Plan

- **Application layer**
  - Application layer protocols
  - The Web and HTTP
  - File transfer: FTP
  - Electronic mail in the Internet
  - DNS — The Internet's directory service
  - Socket programming with TCP and UDP
  - Content distribution

Application layer

The network applications are the reason why we use the Internet, because they are the programs we interact with to connect our host to another host on the network.

There are many popular network applications:

- electronic mail, remote access to computers, file transfers, newsgroups, text chats (in the 1980's);
- Web (mid-1990's);
- instant messaging, peer-to-peer (end of 1990's).

Plan

- Application layer
  - **Application layer protocols**
  - The Web and HTTP
  - File transfer: FTP
  - Electronic mail in the Internet
  - DNS — The Internet's directory service
  - Socket programming with TCP and UDP
  - Content distribution

Application-layer protocols

A network application is made of several distributed software components, each of them running on different hosts.

Precisely, we do not say that programs communicate with each other, because programs are basically texts. The execution of a program gives birth to the notion of **process**. Processes are the communicating entities.

Network applications use application-layer protocols which define the format of the exchanged messages and the actions undertaken by the hosts on the receipt and sending of given messages.

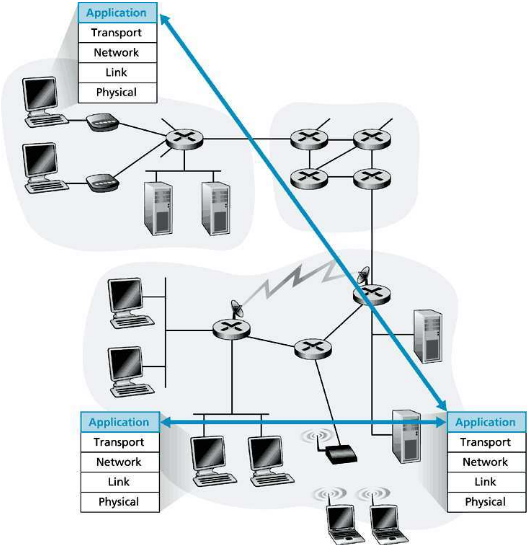
Application-layer protocols (cont)

The figure in the next slide shows the **protocol stacks** at each communicating host.

Applications are processes which interact, within the same layer, with the **operating system** and the user, and use the transport layer services for communicating across the network.

This behaviour is depicted using solid blue arrows.

Application-layer protocols (cont)



Application-layer protocols (cont)

Keep in mind the difference between a **network application** and an **application-layer protocol**:

- a network application is a process;
- an application-layer protocol is the definition of data formats and of the behaviour of communicating network applications.

In other words, a network application **implements** an application-layer protocol, but also do more, like offering to the user a graphical interface etc.

Some application-layer protocols are POP and SMTP for e-mail. The mail client, like Outlook, Thunderbird or Evolution, is a network application.

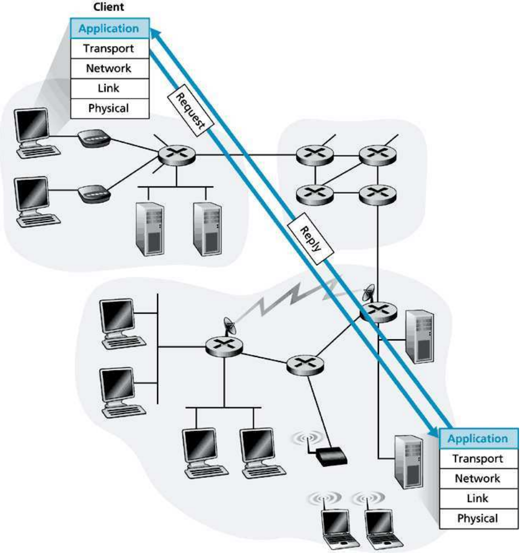
Application-layer protocols (cont)

The figure in the next page shows that a network application is made of a **client** and a **server**.

Usually a host implements both the client and the server, e.g., when an FTP session exists between two hosts, either host can transfer files to the other host.

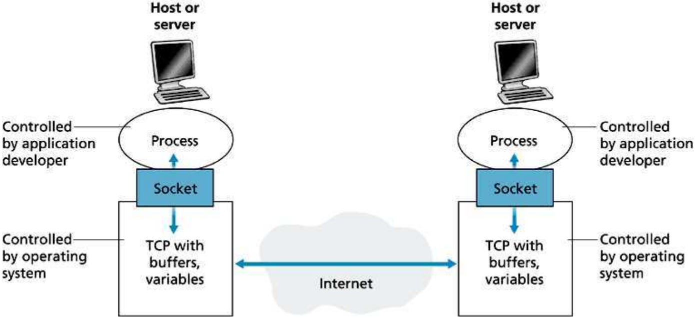
However, the host that initiates the session is labeled the client.

Application-layer protocols (cont)



Application-layer protocols/Sockets

A process sends and receives messages through its **socket**. By analogy, a socket can be considered as an application's door to the network, which is assumed (by the application) to provide a transport service for the messages entering it. Formally, it is an **API** (**Application Programmers' Interface**) between the application and the network.


Host or server
Host or server
Controlled by application developer
Process
Socket
Controlled by operating system
TCP with buffers, variables
Internet
Process
Socket
TCP with buffers, variables
Controlled by application developer
Controlled by operating system

Application-layer protocols/Addressing processes

In order for one process on one host to send a message to another process running on another host, it must identify the receiving process.

For this, two elements are needed on the Internet:

- the **IP address** of the receiving host, which is a 32 bits number globally unique;
- a **port number** within the receiving host, which allows addressing the message to the peer process.

Therefore, a network application must be assigned a port number, e.g., 80 for Web server processes (using HTTP protocol) or 25 for mail server processes (using SMTP protocol).

Application-layer protocols/User agents

A network application usually offers a graphical interface to the user. It is usually called **user agent** because it relays the wishes of the user to the core of the application network, which interacts in turn with the transport layer through sockets.

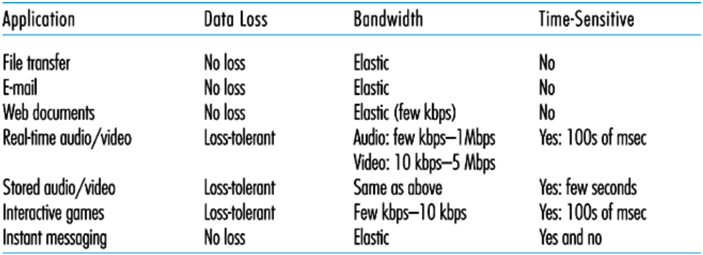
Web browsers and mail readers provide such user agents, while implementing specific application-layer protocols, respectively HTTP (web), SMTP (sending e-mails), POP3 or IMAP (retrieving e-mails).

Application-layer protocols/Transport services

The Internet offers several transport protocols to the network applications:

- **Reliable data transfer.** Some applications, such as e-mail, file transfer, remote host access, web document retrieving etc. require no data loss. Others, such as real-time or stored audio and video playing are **loss-tolerant applications**.
- **Bandwidth.** Some applications, such as Internet telephony, require a minimum bandwidth allocation: they are **bandwidth-sensitive applications**. Others, such as e-mail, can make use of any available bandwidth. They are **elastic applications**.
- **Timing.** Some applications, such as multi-player games, require timing constraints on the end-to-end delay: they are **time-sensitive applications**.

Application-layer protocols/Transport services (cont)



Application	Data Loss	Bandwidth	Time-Sensitive
File transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Real-time audio/video	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
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

Application-layer protocols/TCP services

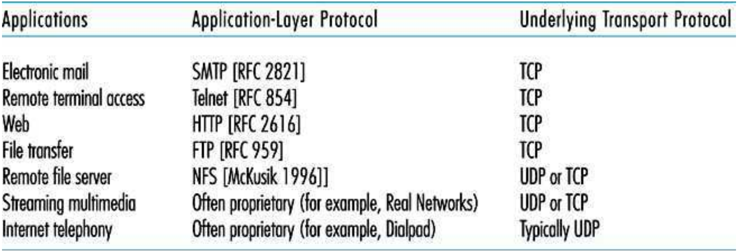
Let us revisit the TCP services:

- **Connection-oriented service.** TCP has the client and the server exchange transport-layer control information before they exchange application-level messages: this is **handshaking**.
  After it, a **TCP connection** is said to exist between the sockets of the two processes.
  The connection is **full-duplex**, i.e., the two processes can send information to the other at the same time. After, the connection must be torn down.
- **Reliable transport service.** The processes rely on TCP to deliver their messages correctly, entirely and orderly.

Application-layer protocols/TCP services (cont)

TCP does *not* guarantee

- minimum transfer rate (due to flow and congestion control),
- end-to-end delay (due to routing protocols and queuing).



Applications	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Remote file server	NFS [McKusik 1996]]	UDP or TCP
Streaming multimedia	Often proprietary (for example, Real Networks)	UDP or TCP
Internet telephony	Often proprietary (for example, Dialpad)	Typically UDP