



Review

BUPT/QMUL
2019-06-03





Network Basics (1)

- What is the Internet?
 - Internet vs. internet
 - Internet vs. www
- What are the major components of Internet?
 - Internet applications
 - Internet protocols
 - Internet addresses
 - Physical infrastructure
- What is protocol?
 - Protocol = A set of rules for communicating
 - HTTP/FTP/SMTP..., TCP/UDP, IP



Network Basics (2)

- What are the two important design concepts of Internet?
 - Layered networking model
 - Client-server paradigm
- Terms for network devices and examples
 - node, host node, link, network component
- Terms for network performance parameters:
 - bandwidth, throughput, delay, jitter, error rate
- Network types according to the switching function in the network
 - Circuit switching network
 - Message switching network
 - Packet switching network
 - Hybrid switching network



Network Basics (3)

- Different channel access technologies
 - multi-access and point-to-point
- Network types according to the range of the network
 - LAN, MAN, WAN, PAN
- Network types according to the user of the network
 - public network and private network
- Layered architecture
 - ISO/OSI model,
 - TCP/IP model
 - Revisory Model
- Devices at different layers
 - Hub, Bridge/Switch, Router, Gateway



Network Programming Basics (1)

- Basic concepts
 - Process: what is a process? Is process equals to program?
 - PID, PPID
 - Special processes: `init(PID=1)`
 - `fork()`
 - function, return values, relationship between parent process and child process
 - `exec()`
 - System call: what is a file descriptor?
 - Special file descriptors: 0,1,2
 - Related system calls and their functions:
 - `open()`, `read()`, `write()`, `lseek()`
 - Signal: what is a signal?
 - Which conditions will generate signals?: 5 conditions
 - What can a process do with a signal? : 3 choices



Network Programming Basics (2)

- IP Addresses

- IP address structure
- Big-endian and little-endian, HBO and NBO
- byte order conversion functions: `htonl()`, `ntohl()`, `htons()`, `ntohs()`

- DNS

- Host entry structure
- System calls for retrieving host entries
 - `gethostbyname()`, `gethostbyaddr()`

- Connection

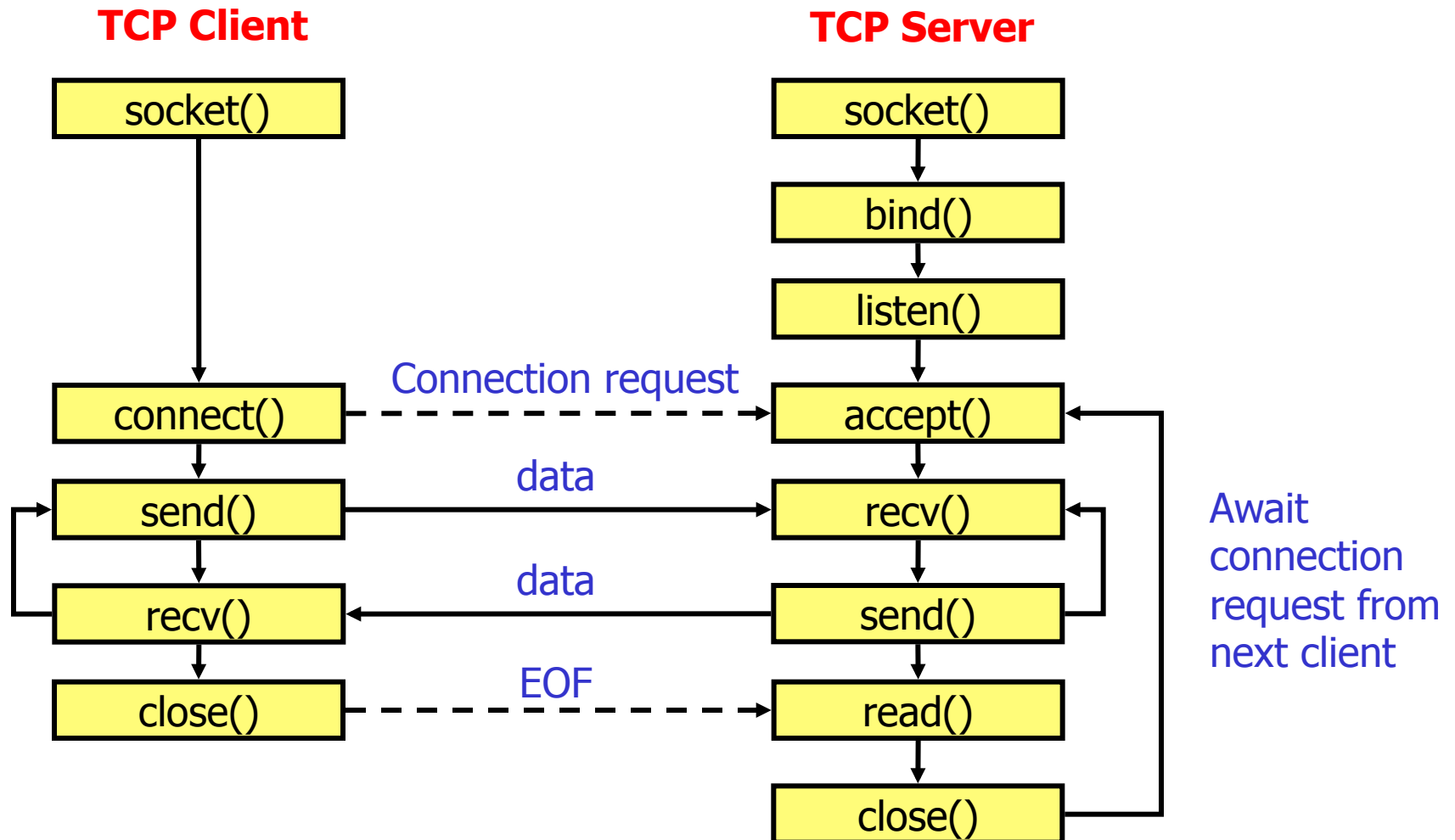
- What is a connection? How to identify an endpoint of a connection?
How to identify a connection?
 - Client-IP, client-port, server-IP, server-port
- Well-known port numbers for typical applications
 - DHCP, DNS, TELNET, TFTP, FTP, SMTP, POP3, HTTP, SNMP



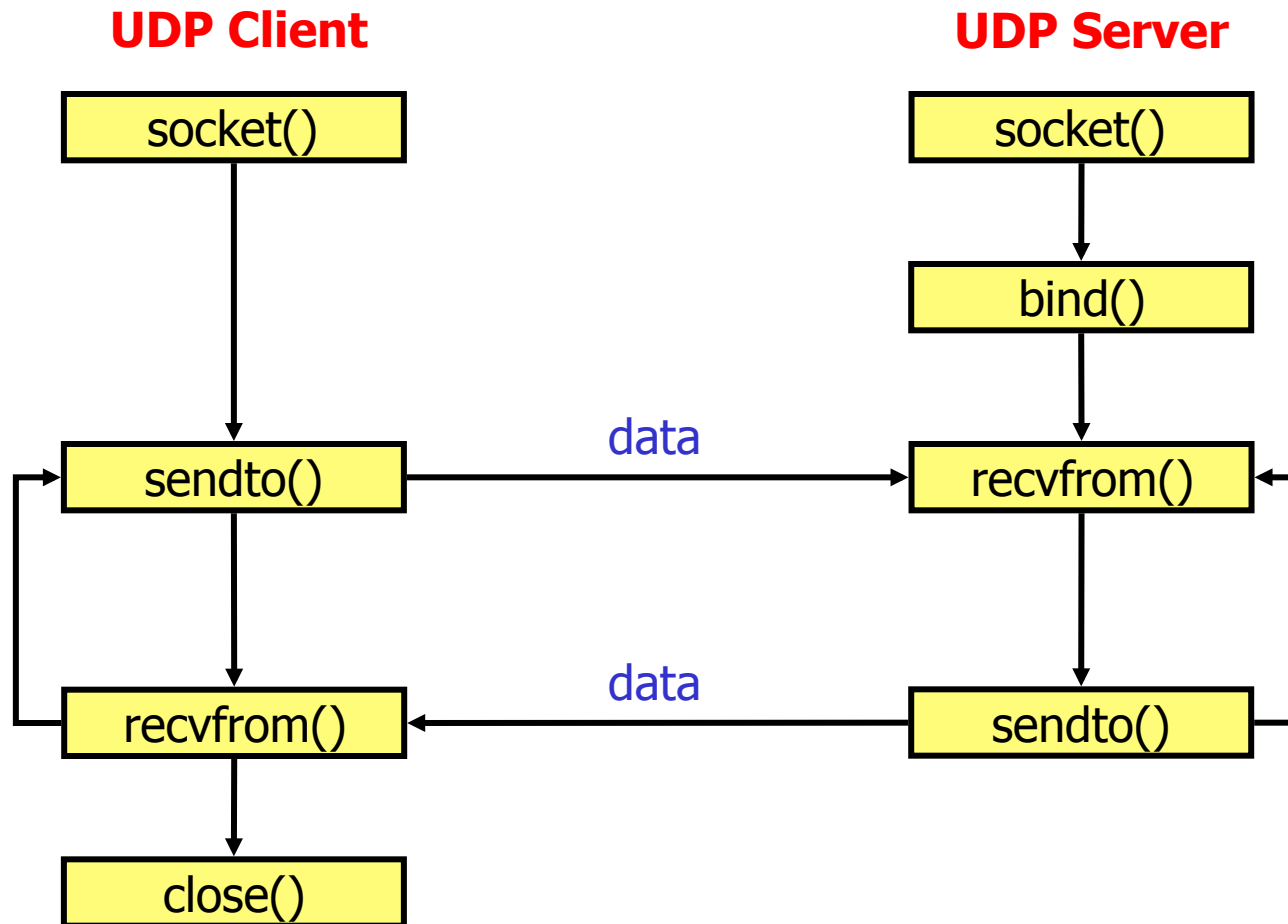
Network Programming Basics (3)

- Sockets interface
 - What is a socket? What is the sockets descriptor?
 - Internet-specific socket address and Generic socket address
 - struct sockaddr vs. struct sockaddr_in
 - how do we use them in the system calls such as connect(), bind(), and accept()?
 - Different sockets interface types
 - SOCK_STREAM, SOCK_DGRAM, SOCK_RAW
- System calls used in sockets programming
 - Socket operation: socket(), bind(), send(),recv(), close()...
 - Byte order operation: htonl(),...
 - Address formats conversion: inet_aton(), inet_ntoa(),...
 - Name and address operation: gethostbyname(),...
- Basic flows of TCP/UDP based sockets API

Overview of TCP-based sockets API



Overview of UDP-based sockets API



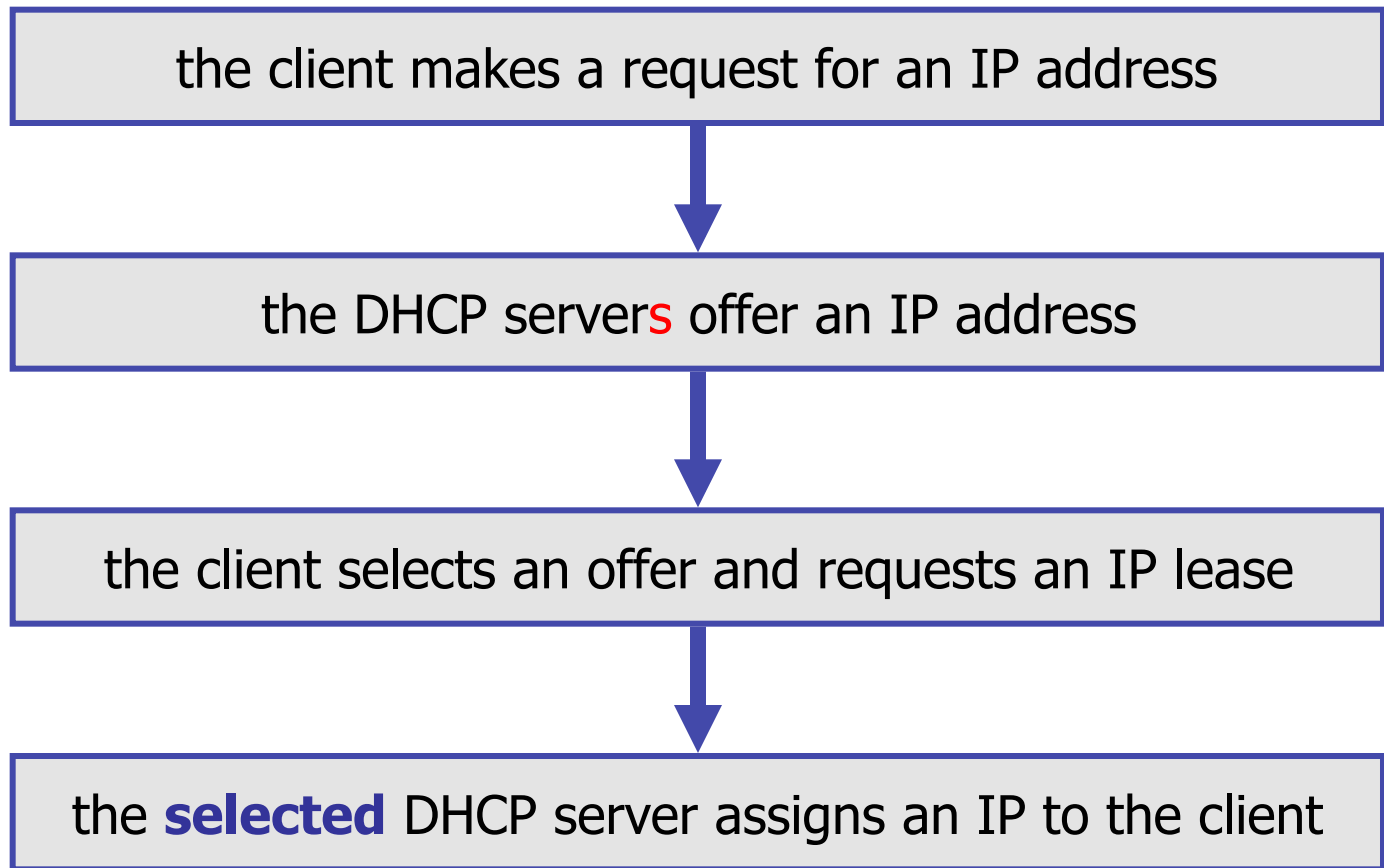


DHCP

- DHCP overview
 - Basic function: **automatic configuration** of remote hosts
 - relationship between BOOTP and DHCP
 - DHCP client(port 68), DHCP server(port 67)
 - DHCP Relay: **forward** DHCP messages between subnets.
 - DHCP lease: **amount of time** that the DHCP server grants to the DHCP client
 - Phases of IP assignment with DHCP
- DHCP message format
- STD and Message Sequence Chart of DHCP Address acquisition
- STD and MSC of Early lease termination in DHCP
- STD and MSC of Lease renewal in DHCP
- Basic address acquisition procedure through DHCP relay



Phases of IP Assignment with DHCP

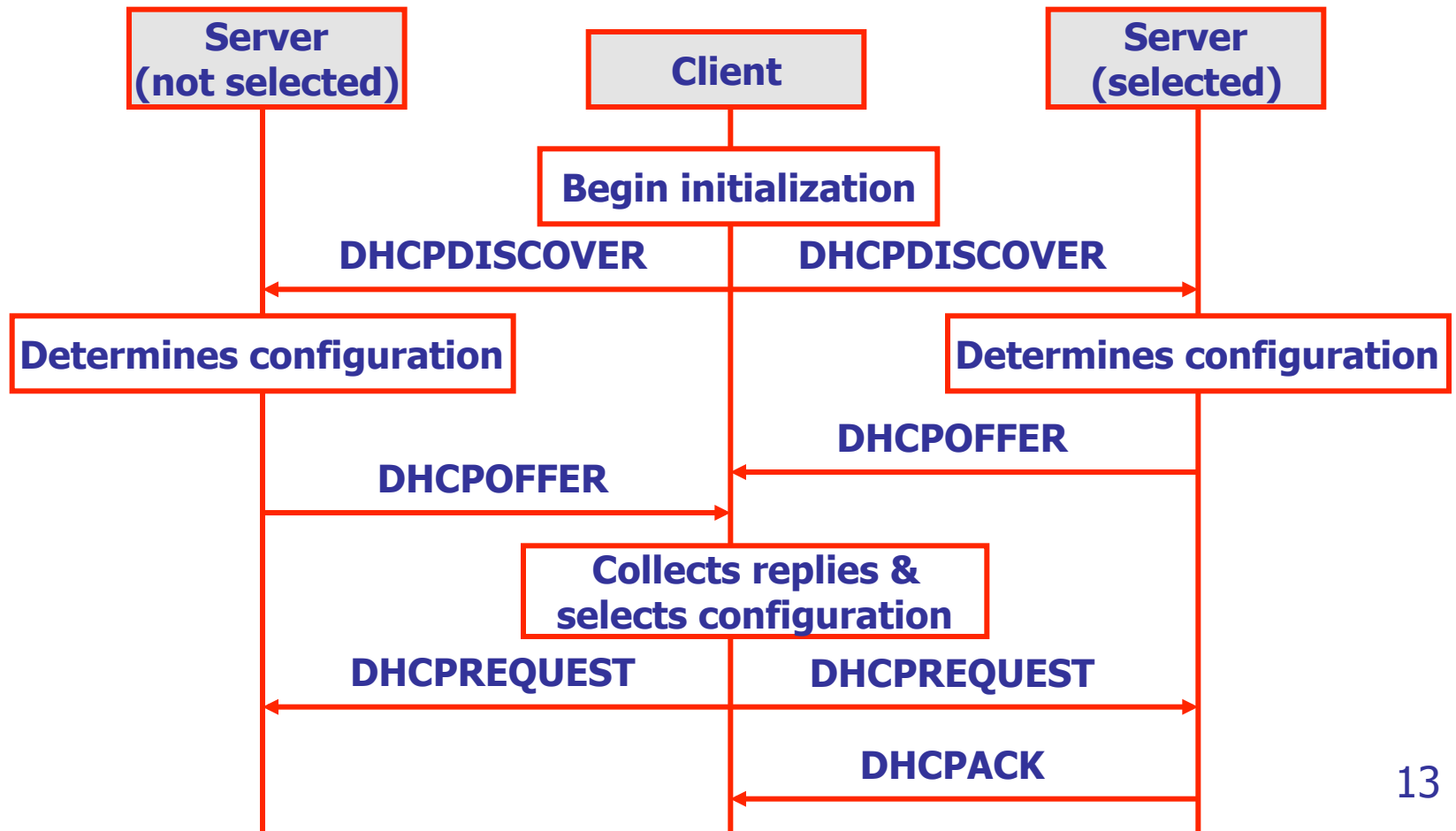




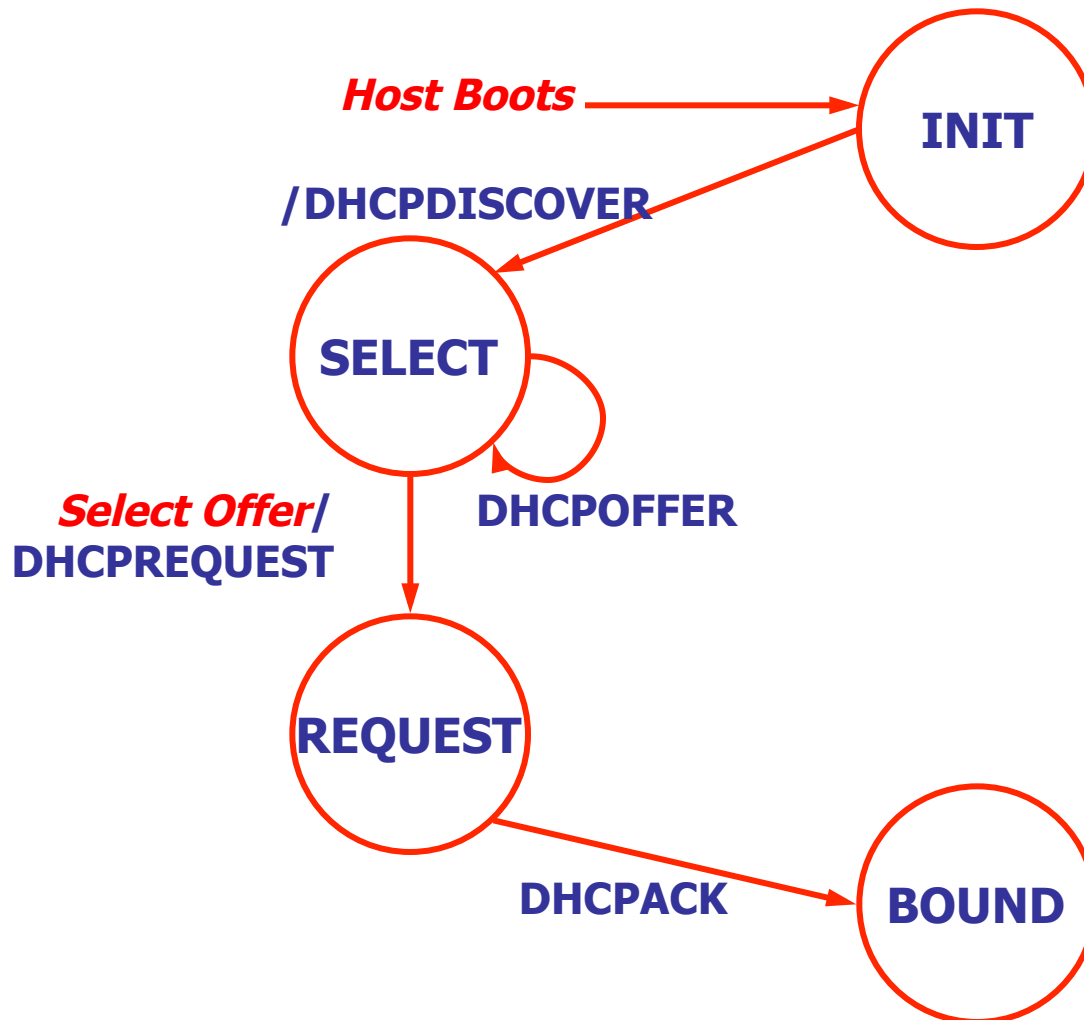
DHCP Message Format

OP (1)	HTYPE (1)	HLEN (1)	HOPS (1)
TRANSACTION ID (4)			
SECONDS (2)		FLAGS (2)	
CLIENT IP ADDRESS (4)			
YOUR IP ADDRESS (4)			
SERVER IP ADDRESS (4)			
ROUTER IP ADDRESS (4)			
CLIENT HARDWARE ADDRESS (16)			
⋮			
SERVER HOST NAME (64)			
⋮			
BOOT FILE NAME (128)			
⋮			
OPTIONS (variable)			
⋮			

Address Acquisition: MSC

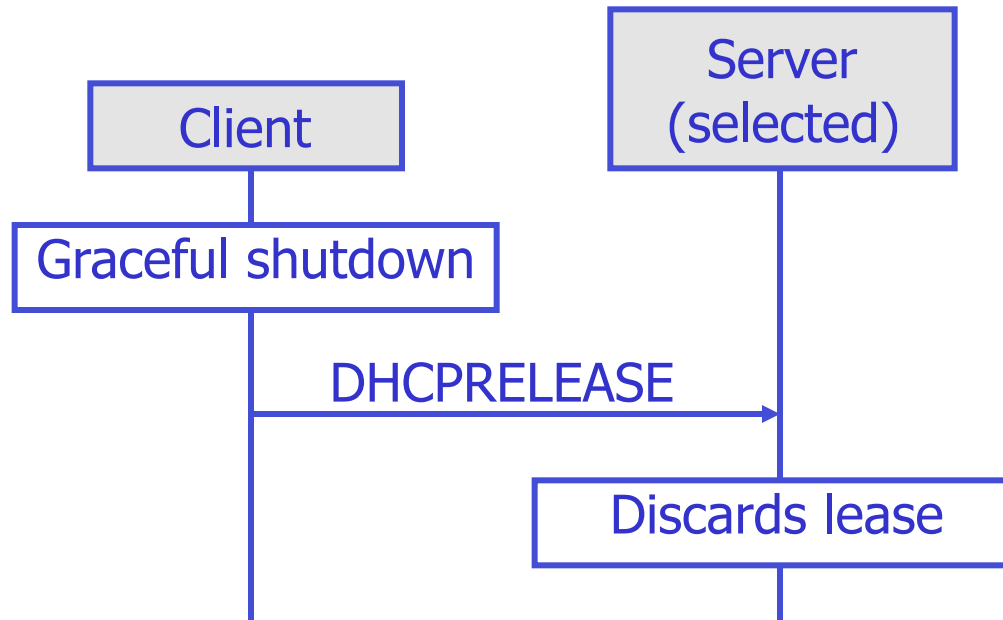


Address Acquisition: STD of the client



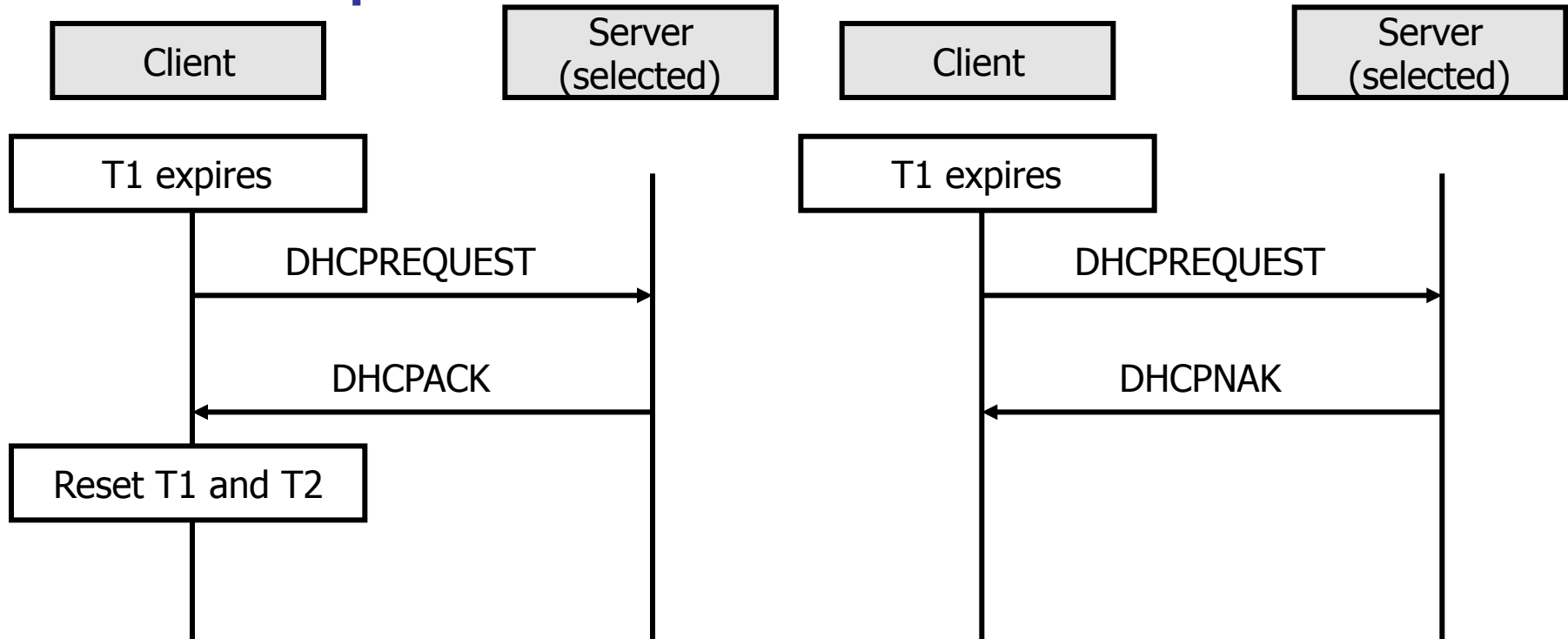
State
Transition
of DHCP
Client

Early Lease Termination: Procedure



Lease Renewal: Procedure (1)

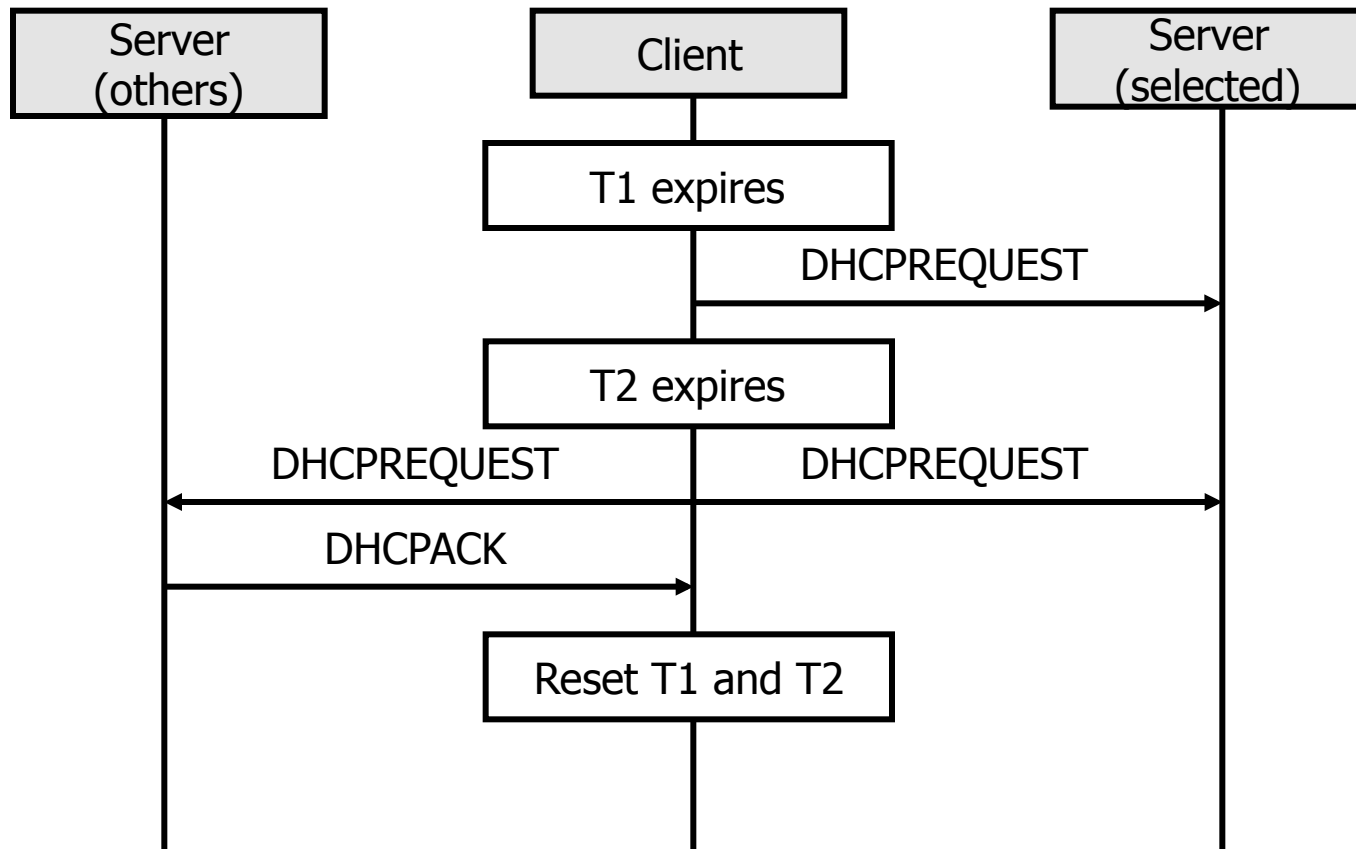
■ T1 expires



- **T1**: the time at which the client enters the **RENEW state** and attempts to contact the server that **originally issued** the client's network address
 - $0.5 * \text{duration_of_lease}$
- **T2**: the time at which the client enters the **REBIND state** and attempts to contact **any** server.
 - $0.875 * \text{duration_of_lease}$

Lease Renewal: Procedure (2)

- Both T1 and T2 expire, address rebound

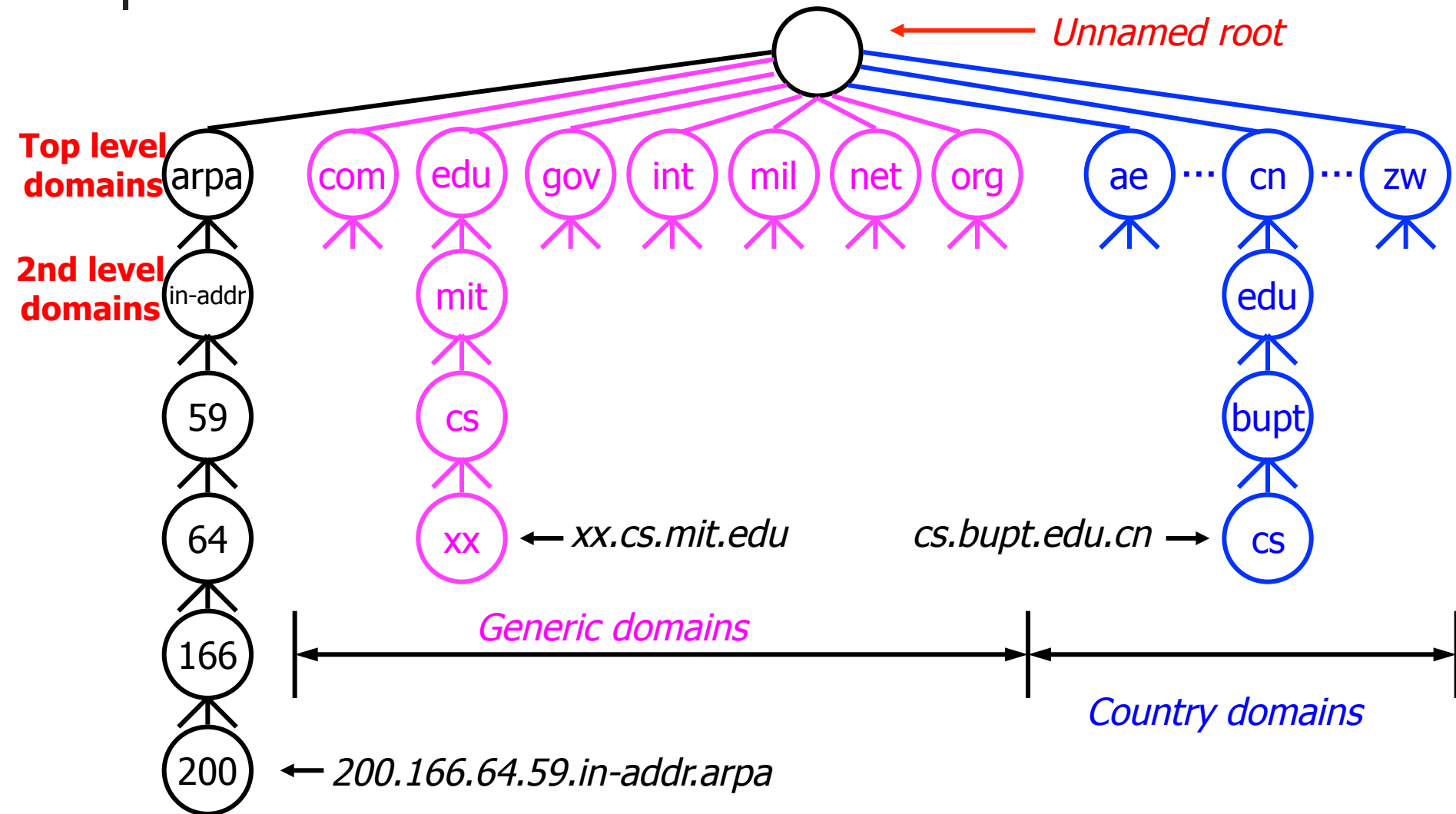




DNS(1)

- Basic functions of DNS: converting between hostname and IP address
- Current status: hierarchical structure, distributed database, general-purpose
- Hierarchical structure of domain namespace
- Important terms
 - Domain / domain name / FQDN
 - Domain namespace, zone
 - Resource Record
 - Name server / primary server / secondary server / caching server
 - Resolver
 - Query / response
 - Standard query / inverse query / pointer query
 - recursive resolution / iterative resolution

Hierarchical structure

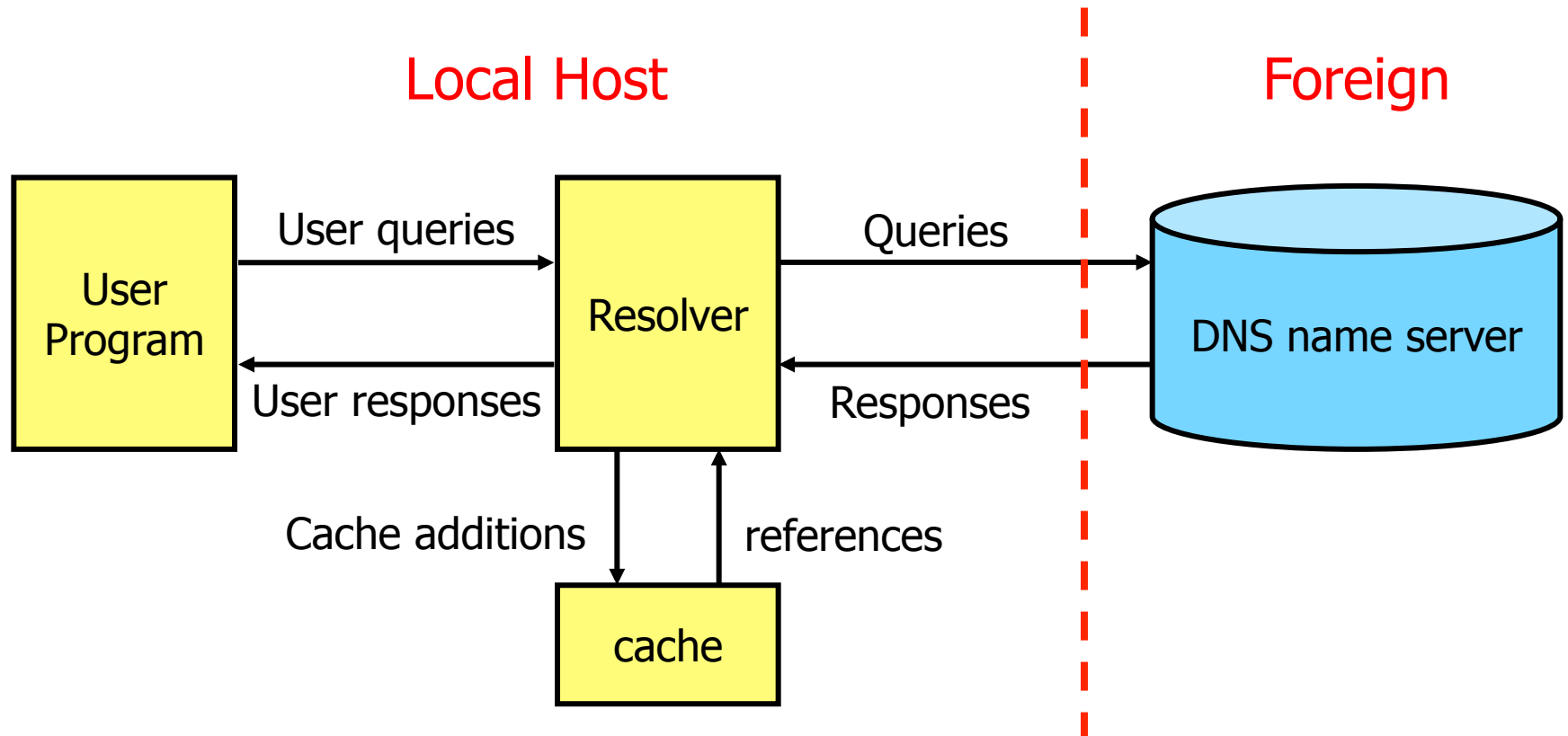




DNS(2)

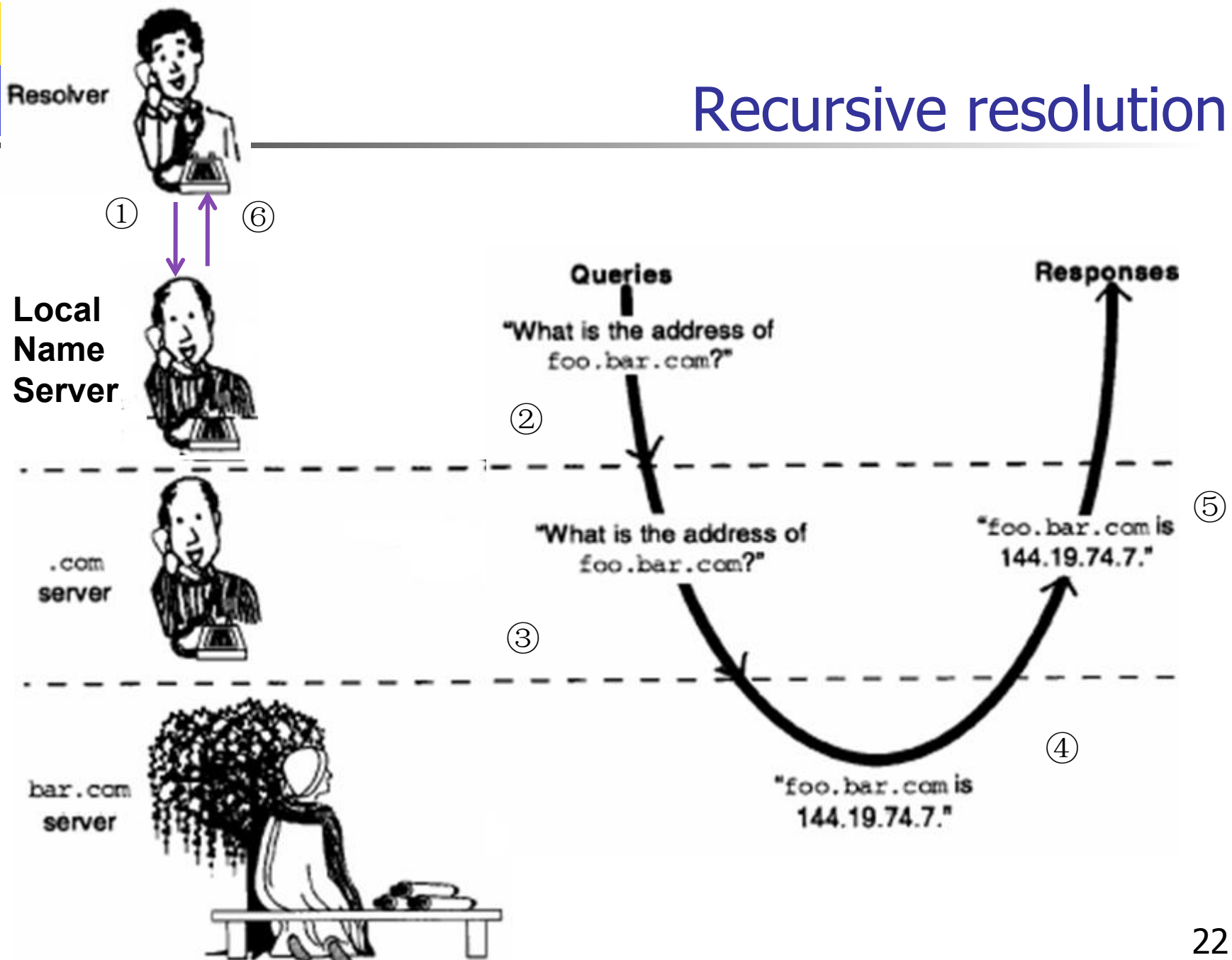
- How does DNS work together with the user programs (e.g. TELNET, FTP, HTTP, SMTP) ?
- Procedure of the recursive resolution and iterative resolution
- What are the mechanisms in DNS that are possible to improve the querying efficiency?
- The ideas of inverse query and pointer query. The comparison between them.
- DNS Message Format
- Types of Resource Record (only the ones highlighted using red color in the lecture notes)
 - A, NS, MX, CNAME, PTR

Communication model between user program, resolver and name server

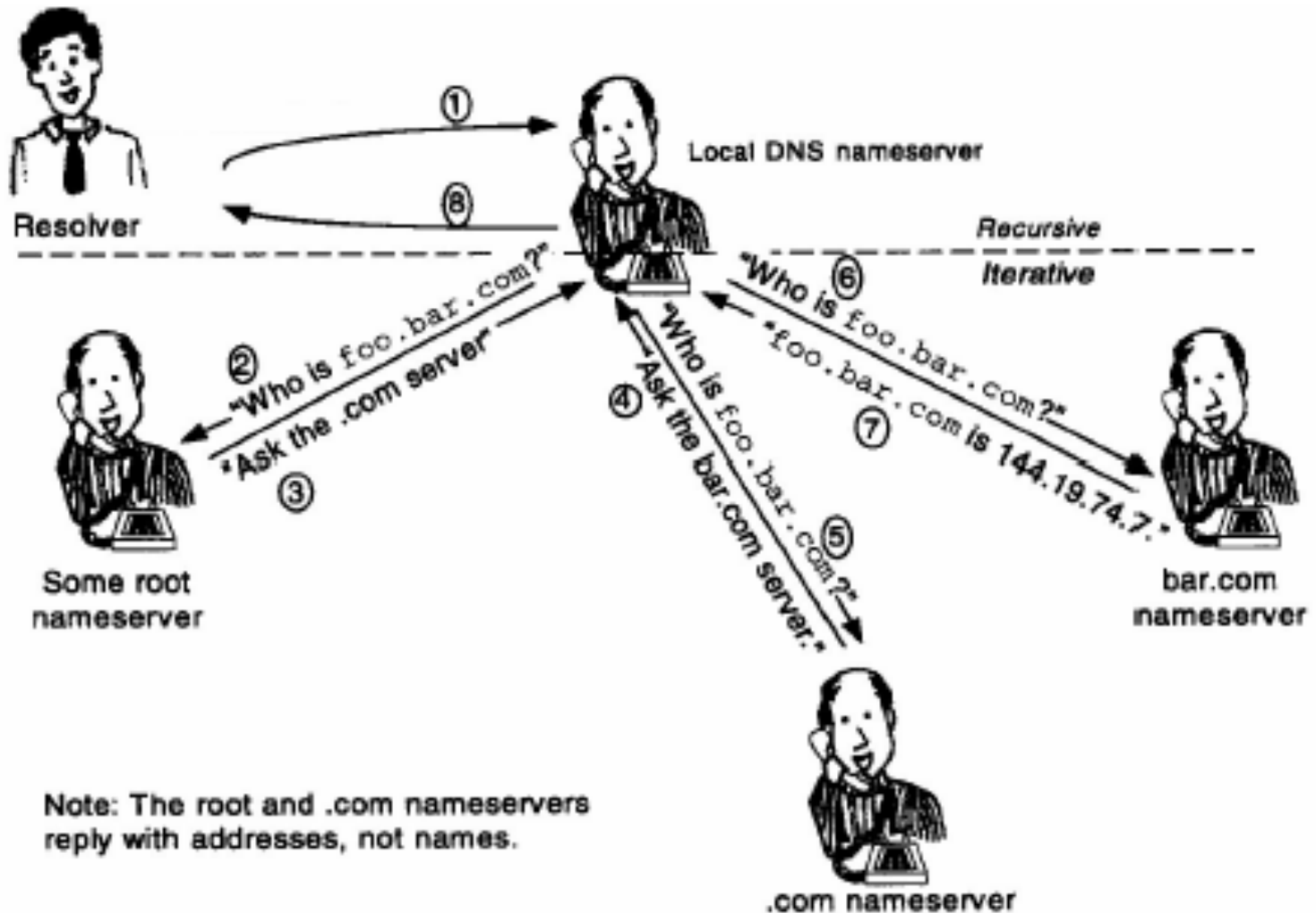


Resolvers are normally implemented in system calls.
E.g. `gethostbyname()`, `gethostbyaddr()`

Recursive resolution



Iterative resolution



- *Query* and *Response* messages, both with *same message format*

ID	QR	OPCODE	AA	TC	RD	RA	Z	Rcode
Question count	Answer count							
Authority count	Additional count							
<p>Question Section (variable number of questions)</p>								
<p>Answer Section (variable number of RRs)</p>								
<p>Authority Section (variable number of RRs)</p>								
<p>Additional Section (variable number of RRs)</p>								

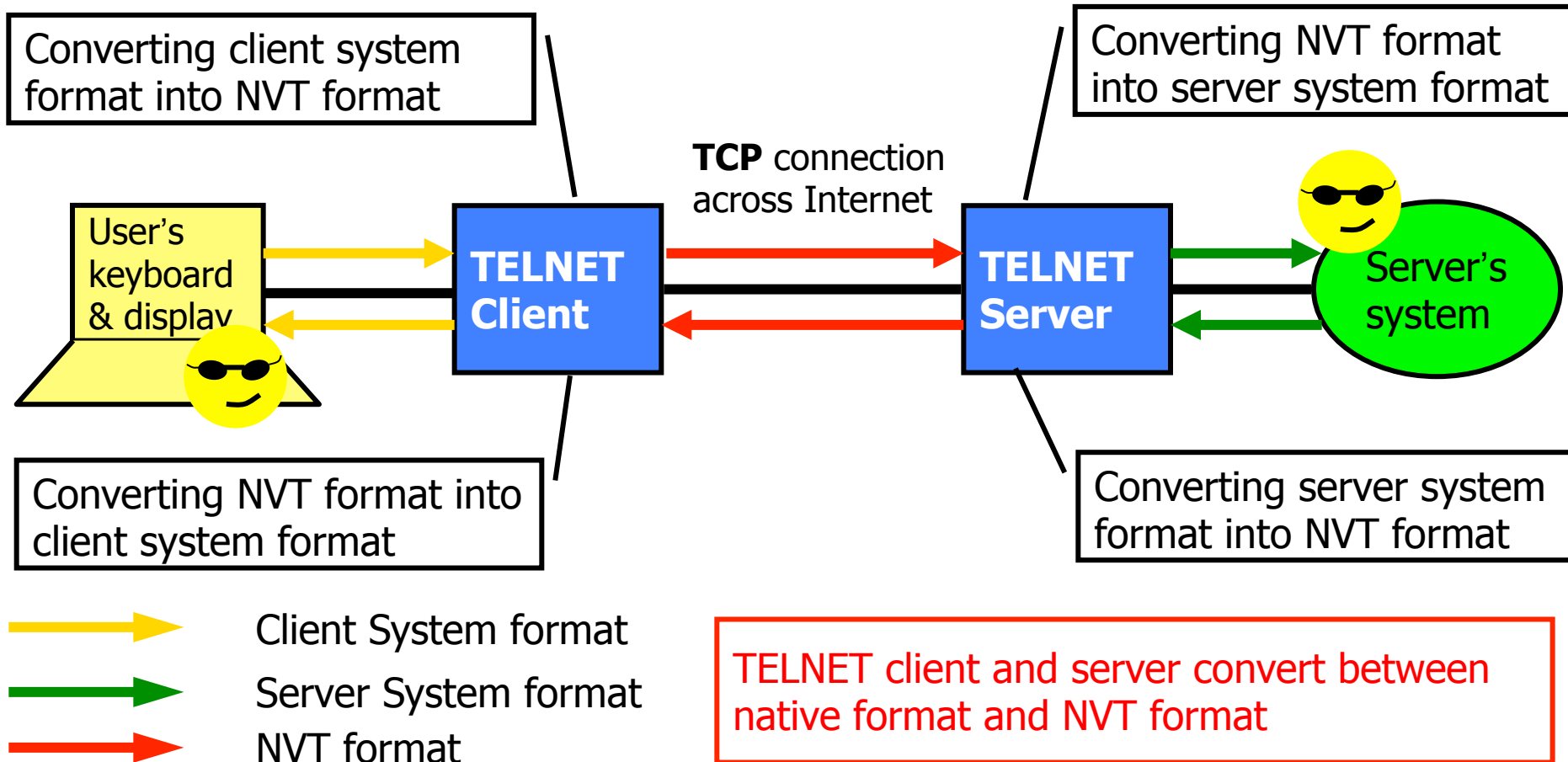


Telnet(1)

- What is TELNET and telnet?
 - TELNET: a **protocol** used to establish a **dumb terminal** session to another computer on the Internet
 - telnet: a **program** that supports the TELNET protocol over **TCP**
- What are the advantages of the idea of option negotiation in TELNET?
- NVT
 - What is NVT? What are its functions?
 - **Network Virtual Terminal**, providing a **standard interface** to remote systems
 - NVT operations

NVT Operation

- Accommodating heterogeneity





Telnet commands

- TELNET control functions:
 - IAC, DO, DONT, WILL, WONT
- TELNET options example
 - Echo modes
 - Binary transmission
 - Line mode vs. character mode
 - Character set
 - Terminal type
- Understand the TELNET session through examples



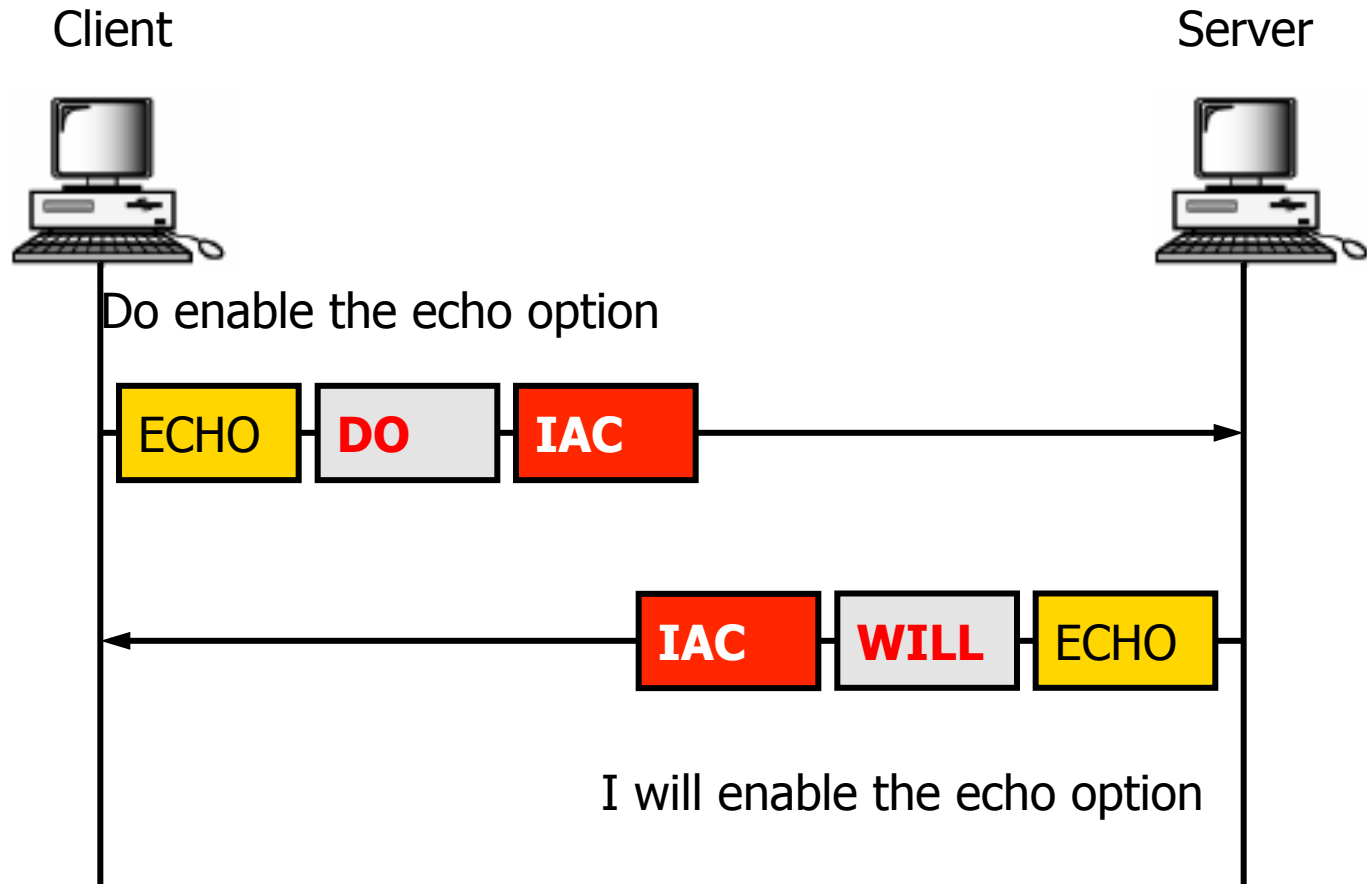
Telnet Control Functions

– DO, DONT, WILL, WONT

- Used for options negotiation
- Examples

Sender	Receiver	Meaning
WILL →	← DO	Sender wants to active a option, and receiver agrees
WILL →	← DON'T	Sender wants to active a option, and receiver refuses
DO →	← WILL	Sender wants receiver to active a option, and receiver agrees
DO →	← WONT	Sender wants receiver to active a option, and receiver refuses

Example: Negotiation of Echo Option





Other Remote Access Technologies

- Remote login in text-based system
 - telnet
 - SSH
 - Rlogin



TFTP/FTP(1)

- TFTP features
 - Read and write files from / to remote computers
 - Minimal overhead ...
- Transfer mode of TFTP: Netascii, Octect
- Retransmission defined in original TFTP protocol
- The sorcerer's apprentice syndrome problem and how to fix it

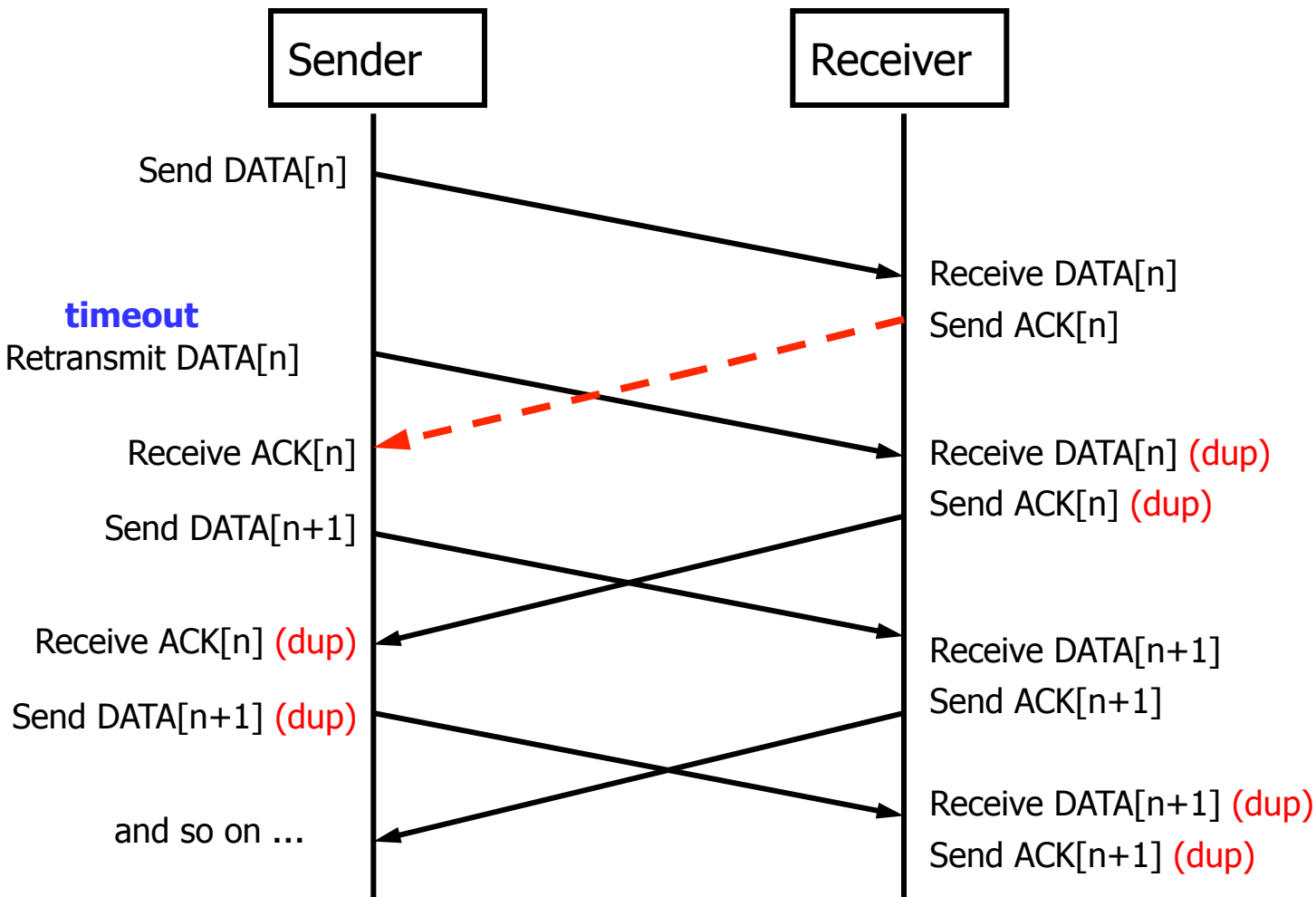


TFTP Operations – Retransmission

- **Symmetric**
 - Both machines involved in a transfer are considered senders and receivers.
 - One sends data and receives acknowledgments
 - The other sends acknowledgments and receives data
 - Each side implement the **timeout** and **retransmission**
 - If a data packet gets lost in the network, the data sender times out and retransmits the last data packet
 - If an acknowledgment is lost, the acknowledgment sender retransmits the last acknowledgment
- The sender has to keep just **one packet** on hand for retransmission, since the **lock step acknowledgment** guarantees that all older packets have been received
- **Duplicate** data packets must be recognized (ignored) and acknowledgment retransmitted
- This original protocol suffers from the ***sorcerer's apprentice syndrome (SAS)***

TFTP Operations

– Sorcerer's Apprentice Syndrome



- Arising when an acknowledgment for a data packet is **delayed**, but not lost

- Leading to excessive retransmissions

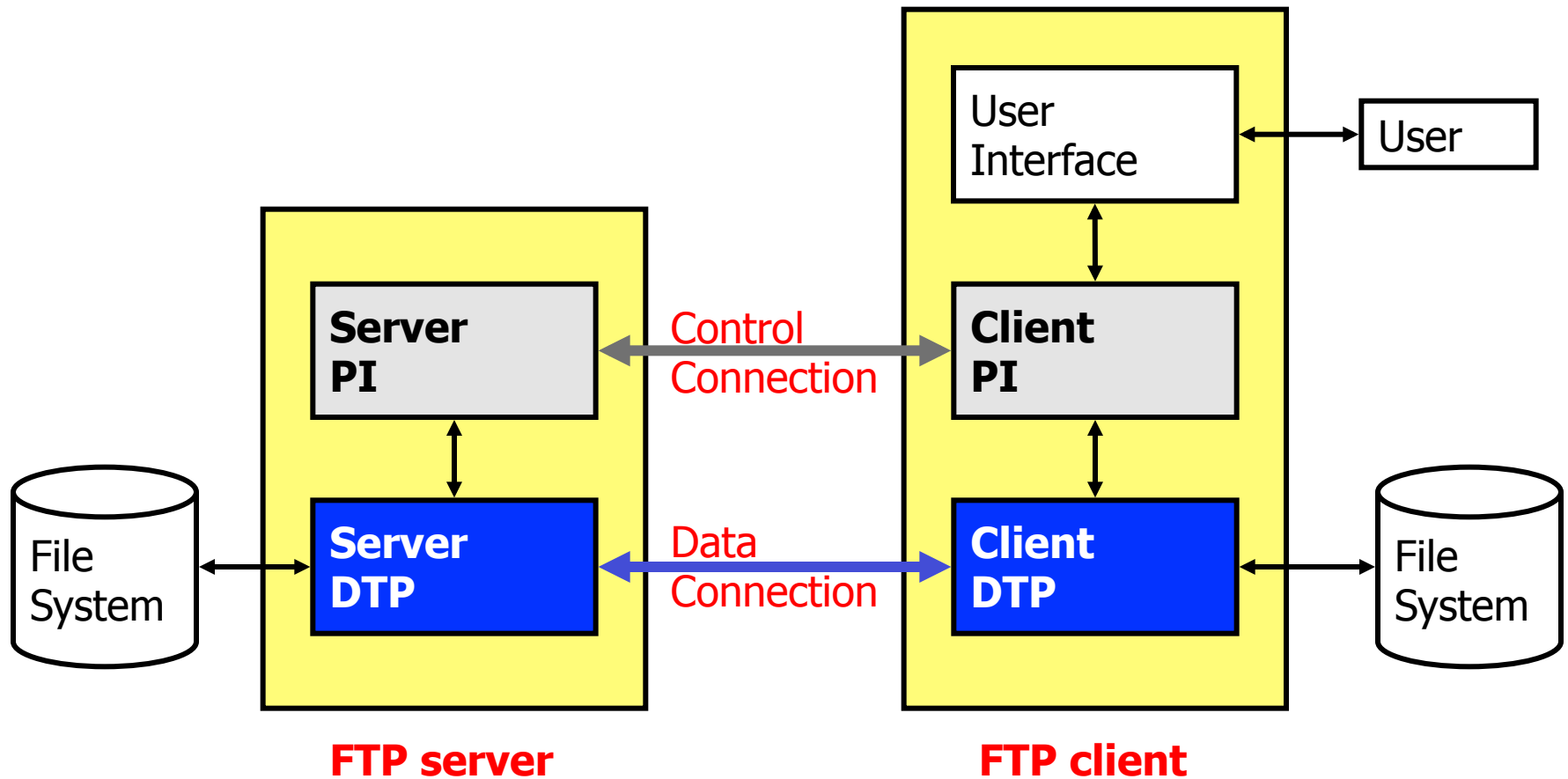
- Once started, the cycle continues indefinitely with each data packet being transmitted exactly **twice**



TFTP/FTP(2)

- FTP features
 - Used to transfer files between hosts
 - Used to manipulate files, ...
- FTP model
- FTP basic control commands and replies
 - USER, PASS, CWD, CDUP, QUIT,...
 - PORT, PASV
 - LIST, RETR, LIST,...
- FTP Control Connection & Data Connection
- FTP Active and Passive Mode

FTP Model

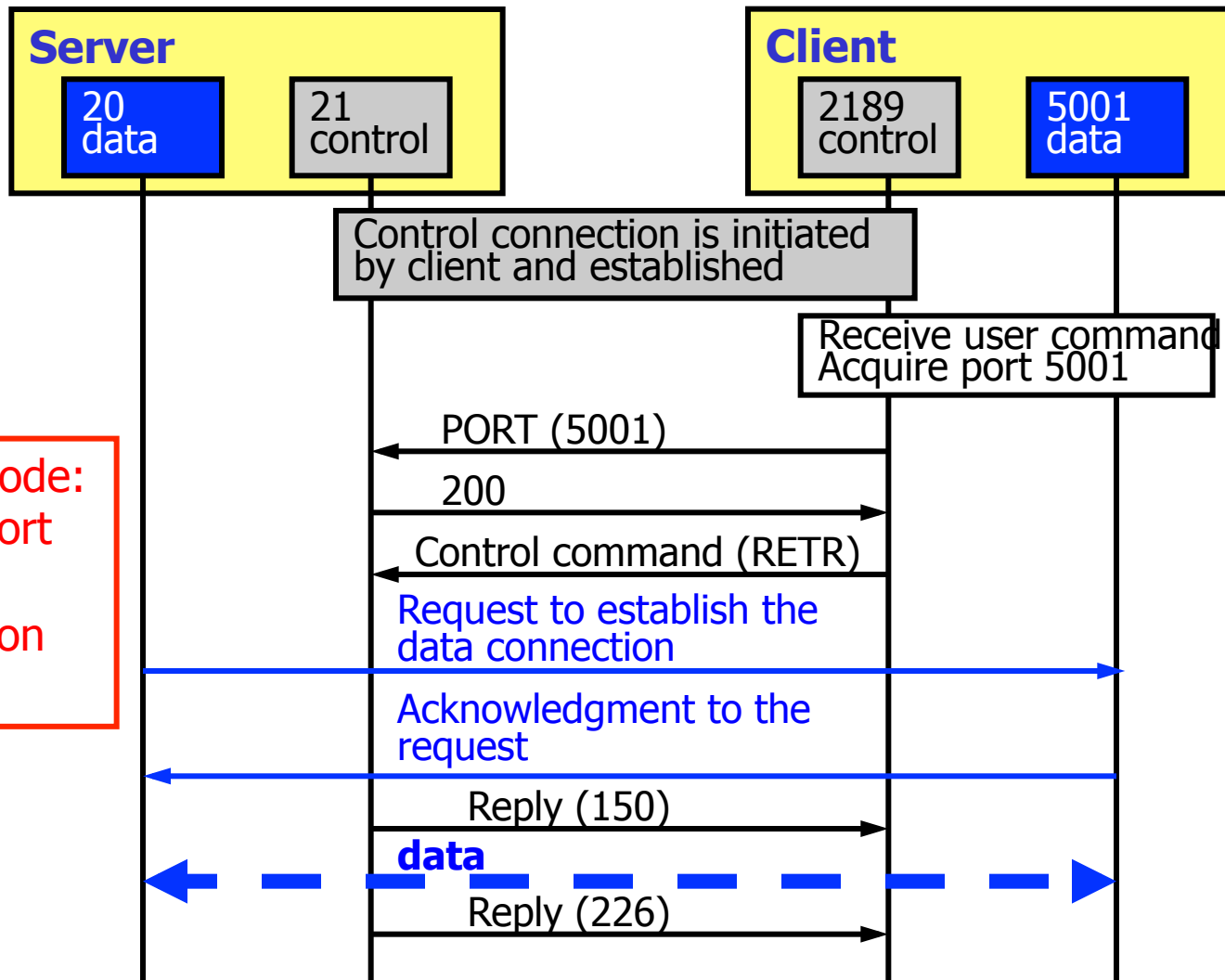




Active and Passive mode

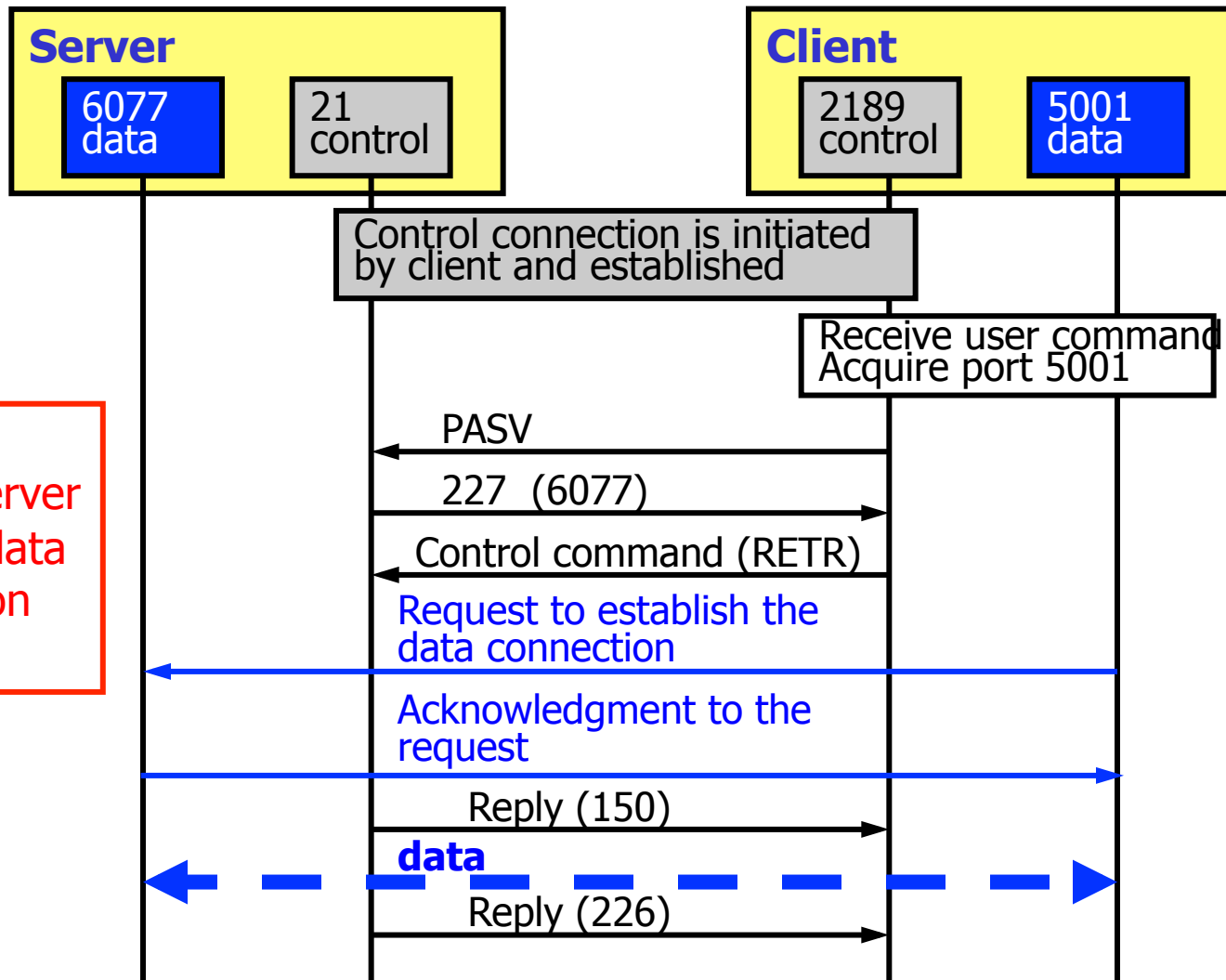
- Typical data connection handling sequence (in **active mode**)
 - Client sets up to “listen” on a unique port
 - Client uses local socket information to send PORT command to server
 - Server responds with “200” reply to acknowledge the port number
 - Client sends RETR, STOR, or other transfer command
 - Server sends preliminary reply
 - Server does active open (“connect”)
 - File data sent over connection
 - Server sends “226” or other reply
 - Server/client closes data connection
- Another mode: **passive mode**
 - Client sends command *PASV*
 - server listens to a specific port and client should access that port₃₆

FTP Control Connection & Data Connection (1)



Active Mode:
Server port
for data
connection
=20

FTP Control Connection & Data Connection (3)



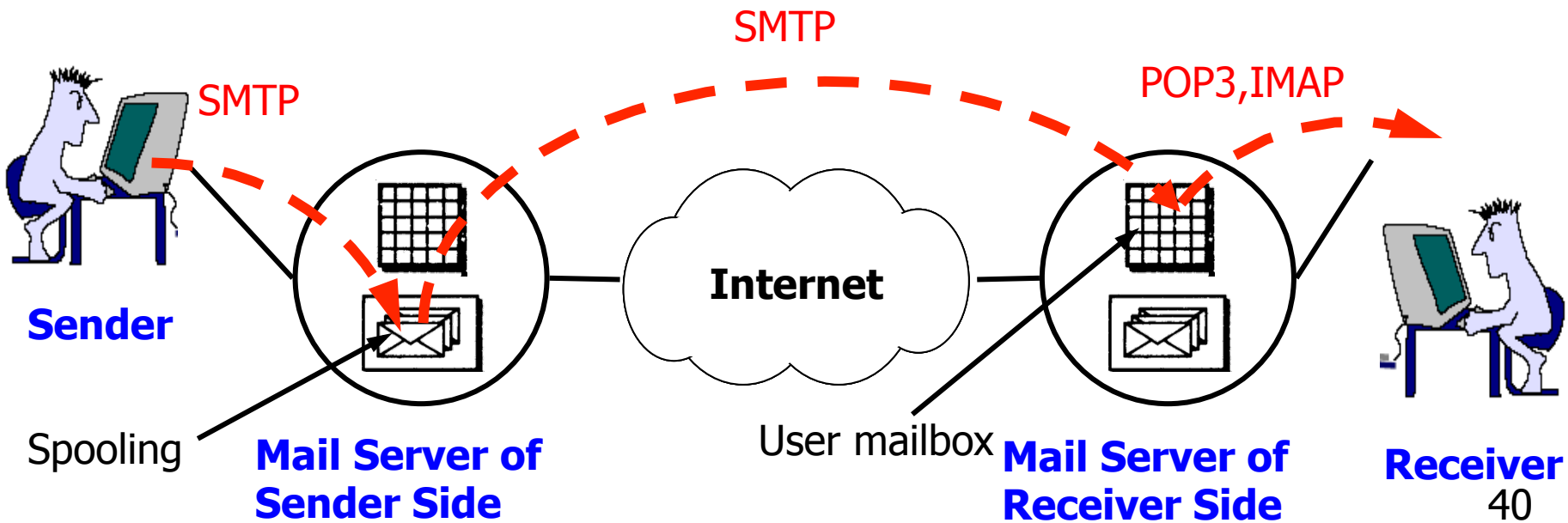
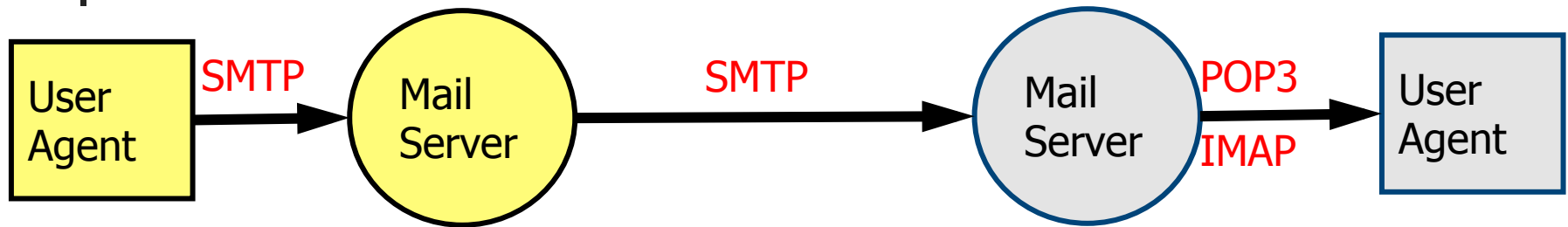
Passive
Mode: Server
port for data
connection
> 1024



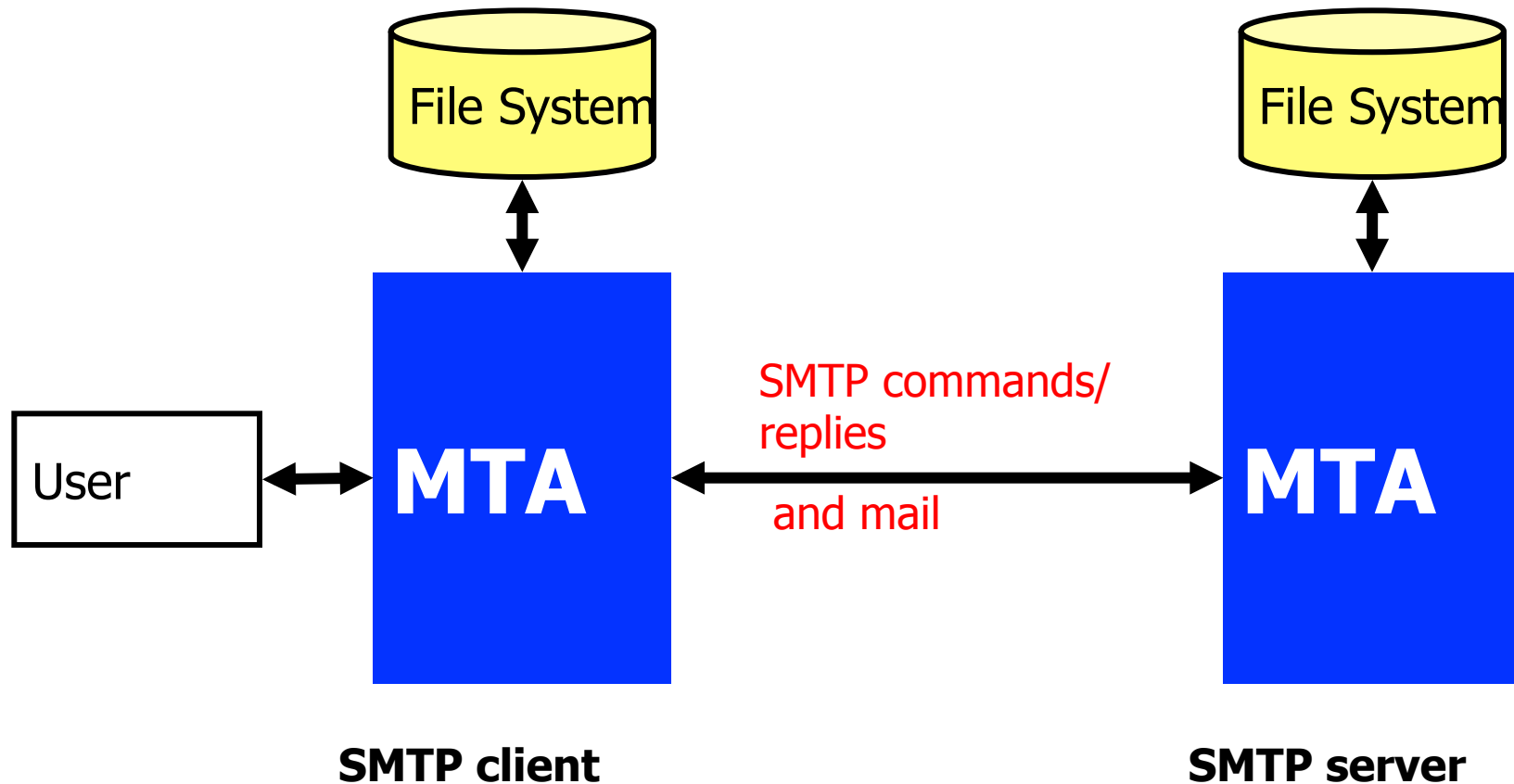
Email(1)

- Email Overview
 - Components of email system
 - Basic functions of email system
 - *Composition, Transfer, Reporting, Displaying, Disposition*
 - Terms: UA, Mail Server, MTA
 - Email address: *mailboxname@domain*
- Message Format
 - Header, blank line, body
- SMTP
 - Basic model
 - Basic commands and replies
 - HELO, MAIL FROM, RCPT TO, DATA, QUIT

Components Of Email System (1)



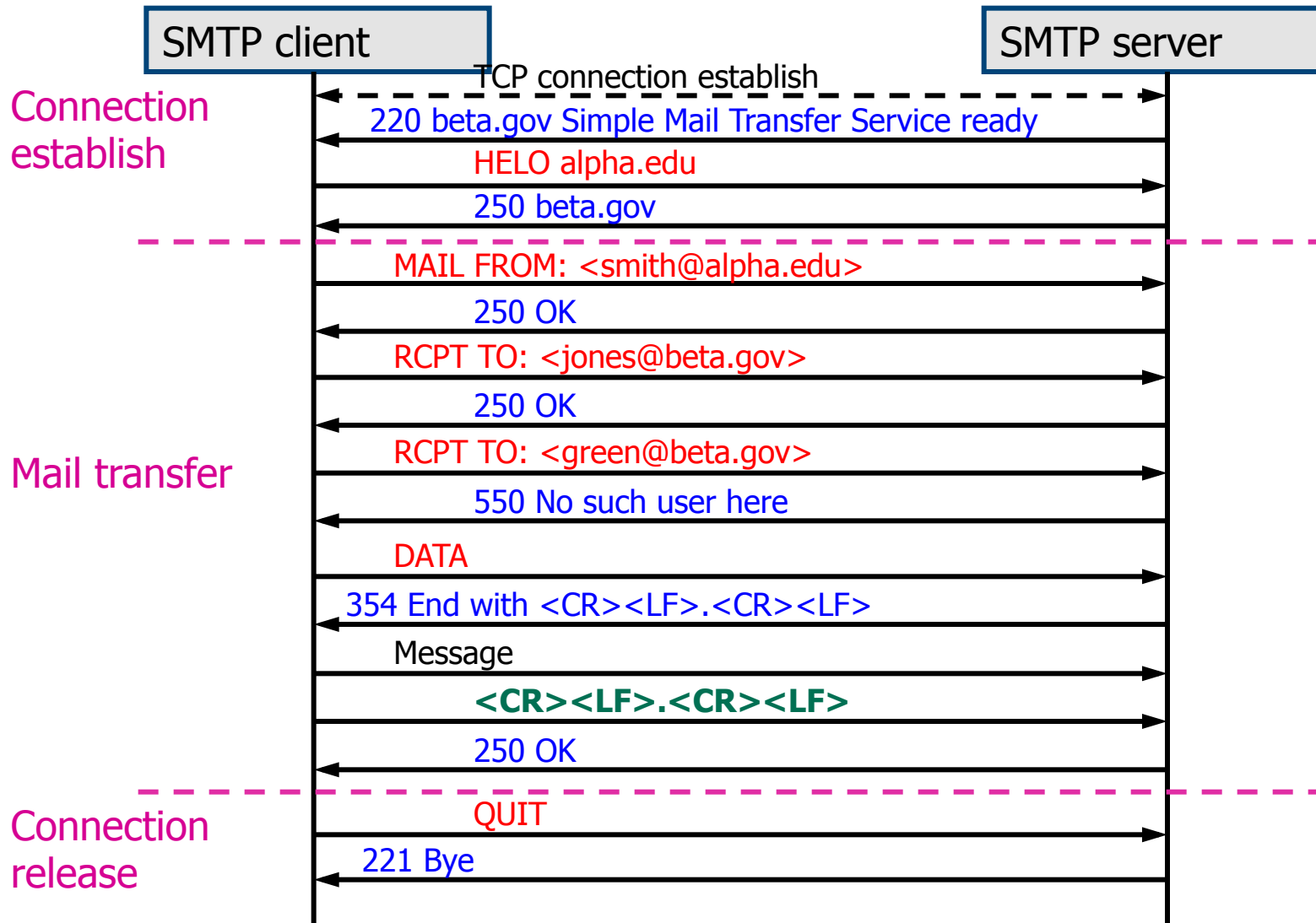
SMTP Basic Model



SMTP Commands and Status codes

– example

- smith@alpha.edu sends a mail to jones@beta.gov, green@beta.gov



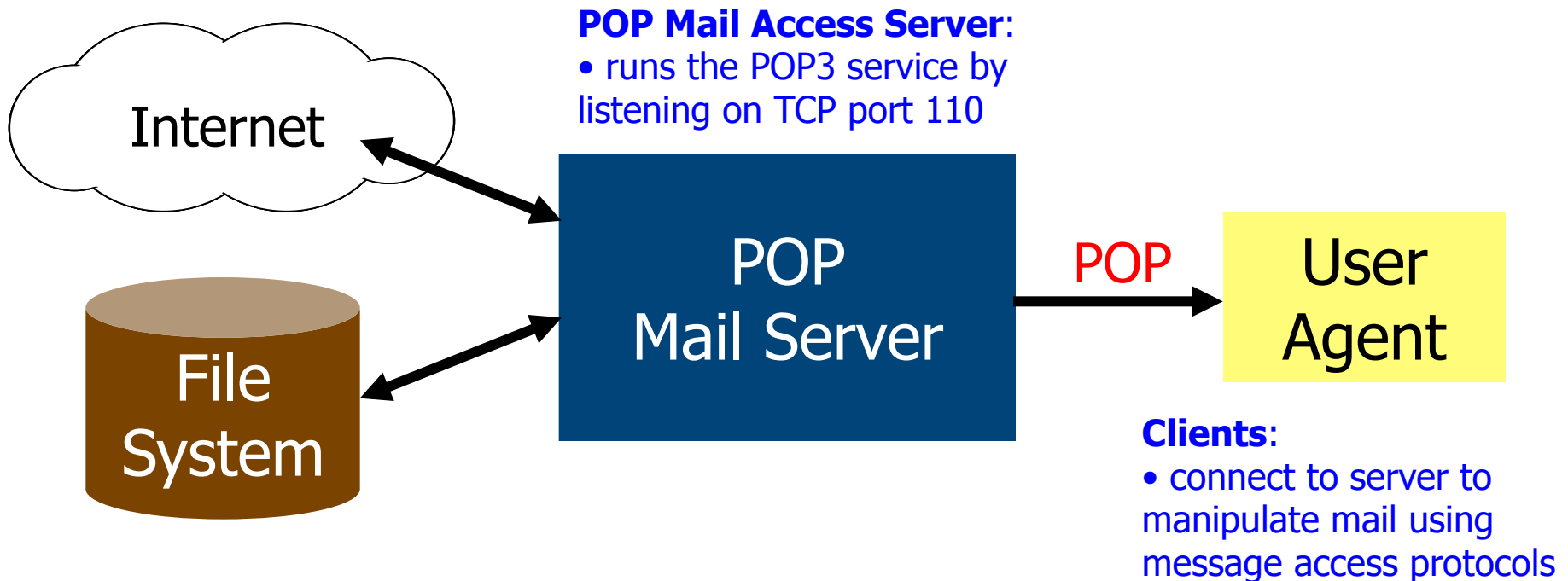


Email(2)

- POP
 - Basic model
 - Basic commands and replies
 - USER, PASS, LIST, RETR, DELE, QUIT, ...
- IMAP
 - Features of IMAP
 - Comparison of POP and IMAP
- Web-based email
 - What are the differences of accessing incoming emails through UAs and web-based manner?
- Message formats
 - RFC 5322: main headers
 - MIME: New headers and main content types
- What are the limitations of SMTP? How is MIME used to offset the limitations of SMTP?

POP/IMAP – Basic Model

- Used to transfer mail from a mail server to a UA





IMAP:

Internet Message Access Protocol

■ Features

- Folders and messages can be stored **either** on the server or on the local computer
- Since folders can remain on server, it is possible to **access** your same mail store even using a dumb terminal character based client like Pine.
- Much **better for mobile users than POP** (since mail remains on the server)
- Can selectively copy messages from the server to the local client based on many criteria



POP vs. IMAP(2)

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is email stored?	User's PC	Server
Where is email read?	Off-line	On-line
Mail Syncing	No	Yes
Direction	One-direction	Bi-directional
Good for mobile users?	No	Yes
Partial message downloads?	No	Yes
Speed	Fast	Low

Web-based Mail: HTTP

- Using web browser, not using UAs
- Can deliver mail message in web page format
- What are the differences of accessing incoming emails through UAs and web-based manner?

北京邮电大学邮件系统



找回密码 在线帮助

 登录入口

用户名:

@bupt.edu.cn

密码:

☒ 记住用户名

☐ SSL安全登录



WWW(1)

- WWW components
 - Client/browser
 - Web server
 - The web access model
- URL
 - Structure
 - Used for different services
- HTML
 - Static vs. dynamic
 - CGI

Structure Of URLs

- A URL consists of three parts:
 - The protocol – for example **http** or **ftp**
 - The DNS name of the host
 - The directory and file name



- Protocol: `http` by default
- Port: `80` by default
- `Index.html`, `index.htm`, `default.htm`, `default.asp` etc. are assumed if no file-name given
- `www.bupt.edu.cn`



HTTP

- Features
 - Application layer protocol for client/server communication
 - Request/response based
 - Stateless, ...
- Transaction
- Main Methods
 - Get, Put, Post, ...
- Performance enhancement of HTTP 1.1
 - Persistent connections
 - Pipelining
 - Enhanced caching options
 - Support for compression
- Cookie: function, four components for cookie supporting, example
- Proxy server, Conditional get



HTTP – HTTP Transaction

Establish connection



Client request



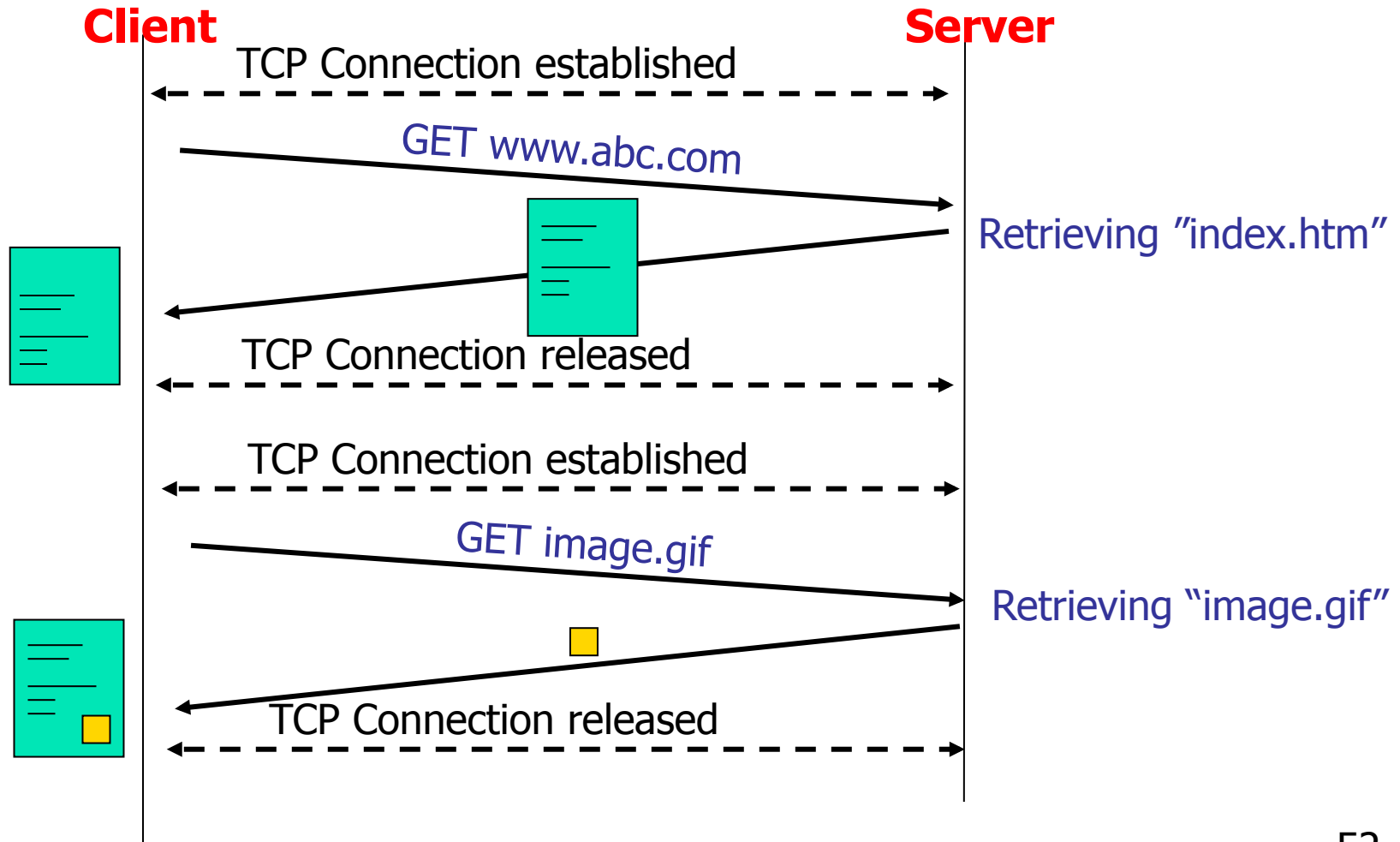
Server response



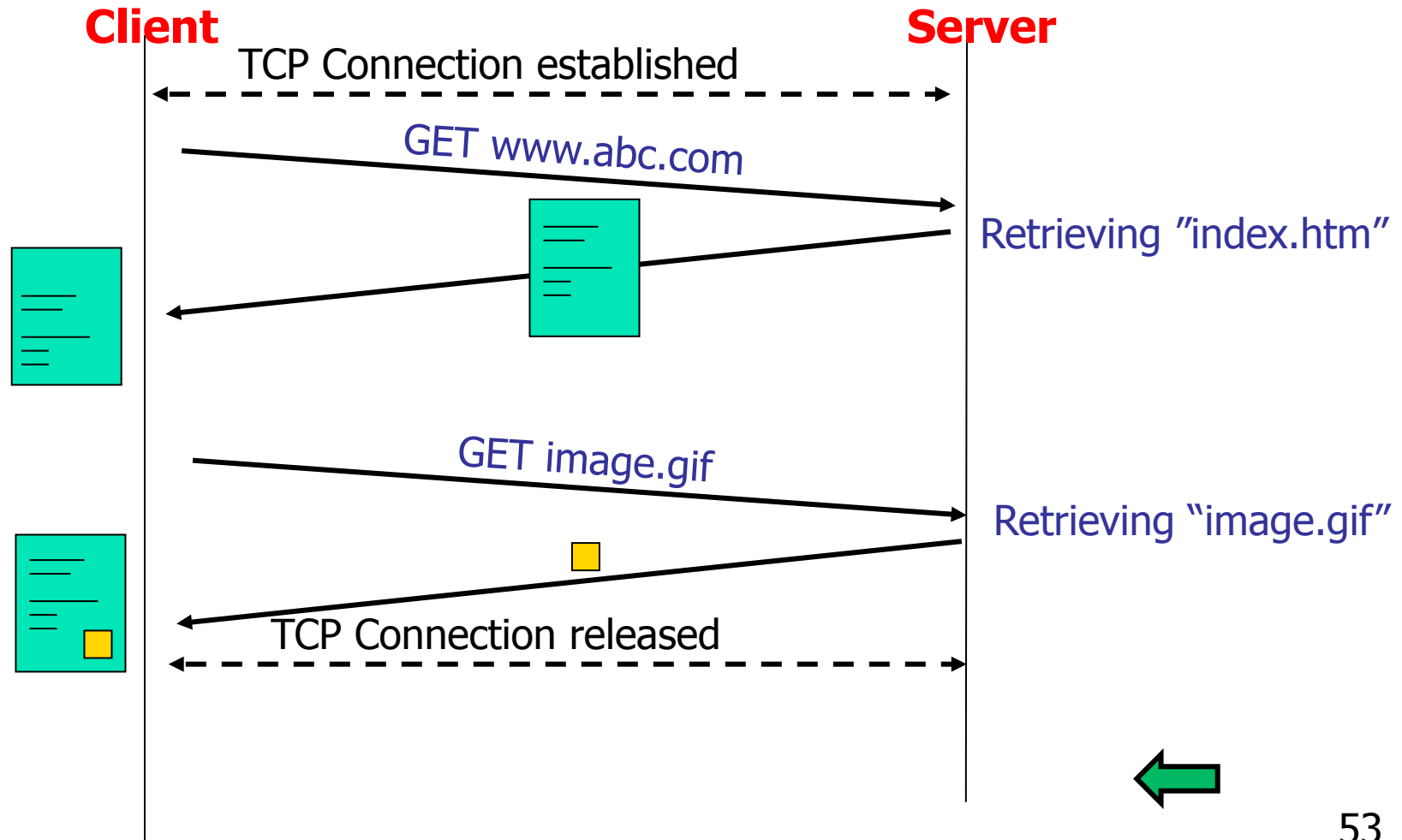
Connection terminated

- TCP connection set up
- uses a port number as application reference
- usually port 80
- HTTP message sent with a request line
- request-line = **method** **URL** **HTTP version**
- server sends HTTP message and optionally requested data
- resp-message = HTTP version **status code** reason-phrase [optional stuff]
- usually the server
- sometimes the client “stops” it
- anything else, whoever notices terminates

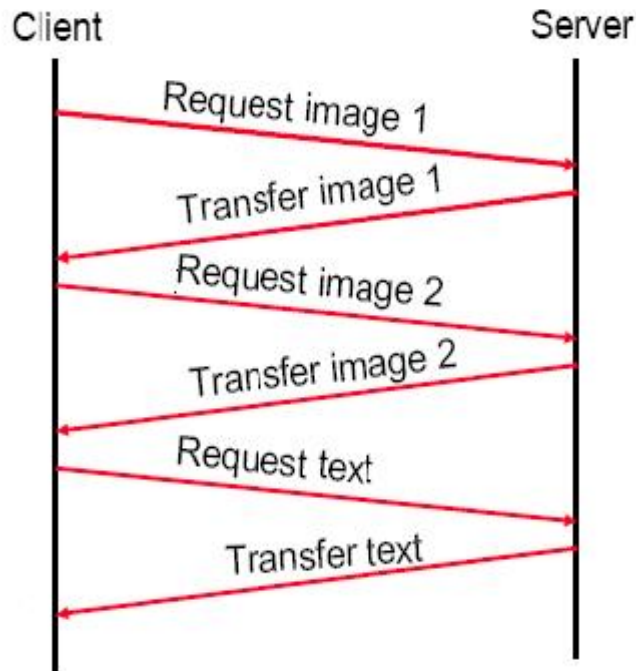
Example of Non-persistent connections



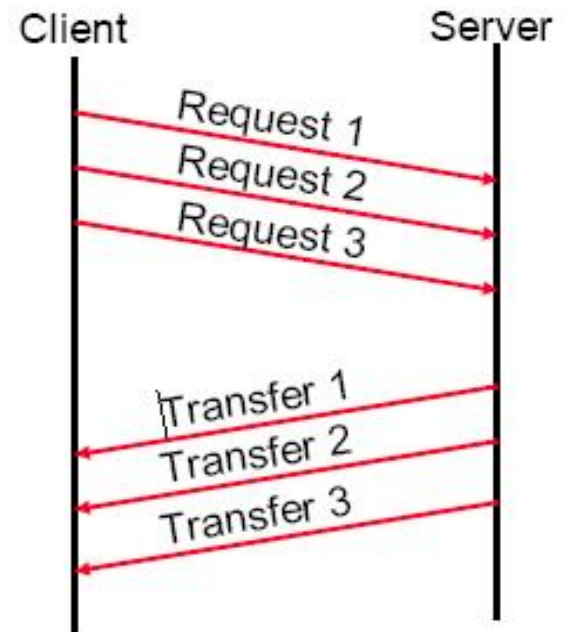
Example of Persistent Connections



Example of Pipelining



Non-pipelining



Pipelining



User-server state: cookies

Many major Web sites use cookies

Four components:

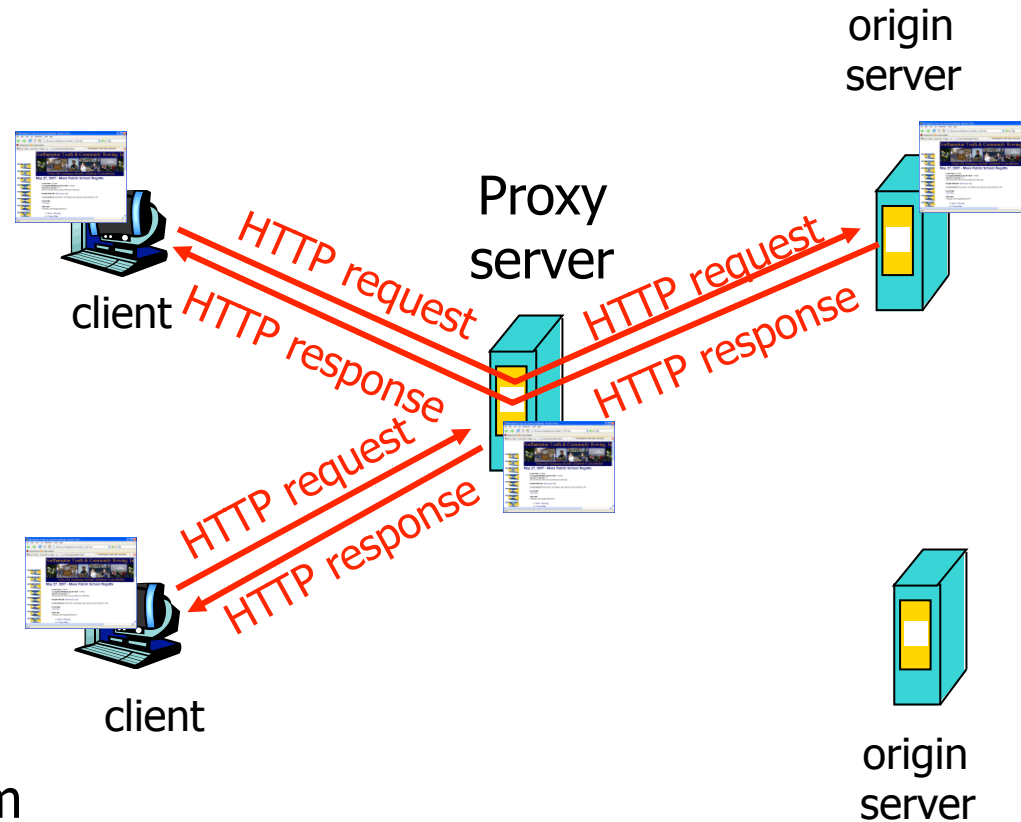
- 1) cookie header line of HTTP *response* message
- 2) cookie header line in HTTP *request* message
- 3) cookie file kept on *user's host*, managed by user's browser
- 4) back-end *database at Web site*

Example:

- Susan always access Internet always from PC
- She visits specific e-commerce site for first time
- when initial HTTP requests arrives at site, site creates:
 - unique ID
 - entry in backend database for ID

Web Caches (Proxy Server)

- Motivation: satisfy client request without involving origin server
- User sets browser: Web accesses via a **proxy server**
- Browser sends all HTTP requests to proxy server
 - If requested file in cache: proxy server returns file
 - else proxy requests file from origin server, then forwards to client



Conditional get

- Server does not send required files if cache has up-to-date cached version
- cache: specify date of cached copy in HTTP request
`If-modified-since: <date>`
- server: response contains no object if cached copy is up-to-date:

`HTTP/1.0 304 Not Modified`

cache

server

HTTP request msg
`If-modified-since:
<date>`

file
not
modified

HTTP response
`HTTP/1.0
304 Not Modified`

HTTP request msg
`If-modified-since:
<date>`

file
modified

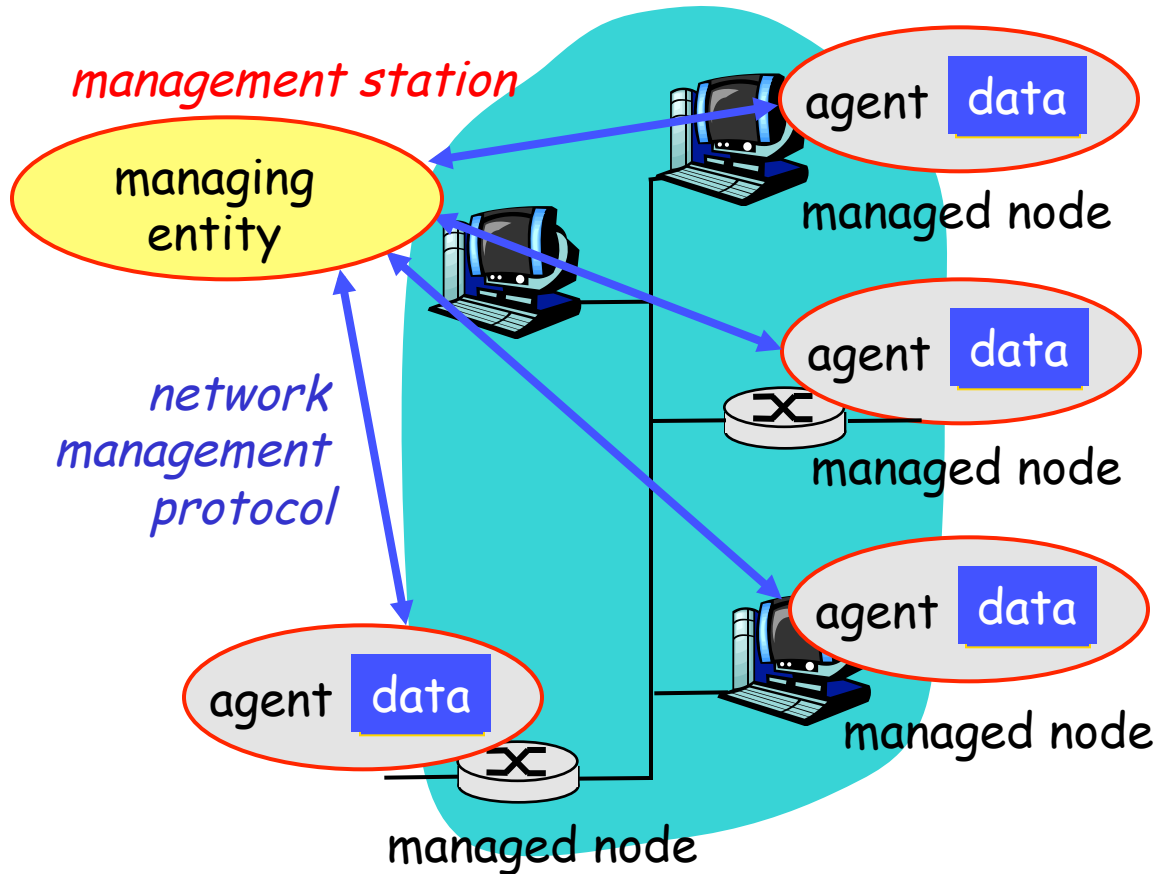
HTTP response
`HTTP/1.0 200 OK
<data>`



SNMP(1)

- Terminologies: SNMP, MIB, SMI, RMON
- Definition, goals and functional areas of network management
 - Functional areas: FCAPS
- Different network management architectures and their pros and cons
 - Centralized
 - Hierarchical
 - Distributed
- SNMP features
 - SNMPv1,v2,v3, RMON1,RMON2
 - SNMP realizes the **F-C-P** functions of network management
- SNMP model and components

SNMP Model



- The SNMP model of a managed network consists of four components:
 - Managed Nodes (**Agent**)
 - Management Stations (**NMS**)
 - Management Information (**MIB**)
 - A Management Protocol (**SNMP**)



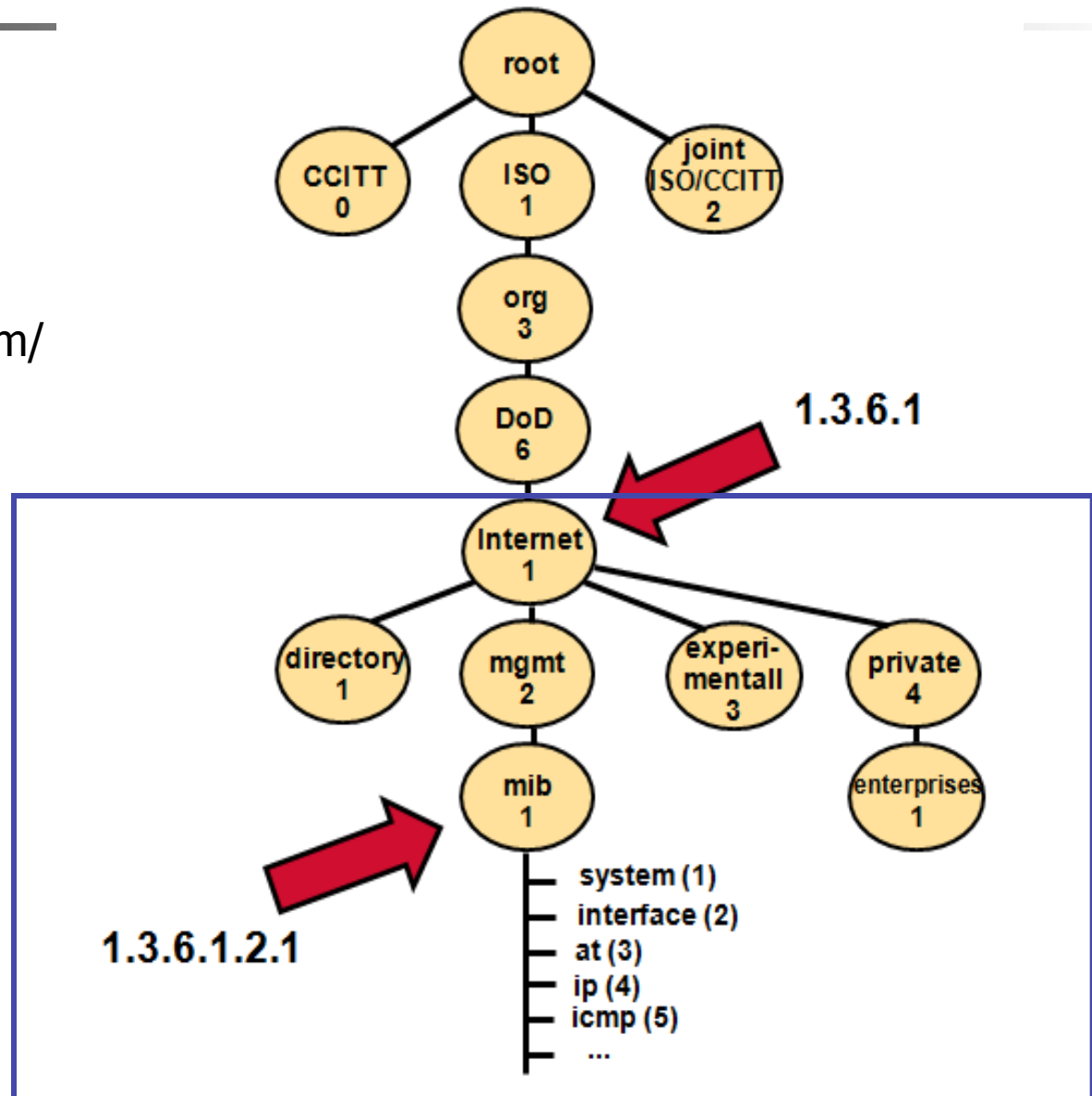
SNMP(2)

- SNMP framework
 - SMI and ASN.1
 - *SMI* defines the rules for describing management information
 - using *ASN.1* for an unambiguous description without inconsistencies
 - MIB hierarchy naming
 - SNMP protocol: traps/polling, SNMP commands
- RMON: Purpose, probe

MIB – ISO Object Identifier Tree

Check out:

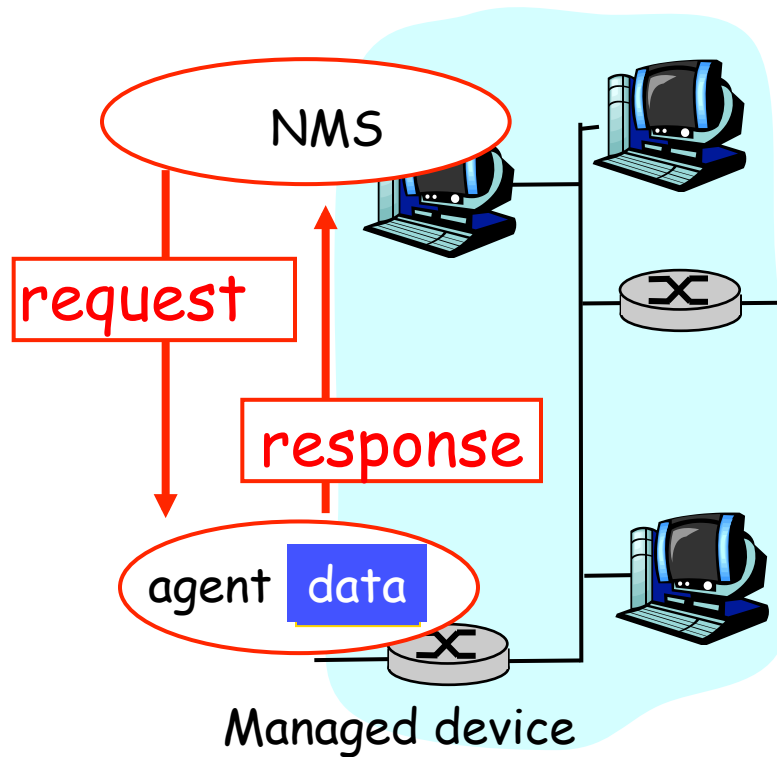
<http://www.oid-info.com/>



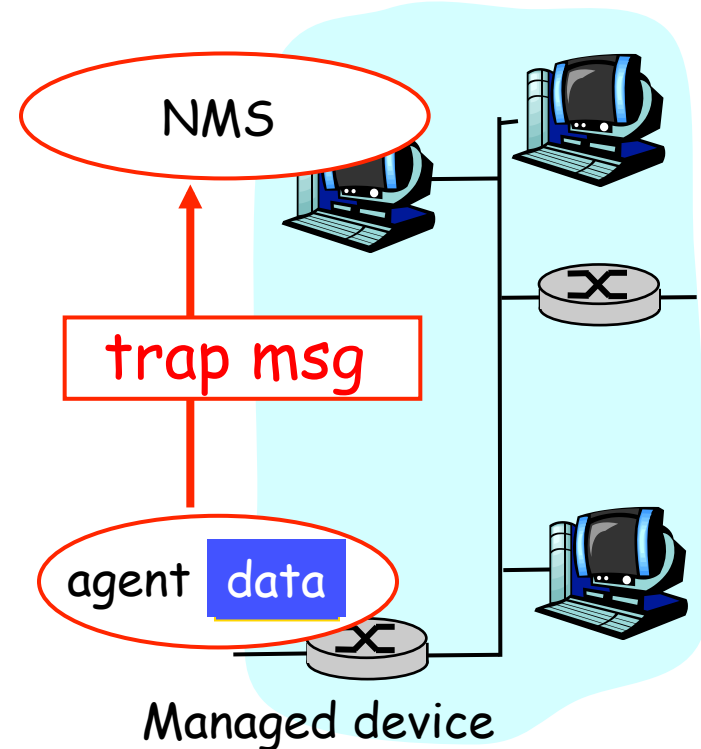
This subtree is
the Internet SMI

SNMP Traps / Polling

- Two ways to deliver MIB information, commands



Polling mode



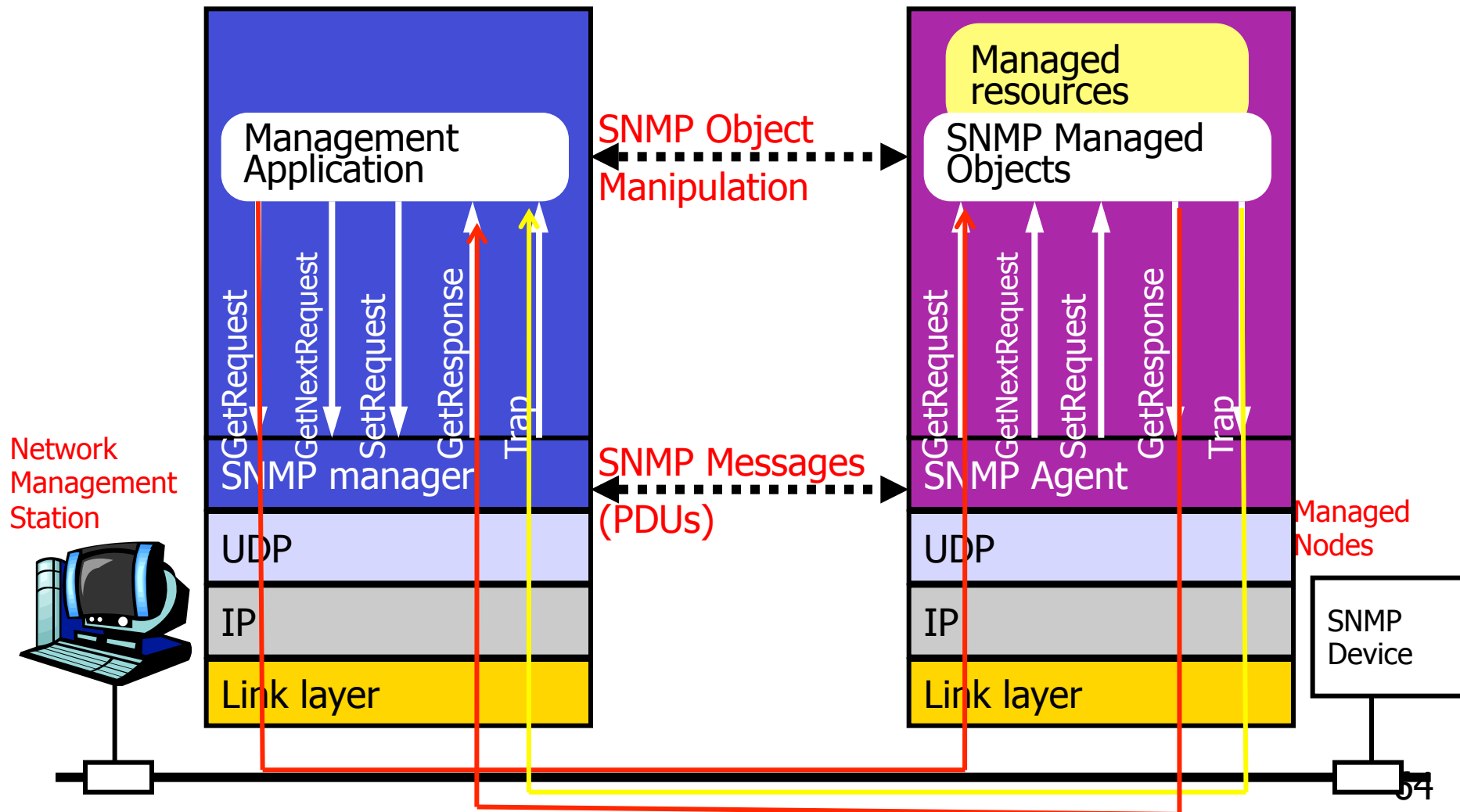
trap mode



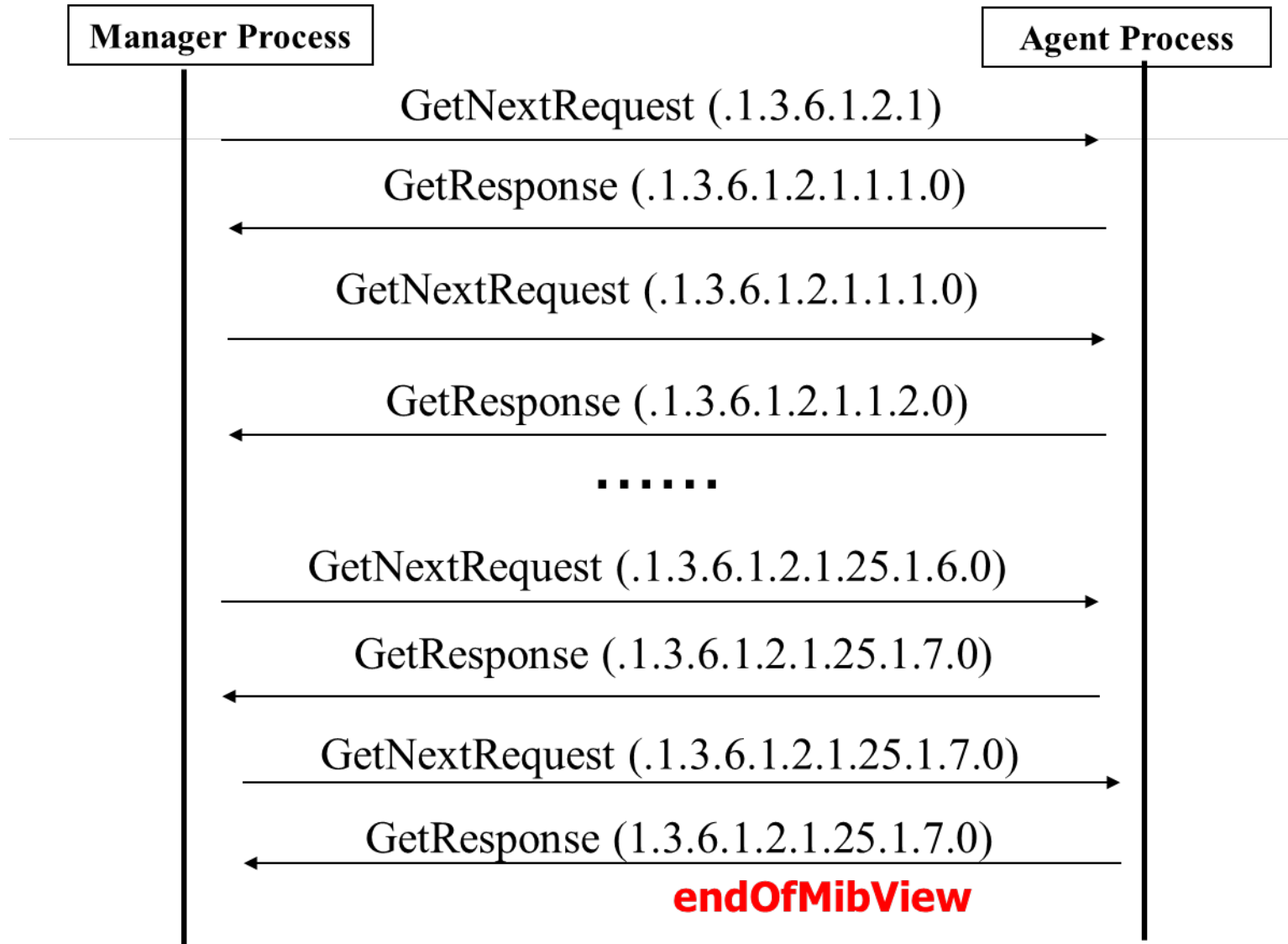
SNMP Commands

Command	Description	Version
GetRequest	NMS-to-Agent: get data (instance)	SNMPv1
GetNextRequest	NMS-to-Agent: get data (next in list)	SNMPv1
GetBulkRequest	NMS-to-Agent: get data (block)	SNMPv2
InformRequest	NMS-to-NMS: MIB information exchange	SNMPv2
SetRequest	NMS-to-Agent: set MIB value	SNMPv1
GetResponse	Agent-to-NMS: value, response to request	SNMPv1
Trap	Agent-to-NMS: report exceptional event to NMS	SNMPv1

SNMPv1 Commands



Example of SNMP Commands (GetNextRequest)





RMON

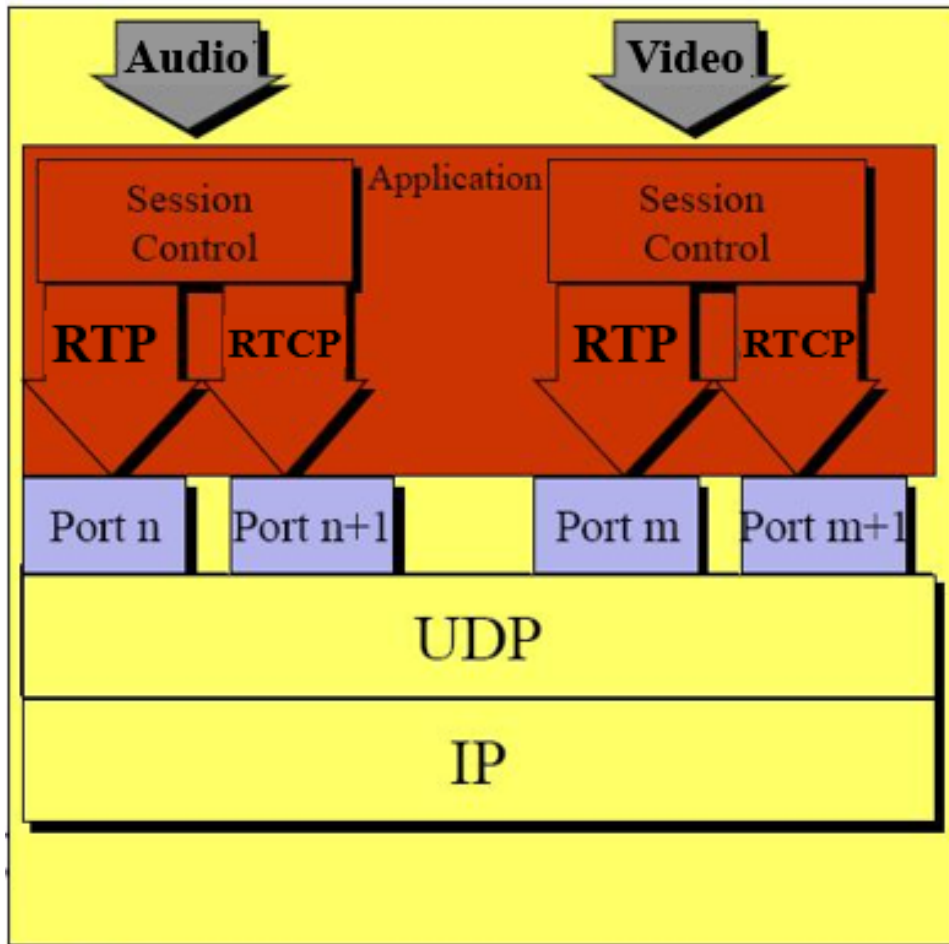
- RMON= Remote MONitoring
- Extensions to SNMP provide comprehensive network monitoring capabilities
- RMON uses remote network monitoring devices known as probes. The RMON specification defines a set of statistics and functions that can be exchanged between RMON-compliant console managers and probes
- RMON is standardized to only operate on Ethernet segments (LAN).
- RMON specifically defines the information that any network monitoring system will be able to provide as part of the MIB.
- RMON1 provides monitoring capability at data link layer in OSI model.
- RMON2 provides monitoring capability above data link layer in OSI model.



Real-time Services(1)

- Real-time services
 - Isochronous services and QoS requirements
 - isochronous pertains to processes that require timing coordination to be successful, such as voice and digital video transmission
 - Jitter compensation via playback buffer
- RTP/RTCP
 - Functions of RTP/RTCP, relationship between RTP and RTCP
 - Definitions: end system, translator, mixer, monitor
 - RTP features, SSRC vs, CSRC
 - RTCP features, relationship of RTCP port number with RTP port number

RTP and RTCP



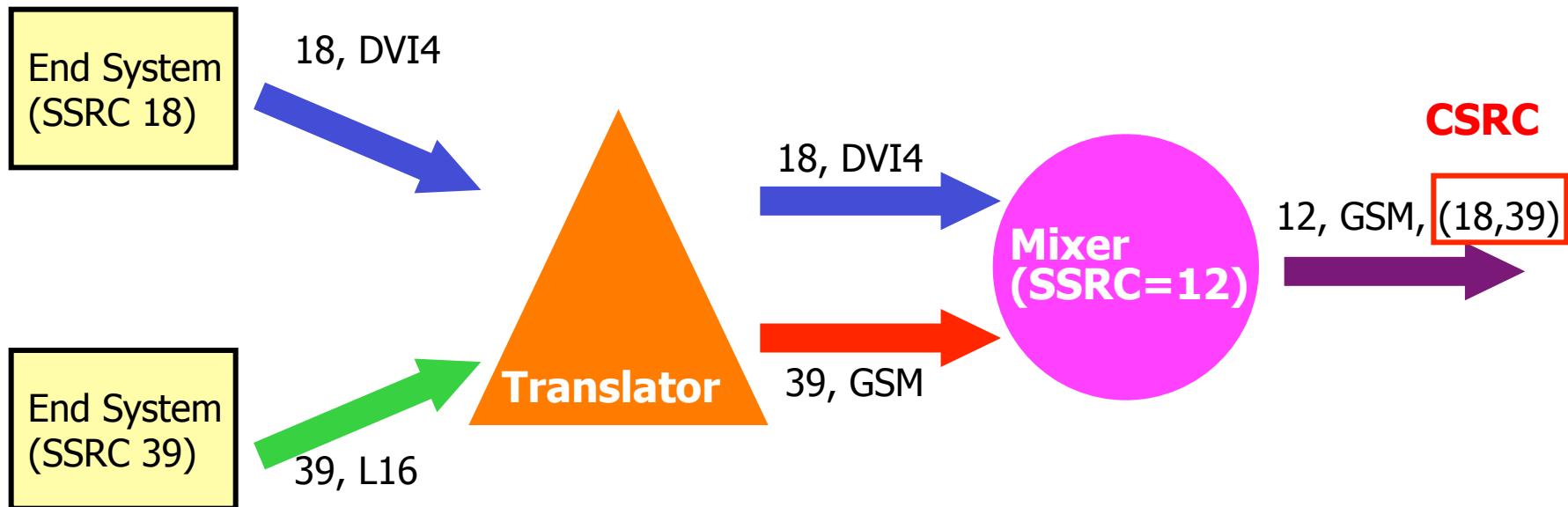
RTP

- For data
- Important fields in header
 - sequence number
 - timestamp
 - synchronization ID

RTCP

- For QoS monitoring and control
- Provides feedback on the quality of the data distribution

Mixer and Translator



SSRC: Synchronization SouRCe Identifier

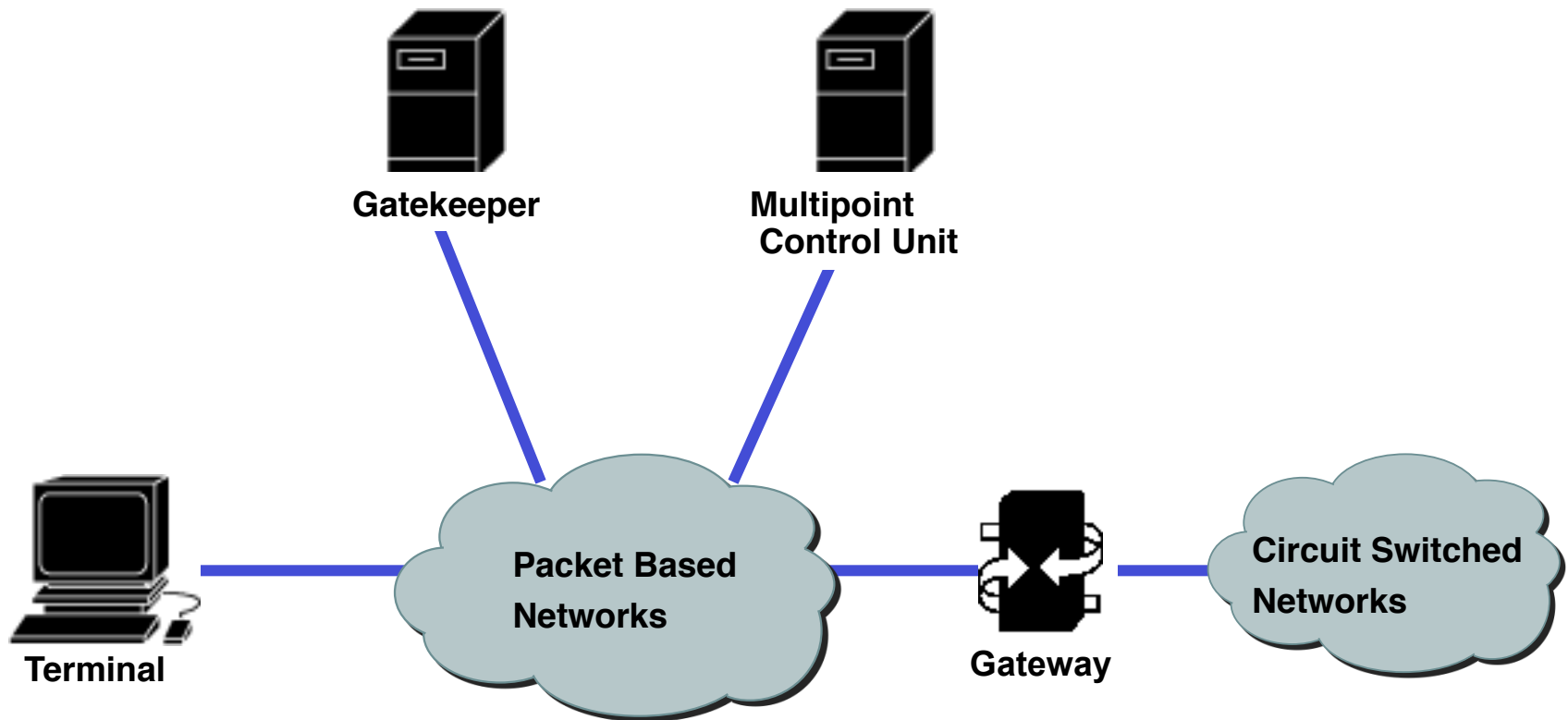
CSRC: Contributing SouRCe Identifier



Real-time Services(2)

- Multimedia signaling protocols
 - H.323
 - Function: Describes terminals and other entities that provide multimedia communications services over Packet Based Networks (PBN) which may not provide a guaranteed Quality of Service.
 - H.323 protocol stack
 - components
 - SIP
 - Function: An application layer signaling protocol that defines initiation, modification and termination of interactive, multimedia communication sessions between users
 - Architecture and components
 - SIP messages, SIP header, SIP URL
 - SIP Messages: INVITE, ACK, BYE,...
 - SIP Header: MIME
 - SIP URL: user@host

H.323 – Components



SIP – Architecture

