Computer Communication Networks

Application Layer

Dr. Raja Vara Prasad

Assistant Professor

IIIT Sri City

Application Layer

Network Applications

Network application development -- writing programs that run on

different end systems and communicate with each other over the network

Example:

Web application → two distinct programs that communicate with each other:

- the browser program running in the user's host (desktop, laptop, tablet, smartphone, and so on);
- the Web server program running in the Web server host.
- in P2P file-sharing system there is a program in each host that participates in the file-sharing community

Network Applications

- do not need to write software that runs on network core devices, such as routers or link-layer switches
- Network core devices do not function at the application layer
- function at lower layers— specifically at the network layer and below

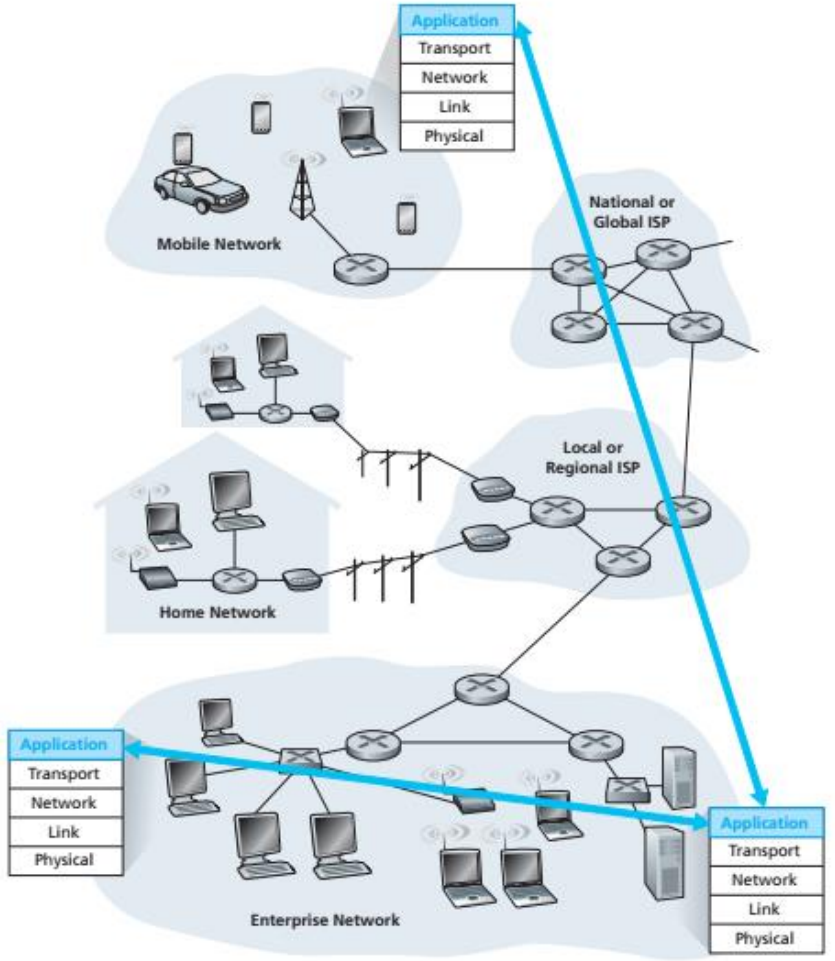


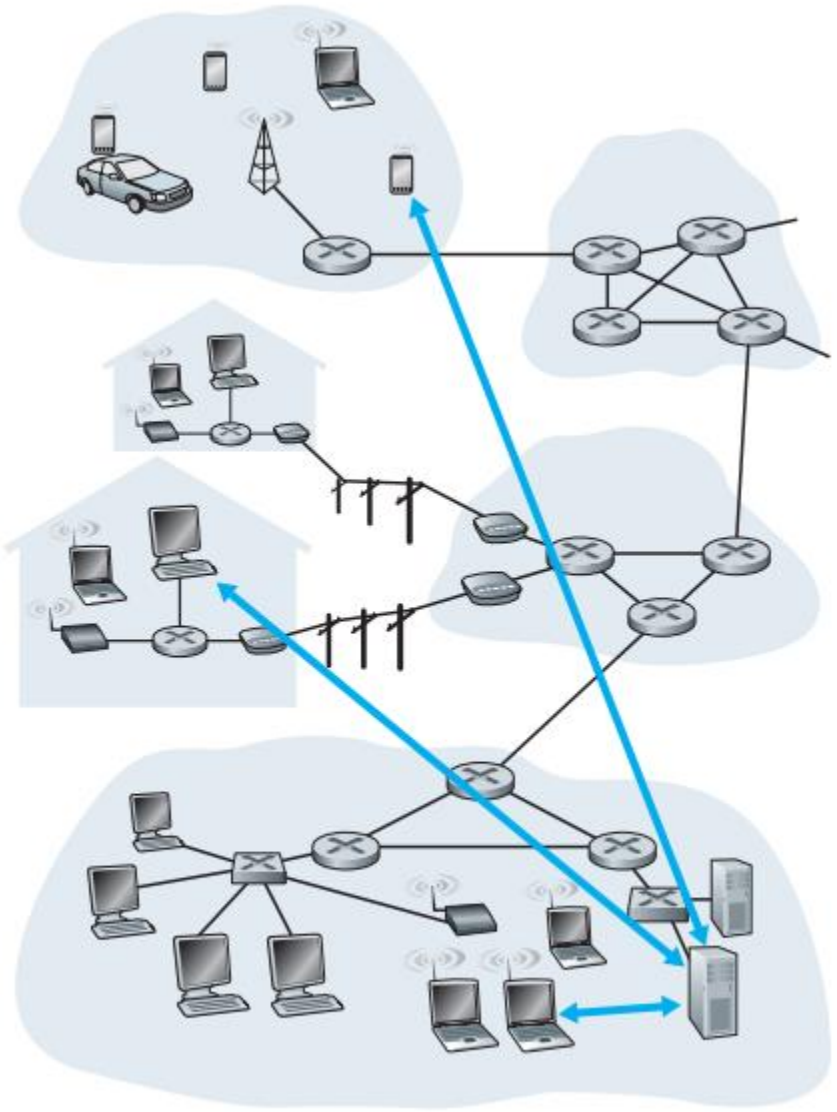
Figure 2.1 ♦ Communication for a network application takes place between end systems at the application layer

Network Applications

- Applications use the services of network (Transport layer)
- For an application developer, architecture and services of network are fixed
- Architectures of applications:
  - Client-Server architecture
  - Peer-to-Peer (P2P) architecture
- Application developer decides on the architecture and services of transport layer to be used.

Client-Server Architecture

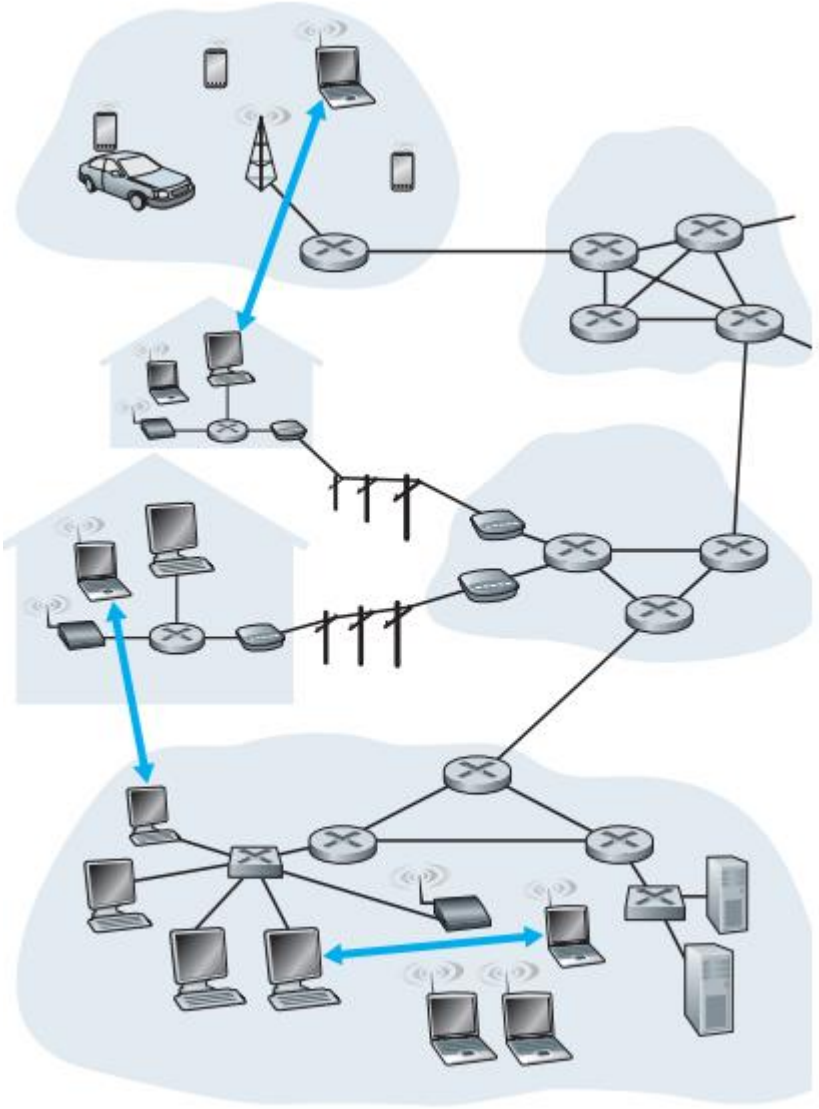
- Server: An end system that serves the requests from various hosts.
- A server is always ON.
- Client: An end system that requests a server for content.
- A client can be either ON-OFF or always ON.
- Example applications using this architecture: web, e-mail, file transfer, etc.



a. Client-server architecture

Peer-to-Peer Architecture

- End systems communicate by a direct connection.
- The end systems are called peers.
- Example applications: skype, internet telephony, torrents, etc
- Advantages:
  - File distribution
  - Self-scalable: can handle growth in traffic
  - Cost effective: no server infrastructure and server bandwidth.
- Challenges in P2P Architecture:
  - ISP friendly: asymmetric data traffic.
  - Security
  - Incentives: Peers should share bandwidth.



b. Peer-to-peer architecture

Processes Communicating

- A process is a program that is running within an end system.
- A client process is a process running on a client and a server process is process running on a server.
- It is the client process and server processes that are actually communicating.
- A process sends and receives messages to and from transport layer through a software interface known as socket.
- A socket is also known as Application Programming Interface (API).

Interface Between the Process: API

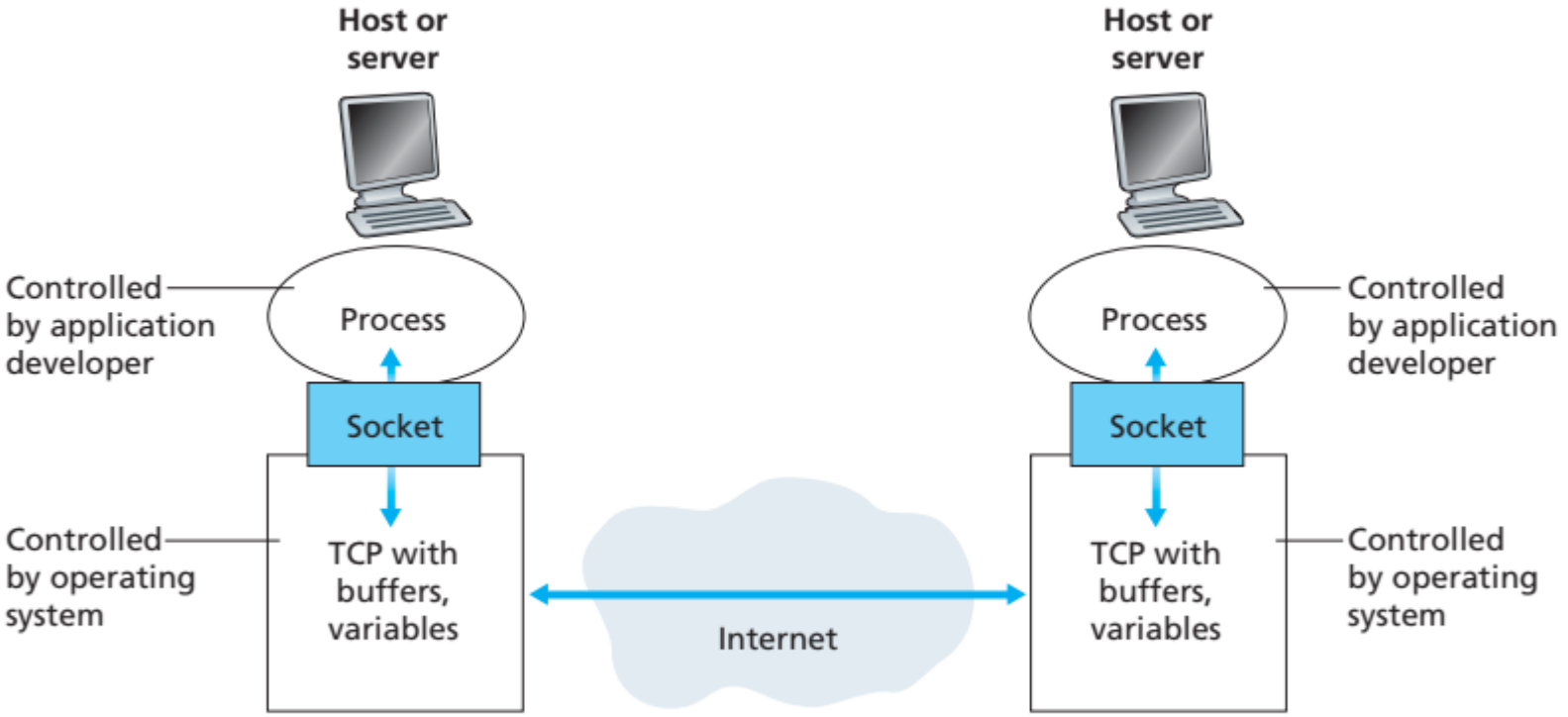


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

Services of Transport Layer

- Reliable data transfer: Guaranteed data delivery service.
- Throughput
- Timing: for example, it is guaranteed that a packet will be delivered no more than 100 msec later.
- security: end-point authentication, encryption and decryption.

Requirements of Applications

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Figure 2.4 ♦ Requirements of selected network applications

Transport protocols

- Transmission Control Protocol (TCP)
  - Connection oriented service: handshaking, full-duplex connection
  - Reliable data transfer service: packets get delivered without error and in proper order.
  - Congestion control
- User Datagram Protocol (UDP)
  - Connectionless
  - Unreliable data transfer service.
  - No congestion control

Applications

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