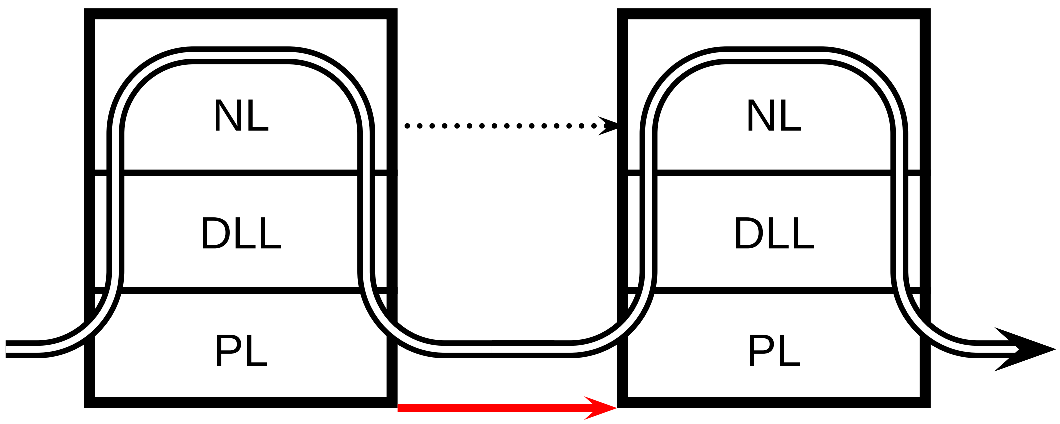
More on protocol stacks

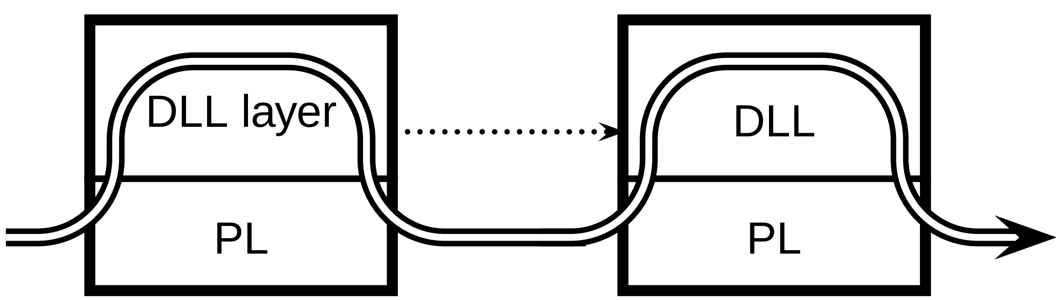
The OSI stack and the Internet stack represent the layers of software in the host stations (source and destination). But a message travels through the network core before reaching the destination host. The switches (etc.) inside the core do not need a whole stack to perform their functions (routing and relaying).

Their stacks are truncated; moreover, they play simultaneously the role of sender and receiver.

Network router stack



Link–layer switch stack



Data Link Layer

The DLL moves NL datagrams over an individual link. The DLL protocol encapsulates a datagram (preceding it with its own header) into a **frame** and passes the frame to the Physical Layer which sends the contents of the frame as independent symbols (e.g. bits) through the physical medium.

There are two types of DLL links:

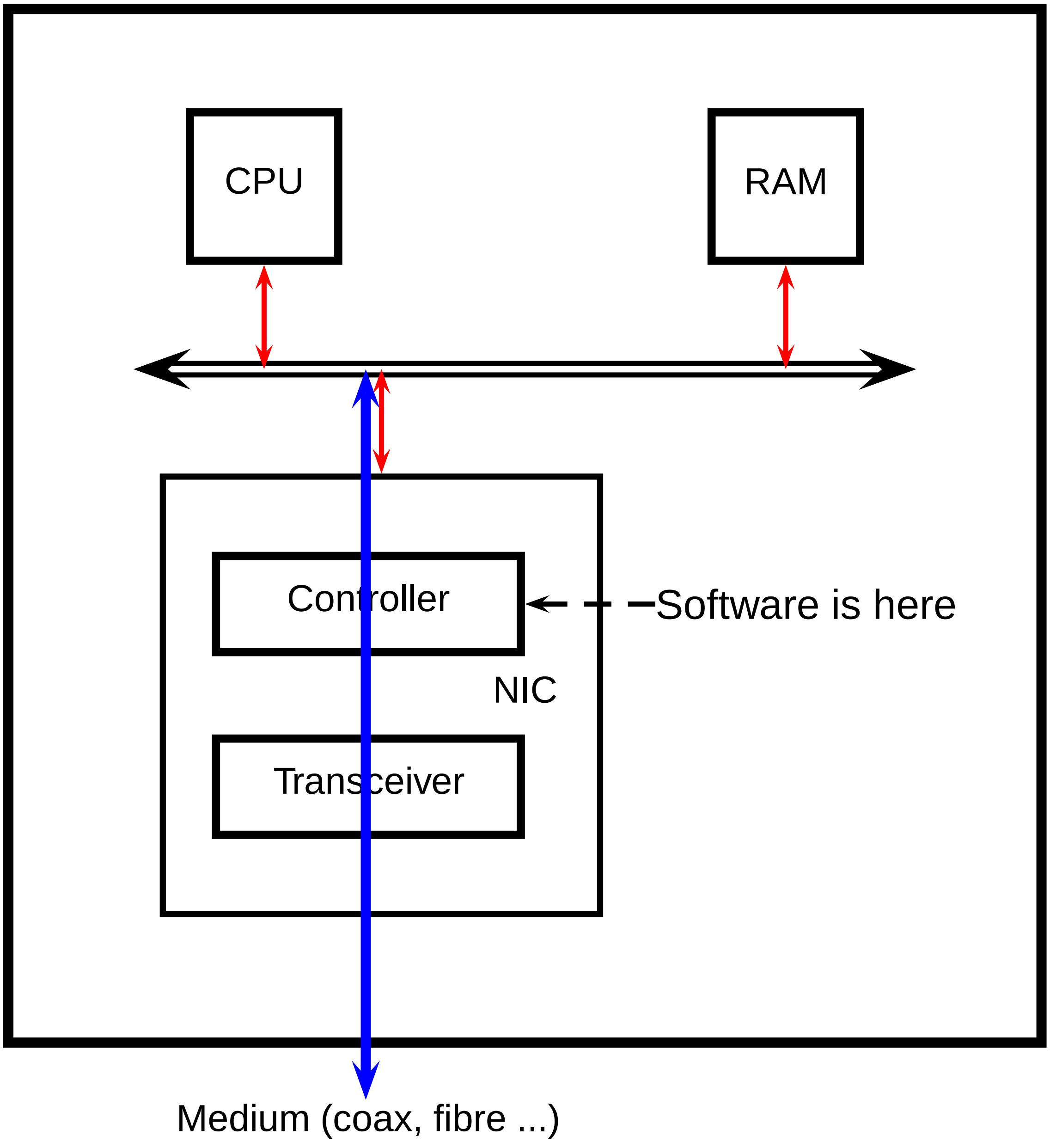
- A point–to–point link has only one node[a] at each end.
- A shared broadcast link ("multiple access") to which several nodes are attached. These nodes have equal rights to the link, even if their actual role may differ (e.g. router vs. host).

[a] Node = host, router, switch, etc.

MAC protocols

A shared broadcast link is controlled by one DLL protocol call **Medium Access Control** protocol. The same protocol must be obeyed by all the nodes on one link, although the same node may use several different MAC protocols if connected to more than one link.

A p2p link does not have a need for a MAC protocol but both nodes connected to it must be compatible (but not always the same, as in 802.11b, 802.11g and 802.11n or in “Fast Ethernet”).

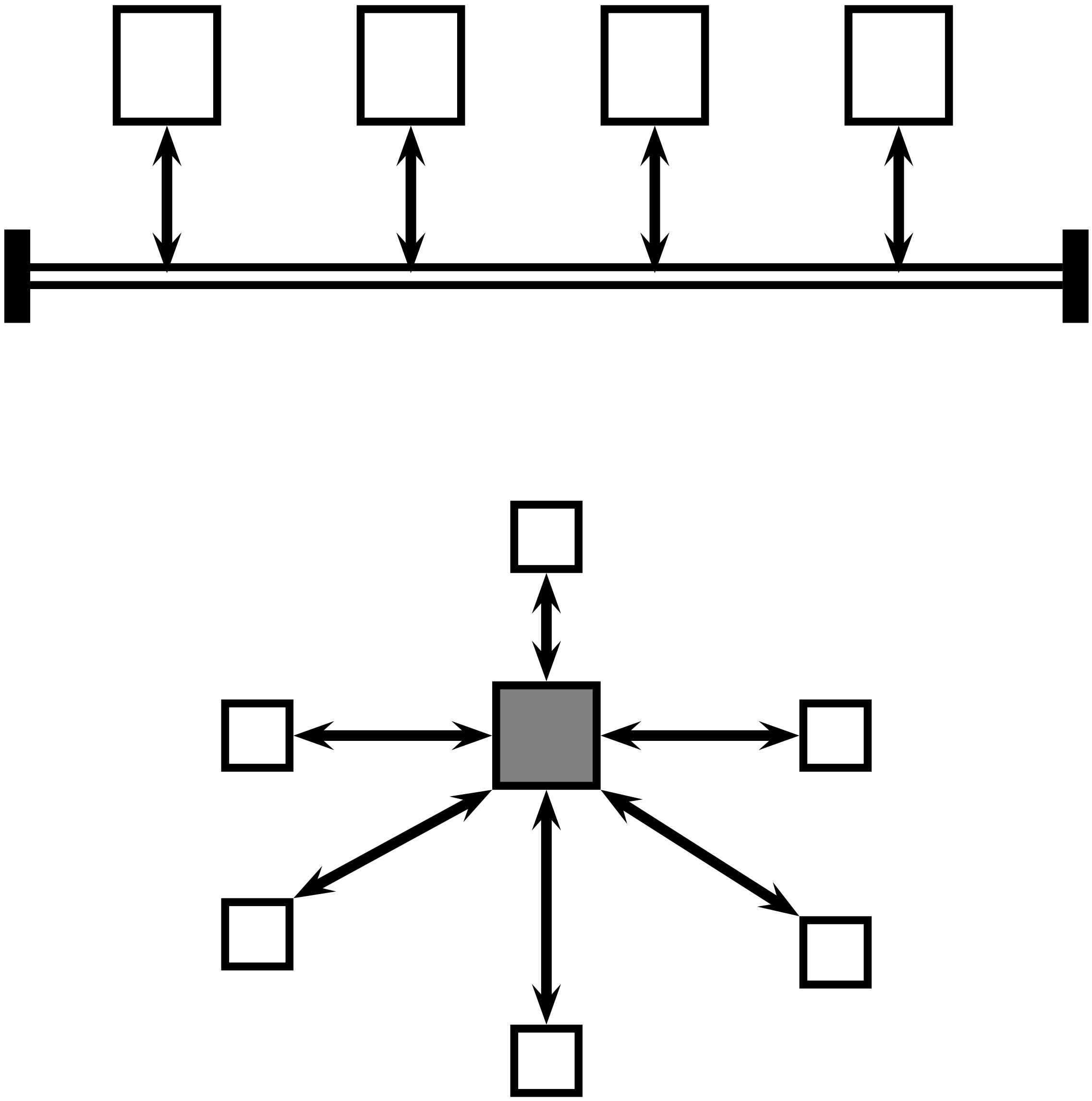

CPU
RAM
Controller
Software is here
NIC
Transceiver
Medium (coax, fibre ...)

Ethernet

Ethernet, the oldest modern local area network (1970s) exists in three variants:

- The original **bus** configuration running at 10Mbps.
- A newer **star** configuration with a central **hub** running at 100Mbps or 1Gbs.
- A star configuration with a central **switch** which can run at any transmission rate (commercial versions run at 10 Gbs but 100Gbs are envisioned). Several switches can be attached to each other, this configuration is a WAN in some sense (it has only 2 layers: PL and DLL).

Ethernet topologies



Address format

The **IEEE** manages DLL addresses, known as **MAC** addresses. Each node has a permanent 48–bit **MAC address** assigned to it by the manufacturer of the NIC. This address is stored in read–only memory and theoretically cannot be modified (but it can be spoofed[a]).

IEEE allows two formats for writing MAC addresses:

HH-HH-HH-HH-HH-HH

HH:HH:HH:HH:HH:HH

where H stands for a hexadecimal digit (0-F).

The address is supposed to be unique worldwide with the first 3 bytes indicating the the issuer of the address (this is called a Burn–In–Address). However, a skilled person may override a BIA address.

MAC addresses are used by Ethernet, wireless 802.11, Bluetooth and most other IEEE 802 networks.

[a]In hub– or star– Ethernet, the response to a spoofed frame is sent to the spoofing host, not to the victim.

Frame format

10101010101010101010101010101010	
10101010101010101010101010101011	
Destination address	
Dest. address	Source address
Source address	
NL protocol type	Data
Data	
from 46B	
to 1500B	
CRC (checksum)	

CSMA/CD protocol

The DLL protocol used by Ethernet is called **Carrier Sense Multiple Access** with **Collision Detection**.

CSMA/CD operates in two modes:

- **Normal** mode.
- **Backoff** or **Collision resolution** mode.

1. In normal mode, a node can start transmitting as soon as the transceiver reports silence.
2. Upon an interrupt from the transceiver, it stops passing bits to it and enters the **backoff** mode and stays in it until it manages to transmit a frame.

Note that when a single node is interested in transmitting a large file, CSMA/CD is always in the normal mode and its efficiency approaches $\frac{1500-96}{1500+26} = 92\%$

When the link is congested, Ethernet is often in backoff mode and its efficiency drops to about 30% but practically never below.

Transceiver

It senses the link at a normal reading rate (twice the transmission rate) and reads bits from it, decoding them appropriately (method depends on the medium used (light or electricity).

It detects transmission errors (crooked bits) and ignores the remainder of a frame if comes damaged.

The main hardware requirement is that the receiver in the Transceiver (also called “adapter”) must also sense any change in the signal energy level (note that it must be clever enough to recognise signal from its own transmitter).

Several possibilities exist:

1. Energy same as during the previous bit. Nothing special happens.
2. Energy level drops to silence (end of frame). The Transceiver starts a 96 bit period of silence. If the link remains silent during these 96 bits, it informs the Controller and waits for a signal to start transmitting.
3. The energy level goes up from silence to standard level (somebody's frame showed up). The Transceiver does nothing.
4. The energy level goes up from standard to abnormally high. Two possibilities:
   - The local transmitter was silent: a **collision** between two (or more) frames transmitted by other nodes. Nothing special: the transceiver waits for event (2).
   - The local transmitter was transmitting. It aborts immediately, sends a 48 bit **jam signal** and informs the Controller which enters the **backoff** mode. This event is called a **collision**.

Exponential backoff

When CSMA/CD enters the backoff mode, it performs a complex algorithm called **collision resolution**. The node stays in backoff mode until it successfully transmits a frame.

While in backoff mode, the Controller:

1. Waits for a random period of time. Typically, **exponential backoff** is used which requires that the wait be a 2^r where $r = random(1, n)$ where n is the number of collisions experienced while in this instance of backoff. When $n = 11$, the protocol resets itself to an initial state.

2. Makes the Transceiver start to retransmit the frame that collided.