Lecture 3,4: Internet Architecture, Delay, Loss and Throughput, Network Architecture, Network Threats, Application Layer @August 31, 2021

Network of Networks

The interconnection of access ISPs amongst themselves. A mesh design linking each access ISP to other is too costly.

Network structure 1 -

Has a global ISP, each access ISP is connected to the global ISP. The global ISP will be complex and for it to be profitable, it needs to charge all access ISPs.

Network Structure 2 -

Has multiple global ISPs which need to be interconnected.

Network Structure 3 -

Has many access ISPs connect to regional ISPs, which connect to tier-1 ISPs (but all global ISPs are not accessible to all regional ISPs).

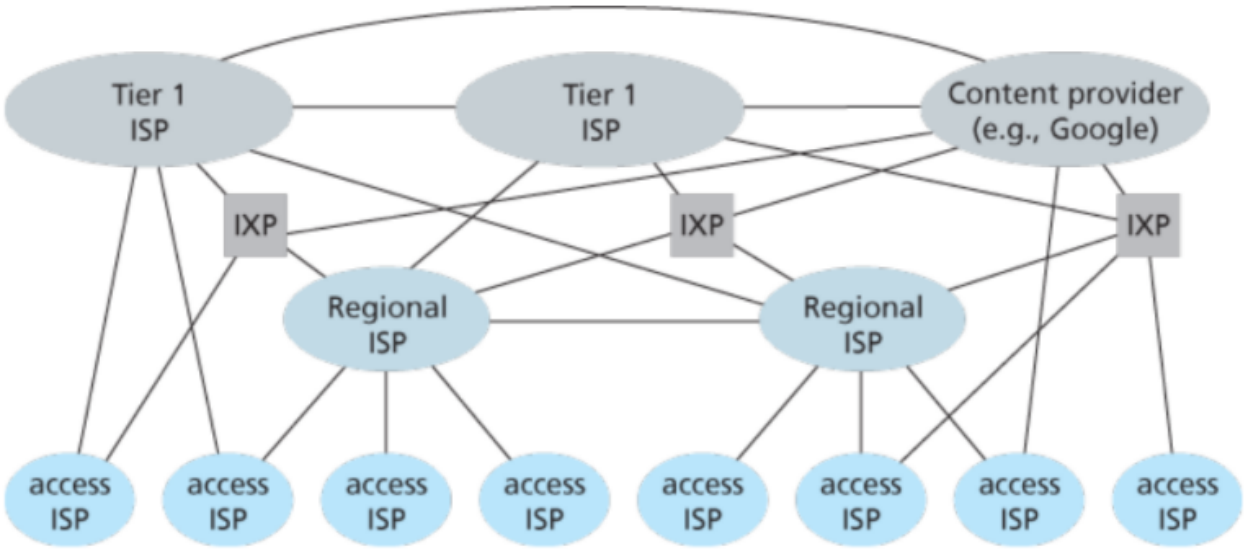
There are about a dozen tier-1 ISPs such as AT&T. Sprint and a customer-provider relationship at each level of hierarchy.

Network Structure 4 -

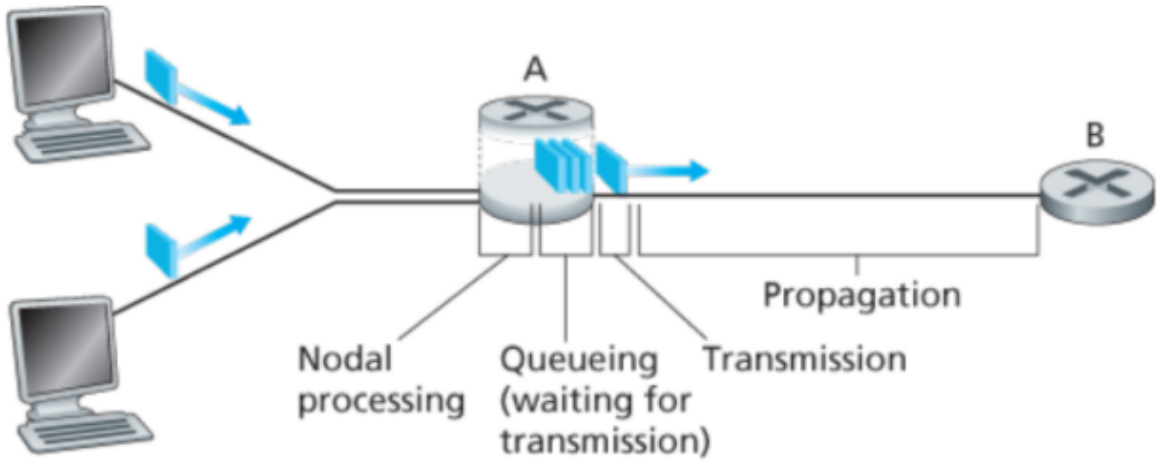
There can also exist a larger regional ISP in the hierarchy; and such a network structure is close to what we have in the Internet. But, we must have **PoP** (**points of presence** - a group of routers in the provider's network from where the customer connects into the provider ISP) at all levels, except, the access ISP level. An ISP may choose to **multi-home**, i.e. connect to more than one ISP of higher level, so that in case of failure of one provider, it can still have access to the internet. To reduce costs associated with these connections, the ISPs at the same level can **peer**, i.e., they communicate amongst themselves without any charges on any of the ISP. A third party company can create an **internet exchange (IXP)**, which is a meeting point for multiple ISPs to peer.

Network Structure 5 -

It adds content provider networks such as google. Google has 50-100 datacenters, each of which houses hundred of thousand of servers or lesser, and are interconnected via peering. They also peers with lower tier ISPs directly or via IXPs. It also connects with tier 1 to reach access ISPs that cant be reached via peering directly. By creating a network of its own, it reduces its payments and has a great control of how its services are delivered to end-users.



Delay, Loss and Throughput in Packet-switched networks



Nodal delay - Time taken for a single hop - made up of 4 delays.

Types of delay -

1. Processing Delay - time required to examine the packet's header, determine where to direct the packet, check bit level errors. Order of microseconds or less.
2. Queuing Delay - the packet is queued onto a buffer, and waits to be transmitted 'til all the packets before it are gone, this depends on the packets already in the queue. The number of packets that an arriving packet might expect to find is a function of the intensity and nature of the traffic arriving at the queue. Order of microseconds to milliseconds.
3. Transmission Delay - length of packet (L) / transmission rate (R) - is the time required to push all of the packet bits into the link. Order of microseconds to milliseconds.
4. Propagation Delay - distance between two routers / propagation delay - the time required from beginning of the link to next router, this depends on the speed of the medium of the link. Order of milliseconds.

Queuing Delay and Packet Loss

Queuing delay can vary from packet to packet, hence, for one entire message we consider the average, variance and probability of queuing delay.

Units of packets arriving at a link = a/sec

Bits per packets = L/packet

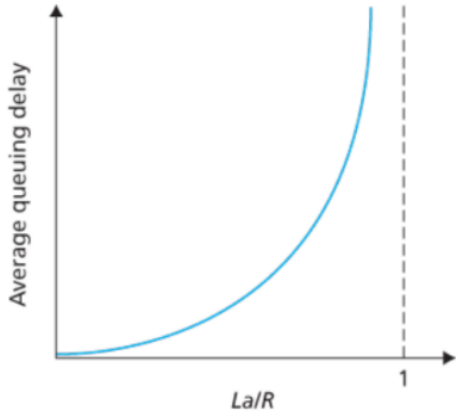
Total bits arriving at a link = La/sec

Transmission rate = R

Traffic intensity = La/R

If La/R > 1 ⇒ Queue will tend to increase

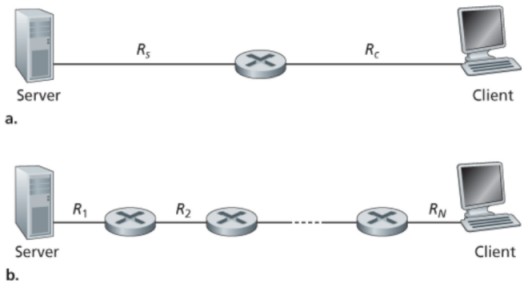
If La/R ≥ 1 ⇒ Nature of traffic (burst or periodic) affects the queue length; usually its random.



Packet Loss - The queuing capacity before a link is finite, so if a packet arrives to a full queue, a router will just drop that packet. The fraction of packet loss increases as the traffic intensity increases.

Throughput -

1. Instantaneous Throughput - At any instant of time, the rate at which host B is receiving the file.
2. Average Throughput - F/T where F is the number of bits in the file and T is the number of seconds all the F bits take to reach host B.



a, b → If Rs < Rc ; throughput = Rs ; If Rc < Rs ; throughput = Rc.

or

throughput = min(Rc, Rs)

Throughput is affected by the slowest link, and the intervening traffic.
Note - Network core in the internet is very fast and not congested, the limiting factor is access net.

Network Architecture

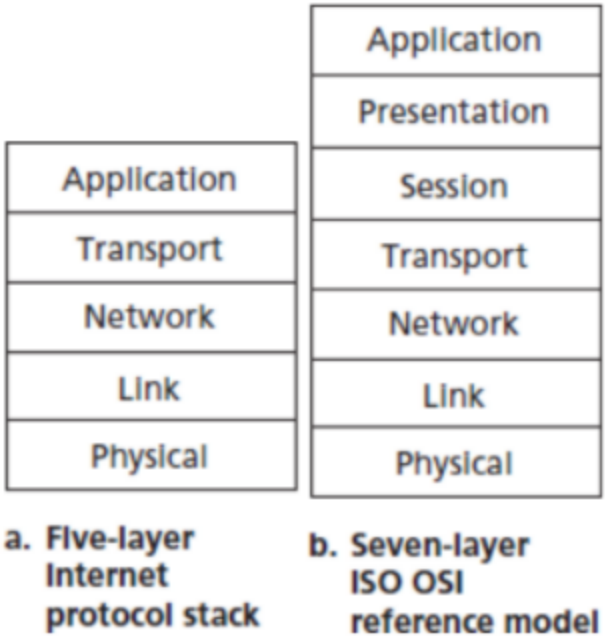
Organizing the network.

Protocol layering - layered architecture - change in one layer can keep rest of the network unchanged.

Service model - a layer offer services to the layers above it by using the services of layer below it.

Structural advantages of layering - modularity, easy to update, well-organized.

Disadvantage of layering - one layer may duplicate the lower layer functionality, layers are interlinked and hence aren't truly independent.



Five-layer Internet Protocol Stack -

- **Application layer -**

  HTTP - web documents request and transfer

  SMTP - transfer of email messages

FTP - transfer of files between 2 end systems.

DNS - Translation of human-friendly names to 32-bit network address.

Application layer packet of information - message.

- **Transport layer -**

  TCP - connection oriented. reliable transport, flow control, congestion control

  UDP - connectionless, unreliable data transfer, no flow or congestion control

  Transport layer packet of information - segment.

- **Network layer -**

  IP → glue that binds internet together

  Several routing protocols

  Network layer packet of information - datagram

- **Link layer -**

  Some link-layer protocols provide reliable delivery, from transmitting node, over one link, to receiving node, like Ethernet, WiFi and DOCSIS protocol - link dependent.

  Link layer packet of information - frames

- **Physical layer -**

  Link dependent - depend on the actual transmission
  medium of the link.

OSI (Open Systems Interconnection) Model

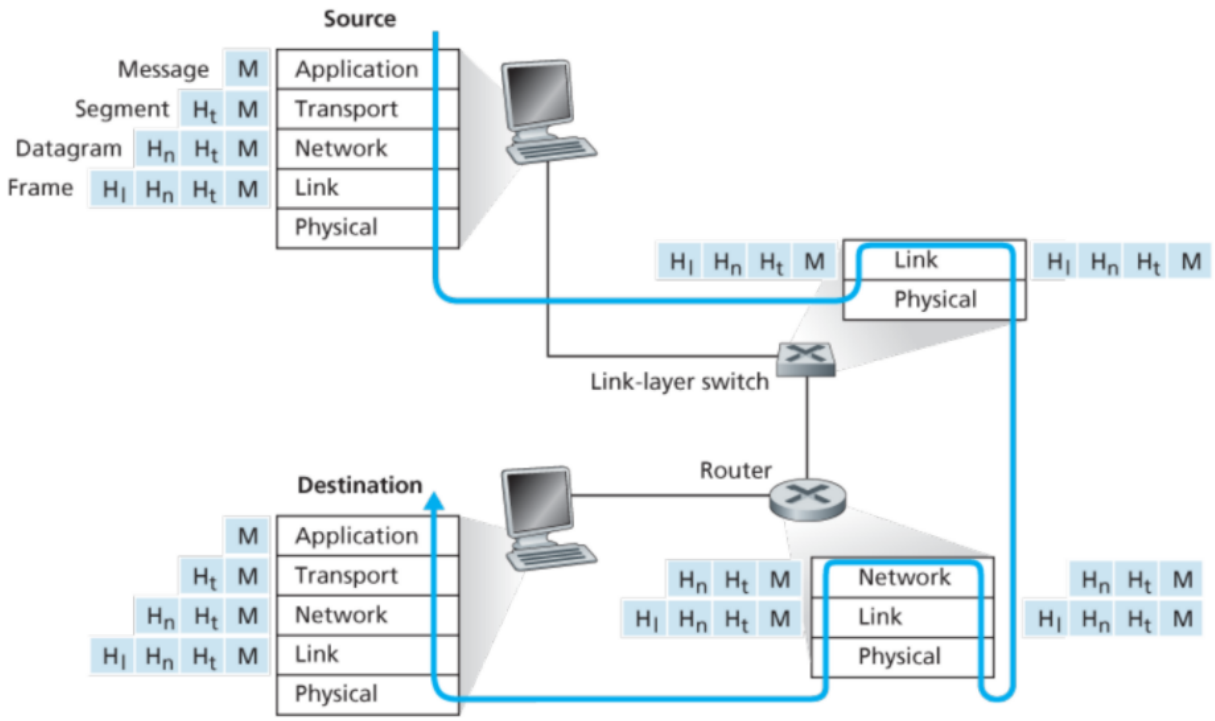
5 layers + session layer + presentation layer

- Presentation layer -

  Data compression and data encryption, data description

- Session layer -

  Synchronization of data exchange, checkpointing.



Network threats

1. Malware - Malicious stuff that can infect our devices - delete our files, install spyware etc.
2. Network of compromised devices - botnet.
3. Goal - Spam e-mail distribution, denial of service attacks.

Types of malware -

1. Virus - needs user interaction, example - a code in the email that user needs to run.
2. Worms - attacks the host without user interaction through vulnerable network applications.

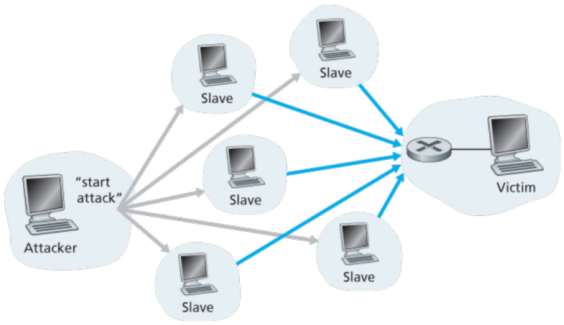
Denial of Service (DoS) attacks -

Renders a host/network unusable to the users.

1. Vulnerability attacks - sending well crafted messages to vulnerable applications.

2. Bandwidth flooding - sending many packets to target host, to prevent actual packets from reaching the host.
3. Connection flooding - establishing a large number of half open or fully open TCP connection with target host.

Distributed DoS - a single source may not be able to generate enough traffic to congest the route to the target host, moreover, if the entire traffic is from one source, it can be detected fairly easily. Hence, the attacker does a DDoS, and controls several hosts and blasts the traffic at the target. A controlled botnet is employed to do a DDoS.



Packet Sniffer -

A passive receiver that is kept in the vicinity of a wireless transmitter to receive every copy of the packet shared. Sniffed packets can be analyzed offline for sensitive information such as passwords, SSNs etc. Packet sniffers are hard to detect but a defense, cryptography can be used to secure our data.

IP Spoofing -

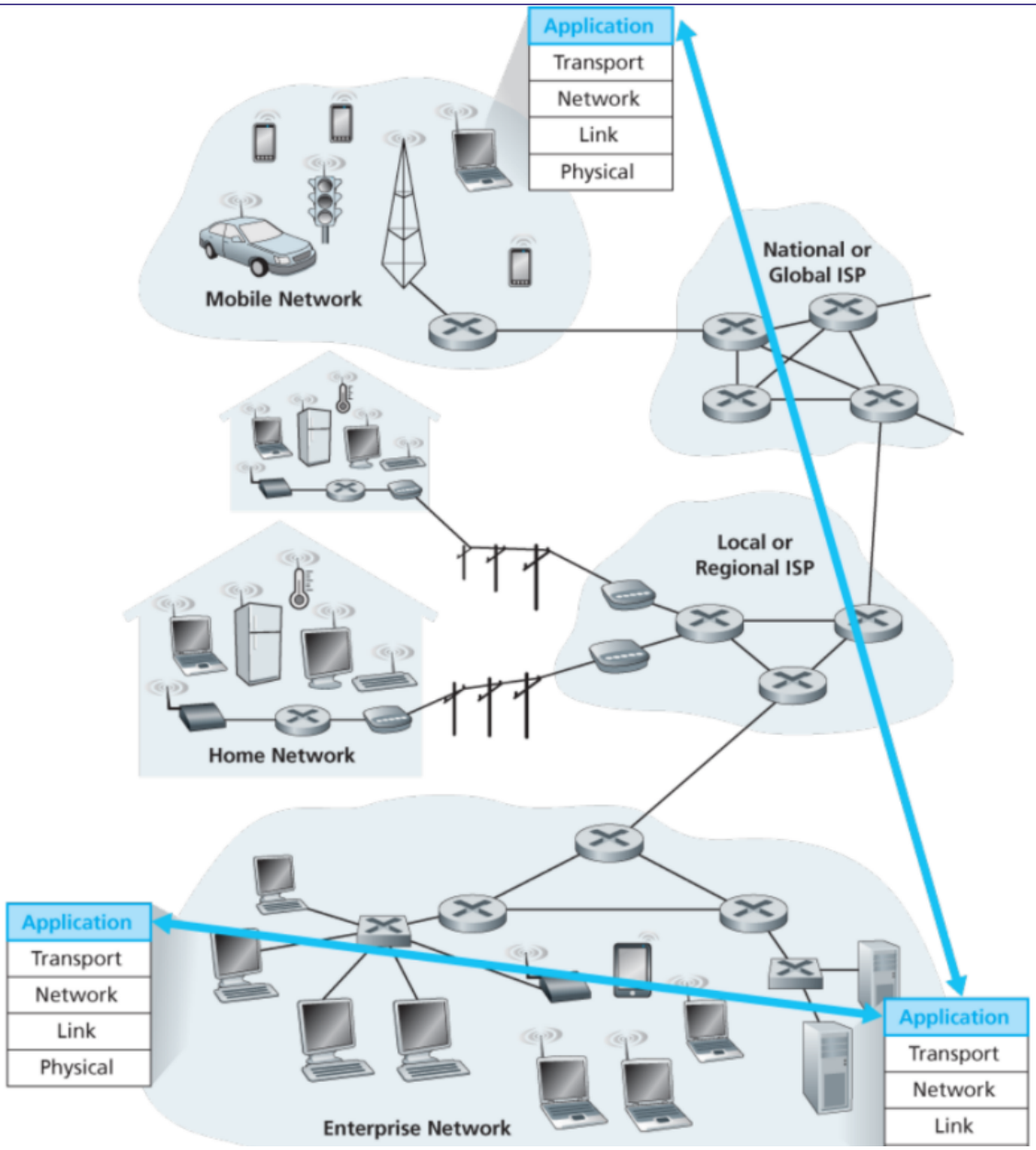
Injecting packets into the internet with false source address. The attacker masquerades as someone else. We do an end-point authentication to validate the source.

History of Internet

Packet Switching → ARPA Network - IBM's SNA → TCP, UDP and IP → Aloha → Ethernet protocol → DNS → World wide web

Application Layer

Application development is writing programs that can run on different end systems and communicate with each other over the network. They need to run on end systems only and not on the network core devices.

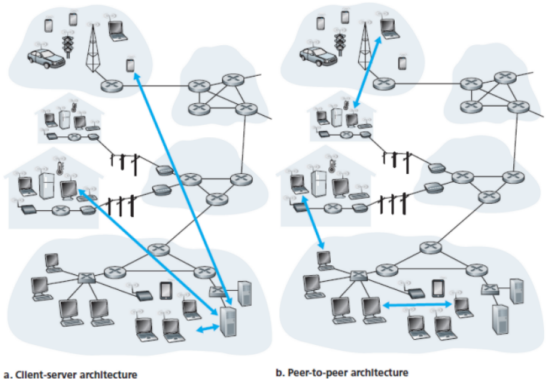


Application developer dictates how the application is structured over various end systems. The two popular application architectures are -

1. Client-Server Architecture - Always on host (called server), responds to requests by clients. Clients don't communicate directly, but via server, which has a fixed and well-known IP

address. Examples are Web, FTP, Telnet and email. **Data center,** hosting a number of hosts, creates a virtual server.

2. Peer2Peer Architecture - The application exploits direct communication between pairs of connected hosts, called peers such as Skype. This architecture is self-scalable, since each new peer generates workload but also increases service capacity. This is also cost-effective.



Process Communication

A process is a program running on one of the systems. The processes across different end systems, possibly having different OS, communicate to each other by sending messages to each other across the internet.

Client and Server Processes -

A client browser exchanges messages with a web server process.

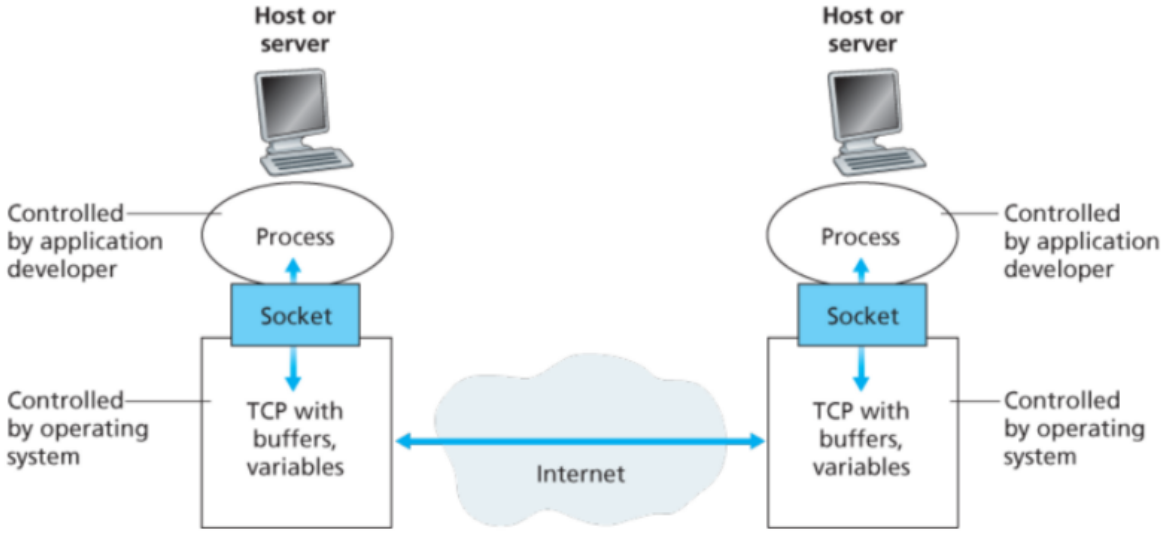
P2P Processes -

When a file is transferred from a process to another, the sender is labelled as the server and the receiver is labeled as client.

Generally, a process that communicates first is considered client and a process that waits to be contacted to begin a session is considered the server.

Socket - A process sends and receives message to/from the network through a socket. The process assumes that there is an underlying transport protocol at the socket for delivery of the message. A socket is the interface between the application layer and the transport layer and is referred to as Application Programming Interface (API). An application developer has very little control over the transport layer services, just the

choice of the transport protocol and the a few parameters such as maximum buffer and maximum segment size.



A receiver's host (IP address) and process (port number) identifier needs to be specified to send the message to the receiving process.

What transport service does an application need?

1. Reliable data transfer - the data transferred should be complete and correct, however, loss-tolerant applications may not need a protocol that provides this service.
2. Throughput - a guaranteed bandwidth is required by bandwidth-sensitive applications. On the other hand, elastic applications can make use of as much or as little bandwidth as available.
3. Timing - the time limit of data transfer is appealing to interactive real-time applications.
4. Security - encryption to ensure confidentiality, end-point authentication and data integrity.

Transport service requirements: common apps

application	data loss	throughput	time sensitive?
file transfer/download	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kbps-1Mbps video:10Kbps-5Mbps	yes, 10's msec
streaming audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	Kbps+	yes, 10's msec
text messaging	no loss	elastic	yes and no

Internet transport protocol services

1. TCP services - connection oriented using handshaking procedure, full duplex communication, reliable data transfer, congestion control - TCP throttles a sending process if the path is congested and also limits each connection to its fair share of network bandwidth.

   Security for TCP - Secure socket layer (SSL)

2. UDP - no frills, lightweight, connectionless, no reliable data transfer service, no congestion control

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

Application-layer protocols

It defines the types of message (request or response) exchanged; the syntax of various message types; the semantics and meaning of the information in various fields; rules to determine when or how a process sends and responds to a message.

Web application ⇒ Standard for document formats (HTML) + Web browser + Web server + application layer protocol (HTTP)

Internet E-mail application ⇒ Mail servers + mail clients + standard for defining structure of email + application layer protocol (SMTP)