2 OSI Layer 1 – The Physical Layer

2.1 Goal of this section

The goal of this section is for readers to learn what the Physical Layer of the OSI model is and what "interfaces" it uses to higher (there is no lower!) layers in the OSI model. Further, readers will learn what matters about the Physical layer, and understand some examples of Physical layer technologies.

2.2 Purpose

The physical layer is about getting individual bits of information from A to B. This can be **unicast** (one A to one B), such as when sending a signal down a wire, or broadcast (one A to lots of Bs) for example using a radio antenna. Layer 1 is the only layer that actually involves travel over distance.

Typical distances covered by the physical layer vary substantially, everything from a few metres to a few Astronomical Units:

- A few metres. Cat 5 Ethernet cable from your Computer to your cable modem wired connection.
- Tens of kilometres. Optical fibre connecting repeater stations under the English Channel
- A few AUs. Deep Space Network 2.3GHz channel used between Voyager 1 and Earth.

2.3 Interfaces Up and Down

There is no interface to the layer below. The Physical Layer is the bottom of the OSI stack.

The interface to the layer above is a bit stream – that is to say the layer above provides Layer 1 with an ordered sequence of bits, and layer 1 delivers that bit-stream to the layer above at the destination. Layer 1 is not responsible for guaranteeing uncorrupted or ordered delivery of the bit stream.

2.4 Services

Layer 1 typically provides the services listed below.

2.4.1 Bit delivery

Note that the bit-stream is digital, but layer 1 is responsible for handling the real physical (analogue) world and so layer 1 handles digital-analogue conversion. Also includes low-level endpoint synchronisation – for example clock synching.

2.4.2 Low-level Buffering

Layer 1 does not expect to receive (or collect) bits *exactly* as quickly as it can send them on the wire / broadcast them so typically each end maintains a small internal buffer.

2.4.3 Rudimentary connection error detection

Layer 1 does not provide any form of error correction or attempts at retransmission if there is an error, but physical problems in the link itself are usually detected at layer 1. For example if an Cat 5 cable is cut and no longer provides a loop of wire, or if an optical cable is snapped and one end no longer receives any light, it is typically layer 1 that can spot this and provides an alert up to higher layers of the stack.

2.4.4 (De)multiplexing

Consider the case where a number of low bandwidth links converge with data that is all destined to go down one high bandwidth link. In this case, the information is **multiplexed** over the higher bandwidth link.

Multiplexing in layer 1 is achieved in two ways:

- **Time division multiplexing** (TDM) simply allocates some number of time-slots, and rotates around the incoming traffic (the first 0.01 seconds transmits the first 1 second of data from source A, the next 0.01 seconds transmits the first 1 second of data from source B, etc.) To do this the switch must be able to buffer the relevant amount of traffic, and does introduce some latency to the transmission.
- **Wave division multiplexing** (WDM) is used in optical networks and runs multiple signals down the same cable at different wavelengths/frequencies. Note that for optical networks it is also theoretically possible to use polarization to provide two independent light waves running orthogonally.

The perfect speed layer 1 switch is simply some very carefully arranged micro-mirrors that deflect laser light from an optical fibre to another optical fibre. This is called an Optical-Optical or Pure-optical switch. (More likely, optical switches receive and terminate the light and then a second laser issues a new beam. These are called Optical-Electrical-Optical switches, and again the goal is to minimise the number of clock cycles taken between receiving and issuing the light.) Note that Pure-optical switches can handle wave-division multiplexing, and the network as a whole needs to be configured such that the incoming light sources to be multiplexed are already the correct frequency.

Multiplexing can also occur by terminating the Layer 1 links and feeding the information up to higher layers of the stack (and having those layers do the multiplexing and then originating a combined feed of data), but this is slow compared to staying in Layer 1.

2.5 What does "Better" mean for Layer 1?

The goal of the physical layer is to get bits from A to B. In this case, better is really measured in the following:

- Bandwidth – which at layer 1 is really speed of clock cycle and ability to multiplex. The line rate is the maximum bandwidth that you can get down a link, and this is the physical limit at which anything using this interface can interact – so layer 1 performance is critical.
- Reliability. There is no error correction at Layer 1 so non-corruption is key.
- Cost. More than other layers – physical building costs are critical.

Note that in the real world no Layer 1 link has perfect reliability – it is always *good enough but not more.* This is because at layer 1 reliability and bandwidth trade off against each other. If your endpoints can detect and turn analog line signals into digital 1s and 0s at a particular clock speed, then why not run the clocks and the line a little bit faster at the cost of blurring your signals slightly? If your light detectors in an optical fibre can distinguish two different frequencies of light with different signals *comfortably*, why not squeeze in another frequency in between? The specifications for Layer 1 push the limits of the physical infrastructure, and as the physical infrastructure gets better, the specs update to squeeze more out of it. We're always at the edge of what is possible.

2.6 Signalling

Whatever the means for transferring information, you typically have a system that can be in 2 states, that is time synchronised in some way.

The simplest signalling system is to have Hi and Lo states corresponding to 1 and 0 respectively (Hi could be potential difference across an electrical loop or a laser light being on). However, many forms of interference that occur along a wire effectively add or remove a ~constant "height", so instead we prefer to use the transitions between high and low to mark our information flow, instead perhaps using a change in height to be 1 and stable height to be 0.

Note that in this case, a long string of 0s is a problem. Without a way for clocks at each end of the link to provide some kind of time synchronisation, very, very slight differences in frequency could result in one sending say 25 0s and the other receiving 26. Two ways to handle this are Manchester encoding, or using a special (unlikely to occur in natural signalling) pattern to replace strings of 0s with something else.

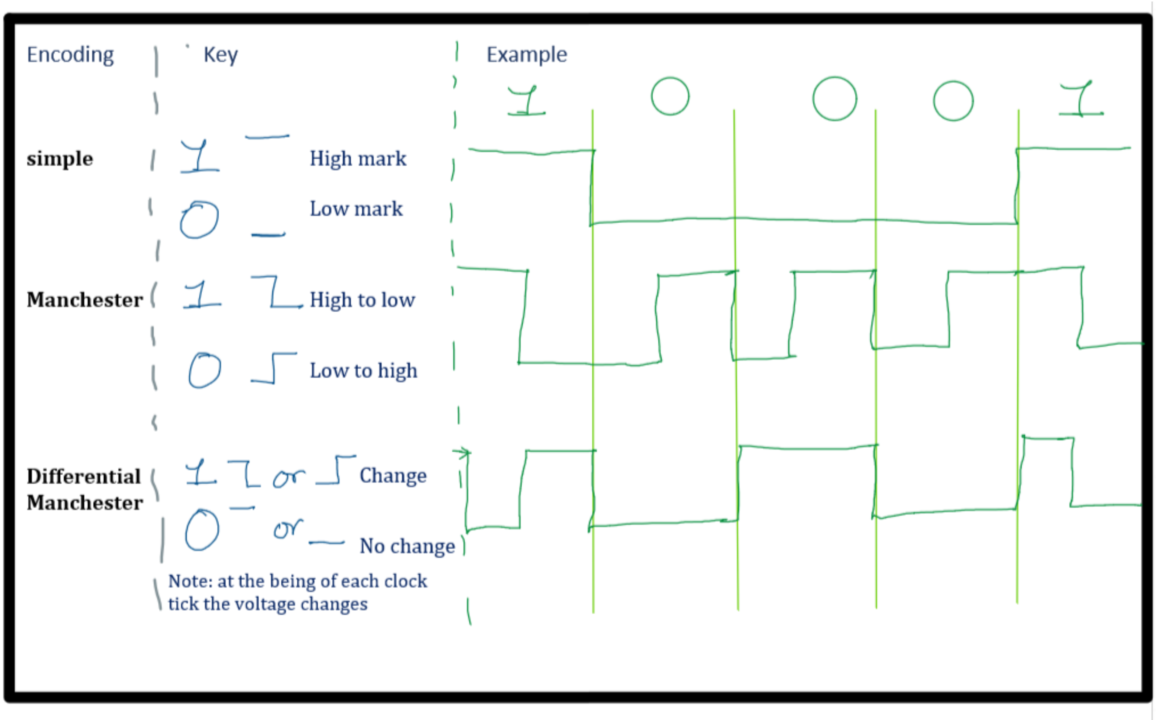
2.6.1 Manchester Encoding

With Manchester Encoding, a 1 is represented by the change High->Low and a 0 is represented by the change Low->High. Between the clock ticks, the signal is reset to the correct Low or High starting state for the next signalled bit to occur.

Differential Manchester encoding works on a similar principle. Here, the signal is always changed on the boundaries between clock ticks, and during that clock tick, a 1 is signalled by no change, and a 0 is signalled by a transition during the clock tick.

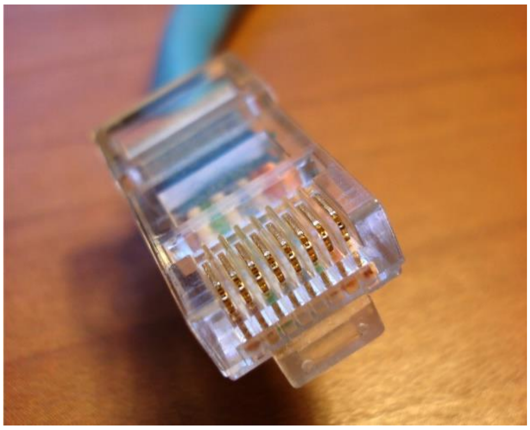
Note that to do either of these, you need transmitters/receivers that can change the state of the line twice every clock tick. If you could get away without these encodings, you could theoretically double your data rate with the same transmitters/receivers.

An example of both: (H/T to Tof Adigun-Hameed for the picture)



2.7 Examples of Layer 1

2.7.1 Category 5 ethernet cabling (*and the buffers in the networking cards to push and pull data off the wire.*)



Physically these types of connection are a pair of wires that can form a loop when they’re connected together at each termination. Signals along those wires are electrical (voltage) pulses driven around the loop by one end and picked up by the other.

To minimise line interference, the endpoint signals half the voltage difference +ve down one side of the loop and half the voltage difference -ve down the other side. The far end subtracts the two to get the full difference. Any interference on the line is likely to affect both cables in the same direction, which is cancelled by the subtraction.

There are 4 loops in the cable, travelling next to each other for the entire cable. To minimise interference across those loops (cross-talk) each pair is twisted and they’re twisted at different frequencies (so on balance any cross-talk roughly cancels out).

One “standard” cable (RJ45 connectors as in the picture) consists of 4 loops (pairs of cables). Historically, the different loops provided transmission in different directions, so people building networks had to carefully use either straight or “cross-over” cables to make sure that across a network, the transmitters were correctly connected to the receivers. For Gigabit Ethernet and above, transmission happens both ways on all lines and this is no longer required.

2.7.2 Optical Fibre (*and the lasers/detectors/buffers at the ends.)*

Physically, an optical fibre is a thread O(µm) of glass, surrounded by lower refractive index material (and then all that surrounded by a protective casing). Light is sent down this at wavelengths in the ~1270-1610 nm spectrum. The advantages of this over copper include high bandwidth and immunity to electromagnetic interference. It is also advantageous in terms of security, both because it is very hard to tap a fibre without the endpoints noticing, and also because glass is of no resale value to thieves.

Extremely high bandwidth can be achieved down an optical fibre through Wavelength Dimension Multiplexing (covered above). This is typically up to 160 channels (wavelengths) spaced across the spectrum mentioned above.

2.7.3 Other examples of Layer 1

- Normal land-line phones have a copper loop. That loop and the software to open and close that loop in your phone and at the exchange are layer 1.
- A child's tin can phone with a taught stretched piece of string is raw "layer 1"

2.8 Electro-magnetic radiation through air (or a vacuum) and the appropriate antennae to broadcast/receive are layer 1Problems / Troubleshooting / Tools

Physical layer problems tend to be physical. These include workers putting spades through cables, people unplugging things and kinked cables. More subtle problems include clocking issues – where the two ends of a link don't agree on when to measure the values being signalled, and multiplex leaking, where information for one timeslot leaks into the next – or interference atsimilar wavelengths on the edge of what a receiver can distinguish.

Physical layer monitoring will usually tell you that something has gone wrong – but as a user, you'll just get an error, and no further diagnostics. Physical layer debugging tools are any physics based investigation tool – such as multimeters, or more likely a purpose built line tester. Often physical links have some kind of indicator of connectivity. For example most RJ45 ports have connectivity lights, indicating that the far end of the cable is plugged into something.

2.8.1 Anecdote: the weirdest bug we ever came across

(Not examinable 😊). One of the more idiosyncratic pieces of networking work that our test team worked on was interoperability with an SLC-96. This was a thing the size of a fridge that was the equivalent of the phone connection box at the end of your road – but for use in rural US and covered in inch-thick armor plating, because random things in the middle of nowhere sometimes get used for target practice. It brought together 96 rural telephone lines to multiplex to a single line and was cutting edge technology in the 1980s.

While testing, we discovered that if you sneezed loudly while on a phone call, while your call was unaffected, any phone call on the line that happened to be physically connected to the SLC-96 *next* to your line was immediately disconnected.

What was going on? The two ends of the multiplexed link were configured with *different* multiple-0 encodings (special patterns of bits to send when you would otherwise be sending all 0s (silence on a phone call), and worse, these were of different lengths. This meant that when there was silence, one end of the link was overwriting the time-division multiplexing on the link, providing tiny amounts of random noise glitching into the next call (not enough to hear). The sneezes, however were providing a sharp boundary between "silence" and "maximum noise", and that maximum noise data, when provided at the start of the next time-slot matched the special bit sequence that signalled "hang up".