

Table of Content

Unit 1 Internet and IP

The 4 Layer Internet Model

The IP Service

 What's in a Datagram

Life of a Packet

 TCP Byte Stream

 Inside the Stream

 The tools

Principle: Packet Switching

 No per-flow State Required: Simple Packet Forwarding

 Efficient Sharing of Links

Principle: Layering

 Definition

 Reasons for Layering

 It helps

 Summary

Principle: Encapsulation

 Definition

 Encapsulation Flexibility

 Reason for Encapsulation

Memory, Byte Order, and Packet Formats

 didn't really understand this lesson..... and left empty temporarily

Names and Addresses: IPv4

 IPv4

 Historical Address Structure

 Today's Address Structure

 IPv4 Address Assignment

Longest Prefix Match

Table of Content

Unit 1 Internet and IP

- The 4 Layer Internet Model

- The IP Service

- What's in a Datagram

- Life of a Packet

- TCP Byte Stream

- Inside the Stream

- The tools

- Principle: Packet Switching

- No Per-Flow State Required: Simple Packet Forwarding

- Efficient Sharing of Links

- Principle: Layering

- Definition

- Reasons for Layering

- Principle: Encapsulation

Unit 1 Internet and IP

The Internet dominant model is a **bidirectional, reliable byte stream**.

Three networked applications:

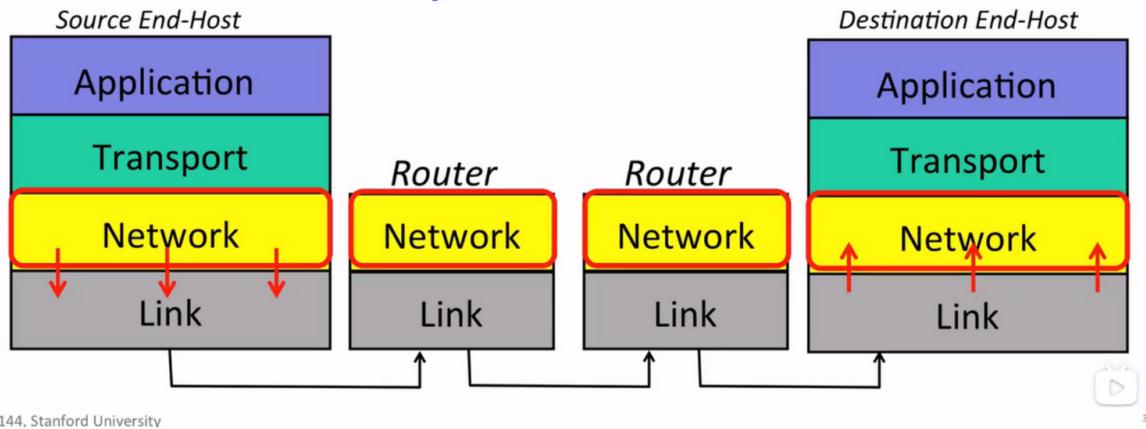
- World Wide Web(HTTP): client-server model
- BitTorrent: peer-to-peer model, *client - Tracker - other clients*
- Skype: Mix of the both, *depends on if Skype clients can directly communicate*

The 4 Layer Internet Model

Data be sent by unit packets.

The network-layered **packet** be drawn as datagram, containing **data, source, and destination**. Source and destination info part is called **header**.

The 4 Layer Internet Model



We use the Internet Protocol(IP) as a must:

- IP makes a best effort attempt to deliver datagrams, but no promises.
- IP datagrams can be lost, delivered out of order, corrupted, no guarantees.
- To deal with the unpraised issued the other protocol on the top of IP needed in Transport Layer (i.e., Transmission Control Protocol).

TCP:

- In Transport Layer
- Makes sure that all data packets delivered **correctly, in order, to right place**.

UDP (User Datagram Protocol):

- Also in Transport Layer
- Does **not guarantee** the correct deliver, no repetition, in order. But **higher efficiency**.

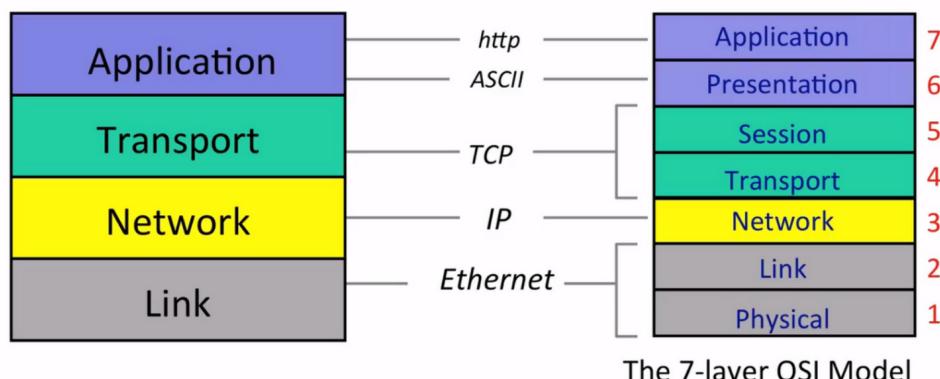
Summary of 4 Layer Model

| | |
|-------------|--|
| Application | Bi-directional reliable byte stream between two applications, using application-specific semantics (e.g. http, bit-torrent). |
| Transport | Guarantees correct, in-order delivery of data end-to-end. Controls congestion. |
| Network | Delivers datagrams end-to-end. Best-effort delivery – no guarantees. Must use the Internet Protocol (IP). |
| Link | Delivers data over a single link between an end host and router, or between routers |

IP is called “thin waist”, for there are many choices on the top and under the Network Layer, but in Network Layer IP is the only choice.

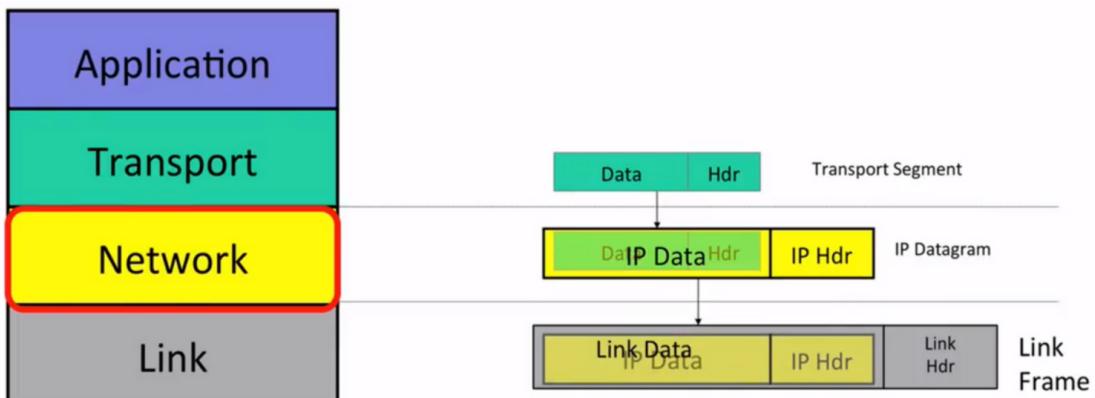
| Application Layer: | Transport Layer: | Network Layer: |
|---|--|---|
| HTTP: - Generate request message according to target web server. - Process the requested content. | TCP: - Divided the HTTP message into segments, in sequence. Then pass them reliably. - Receive the segments and re-structure the segments in sequence. | IP: - Searching for target address, passing while routing. |

The 7-layer OSI Model



The IP Service

The Internet Protocol (IP)

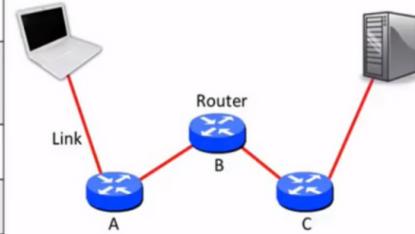


The IP service model:

In which the packet jumps among the routers (in which there are routing tables) in the path towards its destination. Routers in the path don't know its destination but the each other of themselves.

The IP Service Model

| Property | Behavior |
|-----------------------|---|
| Datagram | Individually routed packets. Hop-by-hop routing. |
| Unreliable | Packets might be dropped. |
| Best effort | ...but only if necessary. |
| Connectionless | No per-flow state. Packets might be mis-sequenced. |



The reasons for IP being so simple:

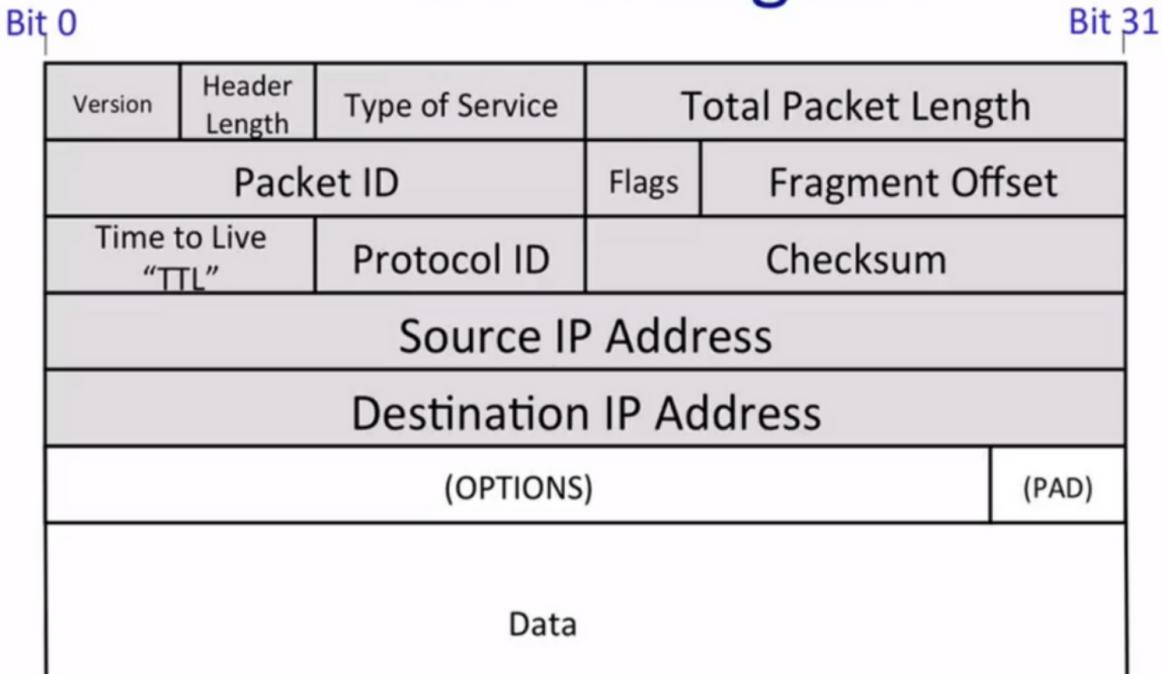
- Simple, dumb, minimal = faster, more streamlined and lower cost to build and maintain.
- The end-to-end principle: Implement features in the end hosts as much as possible.
- Allows a variety of dependable/unreliable services to be built on top.
- Works over any link layer: IP makes few assumptions about the link layer below.

More features of IP service model:

1. Tries to prevent packets looping forever.
2. Will fragment packets if they are too long.
3. Use a header **checksum** to reduce chances of delivering datagram to wrong destination.
4. Allow for new versions of IP
 - IPv4 with 32-bit address
 - IPv6 with 128-bit address
5. Allow for new options to be added to header.

What's in a Datagram

IPv4 Datagram



- The most important parts are **DA (Destination IP Address)** and **SA (Source IP Address)**.
- Protocol ID part tells what is in data field, helping destination host correctly process the packet.
- Version field tells the version of the IP.
- TTL part helps to stop the packet looping in the Internet. Every router will decrement the TTL till it reaches 0.
- The packet ID, Flags, Fragment Offset help router to fragment IP packet into smaller self-contained packets if needed.
- Type of Service tells the importance of the packet.
- Header Length tells how big the header is.
- Checksum is calculated over the whole header to make sure the packet to correct destination.

Application Layer - stream of data

Transport Layer - segments of data (TCP segment)

Network Layer - packet of data (IP packet)

Link Layer - frame of data (Ethernet frame)

In summary, IP provide a deliberately simple services:

- Datagram
- Unreliable
- Best-effort
- Connectionless

Life of a Packet

TCP Byte Stream

Three-way handshake: "SYN, SYN/ACK, ACK"

from Client to Server: Send a synchronized message, "**SYN**"

from Server to Client: Response a synchronized message and acknowledges the client's synchronized, "**SYN/ACK**"

from Client to Server: Acknowledge server's synchronize, "**ACK**"

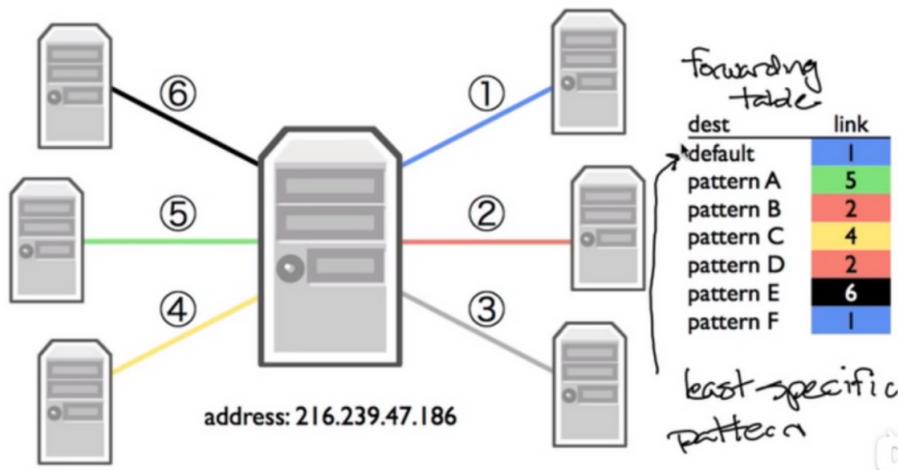
Inside the Stream

Address format: IPAddress:TCPPort (e.g., **192.168.0.1:22**)

Above is how a normal address looks like, IP address in Network Layer and TCP port in Transport Layer.

A forwarding table in a router:

Inside Each Hop



The router will send the packet to the matched entry pattern, which is the **most specific** match.
This is called Longest Prefix Match!

The default route is the least specific route, matches every IP address.

The tools

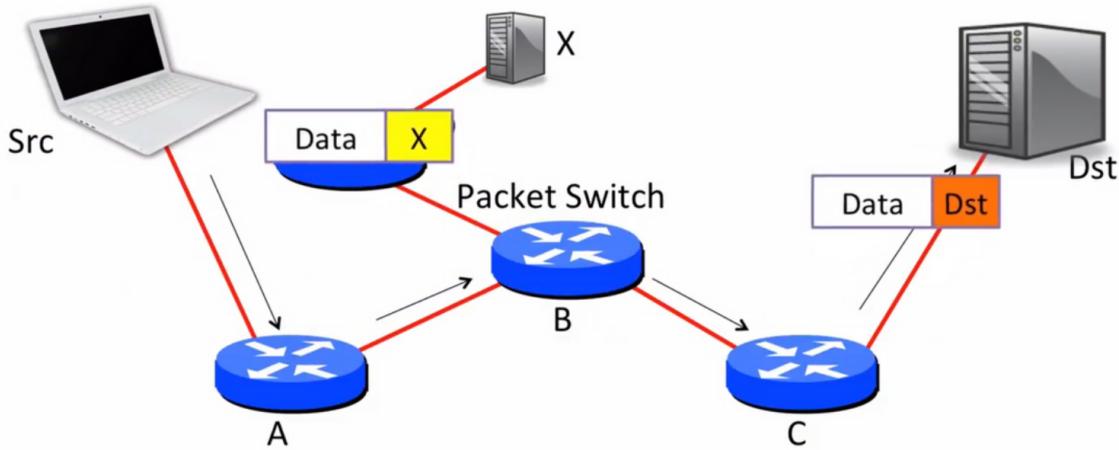
Wireshark: Show all the packets, TCP byte stream establishment and data exchange

Traceroute: Shows the route (all the routers it passed) that the packet took through Internet

Principle: Packet Switching

Packet: A self-contained unit of data that carries necessary information to reach destination.

Packet Switching



The packet will be routed between routers until it reached destination.

Packet switching: Independently for each arriving packet, pick its outgoing link. Send it if the link is free, else hold it for later.

Source routing/self-routing: The source specifies the route. (Not currently common for its big security issues)

Nowadays we use the route table that in each switch, according to which the switch can tell the packet to destination goes to right next hop.

Two consequences:

1. Simple packet forwarding
2. Efficient sharing of links

No per-flow State Required: Simple Packet Forwarding

Flow: A collection of datagrams belonging to the same end-to-end communication, such as a TCP connection.

Packet switching don't need state for each flow, since each packet is self-contained.

- No per-flow state to be added/removed
- No per-flow state to be stored
- No per-flow state to be changed upon failure

In one word, packet switching cares only about switching, do not deal with other issues!

Efficient Sharing of Links

bursty: Occurring at intervals in short, sudden episodes

Data traffic is bursty

- Packet switching allows flows to use all available link capacity.
- Packet switching allows flows to share link capacity.

Principle: Layering

Definition

Layers are separate, functional components, that are hierarchical and communicating sequentially with its direct neighbours, above or below. Each layer provides service to the layer above, using the services from the layers below and its own processing.

A good example of layering is **postal or express service**.

sender >> postman >> local site >> regional centre (near to sender) >>

receiver << postman << local site << regional centre (near to receiver)

In the above, each layer is separated, and does not care about how other layers work apart from its neighbouring layers. Also, they do not affect each other.

Reasons for Layering

It helps

- Separation of concerns and responsibilities
- Allows each service to evolve independently

Summary

1. Modularity
2. Well defined service
3. Reuse
4. Separation of concerns
5. Continuous improvement

Principle: Encapsulation

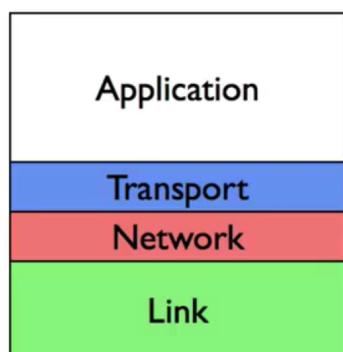
Definition

Encapsulation is a principle that helpful to maintain layering.

It contributes to

- Different protocol layers sharing storage with in the same packet

Encapsulation



- How layering manifests in data representation
- Layer N data is payload to layer N-1
- Example:
 - ▶ HTTP (web) application payload in
 - ▶ a TCP transport segment in
 - ▶ an IP network packet in
 - ▶ a WiFi link frame.

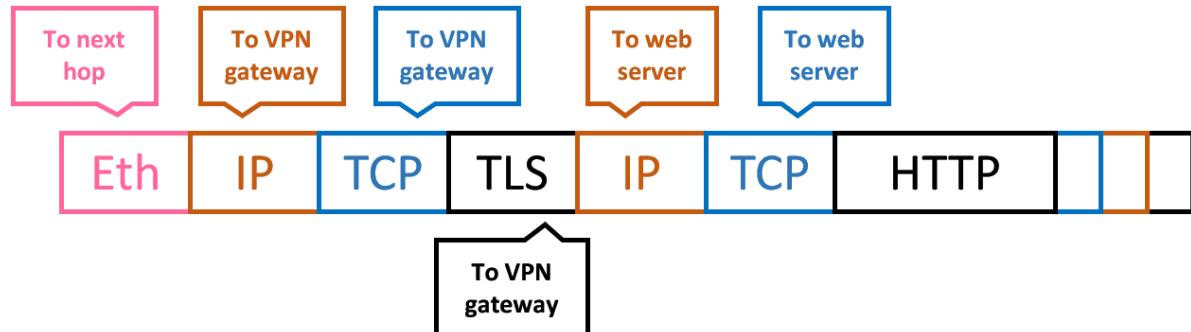


As the figure shows, e.g., a data in a layer is encapsulated as the payload of a data in the next layer .

Note: The order of headers and footers depends on software side (headers-footers) or hardware side (footers-headers).

Encapsulation Flexibility

- Encapsulation allows layer recursively



- Virtual Private Network (VPN)
 - HTTP application payload in
 - a TCP transport segment in
 - an IP network packet in
 - a secured TLS presentation message in
 - a TCP transport segment in
 - an IP network packet in
 - an Ethernet link frame

Reason for Encapsulation

- Encapsulation is how layering manifests in data representation
- Encapsulated payloads
 - help separation of concerns
 - help enforce boundaries/layering
 - simplify layer implementations

Memory, Byte Order, and Packet Formats

didn't really understand this lesson..... and left empty temporarily

Names and Addresses: IPv4

Goal of IP addr: Switch many different networks together, which needs network-independent, unique addr

IPv4

- One addr identifies a device on the Internet, on Network (3rd) Layer
- 32 bits long, separated to 4 octets: **a.b.c.d**
- Netmask: apply this mask, if the mask is the same as an addr, in the same network. It defines how the addr is cut into two parts (network+host)

- e.g., 255.255.255.0 means if the first 24 bits (first 3 octets) of two addrs match, the two in the same network
- e.g., netmask 255.255.255.0 means 1111111.1111111.1111111.00000000
- Smaller netmask (fewer 1s) means larger network

Historical Address Structure

network + host, in which,

- Network to get to correct network
- Host to get to correct device in that network

3 classes of network: class A, class B, class C

| | | | |
|----------------|-------|--------------|-----------|
| Class A | 0 | network (7) | host (24) |
| Class B | 1 0 | network (14) | host (16) |
| Class C | 1 1 0 | network (21) | host (8) |

Today's Address Structure

Classless Inter-Domain Routing (CIDR)

- Addr block is a pair: *addr/count*
- Counts are powers of 2 (*count/2=nR0*), specify netmask length
- 192.168.0.0/16 means the first 16 digits of the 32 digits are network addr
- A/24 has 256 addrs, A/20 has 4,096 addrs

IPv4 Address Assignment

ICANN (Internet Corporation for Assignment of Names and Numbers) is responsible for IANA (Internet Assigned Numbers Authority)

Longest Prefix Match
