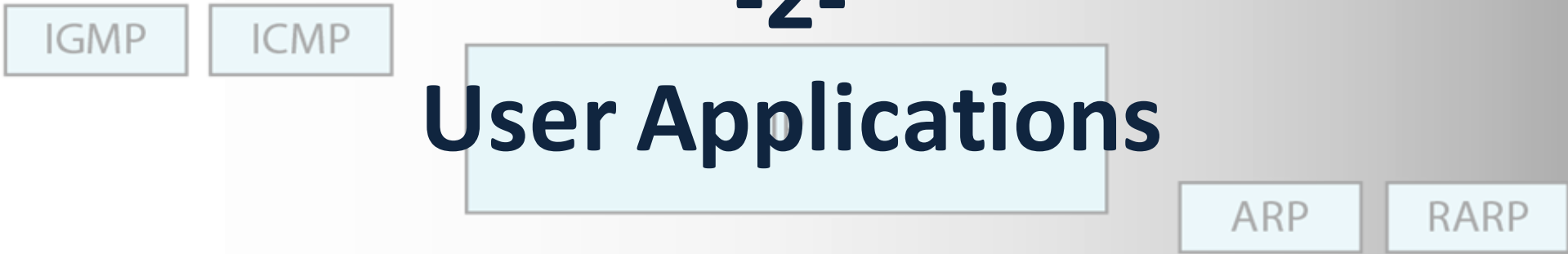


# EITF25 – Internet: Technology and Applications



## Application Layer

-2-



## User Applications

2015, Lecture 10

Kaan Bür

Underlying LAN or WAN  
technology



# Previously on EITF25

## Application Layer (1)

- From domain names to IP addresses: DNS
- Host configuration: DHCP
- Debugging tools: ICMP, ping, traceroot
- World-wide Web: HTTP

# Today: Application Layer

(2)

- File Transfer Protocol, FTP  
*[F26.2]*
- Electronic mail, SMTP, POP, IMAP  
*[S24.1][F26.3]*
- Peer-to-peer networks, P2P  
*[F29.1, F29.5]*
- Voice over IP, VoIP  
*[S25.2-3][F28.4.4-5]*

*\*[Kihl & Andersson: 12.3, 12.5-6, 12.8]*

# World wide web (www)

- 1989-1990: HTTP, HTML by Tim Berners-Lee
- 1991 a first web browser
- 1991 www goes public
  - <http://info.cern.ch>
  - First website
- 1991: The Trojan Room coffee pot
  - <http://www.cl.cam.ac.uk/coffee/coffee.html>
  - First webcam (real time images)
- 1993: Mosaic becomes public



# World wide web (www)

- 1994: Pizza Hut (first online webshop)
- 1994: Yahoo
- 1995: AltaVista
- 1997: AOL instant messenger
- 1997: sixdegrees.com (first modern social net)
- 1997: Google

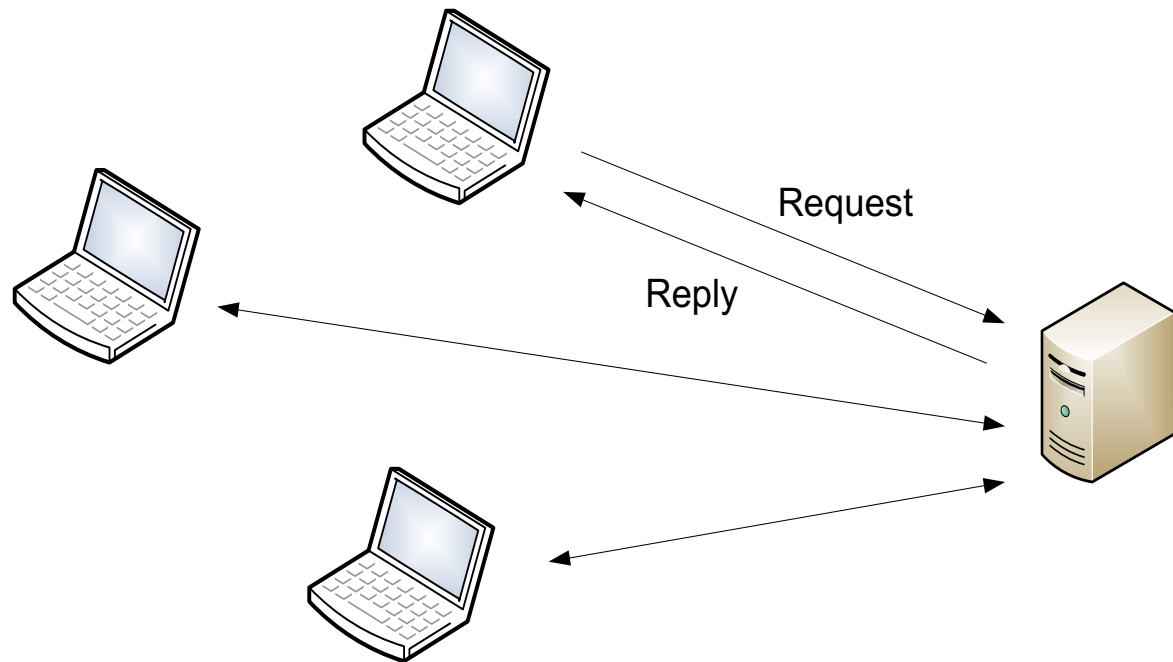
# Then came the digital age...

- 1999: Napster
- 1999: Blogger
- 2001: BitTorrent
- 2001: Wikipedia
- 2003: Skype
- 2003: WordPress
- 2004: Gmail
- 2004: Flickr
- 2005: YouTube
- 2005: Facebook
- 2006: Twitter
- 2008: Spotify
- 2009: Google Docs
- 2009: Angry Birds



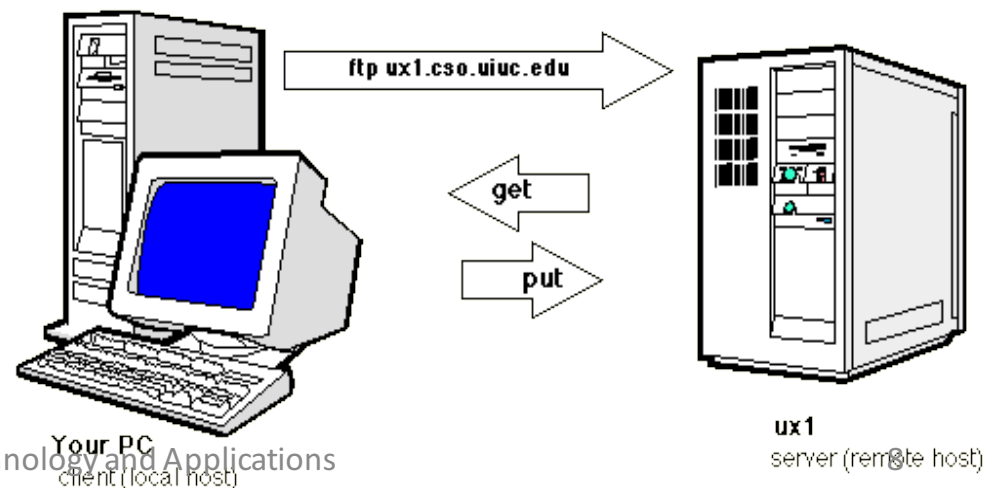
# Client/server paradigm

- Most early applications were based on it
  - http
  - ftp
  - e-mail



# File Transfer Protocol (FTP) - 1971

- File transfer between two computers
- TCP/IP
- Not so straightforward
  - File conventions
  - Data representations
  - Directory structures





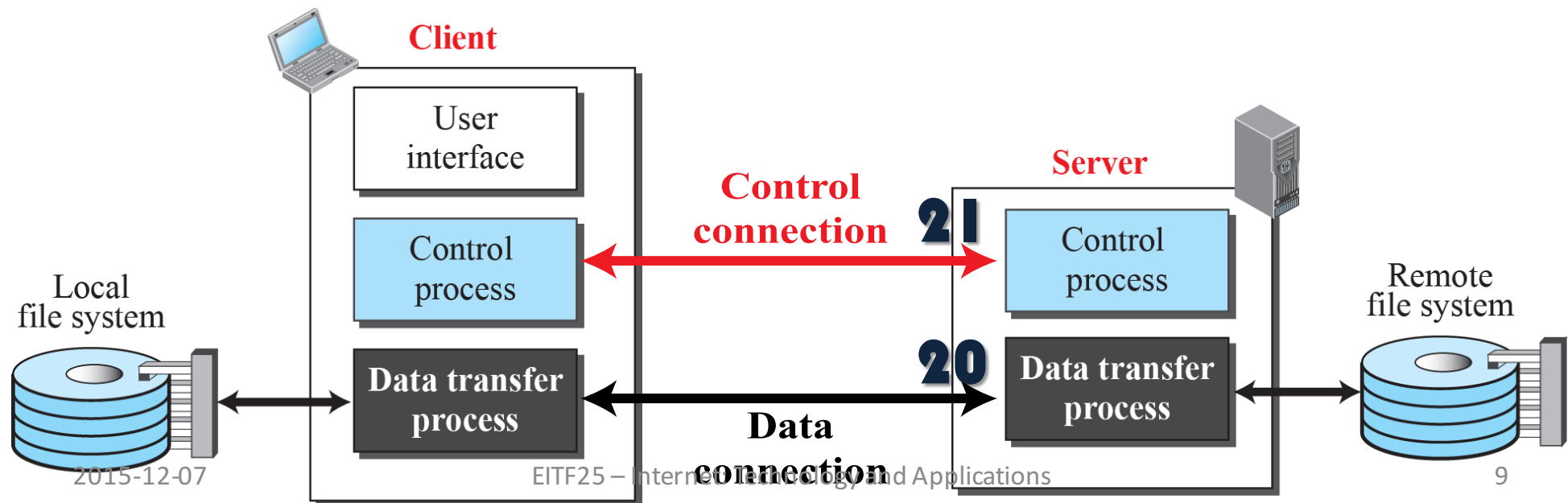
# FTP: basic model

## Control connection

- Open for entire session
- Commands & responses
  - ASCII

## Data connection

- New one for each file



# FTP: Data connection

- Client issues **passive open**
  - Sends **PORT#** to server
- Server issues **active open**
  - server:port20 ↔ client:port #
- All preparations through control connection

# FTP: Control connection

## COMMANDS

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
<b>ABOR</b>		Abort the previous command
<b>CDUP</b>		Change to parent directory
<b>CWD</b>	Directory name	Change to another directory
<b>DELE</b>	File name	Delete a file
<b>LIST</b>	Directory name	List subdirectories or files
<b>MKD</b>	Directory name	Create a new directory
<b>PASS</b>	User password	Password
<b>PASV</b>		Server chooses a port
<b>PORT</b>	port identifier	Client chooses a port
<b>PWD</b>		Display name of current directory
<b>QUIT</b>		Log out of the system
<b>RETR</b>	File name(s)	Retrieve files; files are transferred from server to client






## RESPONSES

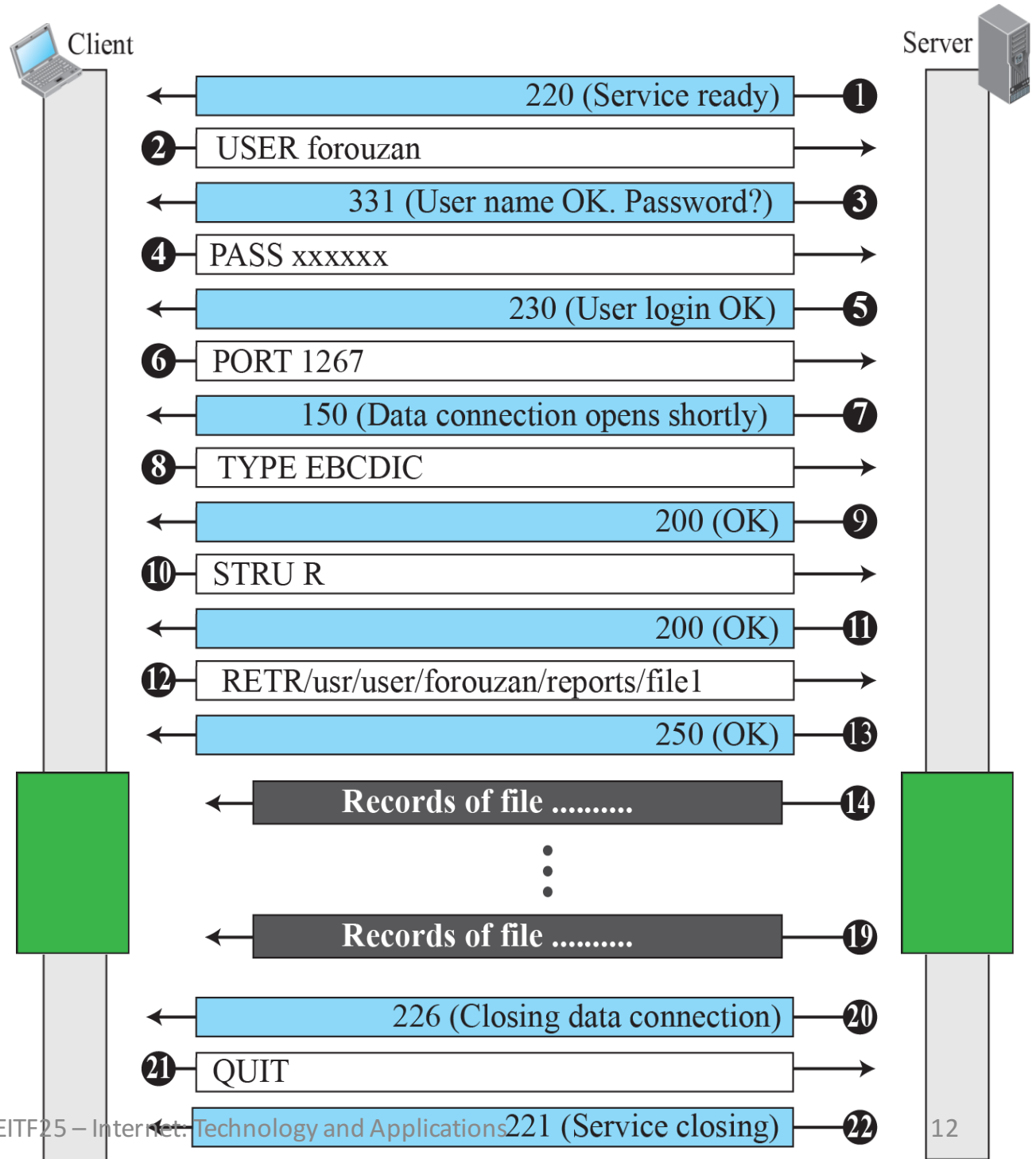
<i>Code</i>	<i>Description</i>	<i>Code</i>	<i>Description</i>
<b>125</b>	Data connection open	<b>250</b>	Request file action OK
<b>150</b>	File status OK	<b>331</b>	User name OK; password is needed
<b>200</b>	Command OK	<b>425</b>	Cannot open data connection
<b>220</b>	Service ready	<b>450</b>	File action not taken; file not available
<b>221</b>	Service closing	<b>452</b>	Action aborted; insufficient storage
<b>225</b>	Data connection open	<b>500</b>	Syntax error; unrecognized command
<b>226</b>	Closing data connection	<b>501</b>	Syntax error in parameters or arguments
<b>230</b>	User login OK	<b>530</b>	User not logged in

# FTP

- Session

### Legend

	Control process (port 21)
	Data transfer process (port 20)
	Command
	Response
	Data transfer

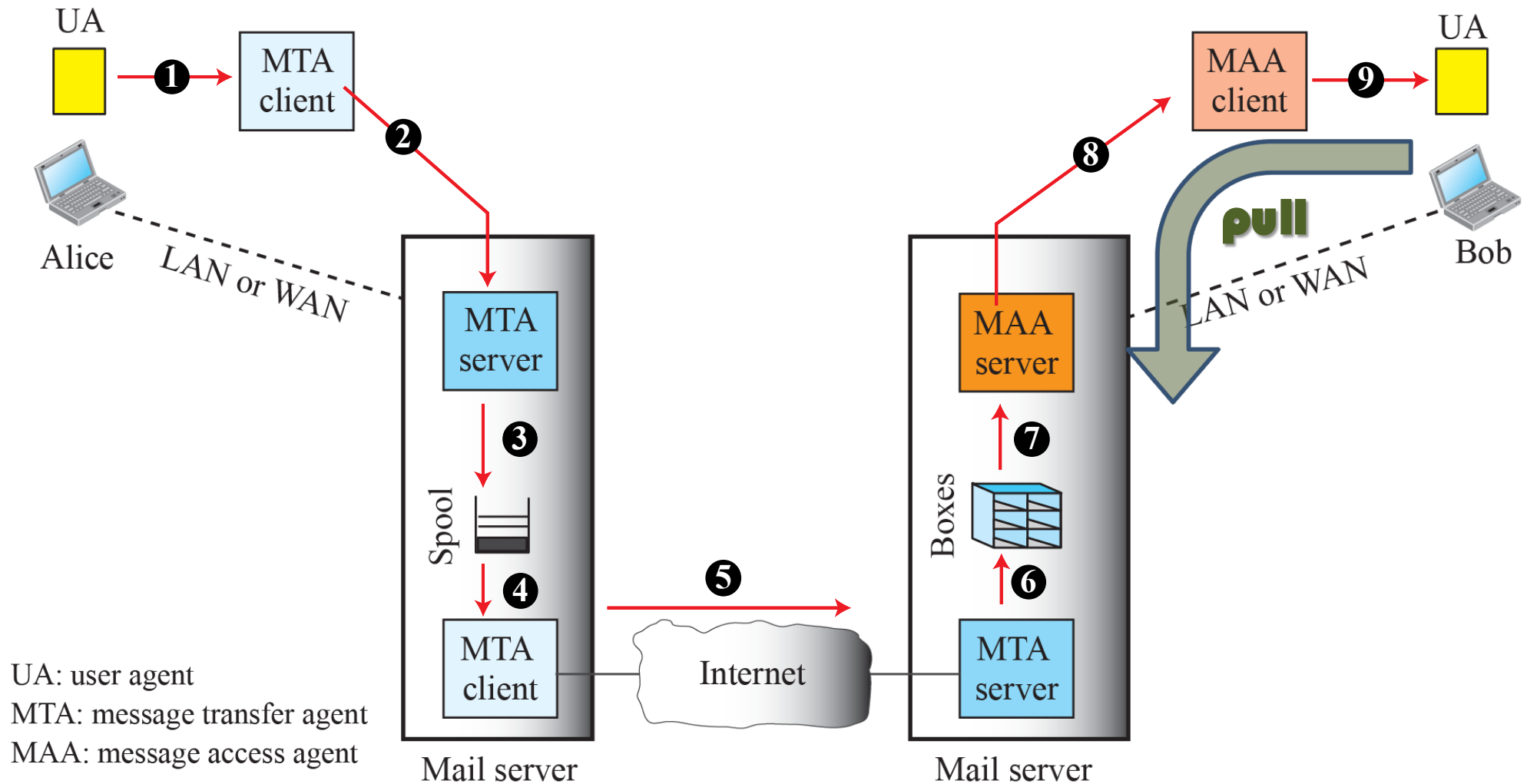


# Electronic mail (e-mail) - 1971

- The first e-mail was sent between two computers in the same room.
- To separate the user from the host computer, the @ sign was inserted. It was unused on the keyboard.

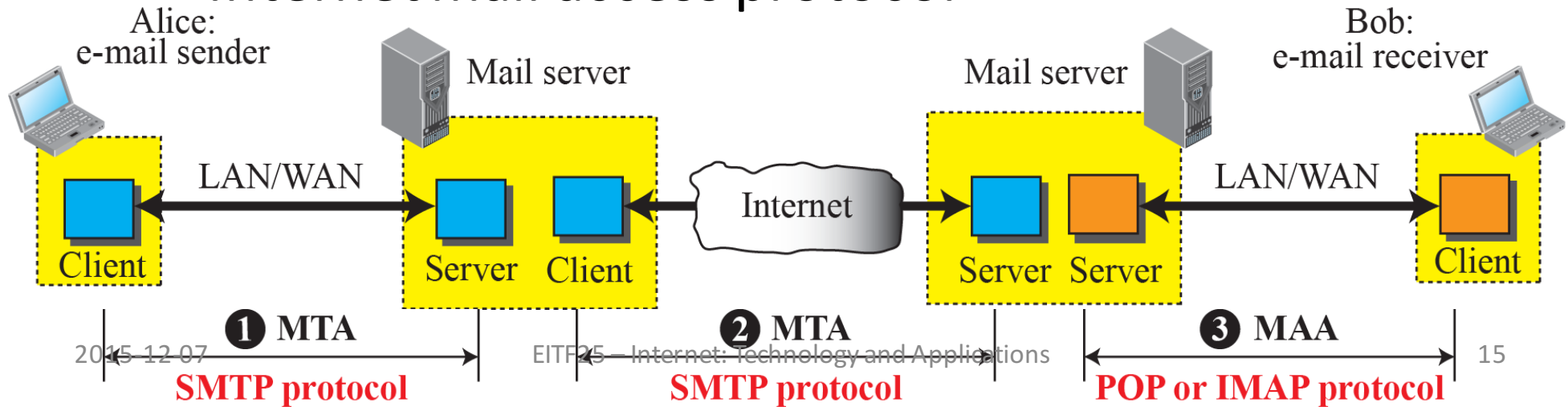
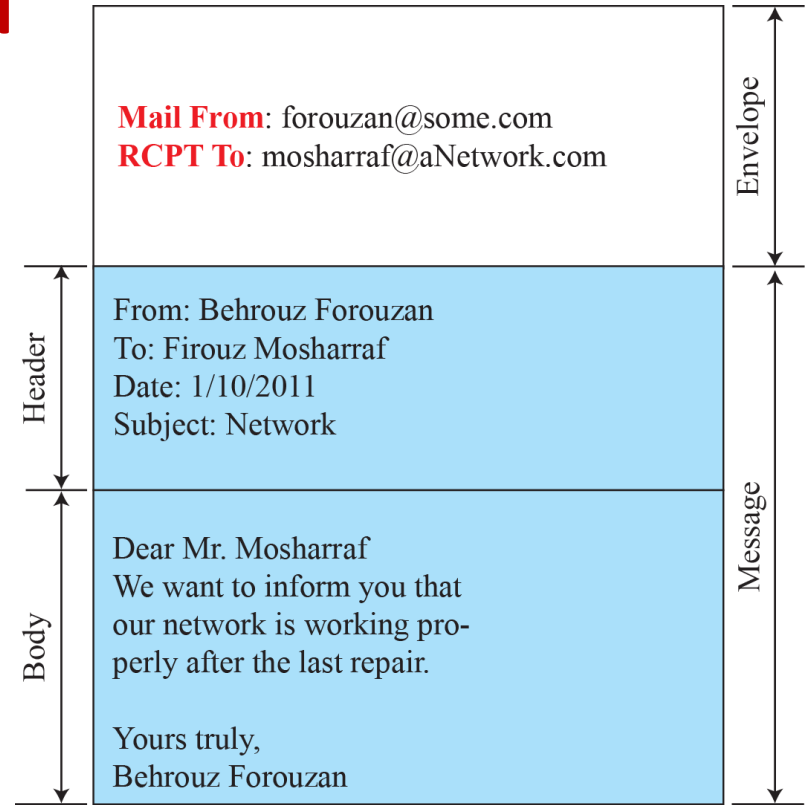


# E-mail: basic model



# E-mail: protocols used

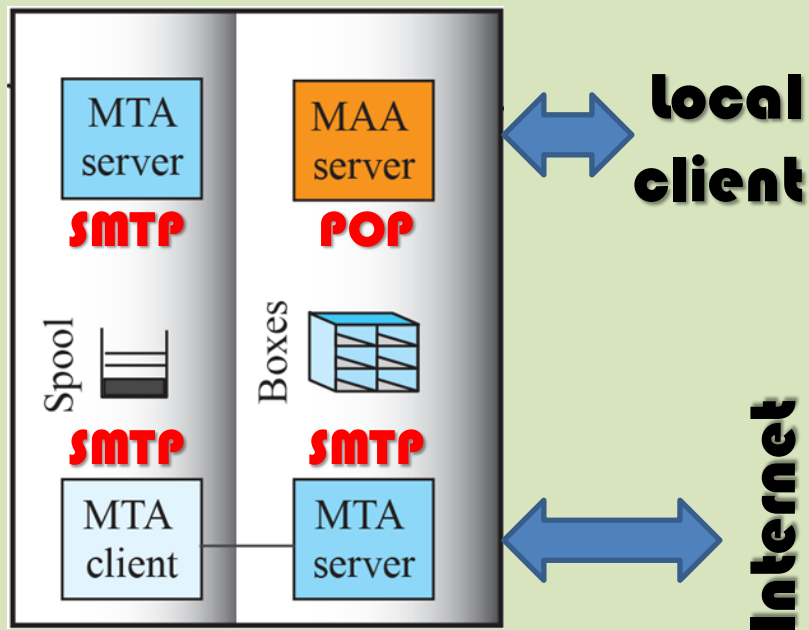
- SMTP
  - Simple mail transfer protocol
- POP
  - Post office protocol
- IMAP
  - Internet mail access protocol



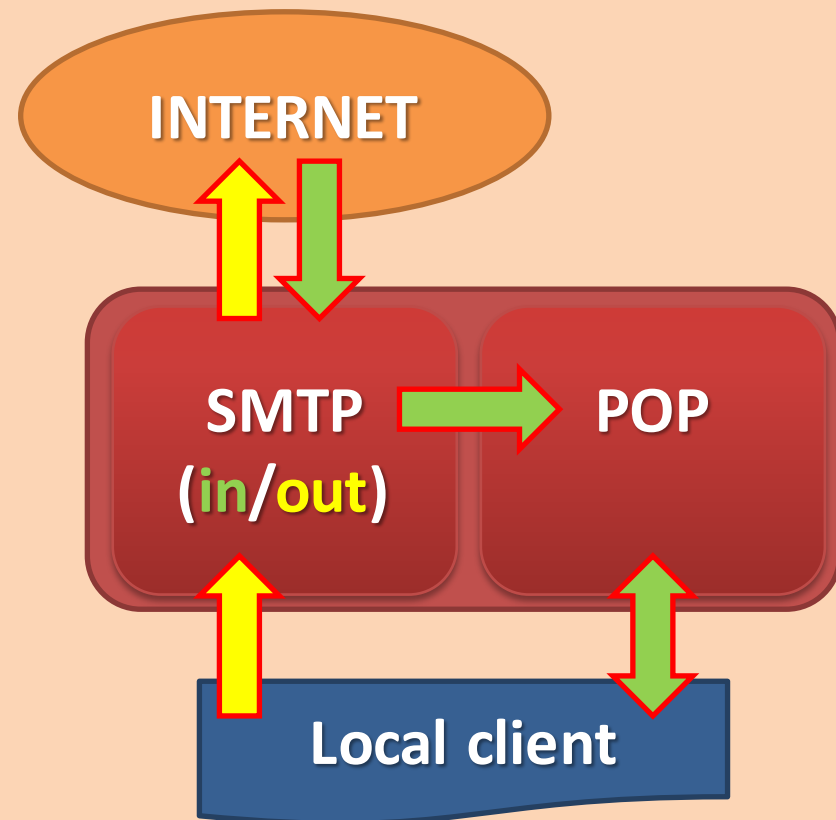
# E-mail: server architecture

## Sender & receiver in one

- Not a very good representation!

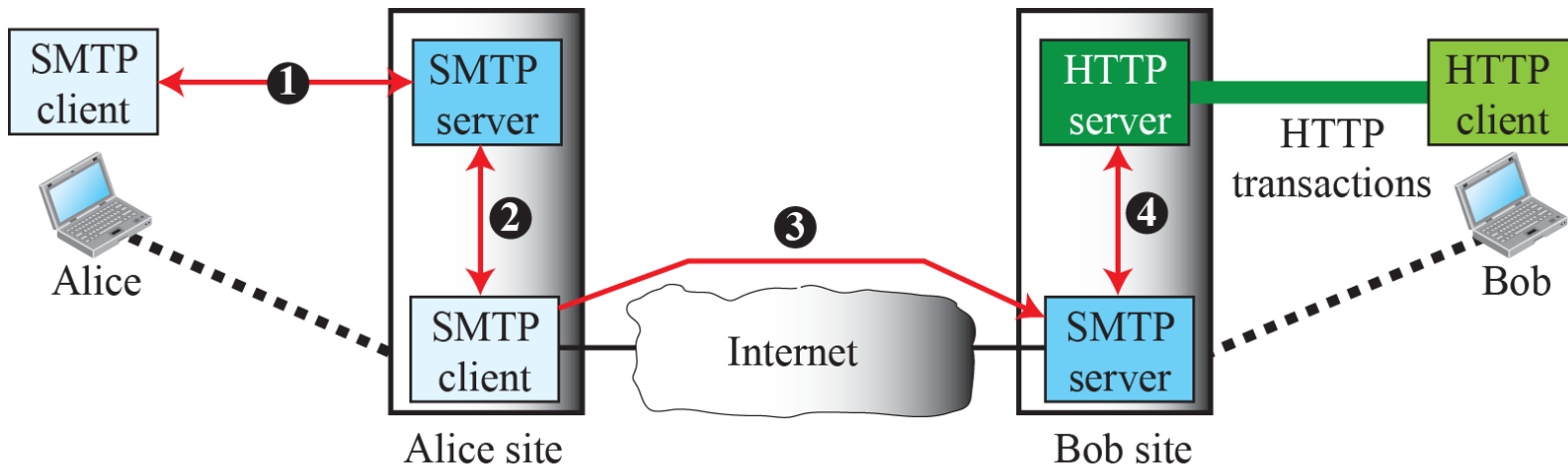


## A holistic view

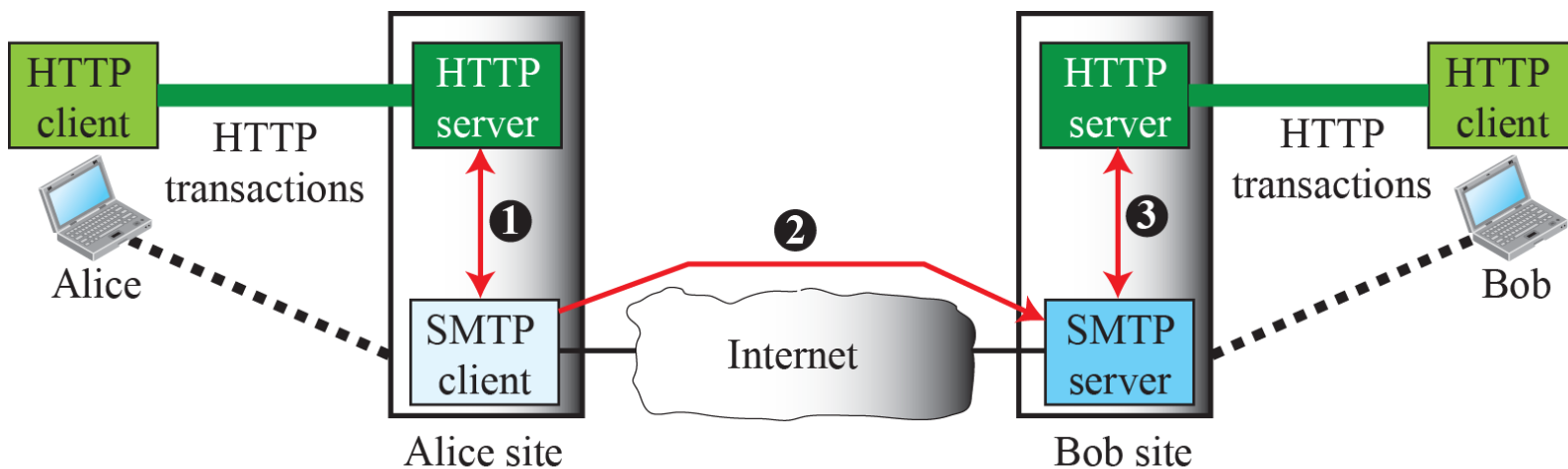




# E-mail: Web-based services



Case 1: Only receiver uses HTTP



Case 2: Both sender and receiver use HTTP

# See you in 15' :)



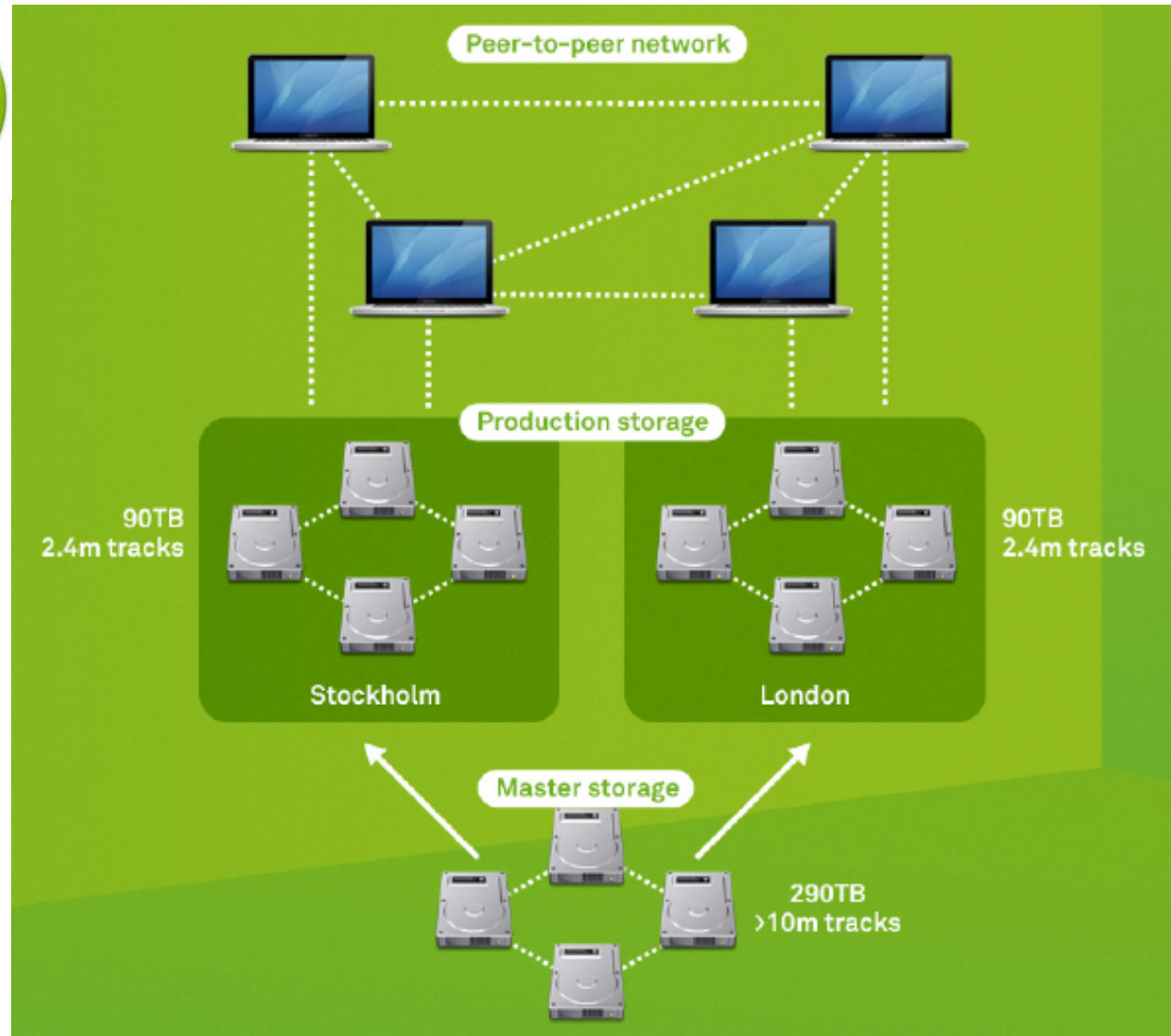
- After the break
  - P2P, BitTorrent
  - VoIP, Skype

# Performance challenges

- Client/server architectures
  - Standardized protocols like HTTP
  - Heavy traffic load on network infrastructure
  - Unicast transmission
  - Delays due to overloaded access networks
  - Single point of failure

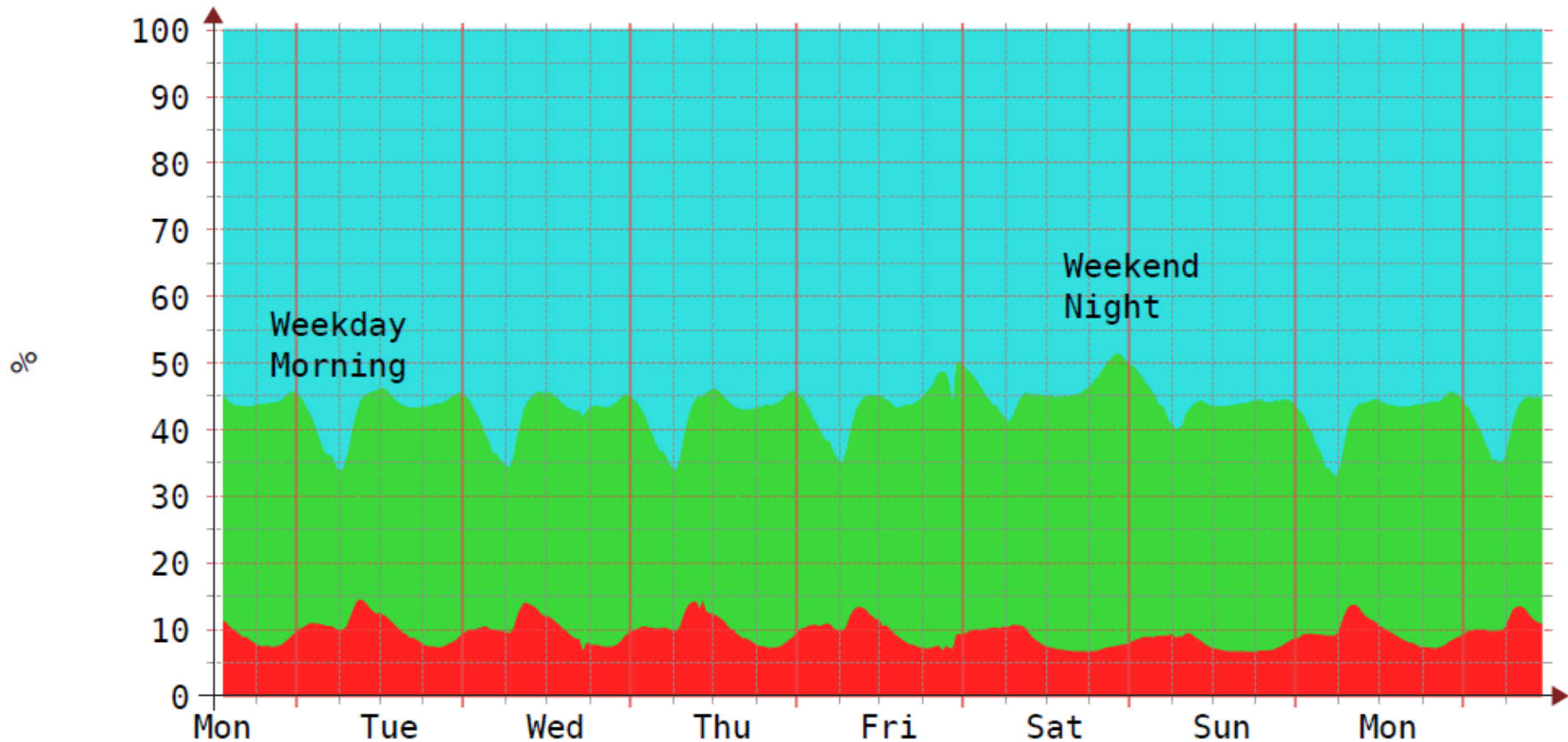
# One solution: P2P (and caches)

- Spotify



# Result: better load distribution

Data source - ratio - by week



KKDIUOL / IUR1 UE1KER

- Server
- P2P
- Cache

Cur:	Min:	Avg:
10.86	6.76	9.62
33.90	23.78	33.86
55.24	48.47	56.53

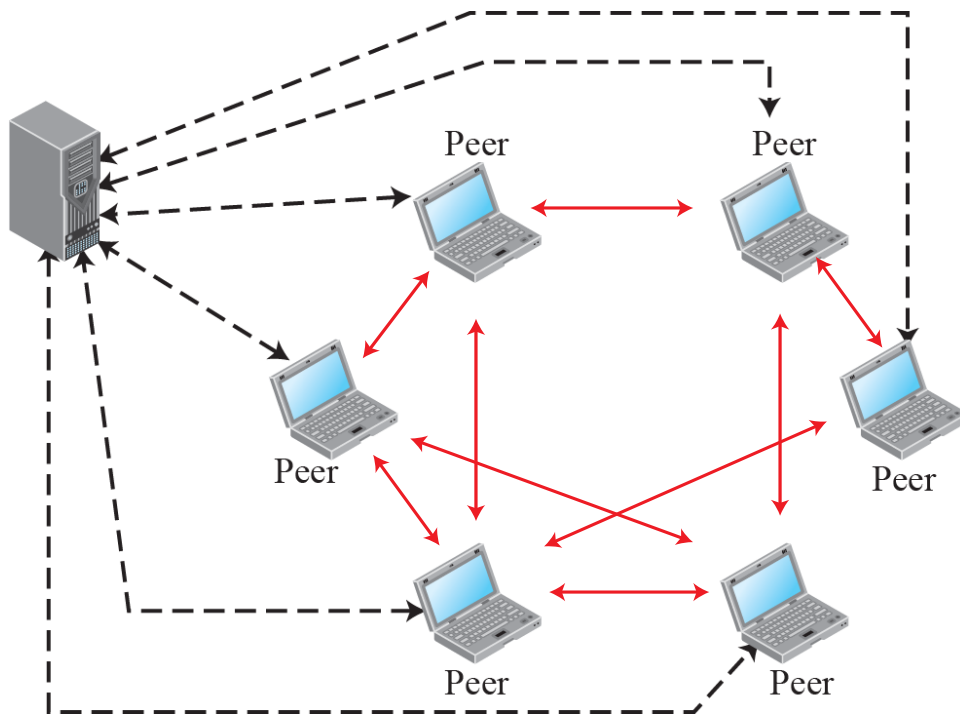
# Peer-to-peer (P2P) paradigm

- Users sharing data form network of peers
- First P2P file sharing 1987
  - WWIVnet bulletin board by Wayne Bell
- Gained popularity 1999
  - Napster by Shawn Fanning

# P2P networks

## Centralised

- Directory server



## Decentralised

- Overlay network
  - Logical on top of physical
- A) unstructured
  - Nodes linked randomly
  - Queries flood network
- B) structured
  - Nodes linked with rules (DHT)
  - More efficient query resolving
- Initial list of nodes provided



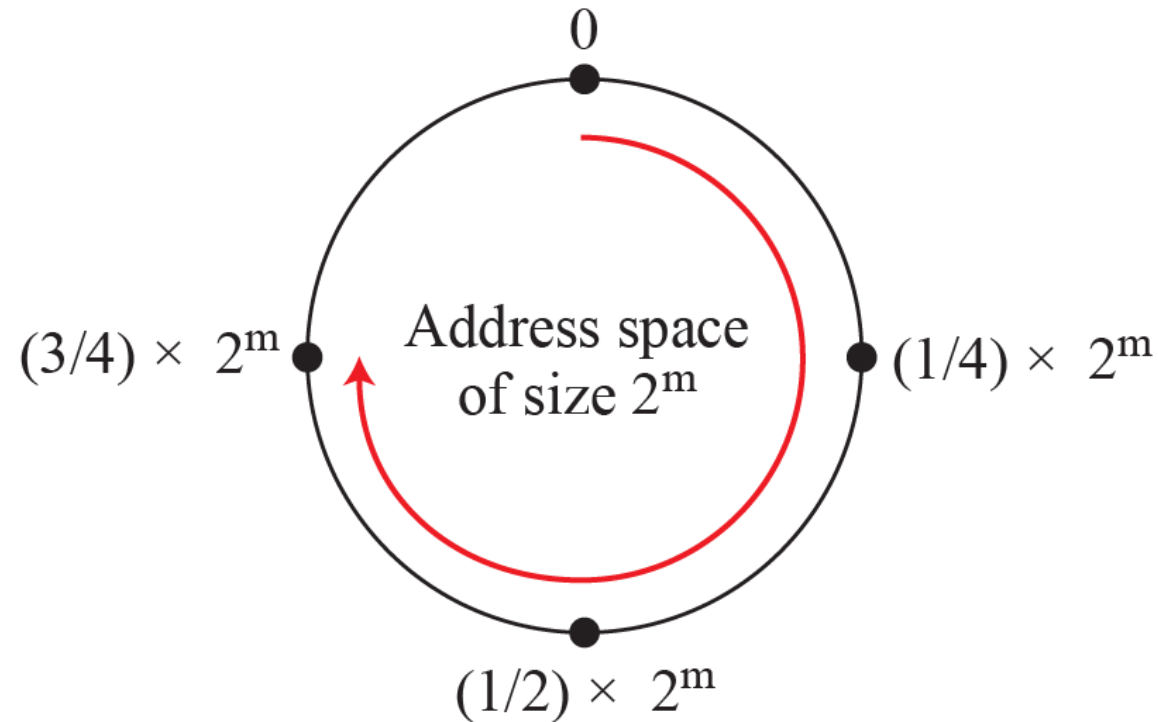
# Distributed Hash Tables (DHT)

- A hash function is an algorithm that maps data of variable length to data of fixed length. (*m bits*)
  - Node ID = hash (peer IP address)
  - Key = hash (file name)
- DHT distributes data among a set of nodes.
  - Each peer is responsible for a portion of data.
- DHT routes a query to responsible node
  - Peers have partial knowledge about whole net.



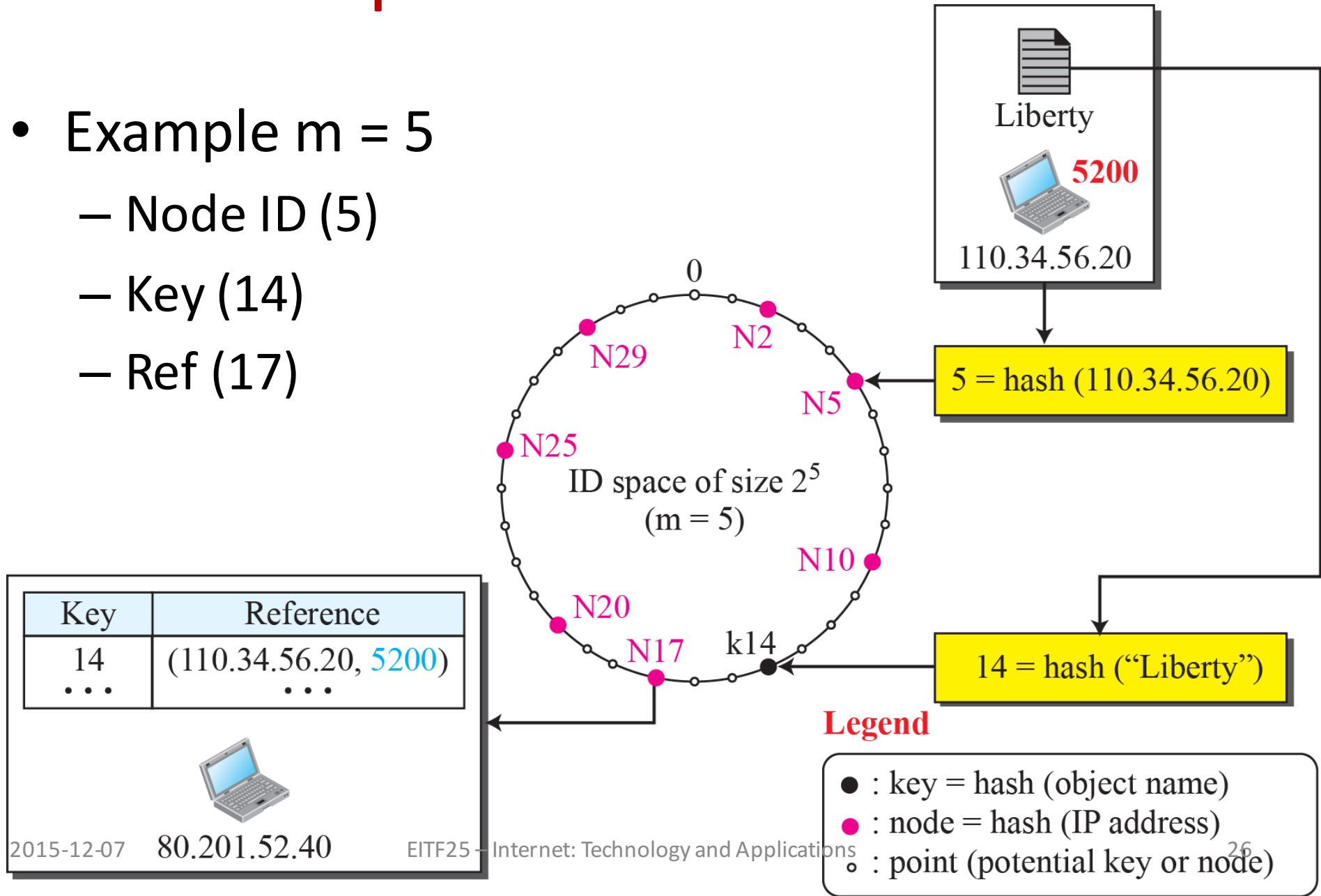
# DHT address space

- Node ID = hash (peer IP address)
- Key = hash (file name)



# DHT example

- Example  $m = 5$ 
  - Node ID (5)
  - Key (14)
  - Ref (17)

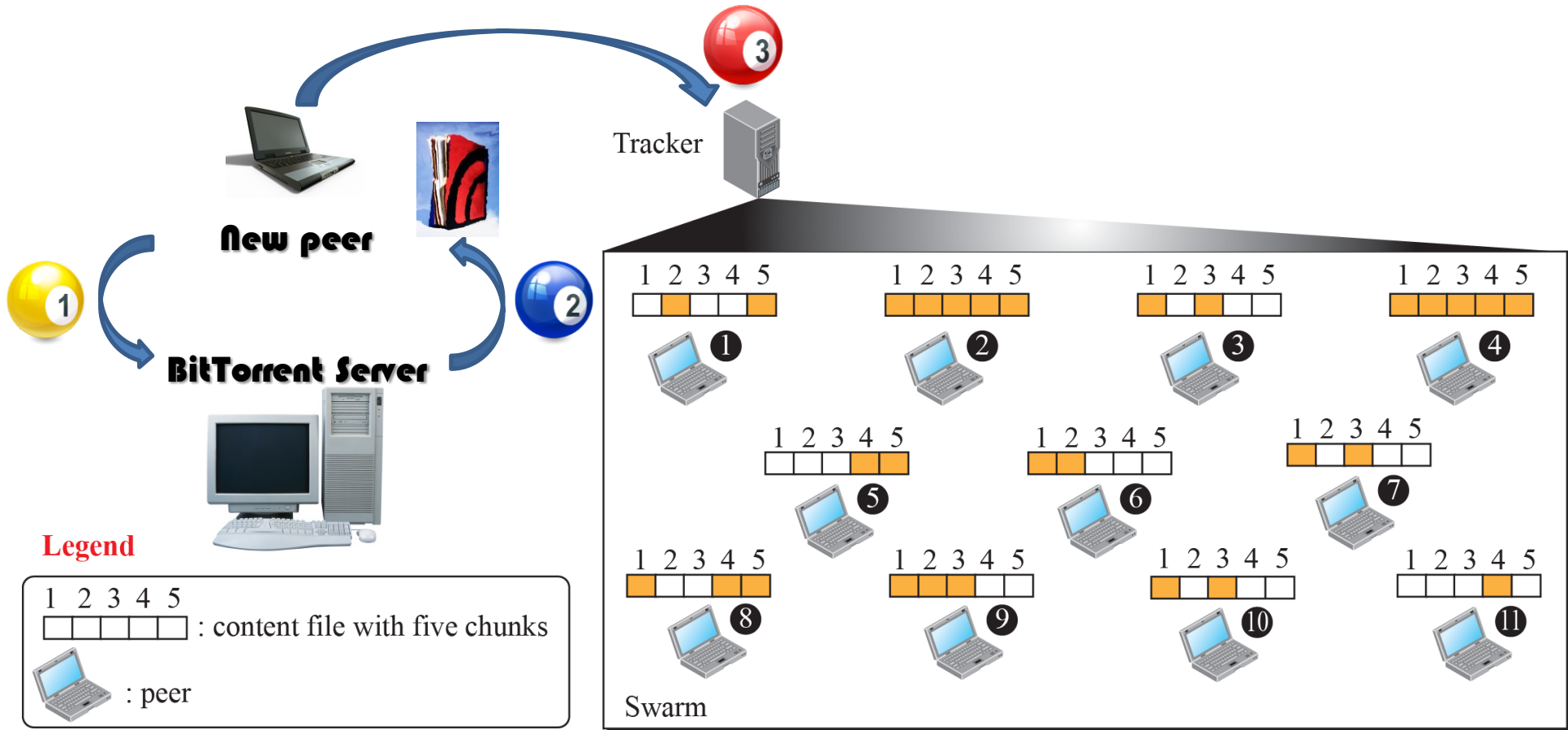


# P2P example: BitTorrent

- Group of peers work together to give all peers a copy of shared file.
  - Torrent
  - Swarm
  - Seed
  - Leech
  - Tracker
- No downloading whole file from one peer

# Tracker

- Provides list of peers for given torrent



**Note:** Peers 2 and 4 are seeds; others are leeches.

# Voice over IP (VoIP)

- Internet telephony
  - Uses two protocols

## *SIP*

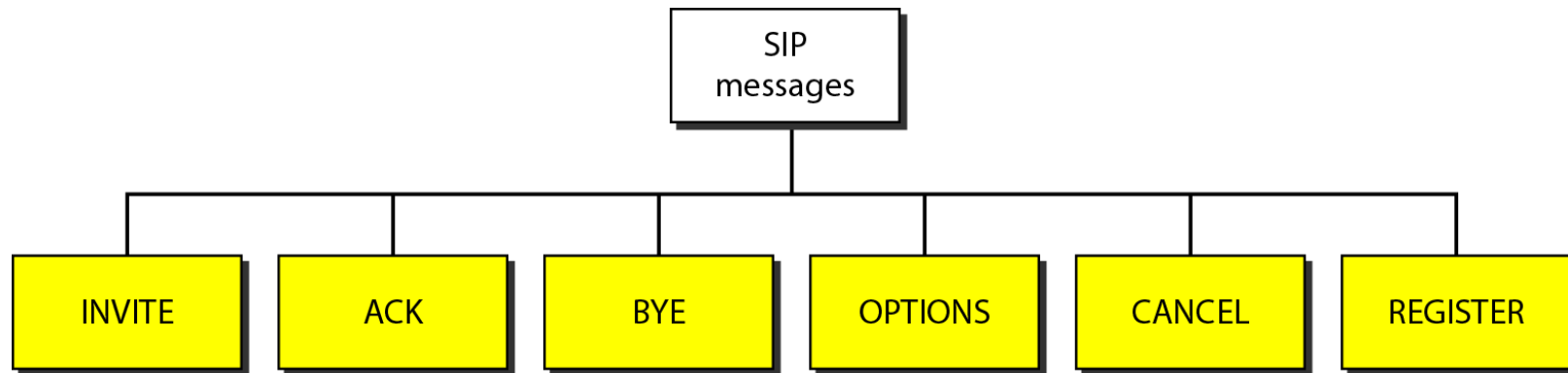
- Session Initiation Protocol
- IETF standard

## *H.323*

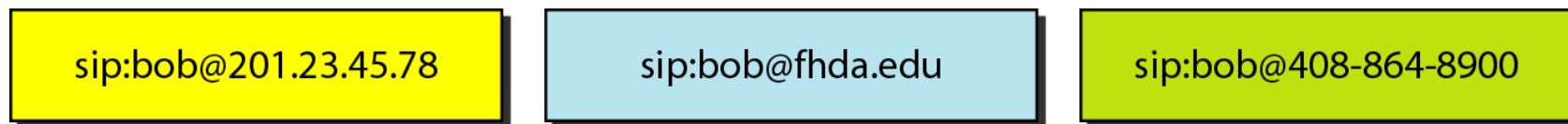
- Communication (telephone, computer)
- ITU-T standard

# Session Initiation Protocol (SIP)

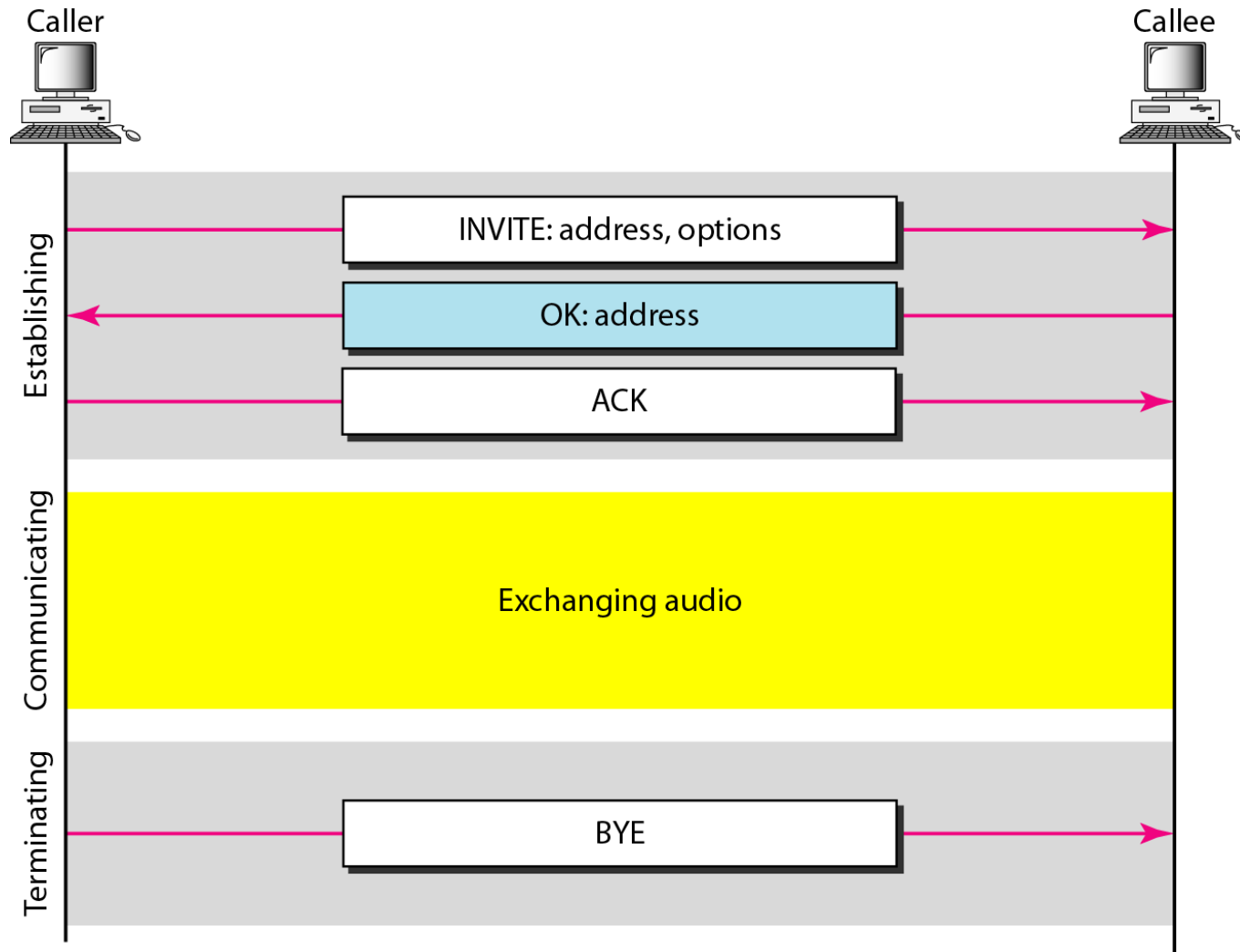
- Application layer protocol
- Multimedia session management
- Text-based messages



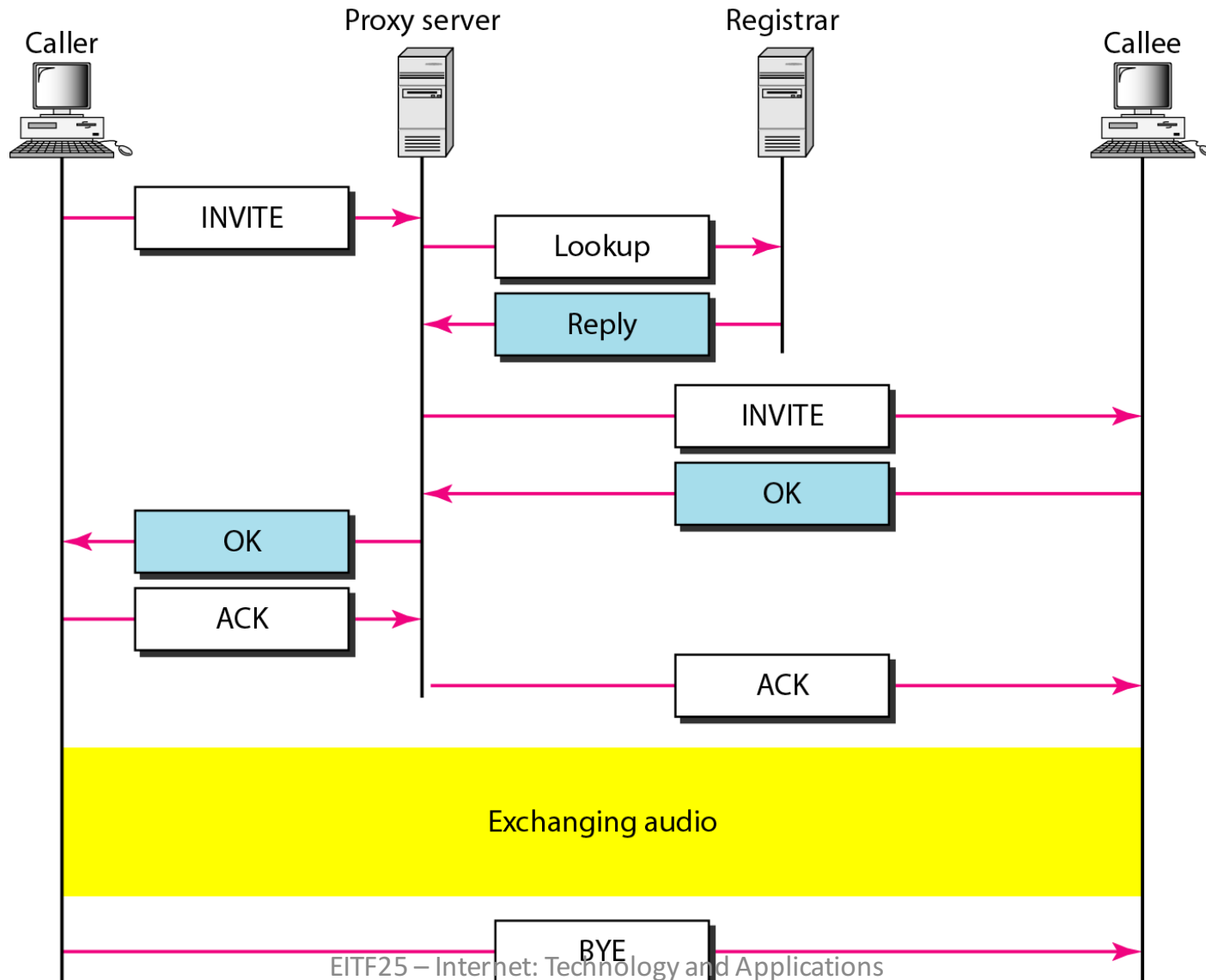
- Various address types



# A simple SIP session



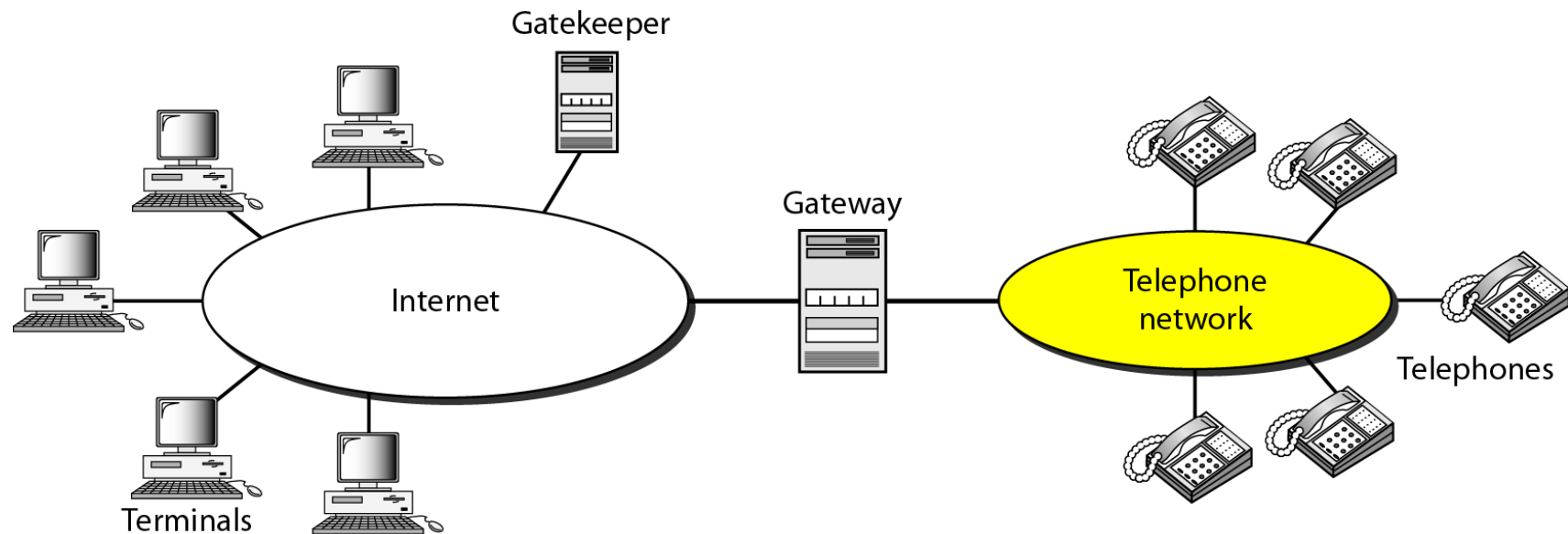
# Tracking the callee



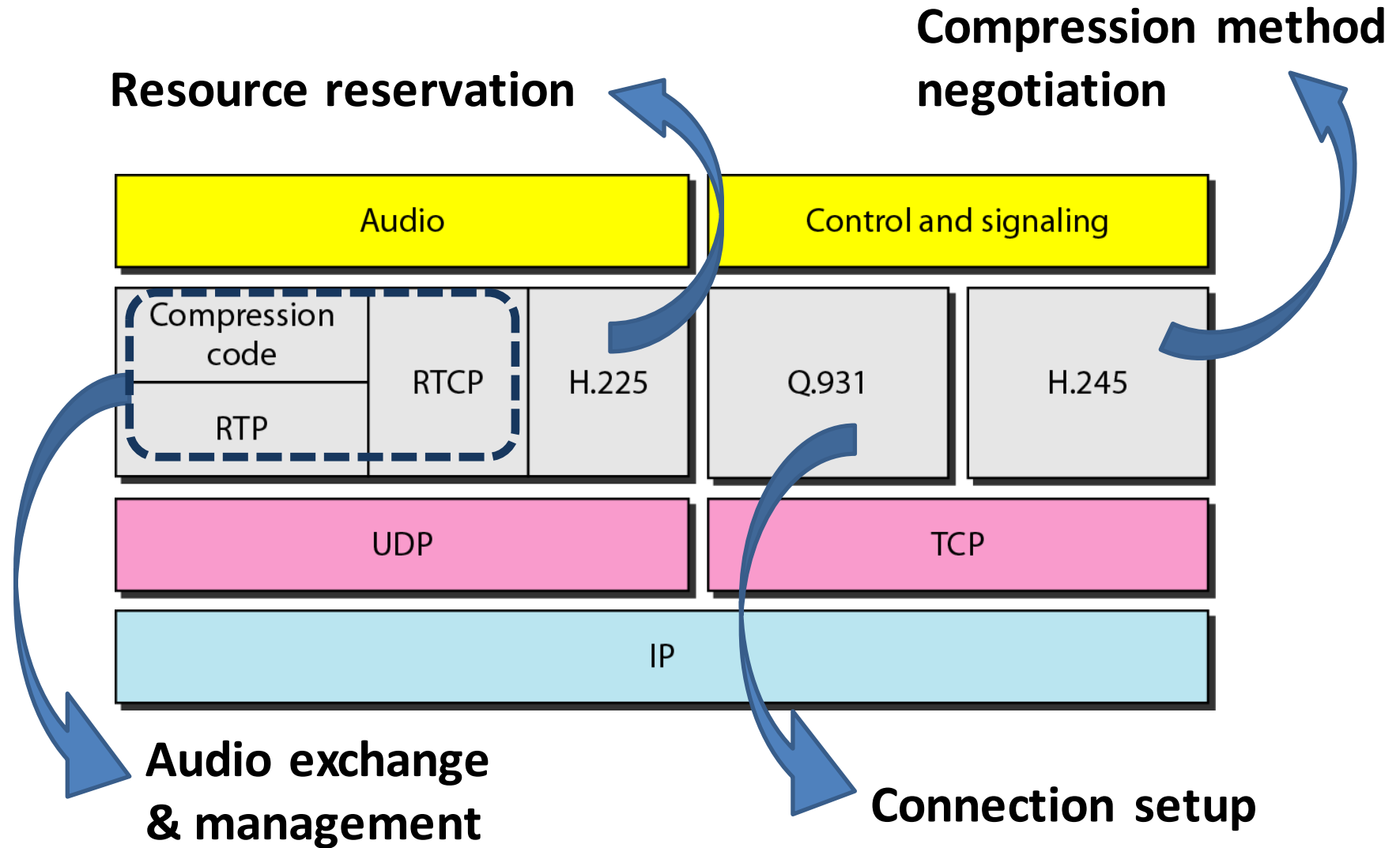


# H.323

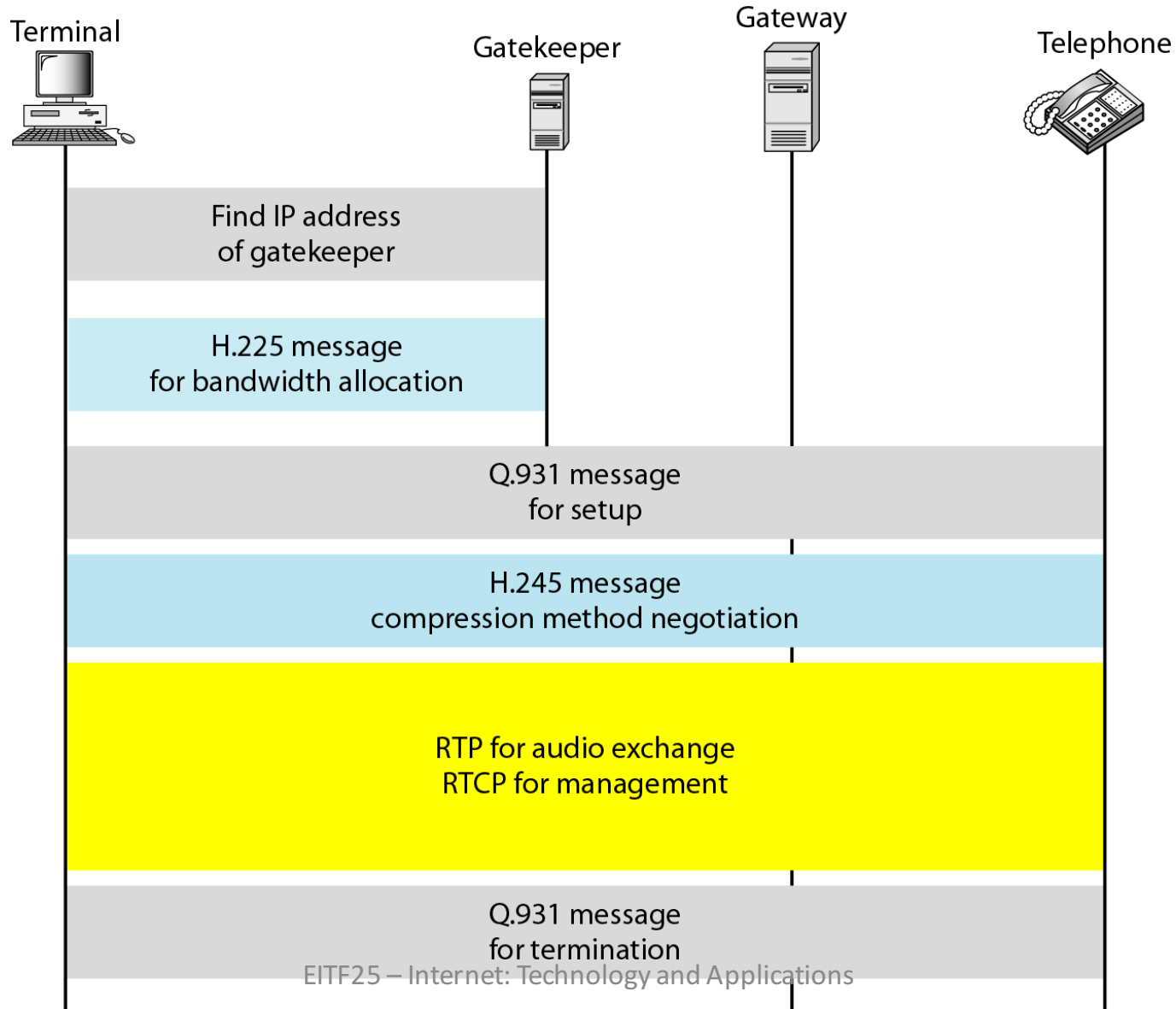
- Communication (telephone, computer)
  - Gateway = 5-layer translation device
  - Gatekeeper = registrar



# H.323 protocols

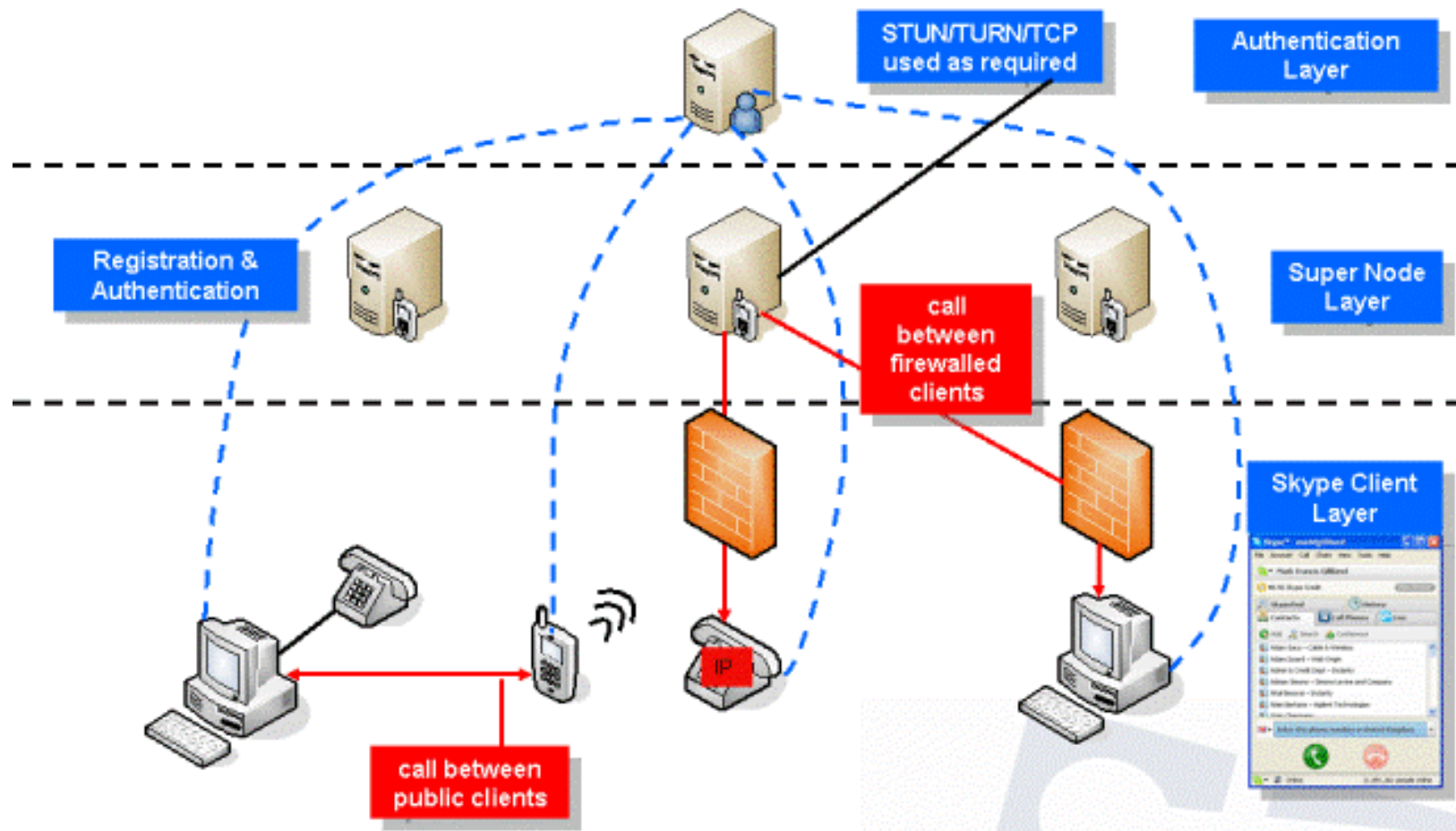


# H.323 session



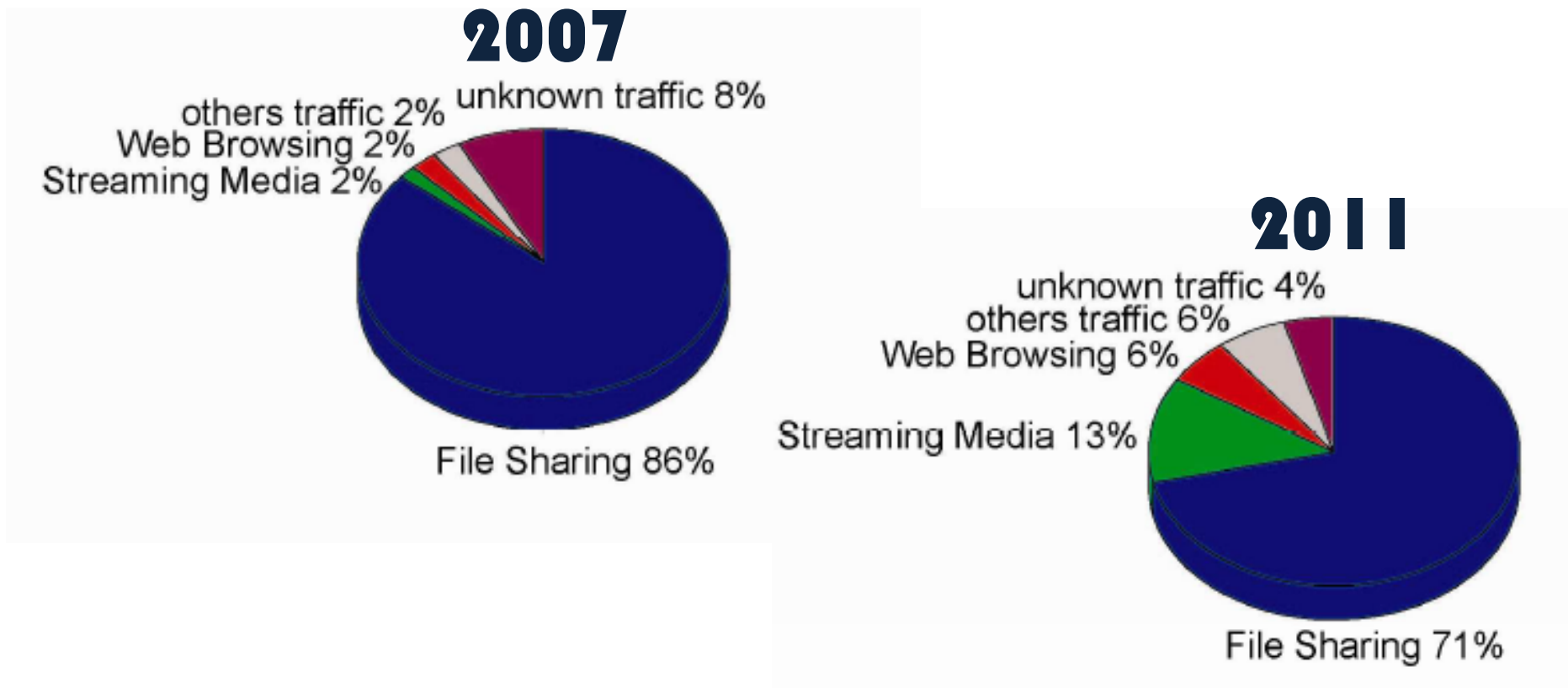
# Counter-example: Skype

## Skype Architecture



# Evolution of Internet usage

- Traffic volumes generated by users



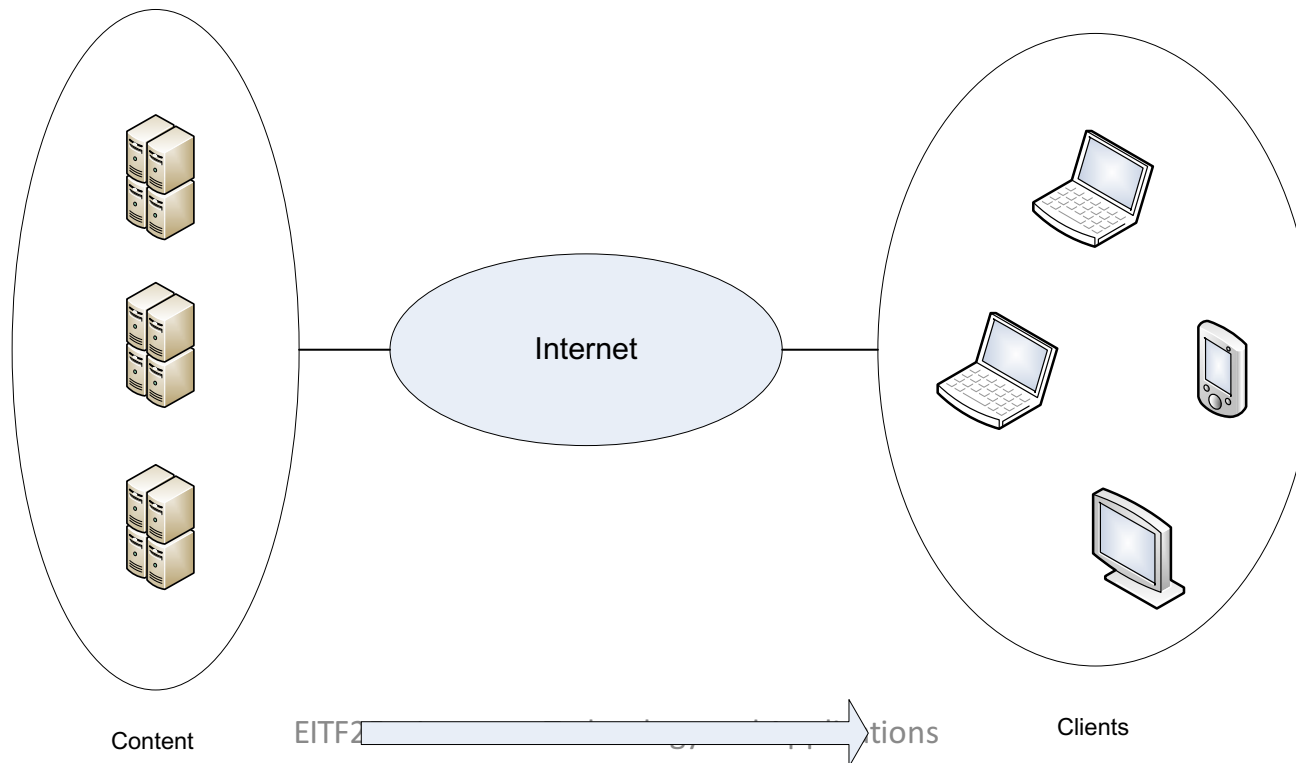
J. Li, A. Aurelius, V. Nordell, M. Du, Å. Arvidsson, M. Kihl:

**A five year perspective of traffic pattern evolution in a residential broadband access network**

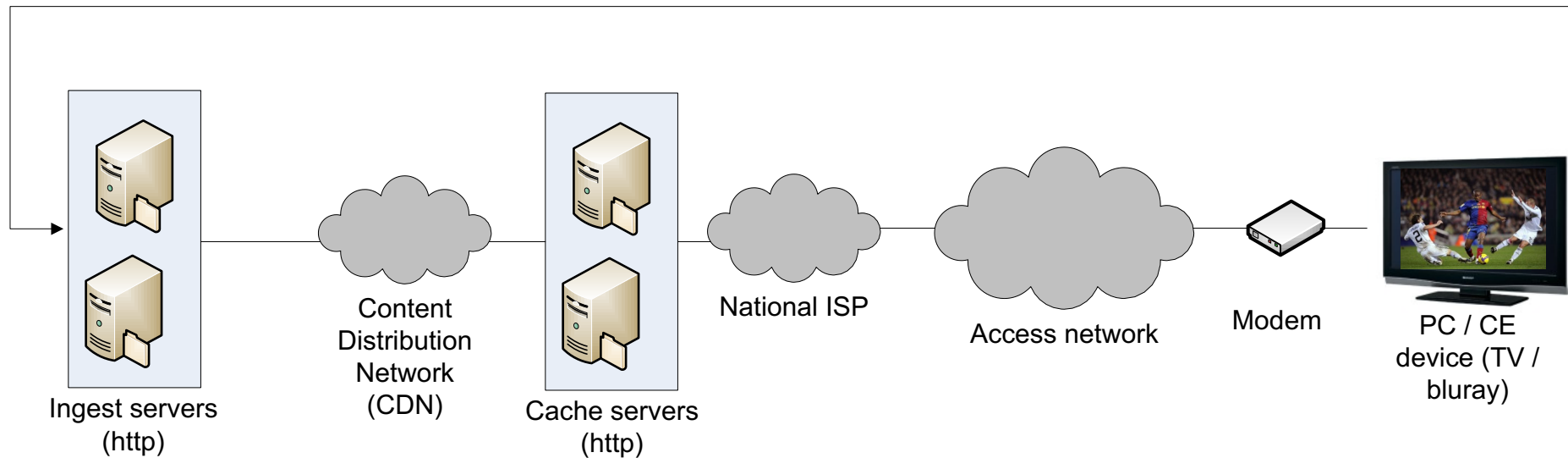
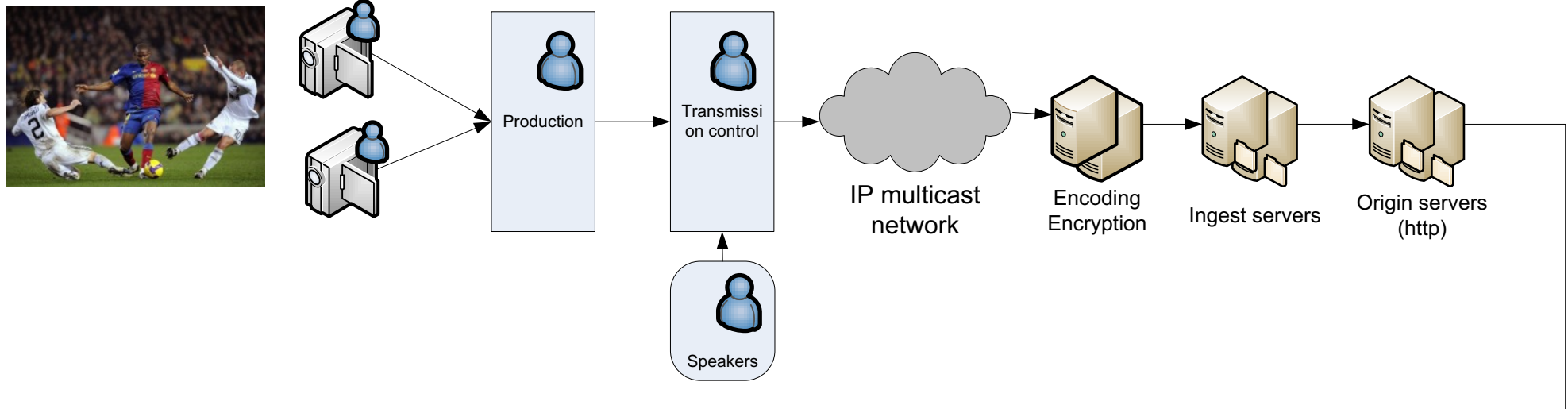
2015-12-07  
2015-12-07  
ETP-25 - Internet: Technology and Applications  
Future Network & Mobile Summit 2012

# Content distribution architectures

- Multimedia
  - Content stored in back-end server clusters (cloud)
  - Distributed to clients upon requested



# Example: Live sport channel at TV4



# Summary: Application Layer (2)

- A little history behind WWW
- Client/server applications
  - ftp, file transfer
  - e-mail, message exchange
- P2P applications
  - Bittorrent, file sharing
  - Skype, Internet telephony
- Streaming multimedia applications

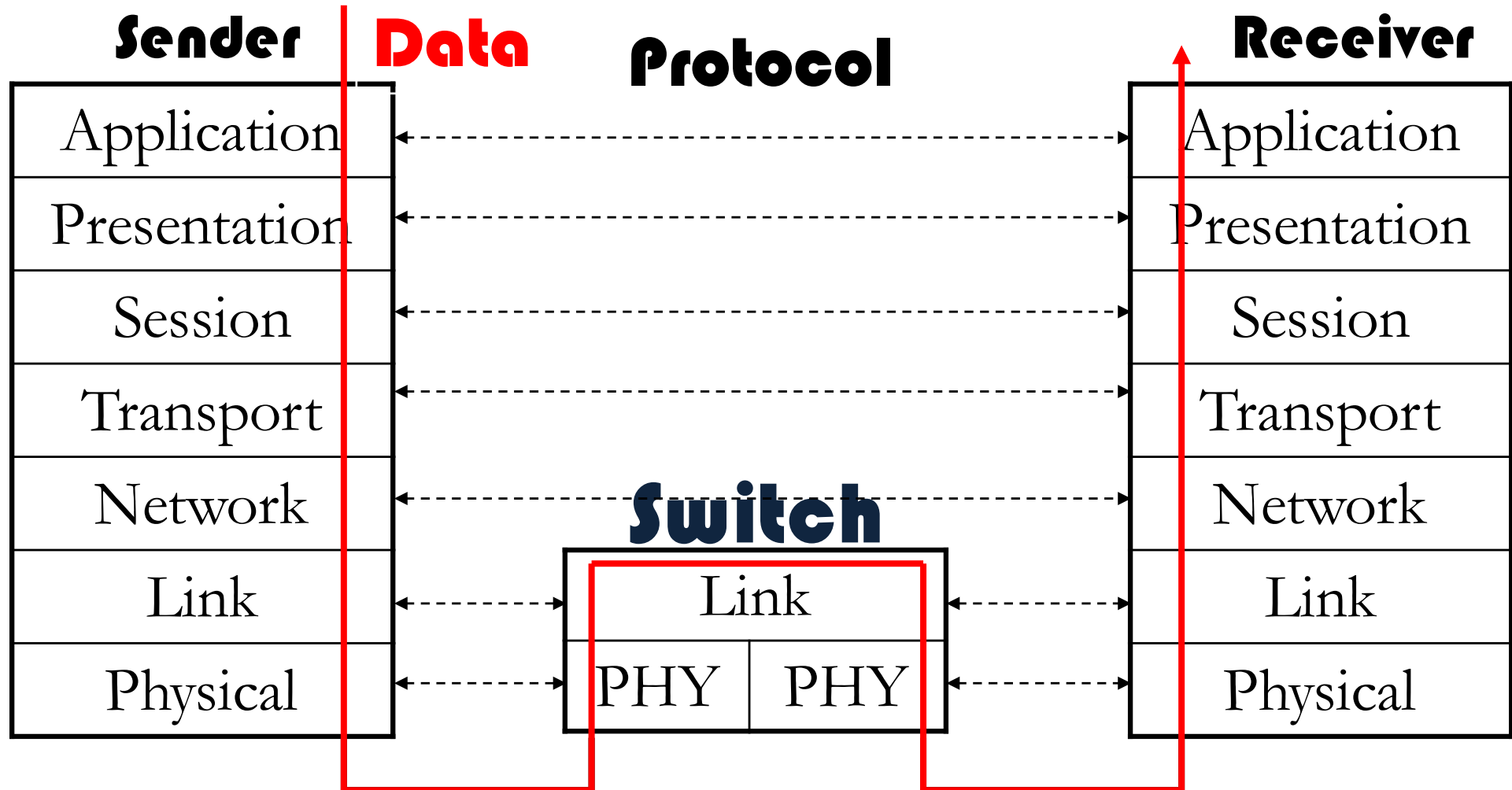


# Appendix

## Connecting Devices

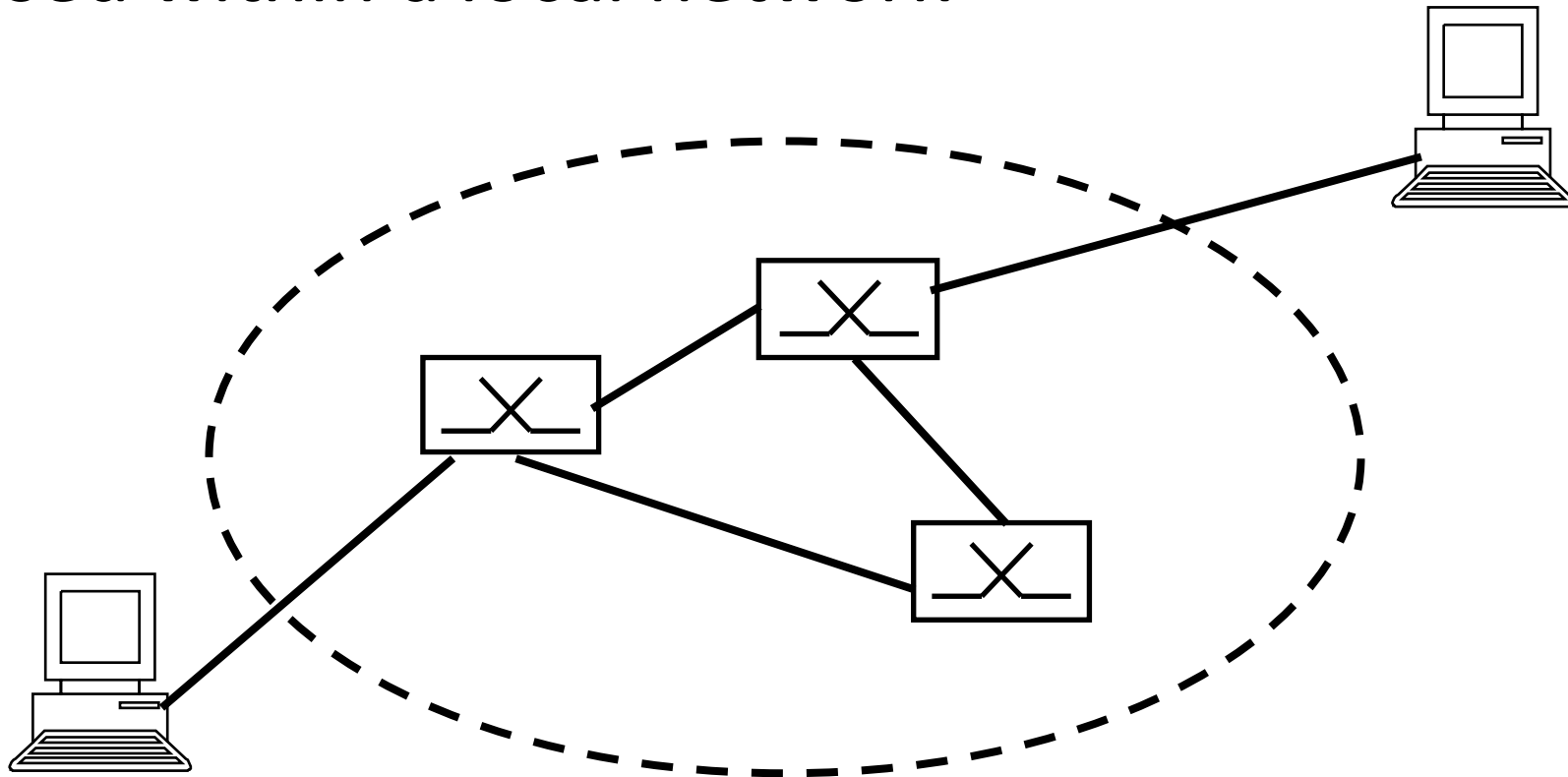
# Connecting devices

(1)



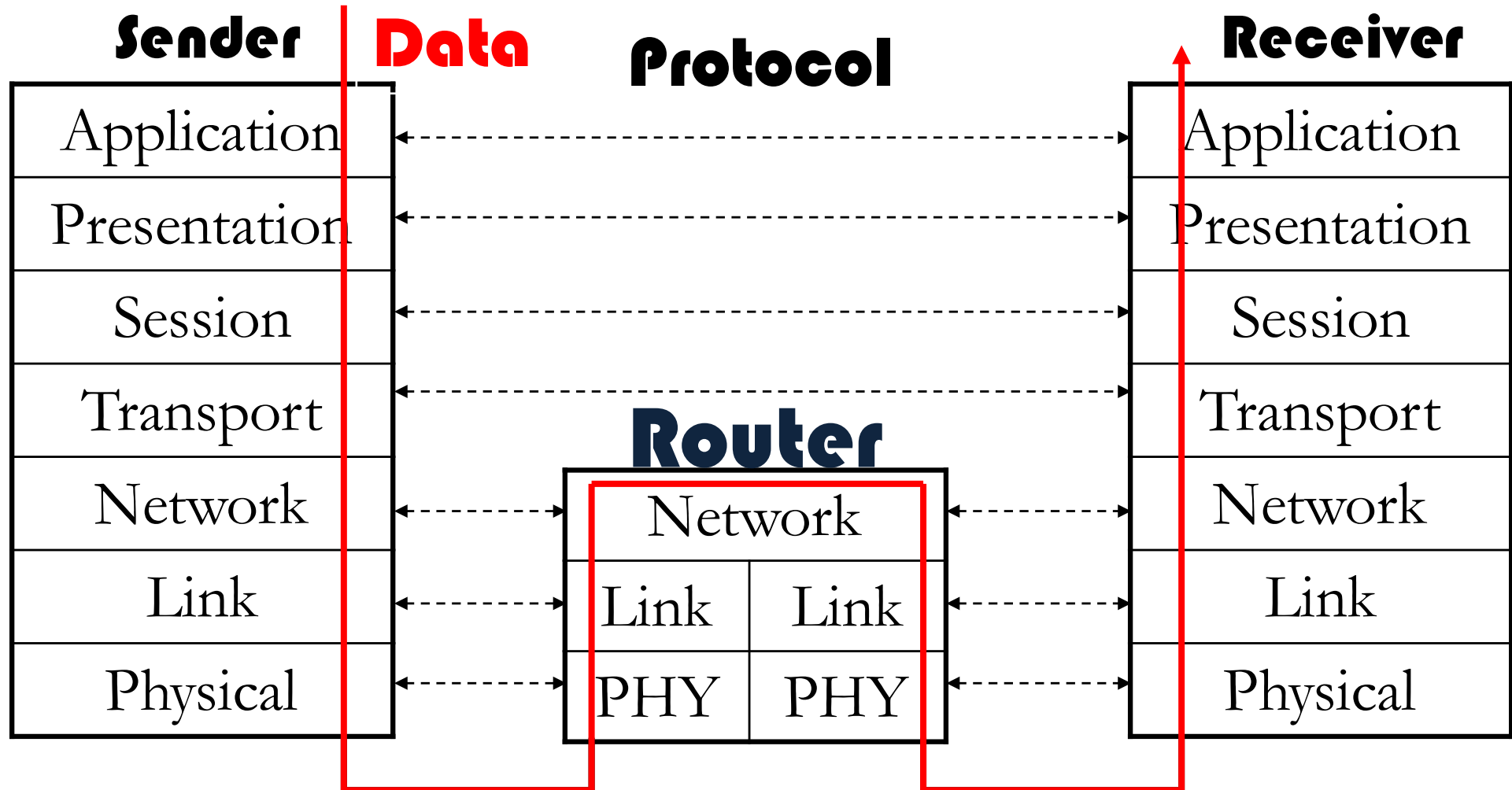
# Switch

- used within a local network



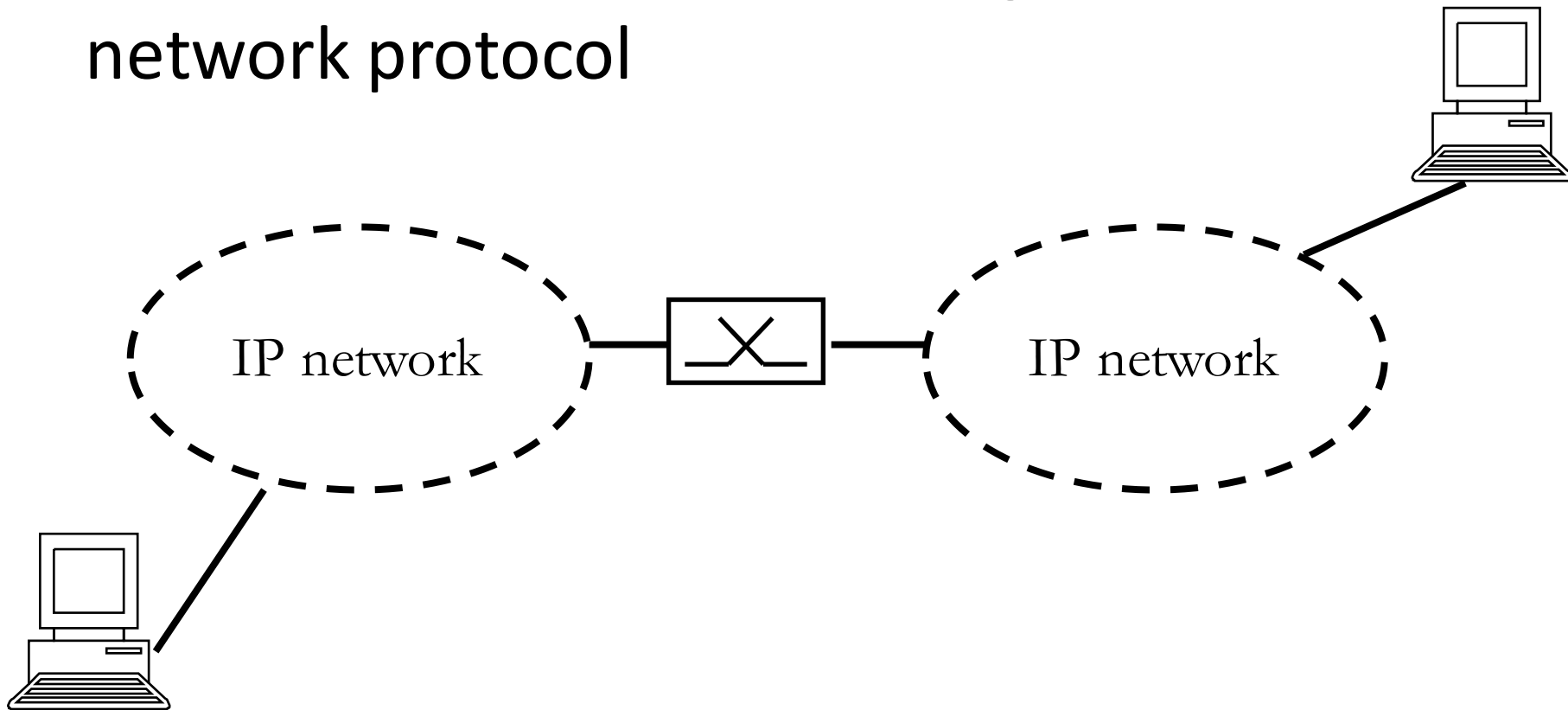
# Connecting devices

(2)



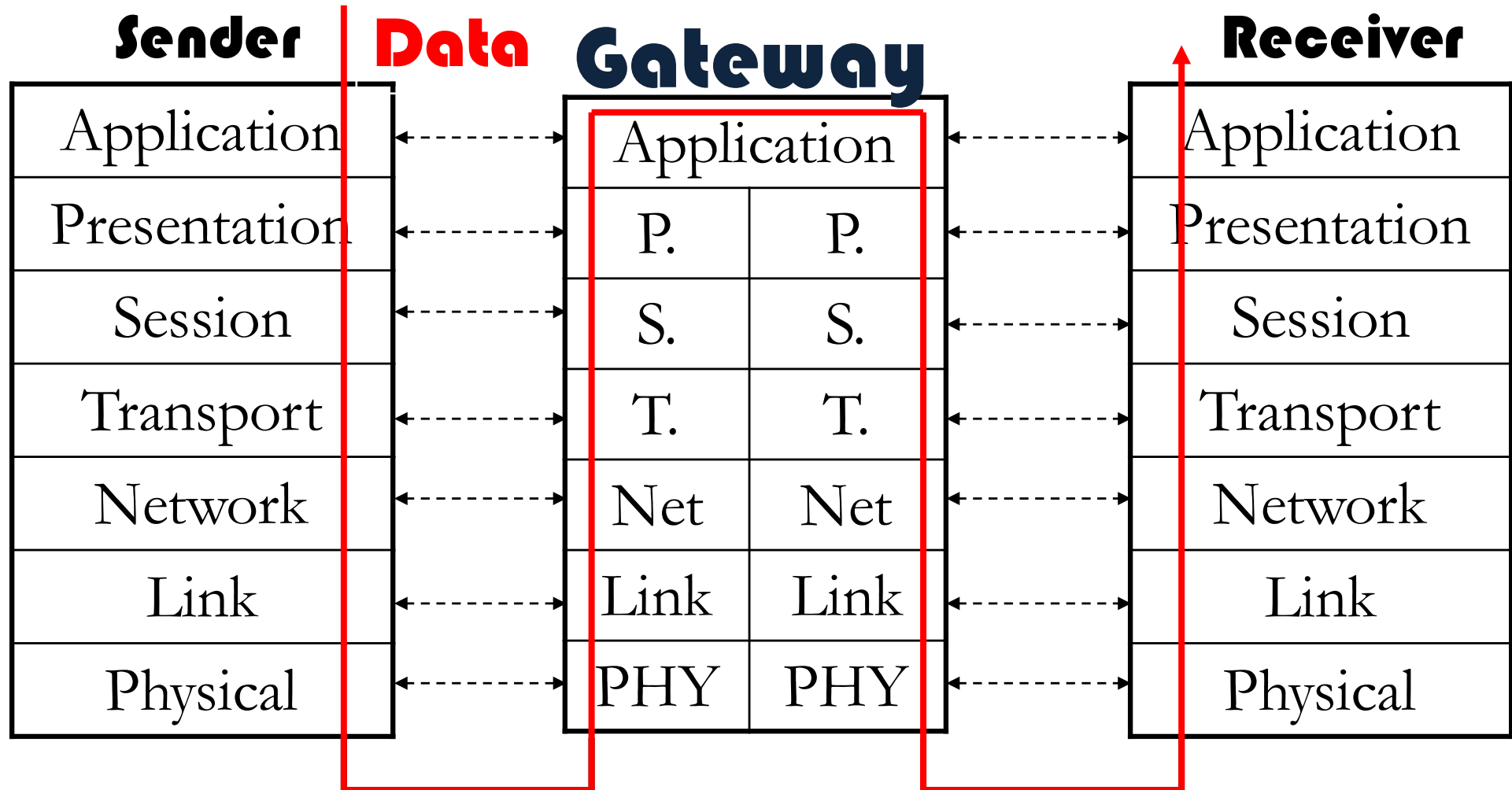
# Router

- used between networks using the same network protocol



# Connecting devices

(3)



# Gateway

- used between networks of different types

