Networking
CM30078/CM50123

Russell Bradford

2022/23

### 1. Networks

#### Ethernet

What are the physical encodings of bits on a 10Mb/s Ethernet?

A simple way would be 0V for 0 and 1V for 1, running at 10MHz

But this has a number of problems

### 2. Networks

#### Ethernet

1. An empty network and a stream of 0s looks the same

And so you could not do carrier sense

2. Bits need to be synchronised to prevent drifting out of step (was that 1000 or 999 0s?)

3. A long stream of 1s is a steady 1V: this is electrically a bad design, an average 0V is best

To connect devices easily you need an AC signal, not a DC one

### 3. Networks

#### Ethernet

So 10Mb/s Ethernet uses a *Manchester Encoding*

* Split the time interval for a bit into two parts
* Low then high voltage is a 0
* High then low voltage is a 1

So the average is 0V

-0.85V for low, +0.85V for high

This voltage is a compromise: a bigger voltage gives a more robust signal that will travel further, but it uses more power

### 4. Networks

#### Ethernet

Easy to synchronise: the transit through 0V is the middle of a bit

This does double the frequency of the signal to 20Mhz

We can use Cat 4 (or better) cable for this

Manchester encoding solves the above problems neatly and actually simplifies the hardware needed

It is described as *self clocking*, as the reading end does not need a clock to determine where the bits are

### 5. Networks

#### Ethernet

What of 100Mb/s Ethernet?

We can’t use even Cat 5e cables with Manchester as it is only specified to 100MHz, and we would need 200MHz

### 6. Networks

#### Ethernet

Instead we start by encoding 4 data bits as 5 physical bits in a *4B/5B* encoding; e.g., 0000 become 11110

| Input | 4B/5B | Input | 4B/5B |
| --- | --- | --- | --- |
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

With some control patterns, e.g., IDLE 11111.

### 7. Networks

#### Ethernet

Hasn’t 4B/5B made things worse: 5 bits where there were 4?

But now we use a *three* level physical encoding *MLT-3*

This has +, 0, and - levels ($\pm 0.85$V), again using transitions to encode bits

### 8. Networks

#### Ethernet

Transitions are cyclical

|  |
| --- |
| - to 0 |
| 0 to + |
| + to 0 |
| 0 to - |

A transition marks a 1, no transition marks a 0

The 4B/5B translation ensures that every chunk of 5 symbols has at least two transitions, so average voltage is roughly 0

E.g., input 0000, with no transitions becomes 11110 with four transitions

### 9. Networks

#### Ethernet

An example. Hex value 0E = 0000 1110



MLT encoding

### 10. Networks

#### Ethernet

Some words:

A physical representation is called a *symbol*

Symbols need not be binary

And need not represent a whole number of bits

The *baud rate* is the number of symbols per second

### 11. Networks

#### Ethernet

100Mb/s Ethernet runs at up to 31.25MHz for a symbol rate of 125MBaud: all 1s output (IDLE) is four transitions (- to 0, 0 to +, + to 0, 0 to -) per cycle ($4 symbols/cycle×31.25MHz=125MBaud$)

This has a symbol rate of 125MBaud for a data rate of 100Mb/s: 80% efficient or 1 physical symbol is $4/5=0.8$ bits

### 12. Networks

#### Ethernet

For Gigabit Ethernet 1000Base-T: 8 bits become $4×3$ physical bits in a continuously changing encoding (not a table lookup)

Each 3 bit chunk is encoded using transitions between 5 levels (PAM-5)

Over all four pairs in the cable simultaneously, in both directions on all pairs

10Gb Ethernet uses a PAM-16 over a very complicated coding (*Tomlinson-Harashima Precoding*)

(SATA and USB 3.0 use 8B/10B; USB 3.1 uses 128B/132B; etc.)

### 13. Networks

#### Ethernet

And then there is Ethernet over optical fibre…

**Exercise** Read about the physical encodings that are used in fibre

### 14. Bridging

Some time ago we talked about bridges joining networks

ARP bridging is fine for joining a pair of small networks, but less so for larger collections of networks

The IEEE 802.1d Ethernet Bridging standard addresses this, dealing with the cases of multiple routes between hosts

### 15. Virtual Bridging

And a common variety is 802.1q *virtual bridging*

More commonly called *Virtual LANs* (VLANs)

This is a kind of reverse of the ARP bridge: it allows more than one network to run on a *single* physical network

### 16. Virtual Bridging



Virtual bridge

A company has two separate sites 1 and 2 with a single dedicated link between them; They want to run two separate LANs, A and B, but not to buy a second link between the sites; They can use 802.1q *tagging*; A packet from LAN A in Site 1, say, arrives at the switch; The switch knows to route the packet over the remote link: it places a 802.1q *tag* on the frame; A tag is an extra four byte header containing a *Virtual LAN Identifier* (VID), a 12 bit integer; The frame type in the physical layer (typically Ethernet) is changed from 0800 to 8100 to indicate a tagged packet; The switch in Site 2 receives the packet, sees the tag, reads and removes it and forwards the packet to its part of LAN A

### 17. Virtual Bridging



Tagging packets in a VLAN

### 18. Virtual Bridging

This generalises well to many virtual LANs and allows many networks to share infrastructure, thus saving on cost

Note: this is quite different from *Virtual Private Networks* (VPNs), which we shall talk about later

**Exercise** Look up the structure of a VLAN tag

**Exercise** The University uses VLANs extensively. Find out about this

**Exercise** How does tagging interact with maximum frame sizes, e.g., in Ethernet?