OSI Layers Overview

Layer	Type	Description	PDU (Protocol Data Unit)	Protocols
Layer 7	Application	*Software I/O*	"Data"	NFS, NIS+, DNS, telnet, ftp, rlogin, rsh, rcp, RIP, RDISC, SNMP, and others
Layer 6	Presentation	*Formats the data into usable format + encryption*	"Data"	NFS, NIS+, DNS, telnet, ftp, rlogin, rsh, rcp, RIP, RDISC, SNMP, and others
Layer 5	Session	*Maintains connections / controls **ports** and **sessions***	"Data"	NFS, NIS+, DNS, telnet, ftp, rlogin, rsh, rcp, RIP, RDISC, SNMP, and others
Layer 4	Transport	*End-to-End Communication using TCP / UDP*	Segments / Datagram	TCP, UDP
Layer 3	Network	*Path Determination (IP) / Defines physical path of data*	Packets	IP, ARP, ICMP
Layer 2	Data Link	*Physical Address (MAC) / Defines the format of data*	Frames	PPP, IEEE 802.2
Layer 1	Physical	*Cabelling / Adapter / Raw binary data stream*	Bits	Ethernet (IEEE 802.3) Token Ring, RS-232

Acronyms (7-1)

- All Prostitutes Seem To Need Double Penetration
- All People Seem To Need Data Protection
- All Presidents Say They Never Did Pot
- All People Say They Never Download Porn

Acronyms (1-7)

- Please Do Not Throw Sausage Pizza Away
- People don't need those stupid packets anyway
- Please Do Not Teach Students Pointless Acronyms
- Programmers Dare Not Throw Salty Pretzels Away

  _source: https://www.osi-model.com

Layer 1 (Physical)

The physcial layer is the lowest layer in the [[OSI Model]] and is responsible for transmitting bits through datastream connections (medium to adapter) which can manifest as **electrical signals** as in physical media like [[ethernet]], **electromagnetic waves** via [[AMI]] or bipolar encoding, or sound carried over air or through line of sight for wireless networks.

> All components that compute or manipulate binary data at the **hardware level** are examples of Layer 1 devices.

Diagnosis: Layer 1 hardware is usually the easiest to troubleshoot, common layer 1 issues will usually be easy to recognise as the OS will display no NIC in the system.

Layer 2 (Data Link)

The data link layer is responsible for controlling / monitoring traffic on a network. This layer is where the first data [[encapsulation]] process begins. Frames are the data of layer 2 which are labeled with forwarding information (*Each Ethernet frame carries a destination MAC address and a source MAC address*).

> - It's seperated between two layers: Logical Link Control (LLC) sublayer provides the logic for the data link. Thus, it controls the synchronization, flow control, and error checking functions of the data link layer. Media Access Control (MAC) sublayer provides control for accessing the transmission medium.

Many services are used in the data link layer for controlling traffic on the network like [[logical link control]] or LLC which controls the **synchronization**, **flow control**, and **error-checking** functions of the data link layer or **LAN switching** based on the MAC address identifier which is obtained by sending an [[ARP]] frame to request the MAC address and stores the MAC address in the **ARP Table** / cache along with the devices's corresponding local IP address.

> switches can make forwarding decisions based on the destination MAC addresss.

Layer 3 (Network)

Multi-layer switches or layer 3 switches are used to make forwarding decisions based on destionation IP addresses using **packets** which contain a source and destination IP address in addition to more forwarding information. These layer 3 packets are used to send data to a **router** on a different subnet / network.

Packets contain forwarding information and TTL fields that the node will check and reference it's [[routing table]] which is constructed and updated at his level to see where to forward the packet in question with added **flow control** like on layer 2.

[[QoS]] (Quality of Service) also takes place at this layer, where traffic is prioritized or limited depending on a pre-defined set of rules.

Layer 4 (Transport)

The two main protocols for transportation are **TCP** & **UDP** with the function of providing *end-to-end communication and reliable delivery*. - **TCP/IP** or the *Transport Control Protocol* is a connection oriented protocol meaning it requires confirmation from the reciever via the TCP 3 way handshake. Client sends a **Synchronization** (SYN) message with a **Seq** # of **9001**, the host will send a **Synchronization & Acknowledgement** (SYN - ACK) message with the **ACK** # of **9002** and it's own **Seq** number of **5001**, with the client finally sending an **Acknowledgement** (ACK) message with an empty SYN number, an **ACK** of **5002**, and the previous **Seq** number of **9002**: (All these flags are stored in the **TCP HEADER** of the layer 4 segment) □ Once the handshake has been established either party can begin sending **packets** with the receiver sending an **ACK** message after each **packet** due to TCP being **Full Duplex**. TCP/IP negotiates transfer speed by sending an ACK message after each group of segments until it can't keep up (doubling each time). All stored in **TCP Header** - **UDP** or the *User Datagram Protocol* is a connectionless protocol meaning it sends segments without ackowledgement

> The primary layer 4 operation is recieving data from applications then segmenting it in addition to it **may** error detection and correction capabilities and sends these segments according to negotiated transfer speed.

Segmentation: TCP receives data from the application (Port information and the size of the data) then negotiates packet size with the network layer (**Layer 3**) and segments the data in to multiple packets adhering to the max packet size.

> Max packet size is 1518 bytes, the transport layer has 5k bytes to sub divide into packets.

Port information stored in **segments**.

□ *frame configuration (https://learningnetwork.cisco.com)*

![[Selection_016.png]] *TCP Header & TCP Flags (https://www.professormesser.com/)*

Layer 5 (Session)

> The Session layer is responsible for allowing software / applications to interact with low level TCP/IP network function, like sending various flags like **SYN** or **FIN** acting as an interface to manipulate the network stack (**TCP Header** manipulation using software): □

Layer 6 (Presentation)

> The presentation layer is responsible for the delivery and **formatting** of information to the application layer for further processing or display.

It's also responsible setting **system-dependent** representation into an independent form for the purpose of maintaining congruence between the client and host, examples of this are system specific **encoding** like ASCII or EBCDIC. Encryption also takes place here usually in the form of **TLS** or the outdated **SSL** for end-to-end encryption with the host.

Layer 7 (Application)

The application layer uses all the familiar high level protocols like bitcoin, dds, ircp, nfs, and ssh. It allows software to interface with these high level APIs and is responsible for client-to-host I/O.