Network models help us to understand how everything fits together.

OSI model breaks the network into 7 layers:

Please   
Do   
Not   
Throw   
Sausage   
Pizza   
Away

Lower layers

Upper layers

|  |
| --- |
| **APPLICATION** |
| **PRESENTATION** |
| **SESSION** |
| **TRANSPORT** |
| **NETWORK** |
| **DATA LINK** |
| **PHYSICAL** |

OSI model is not about specific technologies, but rather how they fit into the network stack.

Upper layers work with very large pieces of information.

APPLICATION -> Where network APIs and apps that access the network live, for example this includes FTP and web browsing

PRESENTATION -> Data needs to be in a format that can be easily understood, anything related to data formats lives here such as image and video files. (Encryption, compression…)

SESSION -> Tracks application processes. This includes remote procedure calls and service requests. Think this layer as building a session between a local app and a remote one.

TRANSPORT -> What if you are transferring very large data and you get end of the transfer and the connection is interrupted. You have to start the whole transfer again and also any other app that wants to use the network would have to wait. So when data gets to TRANSPORT layer, it is broken into manageable chunks. (Each block is segment if you are using TCP, datagram if you are using UDP)

If there’s a problem with data, only one chunk of the data needs to be resent. Also apps can take turns at sending chunks of data rather than one app hogging the host’s resources (multiplexing)

Now we have many blocks of data probably going to different hosts for different applications. How can we track what goes where? -> PORT NUMBERS. Each block of data has a port number associated with the source and destination host (added as part of the header). Client simply generates a random port number and uses this as the source port.

Sometimes we need to add more information such as destination address. Any information we add front of our data is called the header. Any information we add back is called the trailer. As we go, data gets bigger.

NETWORK -> We use IP here as another header. Now block of data is called a packet.

DATA LINK -> Creates a logical link between devices on the same network segment. We need to have a logical link because devices probably aren’t connected directly. They are usually connected with a shared medium like a switch or over Wi-Fi. A well-known protocol here is Ethernet which uses MAC addresses. We use MAC here as another header. Trailer may also be added with error connection information. Now block is a frame. Contains 2 sublayers:

* LLC (logical link control sublayer which is responsible for translating between the network layer and the data link layer)
* MAC (medium access control sublayer is the part that’s responsible for adding headers and trailers to the packet creating the frame and also responsible for error correction)

Router changes the source and destination MAC addresses.

PHYSICAL -> Manages physical network components (radio frequencies, channels…). Encodes data into physical signals.

When data is arrived, process is then reversed. Each layer does its job. Removing headers and trailers and converting the data until it is in a form that the application can understand.

Each layer will only communicate with the layer above and the layer below.