CIS 457 - Data Communications

Nathan Bowman

Images taken from Kurose and Ross book

Application Layer

Application layer is the reason the rest of the layers exist

Goal of computer network is not to send random bits from A to B -- it is to leverage communication to *do something*

Brief history of important applications:

70s-80s: text e-mail, remote access (think `ssh`, but older), file transfers, newsgroups

90s: World Wide Web -- surfing, search, commerce, P2P file sharing (e.g., Napster), instant messaging

2000s: internet telephony, video conferencing, user-generated video (YouTube), movies on demand (Netflix), more online games (WoW), more social networking (Facebook, Instagram, Twitter), ...

We start studying networks at the top because

- applications are the reason for networks and
- applications are the most familiar way we interact with the Internet

Studying application layer protocols and related issues will give us a feel for the same general issues we see in other layers

We will use the term "distributed applications" for applications that have a networking component

One general consideration for distributed applications is communication structure

Recall that all applications are running on hosts, not the network

In general, distributed applications have one of two **architectures**: client-server or peer-to-peer (P2P)

Typical example of client-server is browsing the web

You have a **client** initiates a connection, and, in this case, requests a web page

The **server**, a host residing elsewhere on the network and waiting for connections, responds to your request by sending the page

Once your client (web browser) receives the page, it displays the information using its knowledge of HTML formatting

The client-server architecture requires that the server

- is always on, listening for connections
- resides at a fixed address that can be found by clients
- has enough bandwidth to service requrests from many clients in a timely manner

To deal with the bandwidth issue, servers for major companies are generally not one computer but are instead stored in data centers

We will describe the process of determining a server's address in another lecture

In peer-to-peer (P2P), there is not a server waiting for many clients to initialize connections

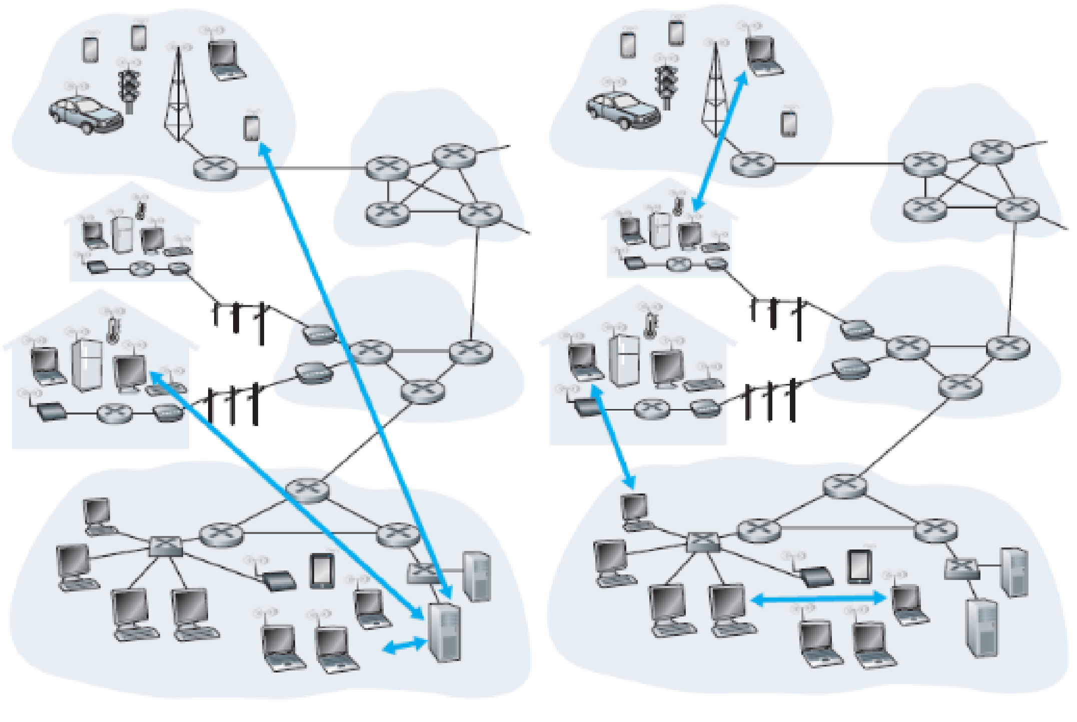
Instead, each process will act sometimes as a client and sometimes as a server

Typical example is P2P file sharing -- each host requests some parts of file from others (acting as a client) and sends other parts of file to others (acting as a server)

Peers come and go from the distributed application, so the connections and addresses are intermittent, unlike with client-server

P2P is inherently scalable, unlike client-server, but comes with its own set of complications

We will talk more about scalability of P2P and file sharing in another lecture



a. Client-server architecture

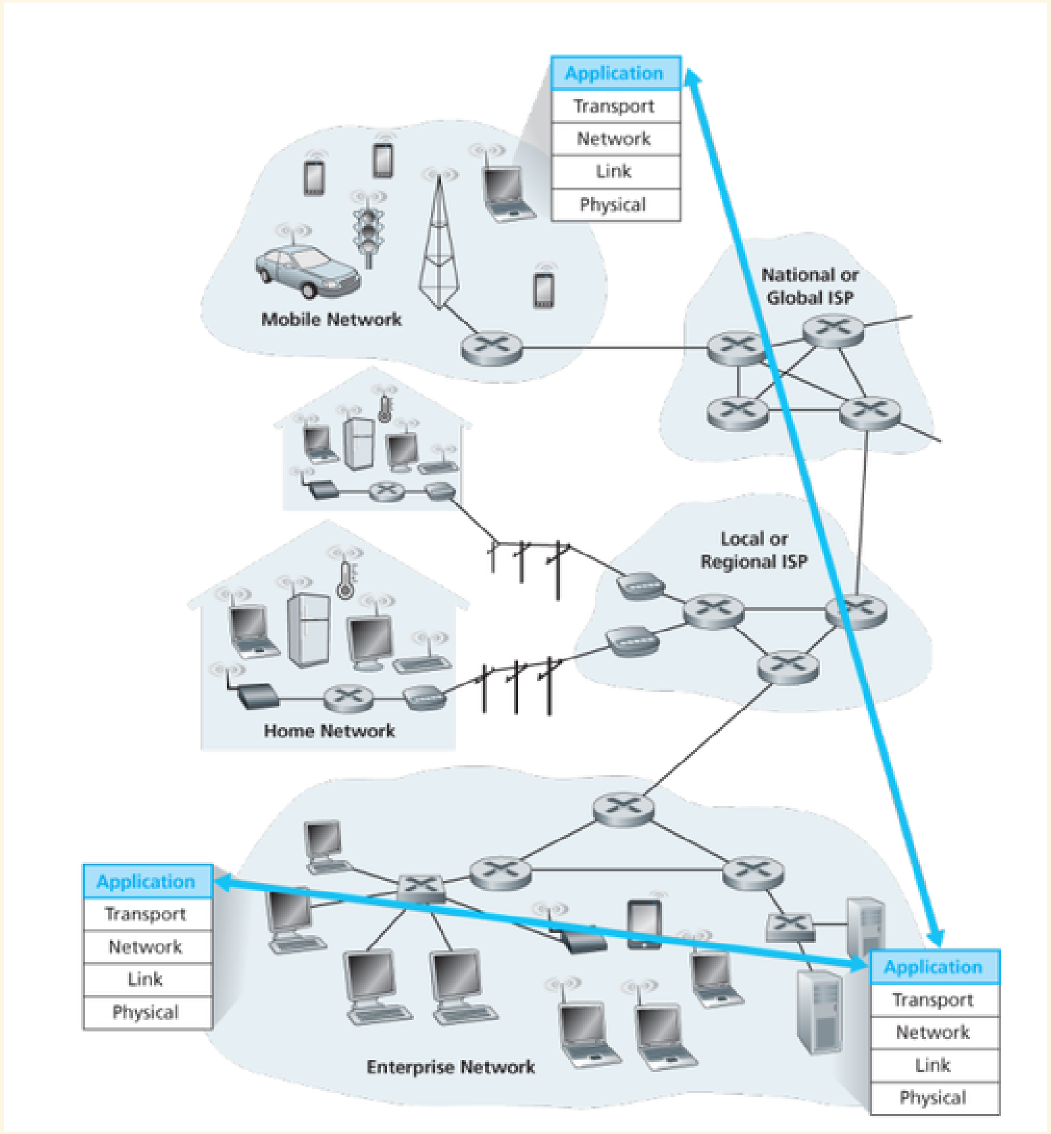
b. Peer-to-peer architecture

Each of these distributed applications is a progam, just like any other application

In a client-server architecture, the client and server are generally running different programs

In a P2P architecture, the hosts (peers) are often running the same program as one another

Any application-layer program is run at the edge of the network -- a router has no idea how email works


Application
Transport
Network
Link
Physical
Mobile Network
National or Global ISP
Local or Regional ISP
Home Network
Enterprise Network
Application
Transport
Network
Link
Physical
Application
Transport
Network
Link
Physical

Clients and servers

When talking about a "server", we often think of a computer that is running elsewhere on the network, but that is not quite right

A client and a server are both processes (an OS term for "running programs"), not machines

For example, the same remote machine could be running a web server and an email server

Clients and servers

It can sometimes be difficult to tell which process is a client and which is a server

For example, in email, the client is the process that sends the mail, which is the opposite of how web clients work

Remember that *the process that initiates a connection is the client*