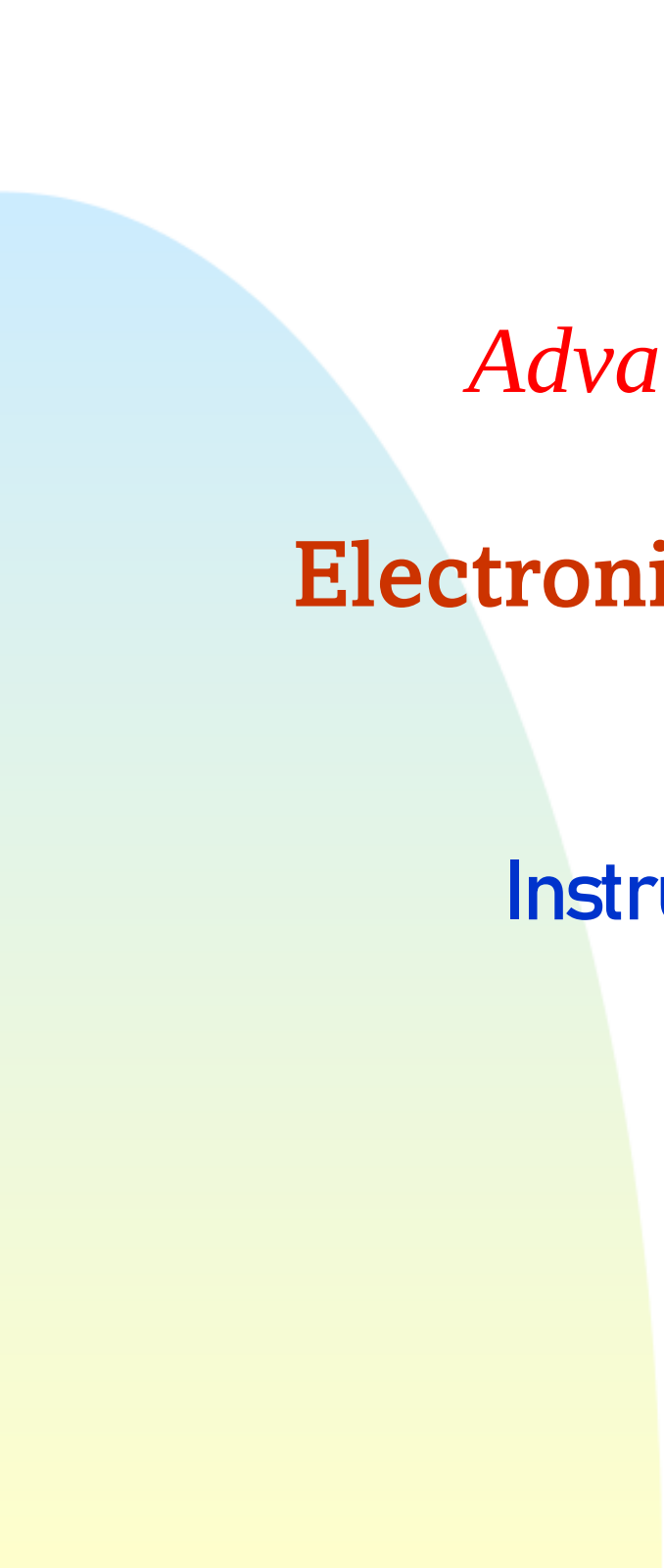


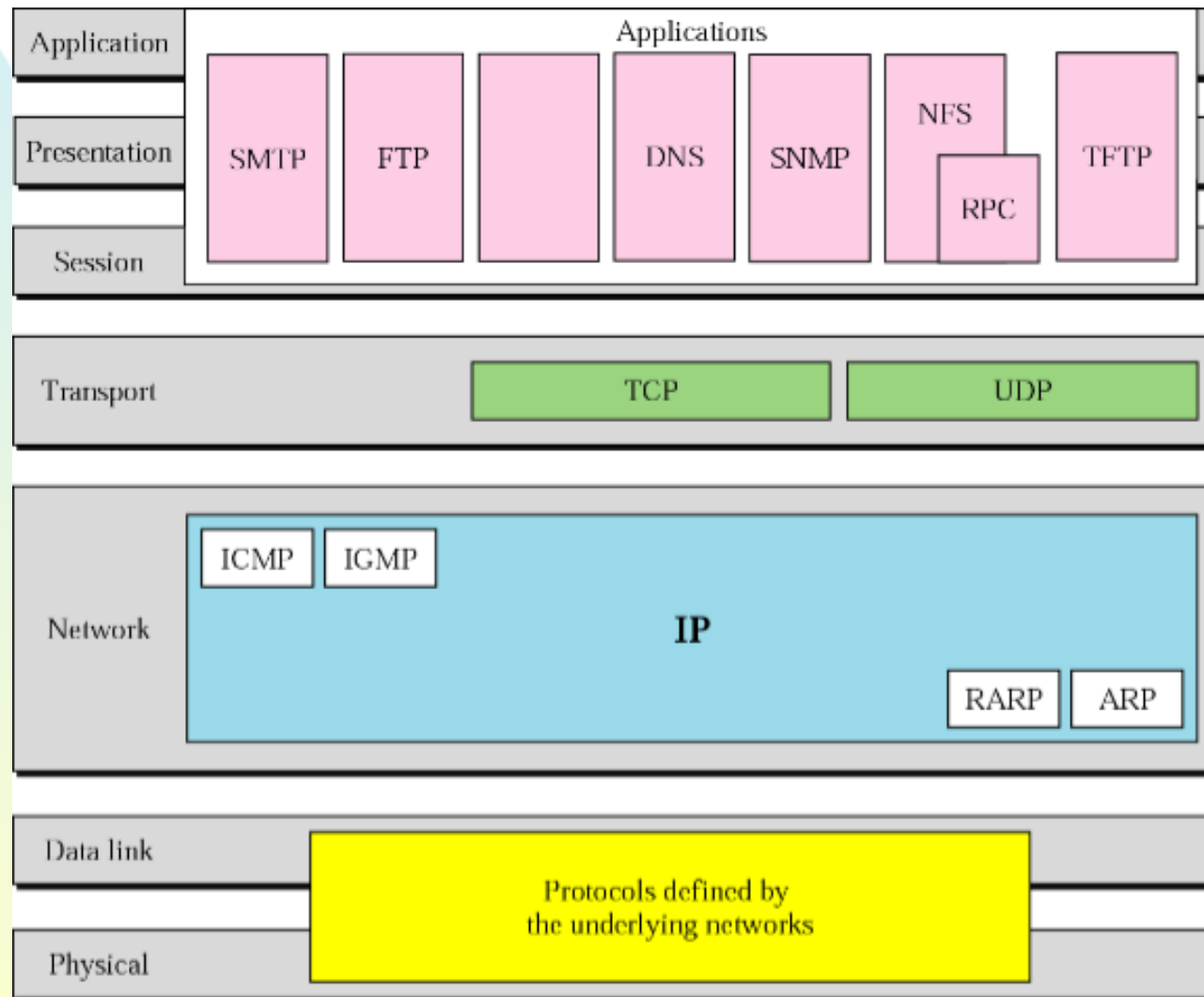


Advanced Internet Technology

Electronic Concept and Application

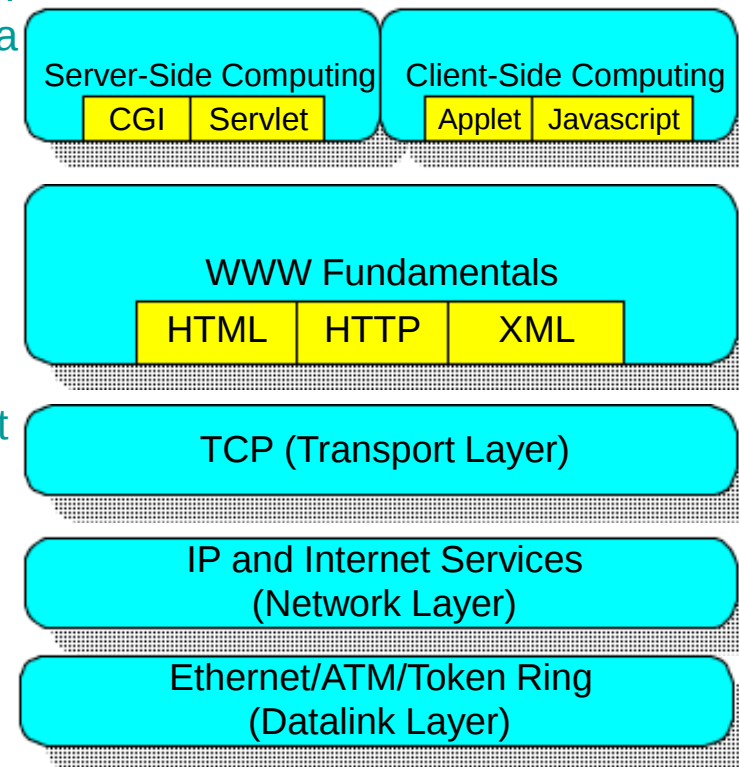
Instructor: Dr.Mohammed Hassan

TCP/IP Protocol Suite



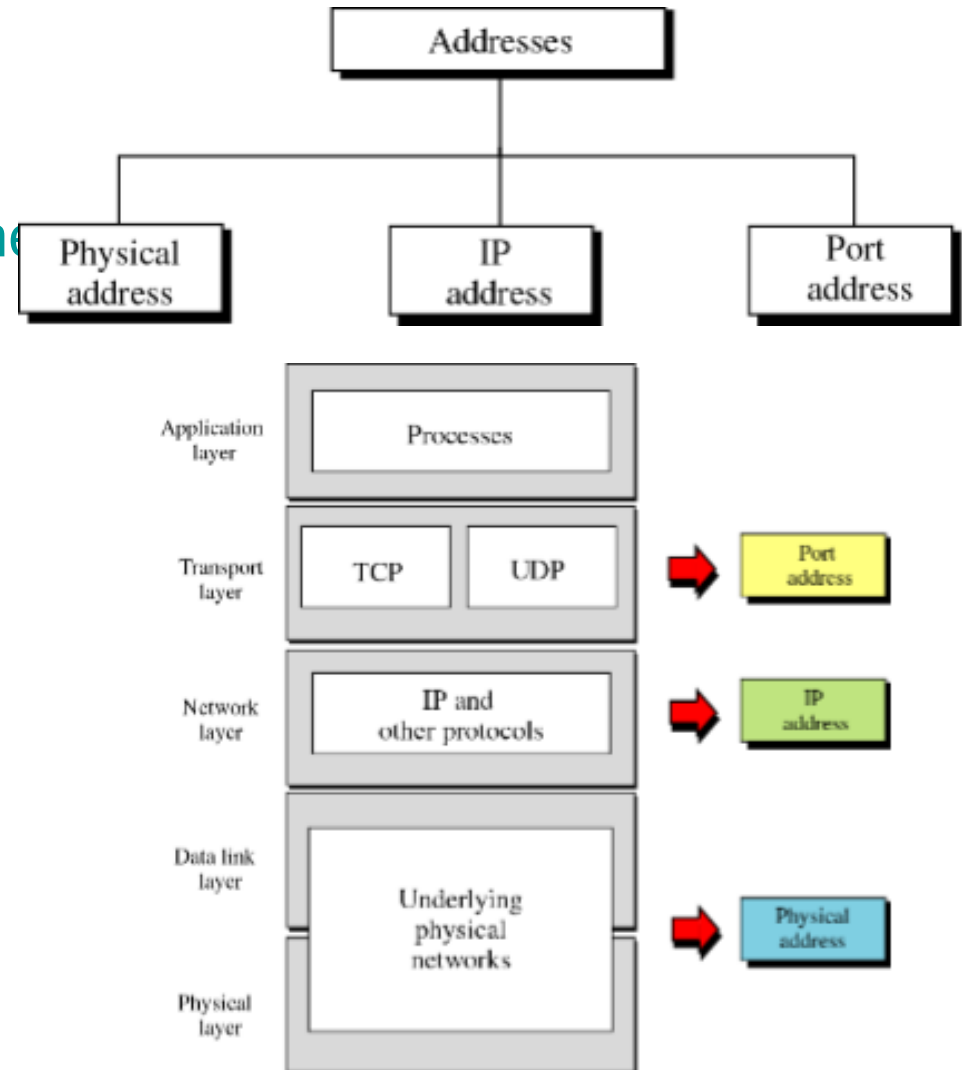
Layered Web Architecture

- Web uses the Internet as the underlying network infrastructure. The complete architecture of the Web is best illustrated using a layered approach.
- **Datalink Layer**: Implements the link interface to the physical medium and provides point-to-point services to move frames around the network.
- **Network Layer**: Overlays a layer of network software to provide uniform network layer services such as routing across different datalink layers.
- **Transport Layer**: Provides end-to-end delivery of packets that resides only in end host. Specifically, it adds services to the network layer to provide reliable transmission.
- **WWW Layer**: Provides the concept of hyperlinking documents across the network. The service uses technologies such as HTML, HTTP, and XML.
- **Server-Side Computing**: Concern with extension of server-side functionality that is beyond delivering of static documents.
- **Client-Side Computing**: Concern with client-side extension to basic web.



Internet Addressing

- Challenges
 - How to send packets from source to destination across the Internet?
 - How to send packets to a specific application?
- Three different levels of addresses
 - Physical address
 - IP address
 - Port address



Application layer

- Principles of network applications
- Web and HTTP
- HTTP Authorization

Some network apps

- E-mail
- Web
- Instant messaging
- Remote login
- P2P file sharing
- Multi-user network games
- Streaming stored video clips
- Internet telephone
- Real-time video conference
- Massive parallel computing (grids)

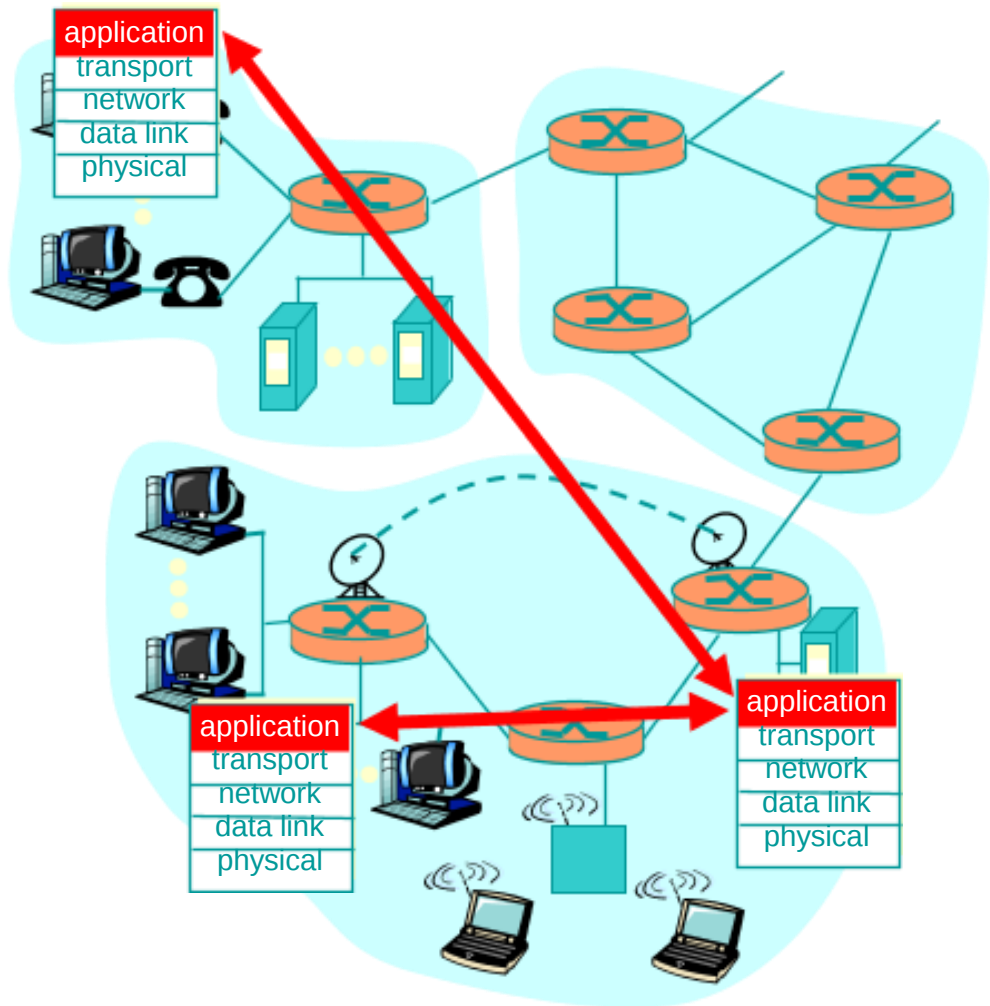
Applications and application-layer protocols

Application: communicating, distributed processes

- running in network hosts in “user space”
- exchange messages to implement application
- e.g., email, ftp, Web

Application-layer protocols

- one “piece” of an app
- define messages exchanged by apps and actions taken
- use communication services provided by lower layer protocols (TCP, UDP)



Client-server paradigm

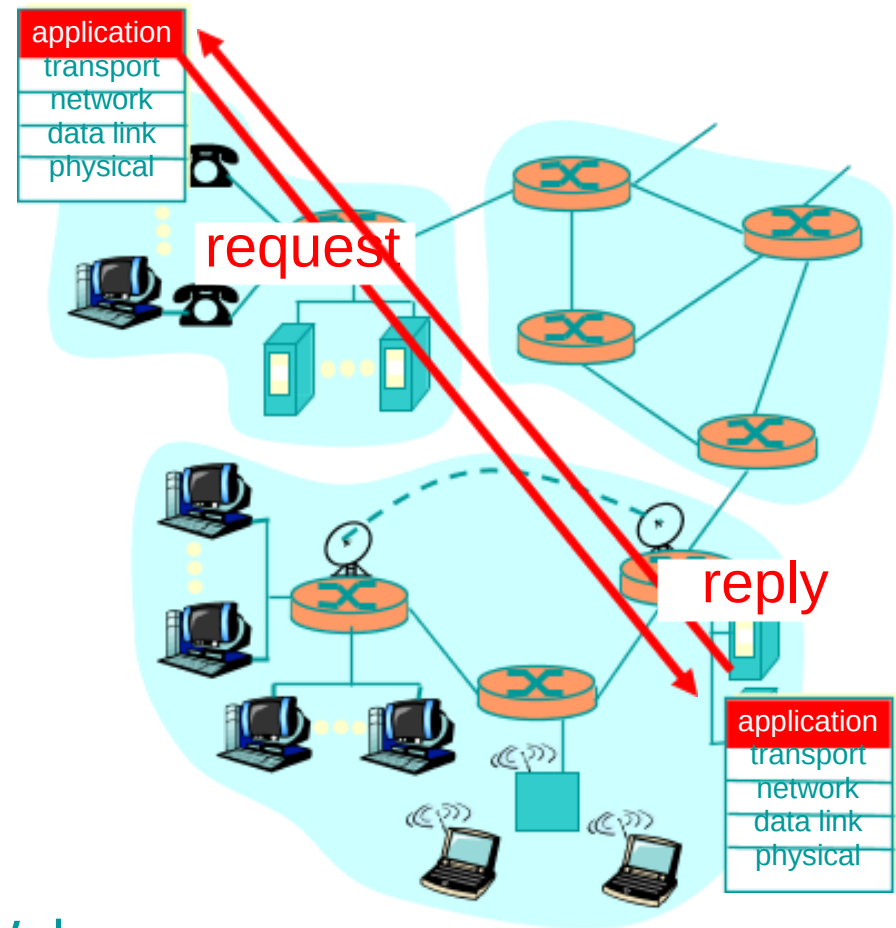
Typical network app has two pieces: *client* and *server*

Client:

- initiates contact with server (“speaks first”)
- typically requests service from server,
- Web: client implemented in browser; e-mail: in mail reader

Server:

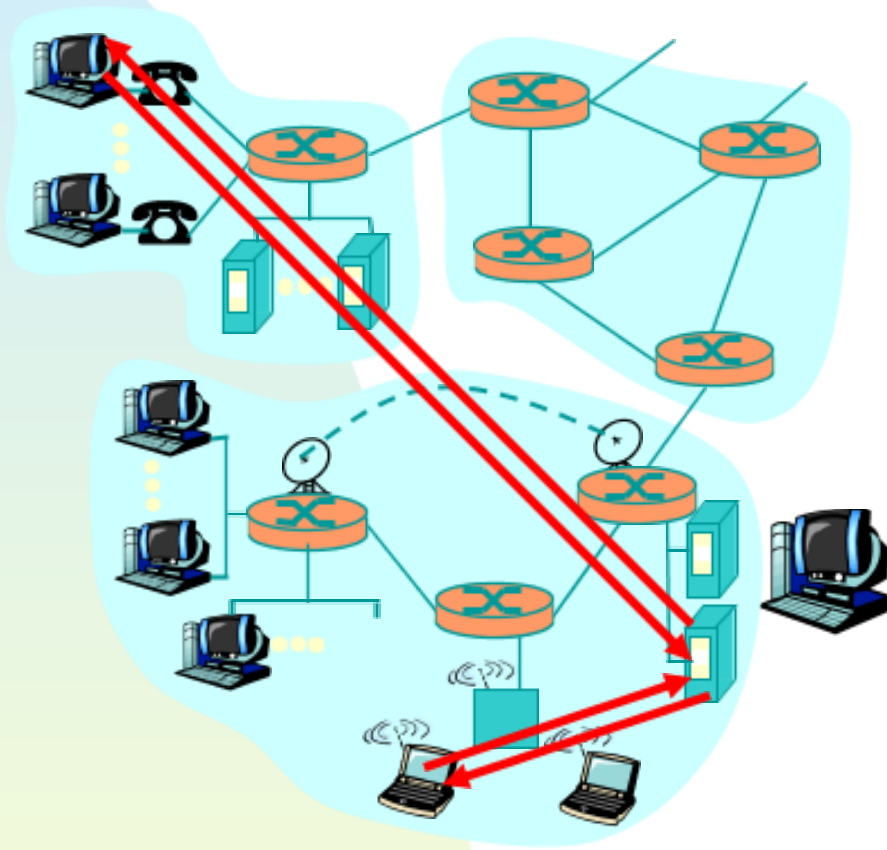
- provides requested service to client
- e.g., Web server sends requested Web page, mail server delivers e-mail



Application architectures

- Client-server
- Peer-to-peer (P2P)
- Hybrid of client-server and P2P

Client-server architecture



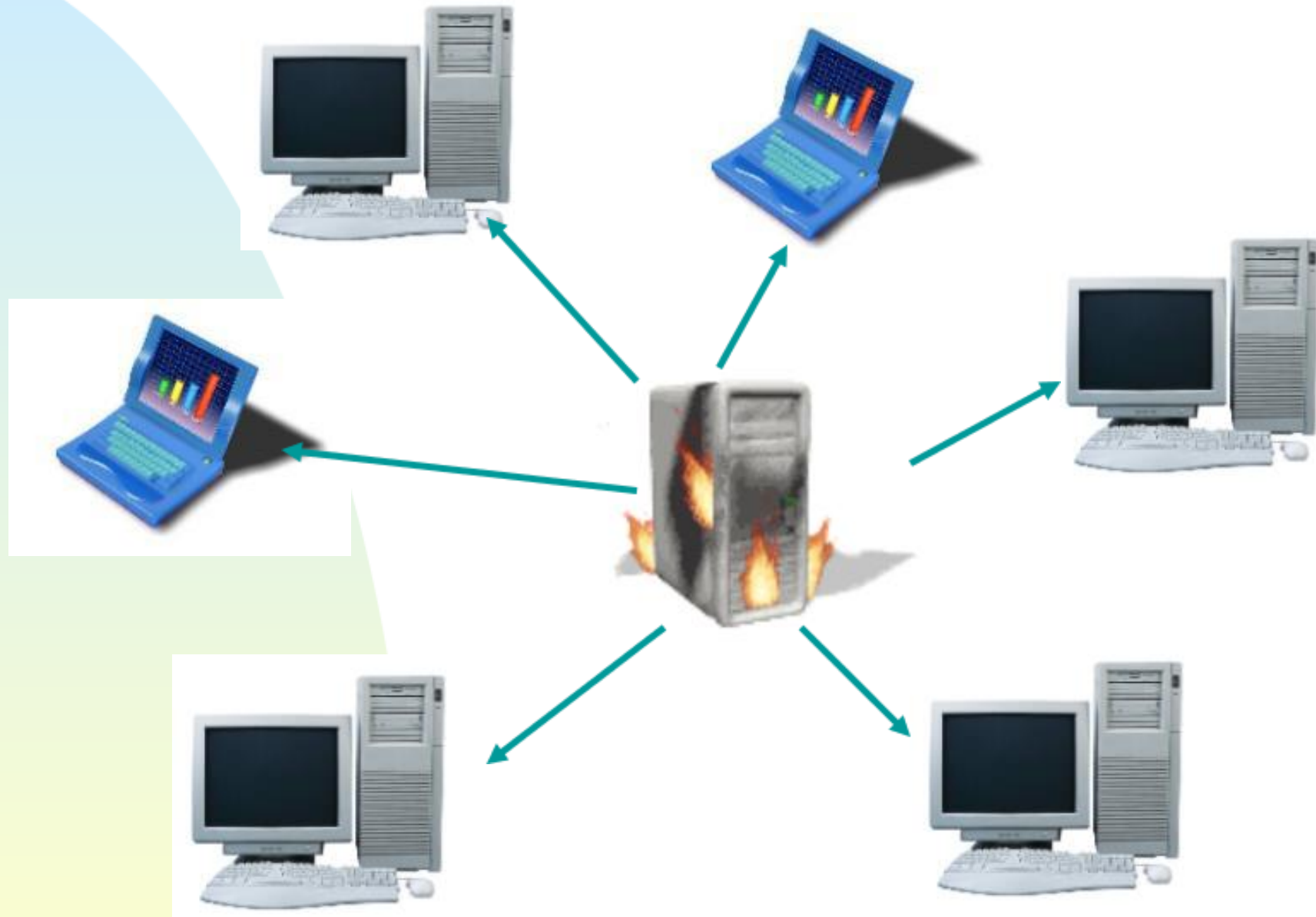
server:

- always-on host
- permanent IP address
- server farms for scaling

clients:

- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Client Server (2)



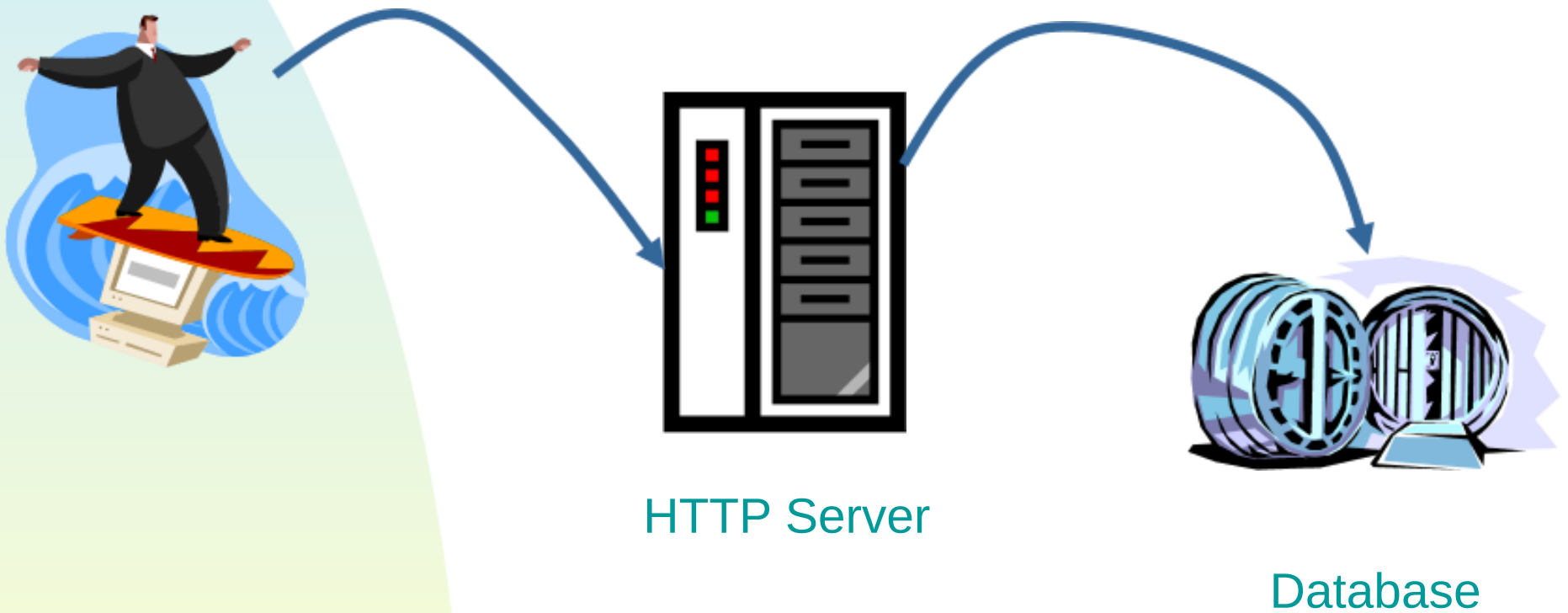
Issues to think about

- Can a node be BOTH a client and a server?
 - what do you think?
 - examples?

Being both a client and a server

- Terms *client* and *server* refer to a service
 - a node (or a program) could be the server of some services and the client of others
- Examples
 - A sports Web *server* might be the *client* of a game-score database
 - A Web proxy server (to a web client) may in turn be a client to an origin web server.

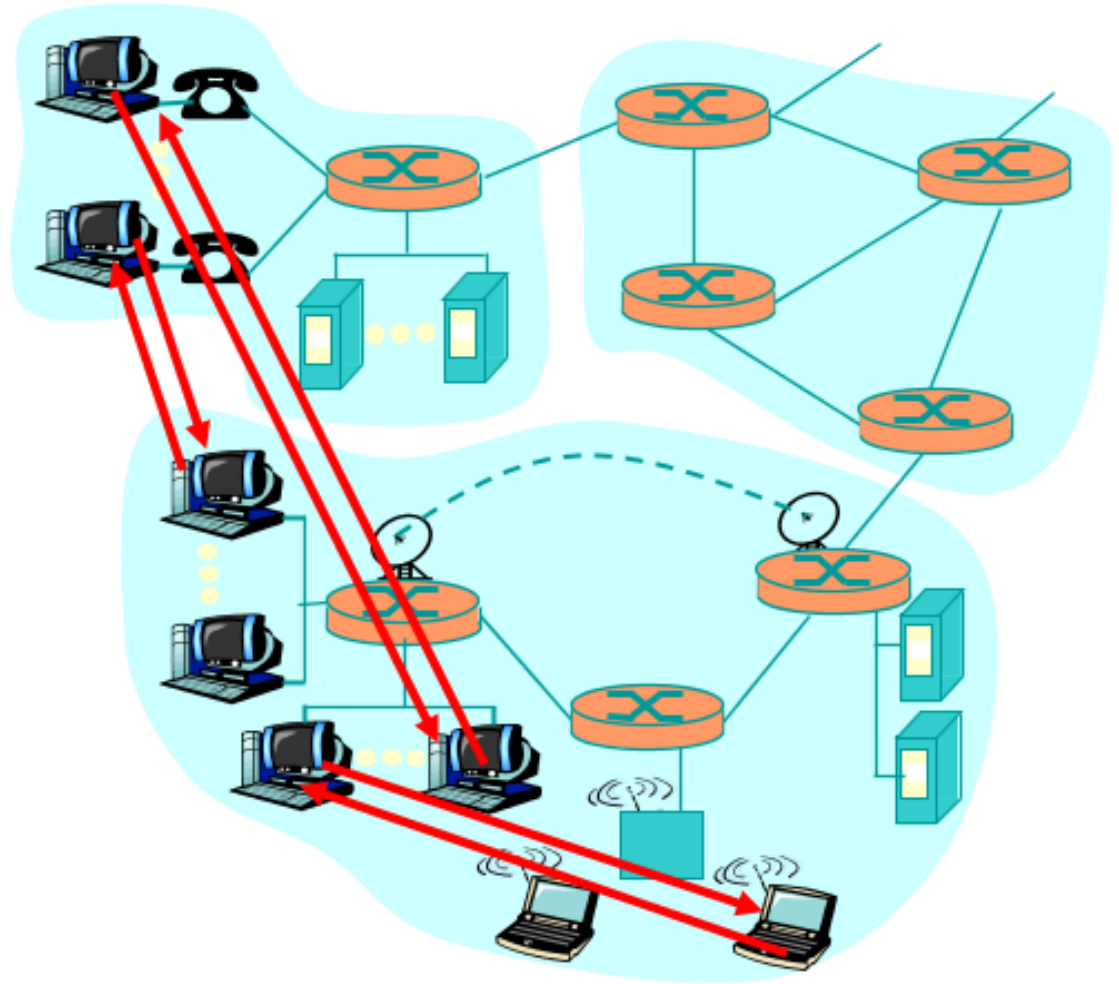
Both server and client

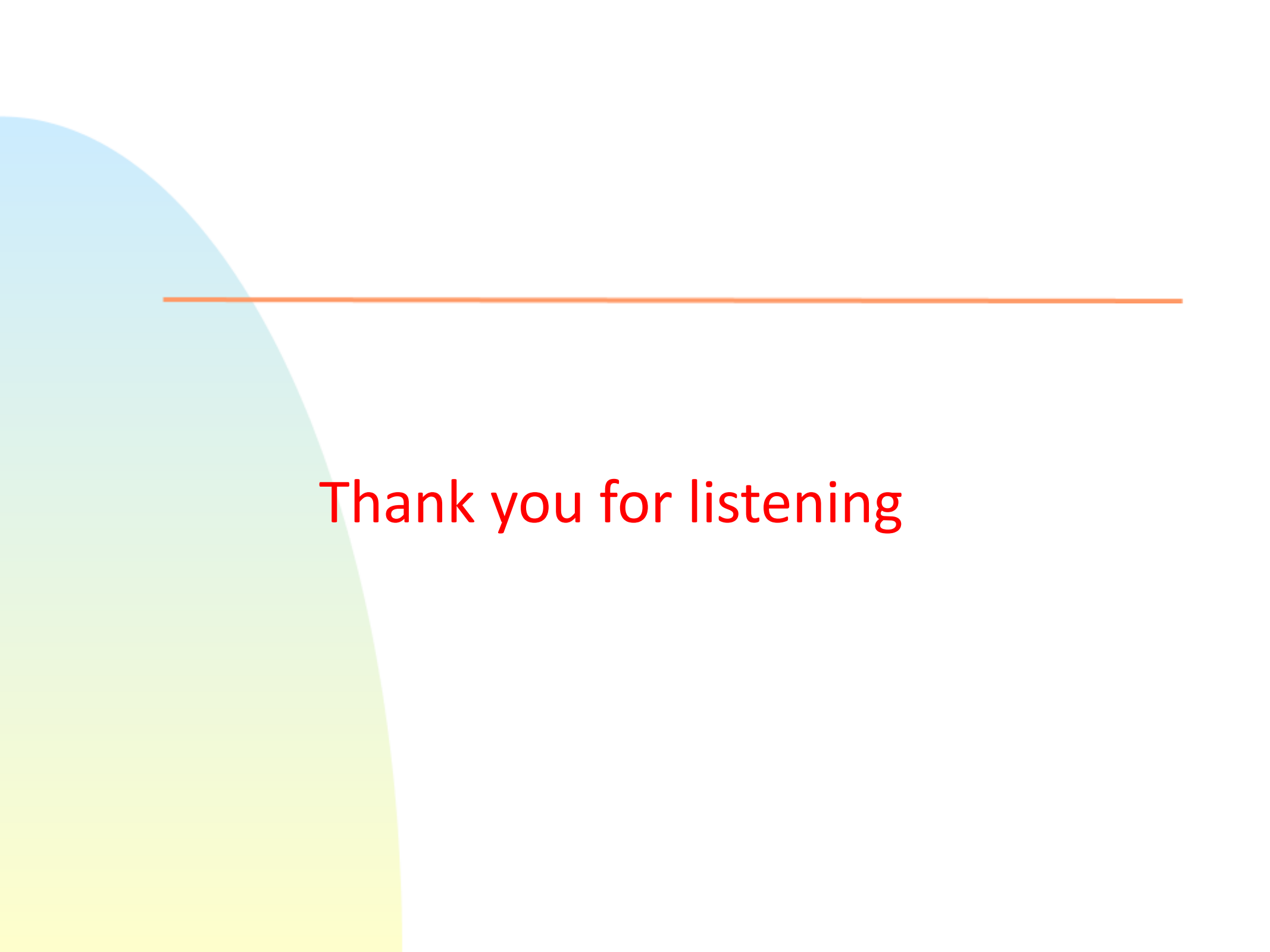


Pure P2P architecture

- no always on server
- arbitrary end systems directly communicate
- peers are intermittently connected and change IP addresses

Highly scalable
But difficult to manage





Thank you for listening