Software Development 3

Assignment 2

CIT185753

Kitman Yiu

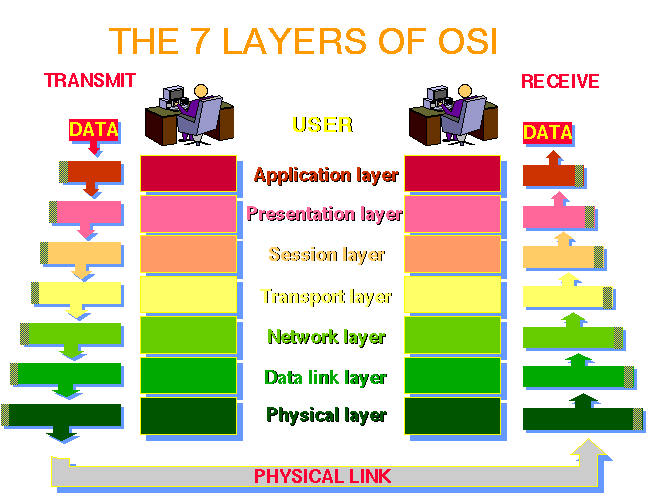
## Network Architecture

Google scholar no definition

No definition in Google scholar

## The history of OSI (The Open Systems Interconnect)

A long time ago, assume you world of Warcraft using mac but your friend is window, and you wanted to play game together. But because the internet didn’t have the common sets of rules so they won’t allow us the send message (eg: chat messages, the position of your friend…etc) from a mac to window, so after that people see that problem, so people was trying to find a common sets of rules that allow send data across from one computer to another computer and the OSI win.



Something: if would need to run through all 7 layers before the information can be send to other computers and each layer has responsible to add some important information such as (the address the data is going to send) to.

How to do? : provide common set of rules of all types of computer

What does OSI do? : allow different types of computer to communicate.

Who :( ITU，International Telecommunication Union）, （ISO，International Standards Organization）, （IAB，Internet Architecture Board）

An OSI model does not actually exists but a model that allow us to understand it my easily. The image form above (1-11) is an image how the OSI models look like, and the follow diagrams explains each layers:

|  |  |  |
| --- | --- | --- |
| 7 | Application | e.g. HTTP, SMTP, SNMP, FTP, Telnet, SSH and Scp, NFS, RTSP etc. |
| 6 | Presentation | e.g. XDR, ASN.1, SMB, AFP etc. |
| 5 | Session | e.g. TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP etc. |
| 4 | Transport | e.g. TCP, UDP, RTP, SCTP, SPX, ATP etc. |
| 3 | Network | e.g. IP/IPv6, ICMP, IGMP, X.25, CLNP, ARP, RARP, BGP, OSPF, RIP, IPX, DDP etc. |
| 2 | Data Link | e.g. Ethernet, Token ring, PPP, HDLC, Frame relay, ISDN, ATM, 802.11 Wi-Fi, FDDI etc. |
| 1 | Physical | e.g. wire, radio, fiber optic etc. |

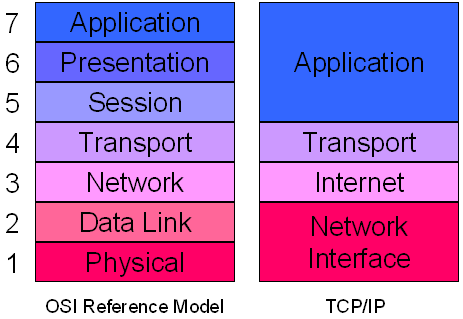
Trying to put this in This is related to Physical layer but don’t know how

局域网LAN，约1千米

城域网MAN，约10千米

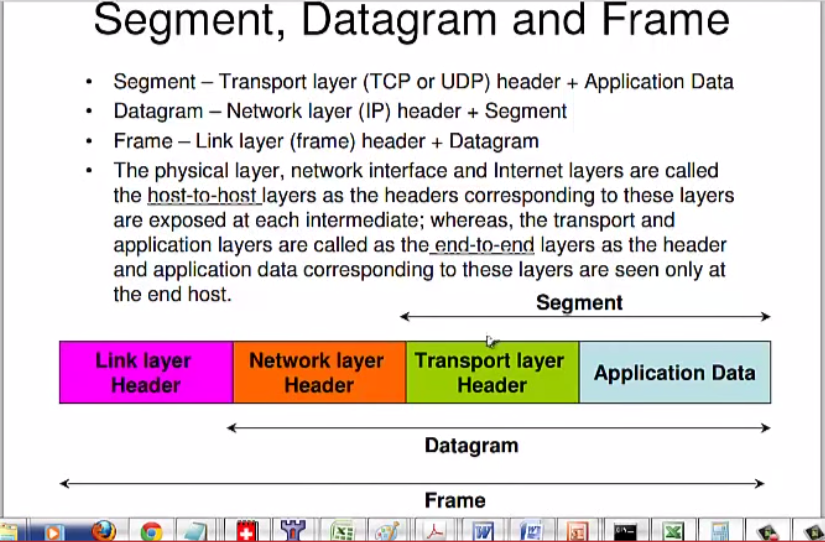
广域网WAN，约100千米以上。

# TCP/IP



Since the ISO does not exists, so we need create an actual model that allow computer to communicate , one of them is called TCP/IP , we will explain what is TCP/IP later The image above is a model that how TCP/IP looks like.

|  |  |
| --- | --- |
| Application layer | BGP, FTP, HTTP, HTTPS, IMAP, IRC, NNTP, POP3, RTP, SIP, SMTP, SNMP, SSH, SSL, Telnet, UUCP, Finger, Gopher, DNS, RIP, Traceroute, Whois, IMAP/IMAP4, Ping, RADIUS, BGP etc. |
| Transport layer | DCCP, OSPF, SCTP, TCP, UDP, ICMP etc. |
| Network/Internet layer | IPv4, IPv6, ICMP, ARP, IGMP etc |
| Physical/ Data  Link layer | Ethernet, Wireless (WAP, CDPD, 802.11, Wi-Fi), Token ring, FDDI, PPP, ISDN, Frame Relay, ATM, SONET/SDH, xDSL, SLIP etc.  RS-232, EIA-422, RS-449, EIA-485 etc. |



As the image from above it should be straghit forward:

**Segment = Application + Transport layer Header**

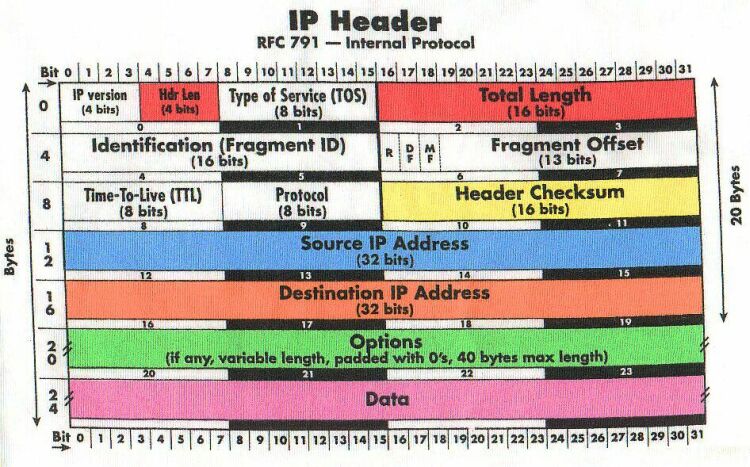
**Datagram = Application + Transport layer Header + Networking layer header**

**Frame = Application + Transport layer Header + Networking layer header + Link layer header**

Since data can be lost .and some data are big (eg:music , video), so people decide to cut data into to smaller parts and add something called header which is placed in front of the data. And the three layers are:

|  |  |
| --- | --- |
| Application Layer: |  |
| Transport Layer: |  |
| Internet Network layers: |  |

The next question is what is inside the all the headers. The following image have a look what is inside those headers.

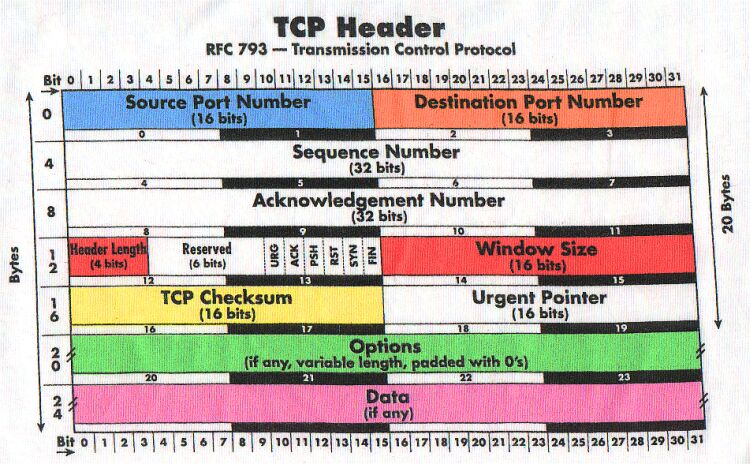


So what are all those data use for the following diagram is to explain what the things use are for:

# A brief of IP description:

|  |  |
| --- | --- |
| Field | Description |
| Version | The version of IP currently used. |
| IHL | IP Header Length (IHL) - datagram header length. The minimum value for a correct header is 5. |
| Type of Service | Data in this field indicate the quality of service desired.  Format of the Type of Service field:   * Bits 0-2: Precedence   111 = Normal Control.  110 = Internetwork Control.  101 = CRITIC/ECP.  100 = Flash Override.  011 = Flash.  010 = Immediate.  001 = Priority.  000 = Routine.   * Bit 3: Delay 0 = normal delay, 1 = low delay. * Bit 4: Throughput 0 = normal throughput, 1 = high throughput. * Bit 5: Reliability 0 = normal reliability, 1 = high reliability. * Bits 6-7: Reserved |
| Total Length | The length of the datagram in byte, including the IP header and data.  This field enables datagrams to consist of up to 65,535 bytes. |
| Identification | An identification field used to aid reassembles of the fragments of a datagram. |
| Flags | If a datagram is fragmented, the MB bit is 1 in all fragments except the last.  This field contains three control bits:   * Bit 0: Reserved, must be 0. * Bit 1 (DF): 1 = Do not fragment and 0 = May fragment. * Bit 2 (MF): 1 = More fragments and 0 = Last fragment. |
| Fragment Offset | For fragmented datagrams, indicates the position in the datagram of this fragment. |
| Time-to-live | Indicates the maximum time the datagram may remain on the network. |
| Protocol | The 8 bits field of the upper layer protocol associated with the data portion of the datagram.  For a complete information please refer to RFC 1700 and the following is some of the protocol numbers:  Decimal              Protocol  1 ICMP (Internet Control Message)  2 IGMP (Internet Group Management)  4 IP (IP in IP -encapsulation)  5 ST (Stream)  6 TCP (Transmission Control)  17 UDP (User Datagram)  27                        RDP  (Reliable Data Protocol) |
| Header Checksum | A checksum for the header only. This value will be recalculated each time the header is modified. |
| Source Address | The IP address of the originated the datagram. |
| Destination Address | The IP address of the host that is the final destination of the datagram. |
| Options | May contain 0 or more options. |
| Padding | Filled with bits to ensure that the size of the header is a 32-bit multiple. |

# TCP



# A brief of TCP description:

Source port - bit 0 - 15. This is the source port of the packet. You can imaging the source port is your home door

Destination port - bit 16 - 31. This is the destination port of the TCP packet. The door of your friends’ home

Sequence Number - bit 32 - 63. The sequence number field is used to set a number on each TCP packet so that the TCP stream can be properly sequenced (e.g., the packets winds up in the correct order).

Acknowledgment Number - bit 64 - 95. This field is used when we acknowledge a specific packet a host has received. For example, we receive a packet with one Sequence number set, and if everything is okay with the packet,

Data Offset - bit 96 - 99. This field indicates how long the TCP header is, and where the Data part of the packet actually starts. It is set with 4 bits.

Reserved - bit 100 - 103. These bits are reserved for future usage.

CWR - bit 104. This bit was added in RFC 3268 and is used by ECN. Used by the data sending part to inform the receiving part that the congestion window has been reduced.

ECE - bit 105. This bit was also added with RFC 3268 and is used by ECN. ECE stands for ECN Echo. Used by the TCP/IP stack on the receiver host to let the sending host know that it has received a CE packet.

URG - bit 106. Tells us if we should use the Urgent Pointer field or not. If set to 0, do not use Urgent Pointer, if set to 1, do use Urgent pointer.

ACK - bit 107. This bit is set to a packet to indicate that this is in reply to another packet that we received, and that contained data.

PSH - bit 108. The PUSH flag is used to tell the TCP protocol on any intermediate hosts to send the data on to the actual user, including the TCP implementation on the receiving host.

RST - bit 109. The RESET flag is set to tell the other end to tear down the TCP connection.

SYN - bit 110. The SYN (or Synchronize sequence numbers) is used during the initial establishment of a connection.

FIN - bit 111. The FIN bit indicates that the host that sent the FIN bit has no more data to send. When the other end sees the FIN bit, it will reply with a FIN/ACK. Once this is done, the host that originally sent the FIN bit can no longer send any data.

Window - bit 112 - 127. The Window field is used by the receiving host to tell the sender how much data the receiver permits at the moment.

Checksum - bit 128 - 143. This field contains the checksum of the whole TCP header.

Urgent Pointer - bit 144 - 159. This is a pointer that points to the end of the data which is considered urgent. If the connection has important data that should be processed as soon as possible by the receiving end.

Options - bit 160 - \*\*. The Options field is a variable length field and contains optional headers that we may want to use.

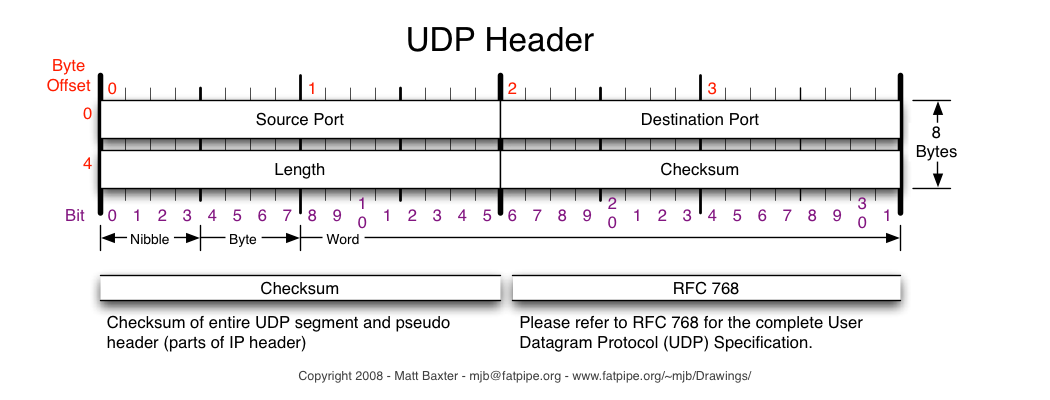
Padding - bit \*\*. The padding field pads the TCP header until the whole header ends at a 32-bit boundary. This ensures that the data part of the packet begins on a 32-bit boundary, and no data is lost in the packet. The padding always consists of only zeros.

THE pros and cons of using TCP/IP and UDP/IP

# UDP

The difference between TCP and UDP. Is what inside the header the following image represent the UDP header information

There are not only TCP/IP and UDP/IP but these are the most common one



## What Slow’s down the Internet

So from the things that we understand above what things are actually slow down the internet when we are playing games

Development need to deal with issues of latency, bandwidth, time synchronization, Protocols overheads and security when designing game

|  |  |
| --- | --- |
| Latency | This is how long a packet takes to travel from the source to the destination.  This is limited by the laws of physics.  An electron can only travel at 1 speed on a copper wire.  Increasing bandwidth has no effect on this.  A photon on a fibre link can only travel at the speed of light.  Even if we could increase this speed, Einstein tells us that this would negatively affect the latency!  Finally, devices that process the data at various hops, e.g. routers, switches and firewalls, also add to latency.  So, the further apart a client and server are, the longer a transmission takes.  Adding in more network devices, e.g. transmissions between different ISP’s, worsens this.  (I’m tyring to cut down this part) |
| Bandwidth | How much data we can transfer at once, i.e. in one packet, from the source to the destination.  We measure this in KBPS, MBPS or even GBPS if you have lots of money. |
| Time Synchronization | Match up the player location to other players so it can see him. |
| Protocols | A set of rules that IT use. This include TCP AND UDP. Most people use UDP |
| Security | Network computer games pose a special security threay since they have to expose the player’s machine to other machines, often over the internet. |

## The solutions

The solutions are listed below

# TCP/UDP

A post called Glenn Fiedler (is a game networking expert with 15 years of industry experience) Was talking about should we use TCP/UDP for games, know why but don’t know how to write it.

<http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/>

# ASIC/FPGA

In warframe forum they recent talked about TCP/UDP has discussed to death, and a new technology uPNP and ASIC has been exists.

**Know why but don’t know how to write it.**

<https://forums.warframe.com/index.php?/topic/376373-network-matchmaking-from-networking-expert/>

# Dead reckoning

The process of calculating one's position, especially at sea, by estimating the direction and distance travelled rather than by using landmarks or astronomical observations.

<http://www.aidanfinn.com/?p=9566>

<https://www.ietf.org/rfc/rfc1129.pdf>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.2134&rep=rep1&type=pdf>

<http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/>

<https://forums.warframe.com/index.php?/topic/376373-network-matchmaking-from-networking-expert/>

<https://www.google.com.au/?gws_rd=ssl#q=what+is+dead+reckoning>