1 GSW… Ethernet

The Ethernet family[1] of protocols are the most popular local area network protocols in the world[2]. This chapter gives a brief introduction to Ethernet, its many variants, and the structure of the frames that Ethernet sends over the network.

1.1 A Brief History of Ethernet

Ethernet was developed by Robert Metcalfe and David Boggs at Xerox's Palo Alto Research Centre and made public in 1976. The word itself was first coined in 1973, to refer to a project previously known as the 'Alto ALOHA net', based on the earlier ALOHA-based wireless network developed at the University of Hawaii. The original version ran at 2.94 MBit/s, with 8-bit destination and source address fields, but this version was never standardised, and is not in use today.

The first Ethernet standard was developed jointly by Xerox, Digital and Intel (and hence became known as the *DIX standard* [3]) and published in September 1980. In February of 1980, the Institute of Electrical and Electronics Engineers (IEEE) formed a committee they called 'Project 802' to establish standards for local area networks. They (eventually) produced a standard called 802.3 for a CSMA/CD based local area network.

Unfortunately, DIX Ethernet and the original 802.3 CSMA/CD LAN are not quite the same. The two standards can co-exist on the same network since they share the same access scheme, the same bit rate and modulation scheme, and the same format of destination and source addresses; but they have slightly different frame formats[4].

The initial speed of commercial Ethernet was chosen to be 10 Mbit/s. This was, at the time, amazingly fast. In the same year, Intel had just announced their latest start of the art processor, the 8080, which ran at 4.77 MHz, and one year later, the IBM PC was launched, with up to 640 kB of memory. A 10 MBit/s Ethernet could then transfer the entire contents of the computer's memory in just over half a second. (25 years later, PCs have one gigabyte of

[1] I say 'family' here, since there are a lot of different varieties of Ethernet. The common factor is that they all share the same frame format (more or less) and where a multiple access scheme is used, it's CSMA/CD. (Some Ethernet networks are just a series of point-to-point links that never have any collisions.)

[2] Although wireless LAN (popularly known as WiFi) is rapidly catching up.

[3] It's about the only pronounceable way to arrange the letters.

[4] Why are the two versions different? The official reason is that the other MAC standards being developed by the IEEE at the time (token bus and token ring) did not have a protocol type field (to tell the receiving node which higher layer protocol to give the frame's data to) in the MAC header, they used a sub-layer called the logical link control (LLC) layer just above the MAC, and it was the LLC that had the protocol type field. It was therefore inconsistent for the 802.3 standard to have a protocol type field in the MAC header. Unfortunately, that decision meant there were now two versions of Ethernet: the official IEEE 802.3 standard version that almost no-one used, and the DIX Ethernet version that almost everyone used. It seems a bit daft to write a standard intentionally different to what the accepted 'industry standard' was, but like much in communication standards, the real reason has more to do with politics, personalities and competition than with solid technical arguments. In order for a new technology to become an IEEE standard, you have to get the agreement of the majority of whoever turns up to the standards committee; and anyone can turn up. Sense has now prevailed, and both versions of Ethernet are now accepted by the IEEE 802.3 standard.

memory, and gigabit Ethernet is standard issue on most high-end motherboards, so it now takes eight seconds to transfer the entire memory contents.)

The original Ethernet cabling scheme was known as 10BASE5. ('10' for 10 Mbit/s, 'BASE' for Baseband and '5' for the maximum length of the cable: 500 meters). This used a large, shielded co-axial cable commonly called 'frozen orange garden hose', to which devices known as MAUs (media access units) were attached; the MAUs then being connected by shorter black co-axial cables (known as *drop cables*) to the computers via 15-pin D-type connectors. This was expensive.

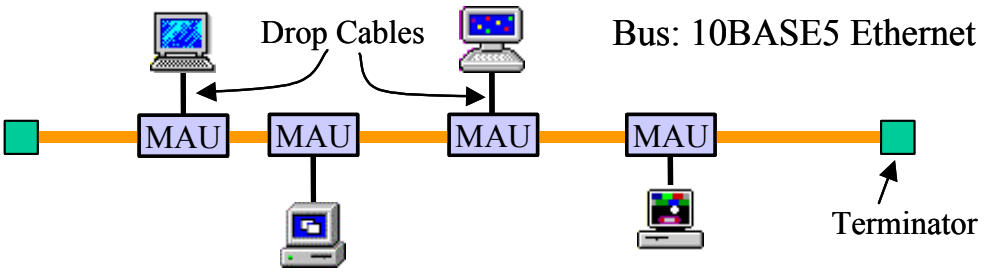


Figure 1-1 10BASE5 Ethernet Topology

In response to customer demand for a cheaper system, in the late 1980s and early 1990s 10BASE2 was standardised, which used a thinner coaxial cable, and linked all the computers directly using BNC T-connectors, without the need for an external MAU. This system became known as *thin Ethernet*, *thinNet* or sometimes *cheapernet*, and it was extensively used in PC-based networks. 10BASE2, as you might expect from the name, could only support up to around 200 meter segments (actually 185 meters), due to the higher attenuation of the cable.

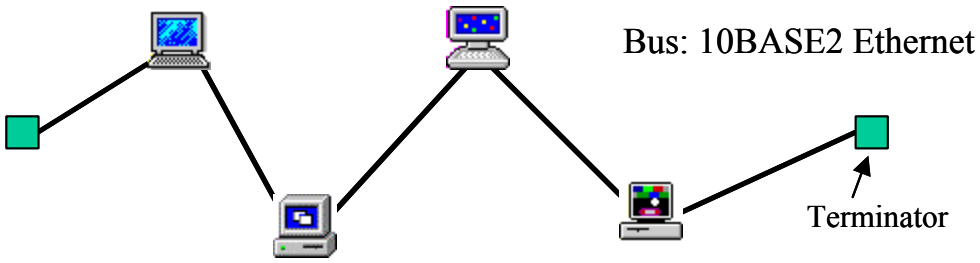


Figure 1-2 10BASE2 Ethernet Topology

Both of these cabling schemes used a bus topology. In the late 1980s, a start-up company called *SynOptics* developed an unshielded twisted pair (UTP) version of Ethernet called LattisNet that became the basis of the 10BASE-T standard ('T' for Twisted pair cable), and this used a star topology. (There is also a 10BASE-F ('F' for fibre), which uses fibre optic cable, again with a star topology.) In most cases, the twisted pair cable from the hub to the stations could be up to 100 metres long, big enough for most offices.

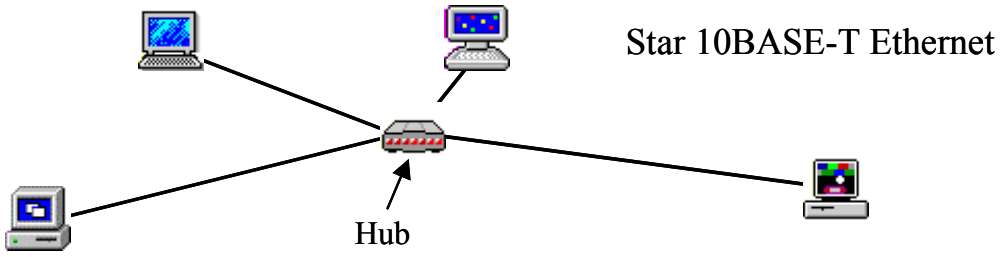


Figure 1-3 10BASE-T Ethernet Topology

There were many advantages to this star-topology over the original bus-topology, not least of which was that companies could install an Ethernet network using cabling that they already had in place: the telephone cables running to most desks. Even if their telephone cable was poor quality, and they had to install new cable, they could run the new networking cables through the same ducts as the existing phone cables, which was easier than trying to find a path for a coaxial 'bus' to go past each computer. In addition, every telephone installation engineer in the country was used to installing twisted-pair cable. This made the system very cheap and easy to install. The 10BASE-T system was standardised in 1990 and took off like wildfire.

(Another advantage of the star topology: with an intelligent central hub, cabling errors were easier to find and less harmful, since a broken cable no longer brings down the entire network. There are some more advantages too: see later in the chapter on Bridging and Switching in LANs. Inventing clever things for hubs to do has made a few people very rich.)

As computers speeded up, 100 Mbit/s versions of the Ethernet standard were developed. By now (the mid-1990s) the star-topology of 10BASE-T had all but replaced the older bus topologies, and there has been no interest in producing higher speed versions of Ethernet based on the original bus topology. Also, as hubs became more intelligent, almost all of them started to offer *full-duplex* links (so that stations could receive and transmit frames at the same time) and buffered frames arriving at the same time for the same destination, transmitting them sometime later when the destination was not busy. This effectively meant that Ethernet didn't use CSMA/CD any more: there weren't any collisions to detect. (About all that was left of the original Ethernet access scheme was the frame format.)

Several varieties of 100 MBit/s Ethernet were standardised, the most popular being 100BASE-TX, which again runs over unshielded twisted pair cable, but requires higher quality cable (known as category-5 or just cat-5 for short[5]) than the cable required for 10BASE-T (cat-3) due to the higher bit rates.

More recently still is the upgrade in speed to 1 Gbit/s, and again the star-based topology is preferred. 1000BASE-T (which can in theory operate over the same cat-5 twisted pair cable as 100BASE-TX, although better quality cables such as cat-5e and cat-6 are likely to give less problems in practice).

A standard for 10 Gbit/s Ethernet on unshielded twisted pair has just been developed (as of 2006), but can only reach 55 metres on cat-6 cable. A new cable type is currently being developed to allow operation at the standard distance of 100 metres, this may be called cat-6a.

1.2 Varieties of Ethernet

At the last count, there were twenty-two different physical layers in the Ethernet standard (both copper and optical fibre) supporting bit rates from 1 MBit/s to 10 GBit/s over distances from 15 metres to 100 kilometres, although many of these systems are comparatively rare nowadays and/or obsolete.

A quick summary of a few of the more important / interesting schemes:

[5] The category system of cabling is standardised by the EIA/TIA. The higher the number, the better quality the cable (lower loss, lower crosstalk). For more details, see the chapter on Copper Cabling.

- 10BASE5: The original 802.3 standard, 10 MBit/s, operating over expensive co-axial cable, with a maximum cable length of 500 meters, capable of supporting 100 stations on a single cable.

- 10BASE2: "Cheapernet", allows 30 stations on a cheaper co-axial cable, a maximum of 185 meters long. Still a bus topology, but stations are now directly connected to the bus.

- 10BASE-T: Uses much cheaper (cat-3) unshielded twisted pair cables, in a star topology, with cables up to 100 meters long.

- 100BASE-TX: Offers 100 MBit/s over cat-5 cables, again up to 100 meters long. This is probably the most common one as of the time of writing these notes.

- 1000BASE-T: The most common form of 'Gigabit Ethernet' claims to provide a ten-times improvement in speed over 100BASE-TX over the same cables (100 meters of cat-5), by using all four pairs to transmit the data in parallel.

- 10GBASE-T: The most recent Etherent standard to use unshielded twisted-pair cabling (cat-6), currently only supporting 55 meters, although the 802.3 committee are working on extending this to 100 meters. Still very expensive.

1.2.1 Hubs and Switches

At this point it's worth talking briefly about the difference between hub and a switch. Both operate at the centre of a star topology, however a hub is a very simple and cheap device that just passes on all data arriving at one port to all the other ports. All stations can therefore listen to all frames being transmitted on the network, just as if they were all on the same bus.

A switch is a lot more complex: it contains a fast internal bus and some buffer memory behind each port. Unicast frames (those transmitted to a single destination address, as opposed to broadcast frames which are sent to everyone on the network) are sent out to the destination only. (In effect, the switch is acting as a multiport bridge – see the chapter on 'Bridging and Switching' for more details.)

This can substantially increase the capacity of the network: for example, consider the diagram below, where station **A** is sending a frame to station **B**, and station **C** sending a frame to station **D** at the same time. If there was a hub at the centre of the network, these frames would collide, all stations would be informed of the collision by the hub, and both frames would need to be re-transmitted.

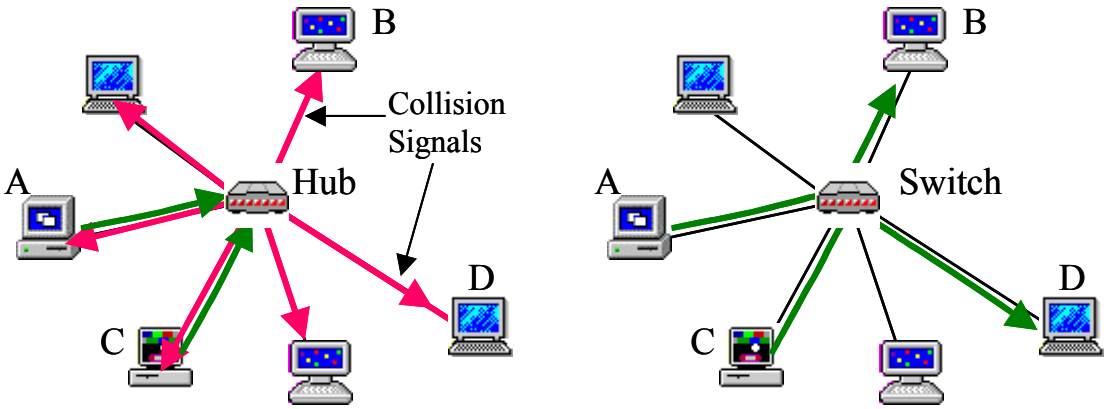


Figure 1-4 Hubs and Switches

However, with a switch, both frames can be send directly to their destinations at the same time, avoiding any collisions. It's almost like the switch is acting as a telephone exchange, sending the frames through to their destinations only, rather than acting as a bus, relaying the frames to everyone.

Thinking of a switch like a telephone exchange misses another important advantage of switches: even when two frames arrive at a switch with the same destination address at the same time, the switch can avoid a collision. The switch can store one of the frames in buffer memory until the other frame has finished being transmitted. The switch can prevent most collisions in this way (only when the buffers in the switch fill up does a switch have to report a collision).

Another advantage of a switch is in security: with a hub all frames on the network are received by everyone, and it's very easy to eavesdrop on someone else's email. With a switch, only those frames going to a specific address (or broadcast or multicast frames) are received, and eavesdropping is much more difficult.

I expect that most current, and probably all future Ethernets will use switches.

1.2.2 Half-Duplex and Full-Duplex

Hubs (and buses) operate with the stations in *half-duplex* mode: they can receive and transmit, but not at the same time. Any frame received by a station while the station is transmitting a frame is interpreted as a collision.

Since switches effectively prevent collisions, they allow stations to operate in *full-duplex* mode. In full-duplex mode, a station can transmit and receive frames at the same time, again increasing network capacity.

1.3 CSMA/CD and Random Backoff

As you might have realised by now, most modern Ethernet networks rarely use CSMA/CD or random backoff. They don't need to: they are connected to intelligent network switches that buffer and delay packets to avoid collisions. However, historically CSMA/CD is very important and it's quite interesting as well, so I think it's worth knowing about.

The original Ethernets, just like any carrier sense multiple access (CSMA) scheme, first listen to the signal on the bus cable to see if anyone else is transmitting a frame. If no-one is, then

the station is free to transmit. During the transmission, the station constantly monitors the state of the signal on the cable to determine whether it is the only station currently transmitting (this is the collision detection (/CD) part of CSAM/CD). If it detects another station transmitting, then the station registers a collision, and stops transmission (see the simplified[6] flow diagram in the figure below).

The station then waits for a random time, and attempts to transmit again. Obviously different stations must wait for different random times, otherwise the collision would just happen again[7]. The problem with CSMA/CD is that the amount of time you should wait (on average) is a function of the number of stations trying to send frames over the network: with more users attempting to transmit, the probability of a collision is higher, and each station should ideally choose a random number with a wider range to minimise the probability of further collisions.

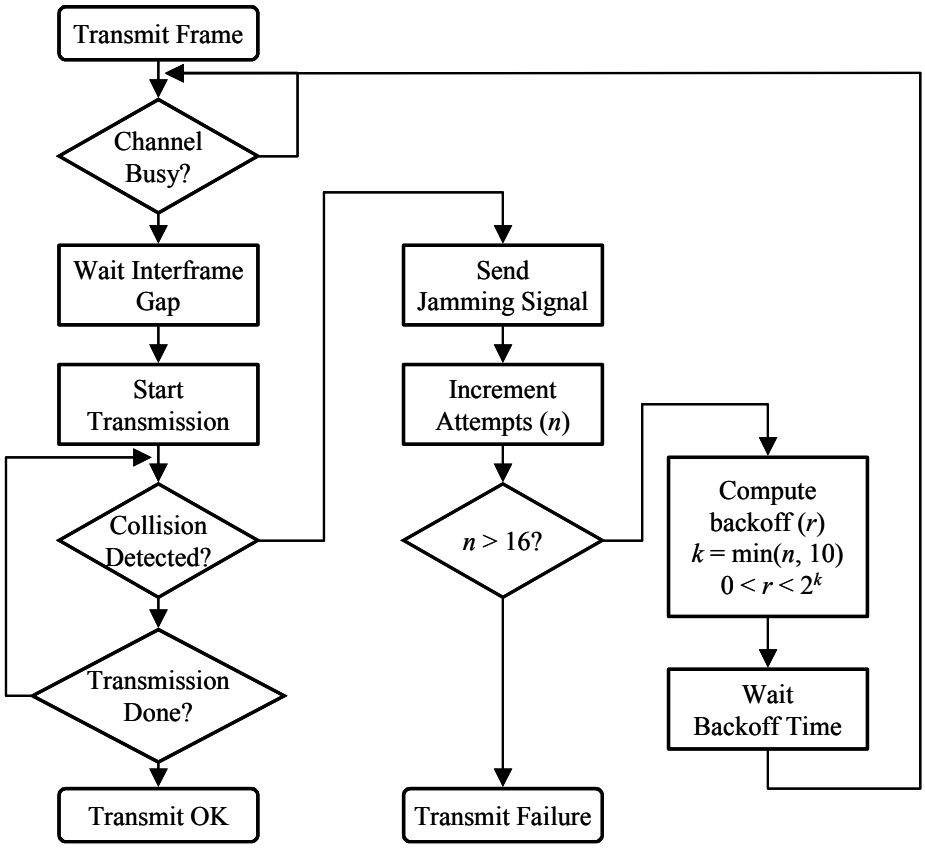


Figure 1-5 Simplified Ethernet Transmission Algorithm

Usually, Ethernet stations do not know how many other stations are also trying to transmit a frame at the same time. The solution is to use *exponential random backoff*. With this technique, if a station detects a collision the second time it tries to transmit a frame, it will wait for (on average) a longer period before trying again. If the second attempt is still unsuccessful, it will wait for a still longer time, etc. This effectively reduces the offered traffic to the Ethernet during times of very heavy load, and prevents the Ethernet from becoming clogged up with collisions at very high offered traffic rates.

[6] 'Simplified' since I've missed out some error conditions such as *late collisions*. For more details, see the standard.

[7] Hopefully you agree this is obvious – although I know of at least one computer manufacturer who got it wrong.

Ethernet stations with a frame to transmit wait at least 96 bit times after the end of the previous frame before attempting to transmit their next frame (this allows time for stations to prepare for the reception of the next frame)[8]. If they detect a collision, they will stop transmitting (after transmitting for at least 512 bit times to make sure all other stations have detected the collision), and then wait for a random time before attempting to re-transmit the frame. This random time is calculated as (512 bit times) r, where $0 \leq r < 2^k$, and $k = \min(n, 10)$, and where n is the number of attempts (so far) to transmit the frame. In other words, it waits either zero or one 512 bit-time period before the second attempt, then zero, one, two or three 512-bit times before the third attempt, then some random number from zero to seven times 512-bit times before the fourth attempt, etc.

Only after failing to transmit the frame without a collision 16 times does the Ethernet MAC give up.

1.3.1 The Channel Capture Effect

There's something rather interesting about this back-off scheme, something that the original inventors of Ethernet didn't know. There are reports[9] that when Bob Metcalfe and David Boggs first proposed the idea of Ethernet to Xerox it was almost rejected, since many people didn't think it would work. Indeed, for about the next twenty years, most analyses of the performance of Ethernet predicted that it would have a poor performance for short frames, with utilisations of around 37% (pretty much the same as slotted ALOHA).

However, real measurements on real networks indicated a much higher level of utilisation, up to around 90%. Ethernet was performing much better than the theory predicted it would. What all the analyses were missing became known as the *channel capture effect*.

A closer look at the flow diagram in the previous section reveals something rather interesting. Any station detecting a collision will tend to increase the amount of time it waits before trying again; however as soon as it succeeds, it will reset its counter of 'how many times have I failed', and therefore only tend to wait for a short time before trying to transmit its next frame.

Imagine a lot of stations trying to send a lot of small frames. Initially, there will be a lot of collisions, and the stations will tend to increase the amount of random time they wait until trying to transmit again. Eventually, one of them will succeed. Now, if this station (the one that's just succeeded) has more frames to transmit, it will only wait for a short time (since it's just reset its failure counter) while all the other stations are still waiting for much longer times.

The net result is that any station that has just succeeded to transmit a frame will probably try sooner than all the other stations to transmit its next frame, and therefore has a better chance of success. This process continues, with most of the capacity of the network being used by this one station (hence the name 'channel capture effect': this one station has effectively captured the channel and is preventing other users from using it).

---

[8] There's an exception to this rule: in gigabit and faster networks, *frame bursting* allows one transmitter to send a series of frames without relinquishing control of the medium. The inter-frame gap is still there, but it's filled with zeros. The first frame transmitted has an extension field if required: this means that collisions will affect this first frame, but not the rest of them. A station can reserve the medium in this way for up to 65,536 bit times before having to cease transmission.

[9] See Wikipedia article on Ethernet

This raises the utilisation of the network, at the expense of being very unfair: one station gets much more than its fair share of the capacity of the network. Various schemes were proposed to try and solve the problems of providing high utilisation and fair access, but by this time (the mid 1990s) most Ethernets used intelligent switches and collisions were no longer a serious problem, and none of these schemes ever made it into the 802.3 standard.

1.3.2 The Minimum Frame Length

Just like any CSMA/CD scheme, in order for Ethernet to work, it is vital that the transmitting station on the network detects any collision (otherwise it couldn't do the random backoff algorithm.) This means that the minimum frame length must be at least the round trip time from one end of the network to the other and back[10].

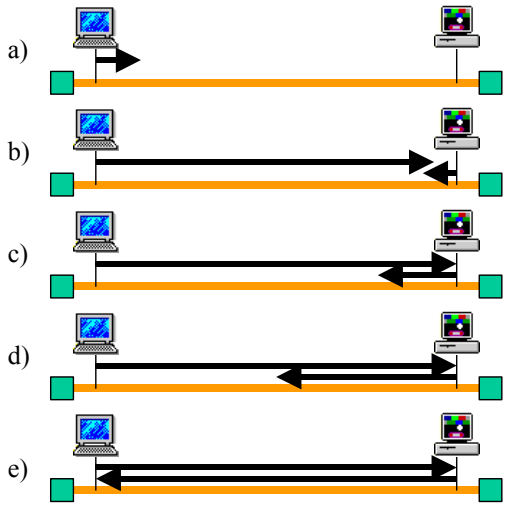
Suppose the bit rate is B, the minimum frame length $F_{\min}$ and the maximum length of the network $L_{\max}$. In the original 10BASE5 network, you could have four repeaters between any two stations, with 500 metre segments between each repeater, and the speed of propagation in the cable was 77% of the speed of light. That makes the total round trip time:

$$rtt = 2 \times 5 \frac{500}{0.77 \times 3 \times 10^8} = 21.6\,\mu s \tag{0.1}$$

which suggests a minimum frame size of:

$$\frac{F_{\min}}{B} > 21.6 \times 10^{-6} \tag{0.2}$$

[10] For more details see the chapter on Contention-Based Multiple Access. In brief: consider the diagram below.



At time a) the station on the left starts to transmit. Just before this signal reaches its destination (at the other end of the wire) the station on the right also starts to transmit a signal as well (time b). Very shortly afterwards (time c), the station on the right will receive the signal from the first station, and detect a collision. However, it is not until the signal from the station on the right reaches right back across the network to the station on the left (time e) that this station detects the collision; and in order to detect a collision, the station on the right must still be transmitting at this time. Therefore, the minimum frame length must be at least the round trip time from one end of the network to the other and back.

where B is the bit rate, here 10 MBit/s. Therefore:

$$F_{\min} > 216 \text{ bits} \tag{0.3}$$

This is a bit of a simplistic calculation, since it doesn't include the effects of delays in the repeaters themselves, which can be up to 19 bit times. Going through eight repeaters (four on the way, and four on the way back) then adds another 152 bits to the minimum frame length required, making a total of 368 bits. It's also important to ensure that there is a reasonable amount of time to detect the collision, which adds a few more bits as well. In order to allow some margin, the minimum frame size was set to 512 bits (not including the preamble or the starting delimiter).

If the packet given to it to transmit by a higher layer protocol is shorter than this, Ethernet has to make it longer or 'pad it out' by adding additional bytes.

1.3.3 How to Detect Collisions

There are two main techniques for detecting collisions in Ethernet networks. The first is applicable to the original bus-topology networks, the second to star-topology networks. In bus-topology networks, stations transmit by placing a known current on the bus, and then monitoring the voltage. As long as only one station is transmitting, then the voltage on the cable will never go above or below IR where I is the current placed on the cable, and R is the resistance of the cable. If the voltage ever becomes greater than this, then two stations must have been transmitting (putting current onto the cable) at the same time, and that means a collision.

One problem arises from the attenuation of the cable. If the bus cable is too long, then by the time the signal from one transmitting station gets to another transmitting station at the other end of the bus, the signal could be too weak for the collision to be detected. This is what limits the maximum length of an Ethernet cable to 500 meters. If a longer network is required, repeaters (which boost the signal levels) are required between 500 meter *segments* to make sure that any two stations more than 500 meters apart can detect each other's collisions. This is also part of the reason why 10BASE2 (CheaperNet) cannot support segments as long: the cheaper cable used has a greater attenuation.

In star-based network it is up to the hub to detect collisions, and this is easy: a collision happens when a signal arrives from more than one station at the same time. If this happens, the hub transmits a jam signal to all attached stations, informing them of the collision.

In a switched star-topology network, the switch usually prevents any collisions, by buffering and storing packets until the output port is available, so there's no need to detect collisions.

(There are more details about these methods in the chapter on the Ethernet Physical Layers.)

1.4 Frame Formats

Originally, the DIX Ethernet and 802.3 frames looked like this:

DIX Ethernet

Preamble	SD	Destination	Source	Type	Data	FCS
7	1	6	6	2	46-1500	4

Original 802.3

Preamble	SD	Destination	Source	Length	Data	FCS
7	1	6	6	2	46-1500	4

Figure 1-6 – DIX Ethernet and Original 802.3 Frame Formats

There is also a minimum inter-frame gap specified, of 96 bit times (9.6 μs at 10 Mbit/s) to prevent one station transmitting continuously. Note the difference: the DIX Ethernet had a type field after the source address; the original 802.3 frame format had a length field here. The DIX frame format was much more popular, and eventually the 802.3 standard was expanded to allow both uses for this field:

802.3-2005

Preamble	SD	Destination	Source	Length / Type	Data	FCS
7	1	6	6	2	46-1500	4

Figure 1-7 –802.3-2005 Frame Format

How can a receiver know whether the two-byte field is a type or a length field? Doesn't this cause a problem? Well, fortunately, no: see the discussion about the type / length field below for more details.

(Note – if this packet is part of a virtual private network (VPN) there will be an additional field of four bytes in-between the source address and the protocol/length field to set a priority for the frame and identify the VPN. In addition, there may be an extension field after the FCS for short packets in gigabit and faster networks to ensure all packets are long enough for collisions to be detected.)

1.4.1 The Preamble

The purpose of the preamble is to allow the receivers at all other stations to lock on to the signal strength and get their clock recovery circuits in phase before the start of the important data. It's the equivalent of coughing or saying "excuse me" to attract attention before you say anything important. It consists of 56 bits of alternating ones and zeros, starting with a one, and ending with a zero. This signal is chosen because it is particularly easy to recognise and synchronise to[11].

1.4.2 The Starting Delimiter

The task of the starting delimiter is to give a clear signal to the receiver that the frame is about to start. It should therefore be different from any pattern of bits in the preamble. It's eight bits

[11] The original Ethernet used Manchester coding, so an alternating series of ones and zeros had transitions only in the middle of the bit times, making it very easy for the receiver to determine where bits start and stop.

long, and contains the bits 10101011. That makes it exactly the same as any byte in the preamble except that the final bit is a one, rather than a zero.

1.4.3 The Destination and Source Addresses

All project 802 LANs share the same addressing scheme. There's much more about these in the next chapter (on Project 802 LANs and the LLC), including the unfortunate issue of bit ordering. Other than noting that these addresses are 48-bits long, I won't say anything more about them here.

1.4.4 The Length / Type Field

This is where the original 802.3 standard and DIX Ethernet diverged. 802.3 had a two-byte length field here, whereas DIX Ethernet defines a protocol type indicating which higher-level protocol sent the frame.

Fortunately, this doesn't cause any problems, since the length of the data field is always between 46 and 1500 bytes, and the higher-layer protocols all have numbers greater than 1536. So, if the number is less than or equal to 1500 it must be a length field, greater or equal to 1536, and it's a protocol type number. In this way, both frame formats can operate together on the same network, although if you want to send information from one station to another, you have to ensure that both are using the same frame format. (Most hardware can do either: the driver software sets up what to use this field for.)

What's the advantage in having the length field there? Supporters of the length field claim that it makes it easier to check the FCS in hardware, which can speed up the reception of frames, and also, there is less chance of a frame fragment being mistaken for a correct, although shorter, frame. In addition, a receiver can tell how long a frame is when it starts to arrive, so it knows not to start receiving a frame if it has insufficient buffer space available, allowing the reception of shorter frames afterwards.

If the two bits following the source address are set to the 'magic number' 0x8100 then this indicates the presence of a virtual LAN tag field after this field, with the 'real' length or protocol number following this: the frame is therefore extended by four bytes. This allows virtual LANs to be set up sharing the same physical LAN.

1.4.5 The Data

Can be anything. Ethernet, as an access layer protocol has no idea what any of it means, it doesn't speak the language. (For the details of what's inside this block we need to look at the higher layer protocols: see next chapters.) However, the data field must be at least 46 bytes long (so that collisions can always be detected) and can't be any longer than 1500 bytes long (some maximum length is needed, otherwise one station could hog the bus, and other stations would have to wait an unreasonable length of time to transmit their frames). If it's less than 46 bytes, then some padding[12] is added.

[12] Often just a lot of zeros, enough to make the data field 46 bytes long. This can add a significant overhead to Ethernet when it is asked to transmit a lot of very short frames.

(There is an unofficial extension to the Ethernet frame structure that allows longer frames sizes (called *jumbo frames*) with payloads up to 9000 bytes long, but this is non-standard, and does not work with all equipment.)

1.4.6 The Frame Check Sequence (FCS)

The FCS is a 32-bit cyclic redundancy check (CRC) sequence, formed by treating the entire frame from the destination address to the end of the data field as a long binary number, then dividing it by a 33-bit generator number; the FCS is the remainder after this division (see the chapter on Error Detection for more details). The receiver can then determine whether there have been any bit errors in the received frame, with only a very small probability of an undetected error. If an error is detected, the entire frame is thrown away and forgotten[13].

1.4.7 The Extension Field

When gigabit Ethernet was introduced, the minimum frame length (64 bytes) was no longer long enough to ensure collision information was received by all stations up to 200 meters apart. However, a lot of software assumed that 64 bytes was the minimum frame size, and this software often didn't know (and should not need to know) the speed of the physical layer. So, for these faster networks, another field is added at the end (after the FCS): the *extension field*. This brings the minimum frame length up to 512 bytes on gigabit Ethernet networks.

1.5 Tutorial Questions

1) True or false:

a) 1BASE5 runs at 1 MBit/s over 500 meters of cable.
b) 100BASE-T2 runs at 100 MBit/s over 200 meters of twisted pair cable.
c) The last field in an Ethernet frame is always the frame check sequence (FCS).
d) The minimum length of any Ethernet frame from the DA to the FCS is 64 bytes.

**2) Which current 802.3 standard has the shortest range? Which has the longest? In both cases, how fast are the networks going?

3) What would the Ethernet preamble and starting delimiter look like if written as a long hexadecimal number?

4) "The Ethernet MAC detects collisions and re-transmits colliding frames, therefore it is an example of a reliable link" - discuss whether this statement is true or false.

5) With a minimum frame length of 512 bits, what is the maximum length that a 100 MBit/s CSMA/CD network could be, using a passive optical hub (i.e. a non-switching hub) and fibre with a propagation speed of two-thirds of the speed of light?

[13] Forgotten in the sense that it is not passed further up the protocol stack, and no negative acknowledgement is passed back to the sender asking for a retransmission. What might happen is that the receiver will increment a counter that's counting invalid frames received for management purposes: if there are suddenly a lot of FCS errors in a network, the network manager would probably want to know: this could indicate either some faulty wiring or a lot of electrical interference.

**6) A 1 Gbit/s CSMA/CD network connecting 25 users operates over a star topology with a maximum cable lengths of 100 meters. What is the minimum frame length required to ensure collisions are detected? (Assume that the speed of propagation in the cables is 2/3 the speed of light.)

Is this a problem for gigabit Ethernet? If so, how does gigabit Ethernet solve the problem?

7) What does an Ethernet MAC layer do if asked to transmit a frame with a data payload of 5 bytes? And why?

8) Give five reasons why the star topology of 10BASE-T was more popular than the bus topology of 10BASE5.

*9) What are the maximum and minimum numbers of symbols transmitted onto the wire by a 100BASE-T Ethernet station during the course of one frame?

**10) What is "jabber", and why is jabber suppression useful? (A good place to look is the Ethernet standard, available free on-line from the IEEE.)

**11) If a (non-switched) 10BASE-T hub detects a collision, it transmits a 'jam' signal to all ports, indicating that there has been a collision. This jam signal is not specified in the standard, and can be any series of ones and zeros… with one exception. What's the exception?

*12) A network layer protocol has a 12-byte long packet to send, so it gives to an Ethernet, which then adds 34 bytes of padding before transmitting the frame. The frame arrives at the other end. How does the receiver's Ethernet layer know to remove the padding? In other words, how can the receiver tell the difference between a short padded frame and a network layer packet that really is 46 bytes long?