the receiver's amplifier to automatically adjust its **gain** to the proper level; while this is happening, some of the preamble may be missed. That's OK!

Layer 2 preamble

- If the pattern is detected, we may have the start of the frame. The receiver's **clock** synchronises with the edges of the received waveform so the bits can be accurately sampled (mid-bit-period).
- The receiver monitors the sequence until **two consecutive one bits** are seen - this is the SFD
- At this point, the receiver's sampling clock is properly synchronised, and knows that the frame will start on the very next bit. We can then directly read the rest of the frame header.
- However, if our expected SFD pattern is not seen, and the bit sequence deviates from the 1010 pattern, we revert to searching for this pattern again.
- Now we will look at the next part of a typical Layer 2 header: the addresses. But first, a quick diversion: standards!

IEEE 802 working groups

- It is essential that Layer 2 (and Layer 1) protocols agree on common signalling standards
- The IEEE has standardised a number of different variable packet size Layer 2 protocols and associated sub-protocols, which form the basis of most of the networking products and hardware in use today
- Each standard is managed by a working group. Some of these include:
  - 802.2 - Logical link control sublayer (common amongst several networking technologies - NOT used in Ethernet II)
  - 802.3 - Ethernet
  - 802.11 - WiFi
  - 802.15 - Wireless personal area networks, including 802.15.1 (Bluetooth) and 802.15.4 (Zigbee)
- Not all working groups are active (for example, the 802.2 group is disbanded)

Addressing

- Ethernet and WiFi (and Bluetooth) all use the same medium access control (MAC) address format:
  - It consists of 6 bytes, normally written as 6 hexadecimal numbers: e.g. `00:0a:f7:c6:44:50`
  - The first 3 bytes are called the organisationally unique identifier (OUI) and indicate the vendor (e.g. `00:0a:f7` = Broadcom) - larger vendors have multiple prefixes assigned
- Nominally, each MAC address is **globally unique** - $2^{48} = 281474976710656$ possible MAC addresses
- However, the MAC address of most modern network interfaces can be modified in software
- Mobile devices often randomise their MAC address when they connect to a new network to make tracking more difficult

Addressing

- In an Ethernet or WiFi frame, the destination MAC address immediately follows the end of the SFD. This allows the frame to be discarded early if the destination address does not match the receiving station's MAC address.
- Broadcast frames (e.g. ARP requests - see later) use a destination address which is all-ones (**ff:ff:ff:ff:ff:ff**).
- Multicast frames have the first (most significant) bit set to 1; regular unicast addresses have the first bit set to 0.
- The destination MAC address is immediately followed, in turn, by the source MAC address.
- The remaining protocol fields in the header will be discussed in detail later as they are different for each protocol - except for...

Address resolution

- How does a station X determine the Layer 2 (MAC) address is associated with a given Layer 3 (IP) address?
- **Address resolution protocol (ARP)** is a critical helper protocol which is used to perform this critical function. It sits directly on top of Layer 2 (no network or transport layer)
- A *broadcast* is sent by by the querying station X to all recipients in the local network, asking "**Who has** network-layer address corresponding to this IP address A.B.C.D?"
- This ARP request is a Layer 2 frame with
  - Source address equal to X's MAC address (e.g. `0a:0b:22:77:19:91`)
  - Destination set to the broadcast address (`ff:ff:ff:ff:ff:ff`).

Address Resolution

- *Only* Y, the owner of IP address A.B.C.D will send back a reply to X, with
  - Source field of the Layer 2 frame set to Y's MAC address
  - Destination equal to X's MAC address
- X and Y both cache the IP:MAC mappings, retaining the information for a period of time
- Once the ARP lookup is complete, X forwards its datagram to Y, encapsulated in an Ethernet frame with
  - Source set to X's MAC address
  - Destination set to Y's MAC address

The frame check sequence

- In general, the very *last* field, the **frame check sequence**, follows the payload. It is a checksum which is computed over the entire frame as the packet is transmitted
- The algorithm used for the checksum ensures that **small** changes to data being checksummed (e.g. single-bit inversions, deletions or insertions) will result in a **completely different checksum result**
- As the frame arrives at the receiver, the checksum is also calculated
- If the result differs from the value included in the frame, that means that something in the frame has been corrupted
- The frame will therefore be *discarded*
- What happens next depends on the protocols (more to come...)

Medium access control

- A key function of the data link layer is sharing access to the physical medium
- Some physical media are not shared at all - for example, in modern Ethernet networks, medium contention does *not* arise as each station has its own dedicated physical channel in each direction to a central **switch**, which routes frames between stations
- This was not the case in the original Ethernet, which used a shared coaxial cable to interconnect hosts. Similarly, in wireless networks, the medium is shared - a transmission can potentially be heard by multiple stations
- In this case, it is possible for **collisions** to occur when two stations start to transmit at the same time

Medium access control

- A station wishing to transmit a frame in an Ethernet network first listens to the medium to see if it is idle or occupied
- If it is occupied by another transmitting station, it will wait until the transmission is finished, plus another short interval called the **inter-packet interval**, before attempting transmission
- Otherwise, it transmits immediately.
- During transmission, the station listens to the medium
- Critically, there is a minimum frame length - packets shorter than this are **padded** to ensure that the transmission continues for the minimum period of time.

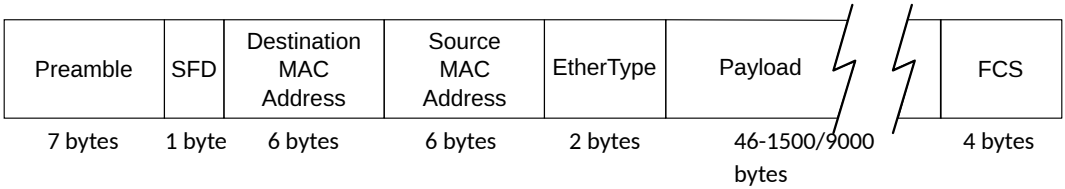
Carrier sense multiple access / collision detection

- In this way, if two frames are transmitted simultaneously, it is possible to guarantee that it is detected before the end of the transmission
- **A jam signal** is then sent, and then both stations wait a random time between 0 and then repeatedly retry
- For transmission N (where $N = 0$ was the original attempt),
  - $1 \leq N \leq 10$: choose k randomly from the range $0 \leq k \leq 2^N$
  - $11 \leq N \leq 15$:, choose k randomly with $0 \leq k \leq 2^{10}$ (i.e. 1024)
- This is known as **binary exponential backoff** - the probability of further collisions decreases exponentially with up to 10 retransmissions
- If collisions are still happening after $N = 15$ then the network has serious problems.
- This strategy is called **carrier sense multiple access** / **collision detection** (CSMA/CD).

Ethernet

- We will now have a look at how this is specifically implemented in Ethernet - and why it is no longer especially relevant
- Ethernet was developed by Xerox, DEC, and Intel in 1970s, and adopted (with minor changes) as a standard by the IEEE as **IEEE 802.3**
- 10 Mb/s, 100 Mb/s, 1 Gb/s, 10 Gb/s, 40 Gb/s, 100 Gb/s, 200 Gb/s, 400 Gb/s speeds available
- Originally bus topology (see the preceding section on CSMA/CD), now evolved into a star topology
- Can run over copper (e.g. 1000base-T/TX) or optical fibre (e.g. 1000base-F/FX)

Ethernet frame format



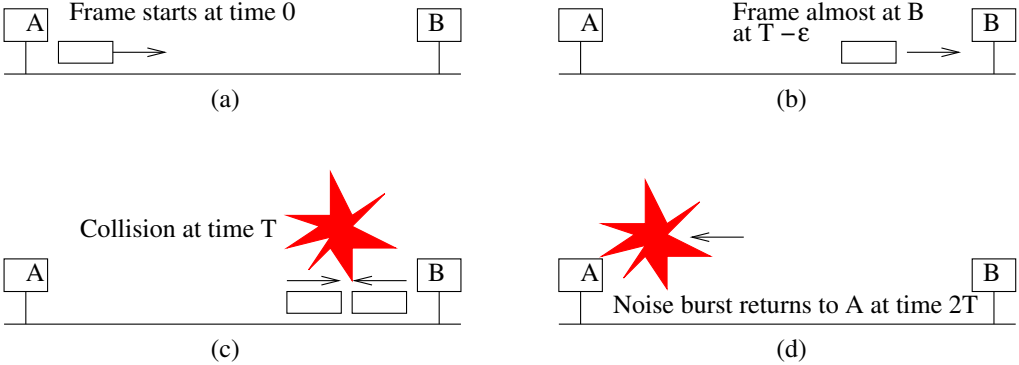
- The Layer 2 payload (MTU) may be between 46 and 1500 bytes in length by default (minimum size is to ensure that CSMA/CD can work!)
- If the **jumbo frames** option is enabled, the payload can be up to 9000 bytes in length - but you may run into issues with fragmentation
- For this reason, many networks stick with the default of 1500 bytes.
- If the value of the **EtherType** field is less than 1536, it indictes the frame length, and this is an old-style 802.3 frame; modern Ethernet II frames will have a value $>$ 1500 to indicate the type of payload (e.g. 0x0800 = 2048 decimal = IPv4). Yes, this is obscure - due to backwards compatibility!

Physical interface operation

- The physical interface ‘sees’ all incoming frames; however, it only accepts (passes to the network layer):
  - Frames addressed to its own unicast address
  - Frames addressed to the broadcast address
  - Frames addressed to any multicast address it has been programmed to accept

Minimum frame size

- Ethernet specifies a minimum frame size of 64 bytes (46 bytes payload) so that CSMA/CD can be guaranteed to detect collisions



Switched ethernet

- There are a number of problems with bus-topology Ethernet
  - Collisions become more frequent as traffic levels increase, drastically reducing efficiency
  - Access to the medium is not perfectly fair - depends on station placement!
  - Damage to the cable or to a single host can result in total loss of the network
  - These problems can be avoided using a **star topology** and a central *switch*

Switched Ethernet

- Ethernet switches only forward data to the destination, without occupying the entire network
- Incoming frames are forwarded to the port corresponding to the destination
- Data can be sent simultaneously in both directions - **full-duplex transmission**.
- Importantly, **collisions are no longer possible**.
- Broadcast/multicast frames are still sent to everyone
- Two mechanisms are used to implement Ethernet switches:
  - Cut-through; and
  - store-and-forward

Cut-through switches

- Cut-through switches only examine the preamble before forwarding the frame to the destination (based on a *look-up table* which is *learned* by the switch when devices are connected or disconnected
- Cut-through switches will happily forward corrupt packets
- Cut-through is cheaper and offers the lowest latency, at the cost of some corrupted frame transmissions

Store and Forward switches

- Store-and-Forward switches buffer incoming Ethernet frames, verify the CRC, then forward if good
- Slower than cut-through but prevent the forwarding of corrupted frames
- Modern switches are typically a hybrid between these two techniques

Modern Ethernet switches

- Hybrid switches initially configure paths through the switching fabric as store-forward paths
- If the checksum error rate is sufficiently low, the switch is changed to use cut-through
- In this case, a running total of CRC errors is maintained, without slowing forwarding in the cut-through switch
- If the error rate gets too high, the switch can revert to store-and-forward

Learning the association between ports and stations

- How do Ethernet switches learn which host is attached to which port on the switch?
- Suppose station A and station B are connected to an Ethernet switch on port 1 and 2, respectively, and Station A wishes to send a frame to station B (assuming it knows B's MAC address)
- Station A will construct an Ethernet frame, with Station A's MAC address as the source and Station B's MAC address as the destination address, and send it to the switch
- The switch will then record that a host with Station A's MAC address is attached to port 1. But it doesn't yet know where Station B is connected.
- Therefore, it floods the frame out of **all** ports except for port 1

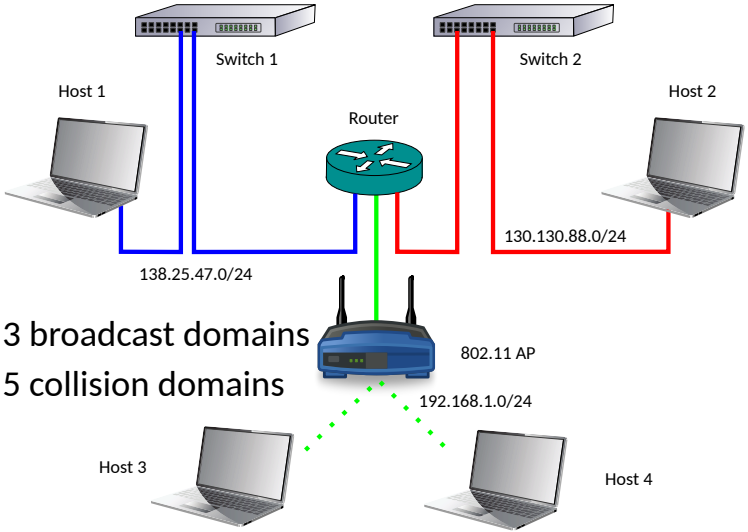
Learning the association between ports and stations

- As this is not a broadcast frame, only Station B will accept the frame. If it wishes to send a response back to station A, then it will construct a frame with its MAC address as the source and Station A's MAC address as the destination, before sending the frame to the switch
- The switch then learns that Station B is connected to port 2; it already knows where Station A is so it can forward the frame to its destination.

Broadcast domains and collision domains

- In a Layer 2 network, broadcast frames (such as ARP requests) sent to a switch will always be sent out to all ports on the switch
- Importantly, broadcasts do *not* leave a given Layer 2 network - they will not be forwarded via routers
- Therefore, we can say that routers separate *broadcast domains*
- Each port of a switch (or rather, the Ethernet segments between the switch and each host), is said to be a separate *collision domain* - even though collisions are not possible
- This is *not the case* for a WiFi access point, where the medium is shared - the network consisting of the AP and its client devices is a single collision domain, since any two stations can transmit simultaneously causing a collision.
- **This is also not the case for older non-switched Ethernet hubs**, which are strictly a Layer 1 device (amplifying and copying the signal rather than the frame).

Broadcast domains and collision domains



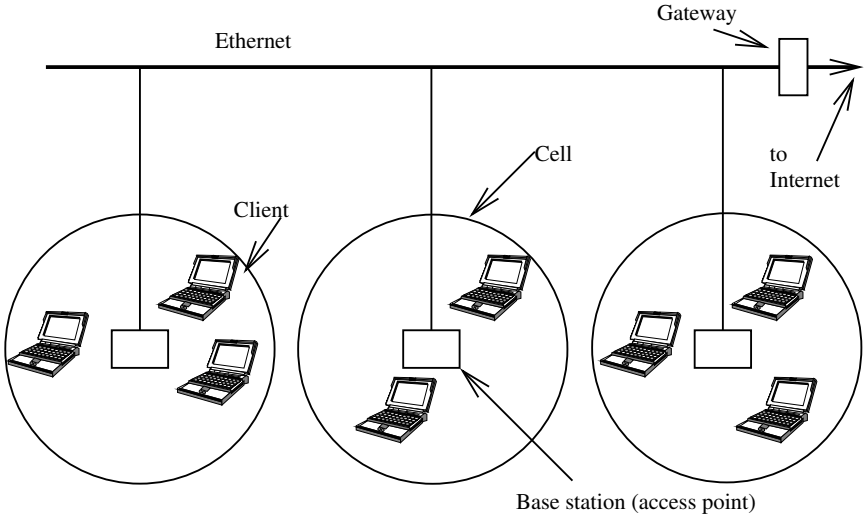
WiFi Networks

- IEEE 802.11 is a standard for wireless networking
- 802.11 uses various spread-spectrum techniques (Frequency Hopping, Direct Sequence and Orthogonal Frequency Division Multiplexing)
- Originally FH at 1 or 2 Mb/s, 802.11b is DSSS at up to 11 Mb/s (both 2.4 GHz)
- Modern WiFi standards operate at speeds of up to several gigabits per second, using a combination OFDM and other advanced modulation and coding techniques
- Forward error correction is employed to eliminate many errors at the physical layer - but errors still occur at a vastly higher rate than Ethernet
  - Since it is *far more likely* for frames to be dropped due to physical layer errors, 802.11 provides a retransmission mechanism for fast local repair.

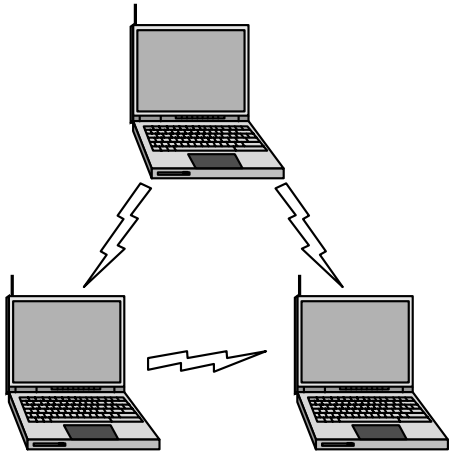
802.11 Access Modes

- 802.11 operates in either
  - *Infrastructure mode*, in which a user *associates* with a central *access point*; or
  - *Ad-hoc mode*, in which users directly communicate with each other.
- Infrastructure mode is the most common mode as it is more efficient and easier to manage

Infrastructure 802.11 Networks



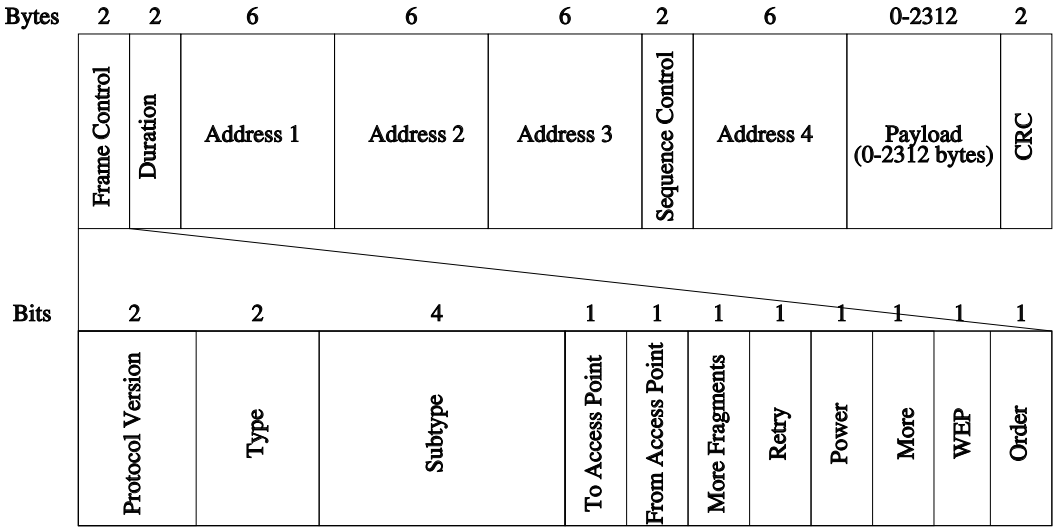
Ad-hoc 802.11 Networks



802.11 characteristics

- 802.11 transceivers are half-duplex: they can either transmit or receive, but not simultaneously.
- Hence is not possible to directly perform collision detection as with Ethernet
- If two nodes attempt to transmit to the same receiver node, the receiver node can hear at most one transmission
- Transmission range of a wireless node is limited, so ‘carrier sense’ is not possible: hidden/exposed node problem results

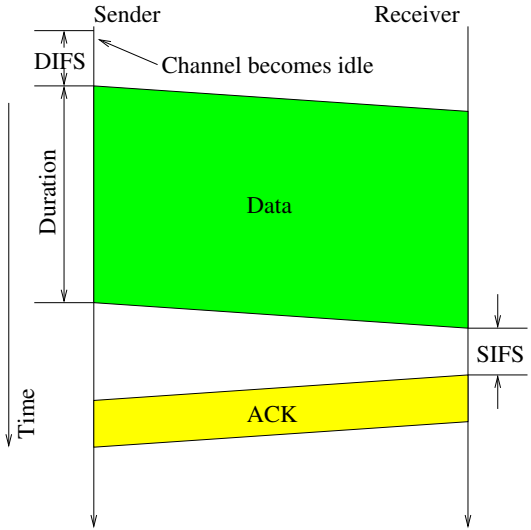
Frame Structure



Distributed Coordination Function

- 802.11 MAC uses DCF: a carrier-sense multiple-access scheme with collision avoidance
- As with Ethernet, the medium is sensed prior to transmission. When it is idle, the sender waits a short period (the distributed inter-frame spacing, DIFS) and transmits
- If the medium is idle, a frame will be transmitted. The frame includes a 2-byte length (duration) field indicating the expected length of transmission

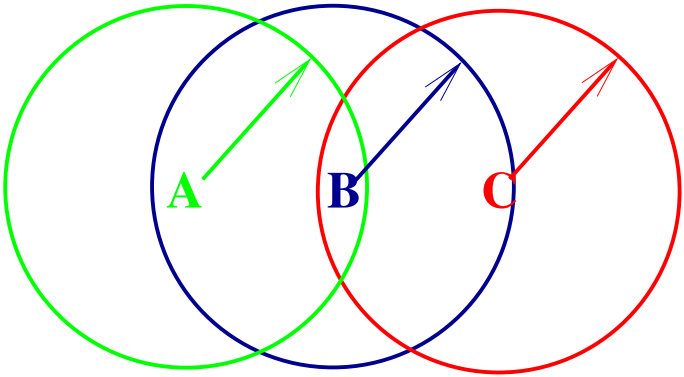
Frame Timing



Distributed Coordination Function

- If it has been successfully received, an ACK frame is sent after a short period (the short inter-frame space or SIFS)
- Hence 802.11 uses positive ACKs rather than Ethernet's passive collision detection - it is a stop-and-wait protocol
- Timeouts will result in exponential backoff, as for Ethernet

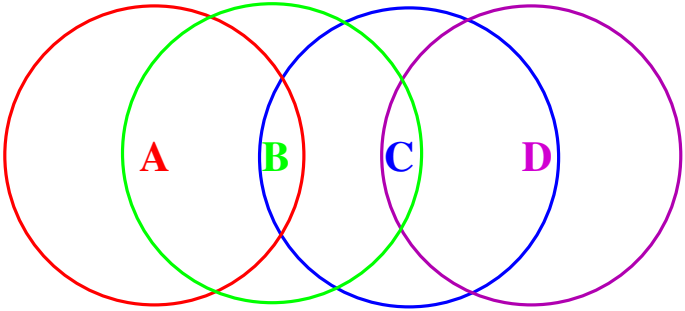
Hidden node problem



Hidden node problem

- Suppose that node A and C both wish to transmit to node B
- Node B can see both A and B, but A and C cannot see each other as they are out of range
- Thus it is possible that A and C transmit simultaneously, and can never detect that there has been a collision
- Node B receives a corrupted packet, which is ignored.

Exposed node problem



Exposed node problem

- Suppose node B wishes to transmit to node A.
- A will happily receive the transmission, as will node C, which is waiting to transmit to Node D
- C will think that transmission of its packet will interfere with B's transmission - and will wait until it has finished before sending
- However, C is out of range of A, therefore its transmission would have no effect - time and capacity are thus wasted.

RTS/CTS

- To address these problems, DCF adds optional RTS/CTS flow control
- Sender and receiver exchange control frames with each other before sender transmits data
- Operation:
  1. Sender sends Request to Send (RTS) frame
  2. Receiver replies with a Clear to Send (CTS) frame
- These frames indicate how long the sender wishes to hold the medium

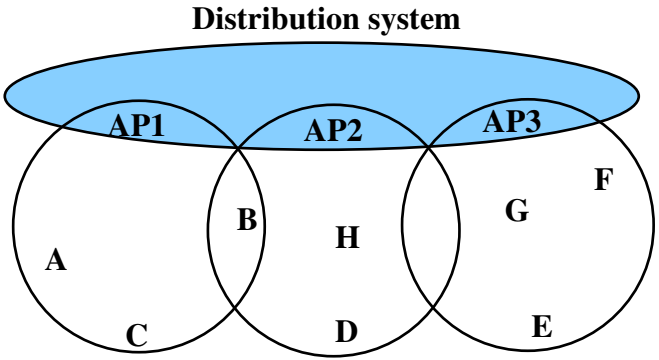
RTS/CTS

- If a node sees the a CTS frame
  - This node is close to the receiver
  - Should not transmit for the time indicated
- If a node sees RTS but not CTS
  - This node is **not** close to the receiver
  - Can transmit
- Solves both hidden node problem and exposed node problem.

Infrastructure Wireless LAN

- Each access point has limited coverage - the *Basic Service Set* (BSS)
- Each node associates themselves with an access point
- Communication within cell
  - enabled by access point
- Communication across cells
  - Enabled by distribution system

Distribution System



Association with an AP

- A node executes this procedure when
  - it joins the network
  - it is required to switch to a new access point, due perhaps to mobility

1. The node sends a probe frame
2. All access points within reach reply with a probe response frame
3. The node selects one of the access points, and sends that access point and association request frame
4. The access point replies with an association response frame

Mobility