Review

BUPT/QMUL

2021-06-03

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? Typical protocols of network layer, transport layer and application layer
  - 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 (bps, Bps), throughput, delay (latency), jitter, error rate (PLR, BER, FER, PER)
- Network types according to the switching function in the network
  - Circuit switching network
  - Message switching network
  - Packet 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 (range, channel access technology, examples of them)
- Network types according to the user of the network
  - public network and private network
- Layered architecture, relationship between them
  - 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()
  - file descriptor: what is a file descriptor?
    - Special file descriptors: 0,1,2
    - Related system calls and their functions:
      - open(), read(), write(), lseek()

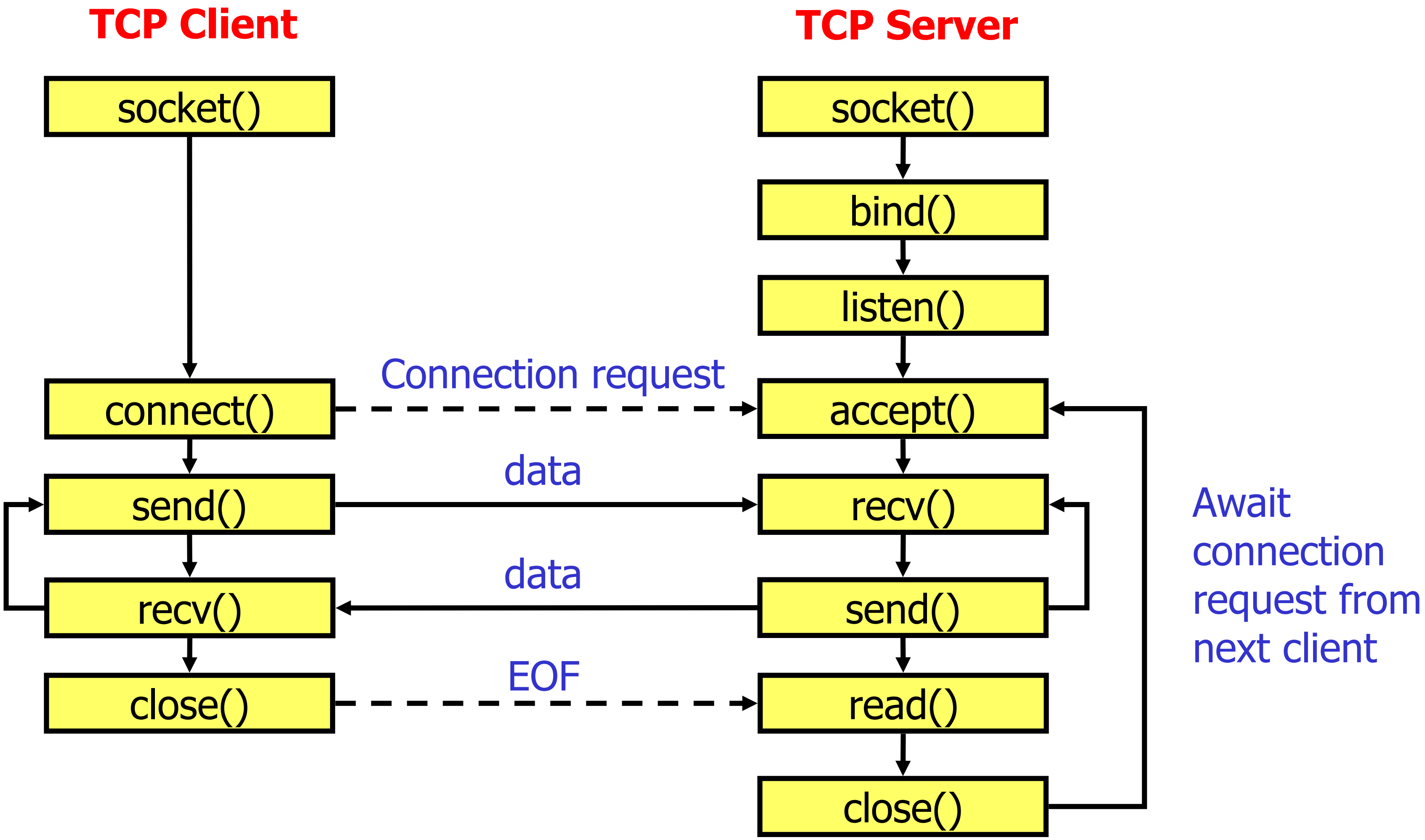
Network Programming Basics (2)

- IP Addresses
  - IP address structure
  - Big-endian and little-endian, HBO and NBO
  - byte order conversion functions: htonl(), ntohl(), htons(), ntohs()
- DNS
  - Function of 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?
    - Socket address
    - socket address pair (client-IP, client-port, server-IP, server-port)
  - Well-known port numbers for typical applications
    - DHCP, DNS, TELNET, TFTP, FTP, SMTP, POP3, HTTP, SNMP

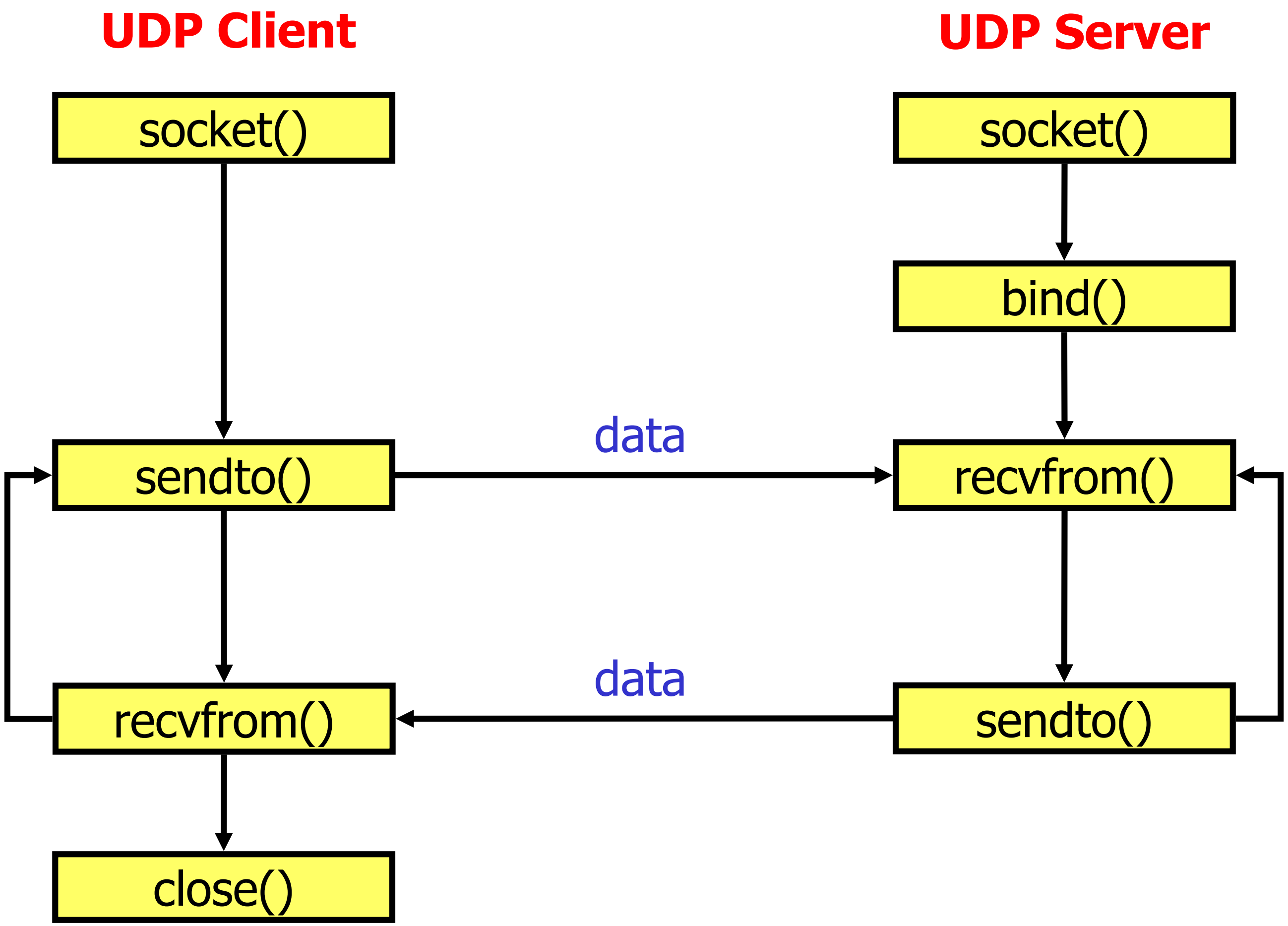
Network Programming Basics (3)

- Sockets interface
  - What is a socket? What is the sockets descriptor?
  - Internet-specific socket address and Generic socket address
    - struct sockaddr_in vs. struct sockaddr
    - how do we use them in the system calls such as connect(), bind(), and accept()?
  - Different sockets interface types
    - SOCK_STREAM, SOCK_DGAM, 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



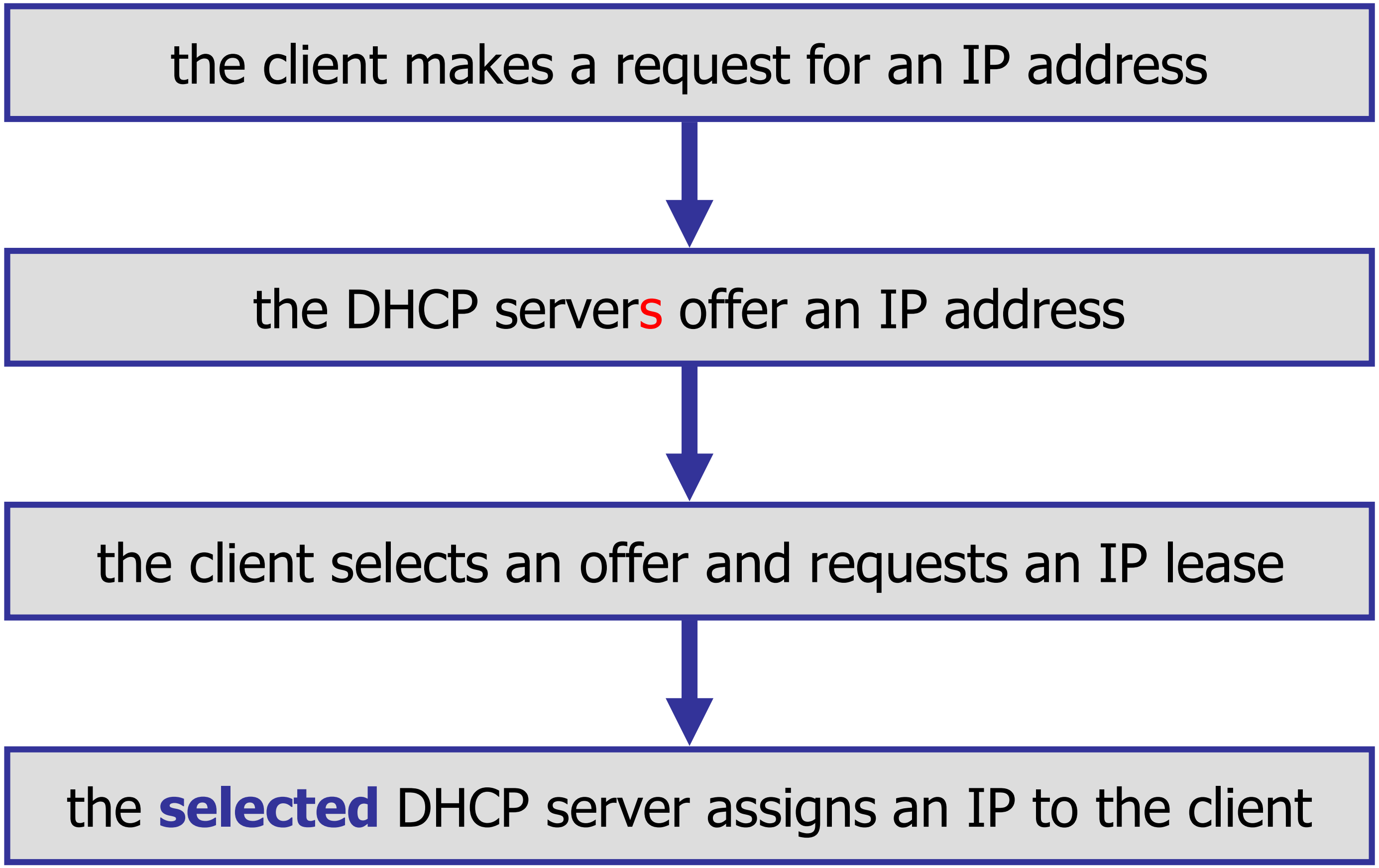
Overview of UDP-based sockets API



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 permission to use a particular IP address
  - 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



the client makes a request for an IP address

the DHCP servers offer an IP address

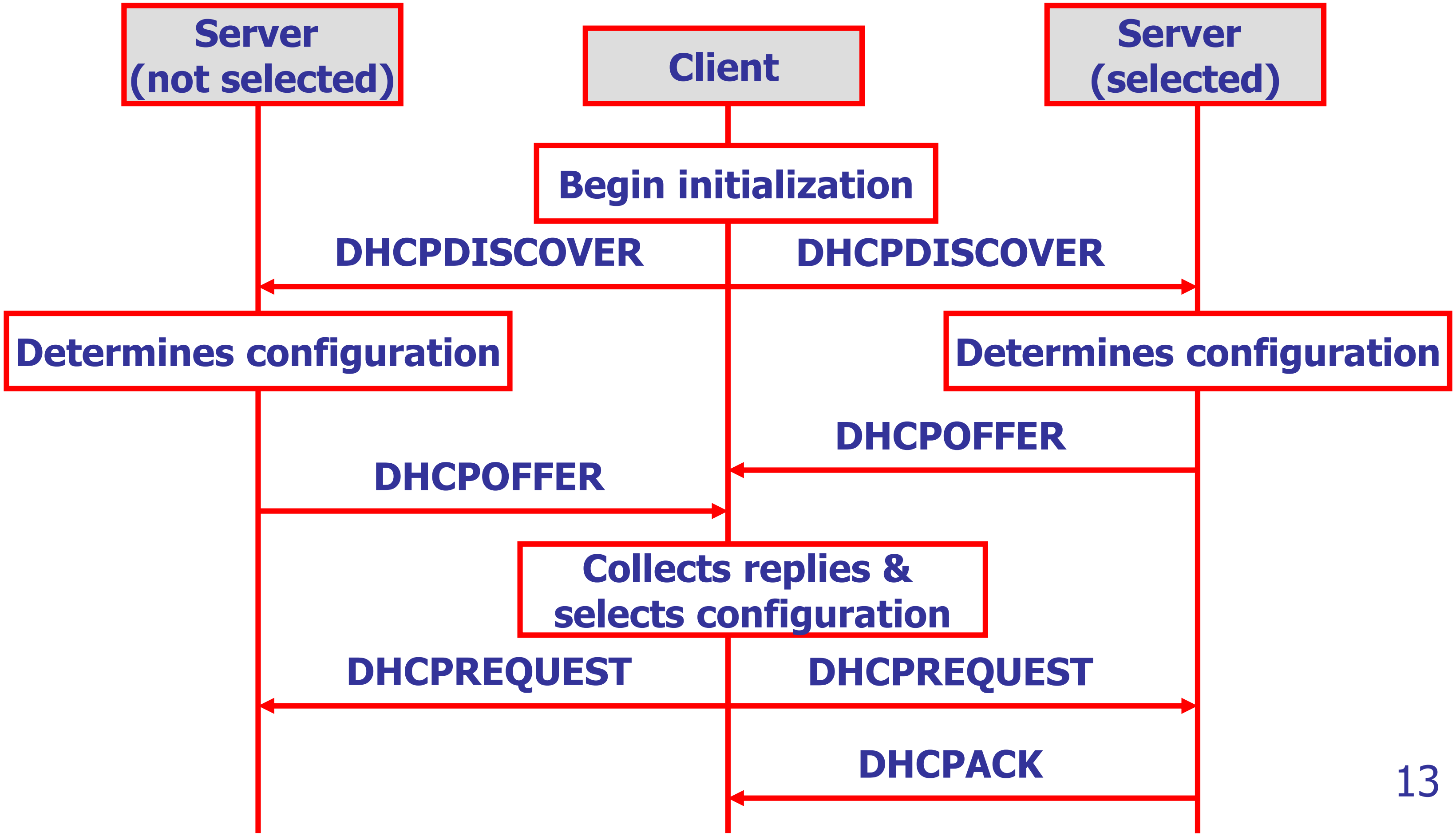
the client selects an offer and requests an IP lease

the **selected** DHCP server assigns an IP to the client

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


Server (not selected)

Client

Server (selected)

Begin initialization

DHCPDISCOVER

DHCPDISCOVER

Determines configuration

Determines configuration

DHCPOFFER

DHCPOFFER

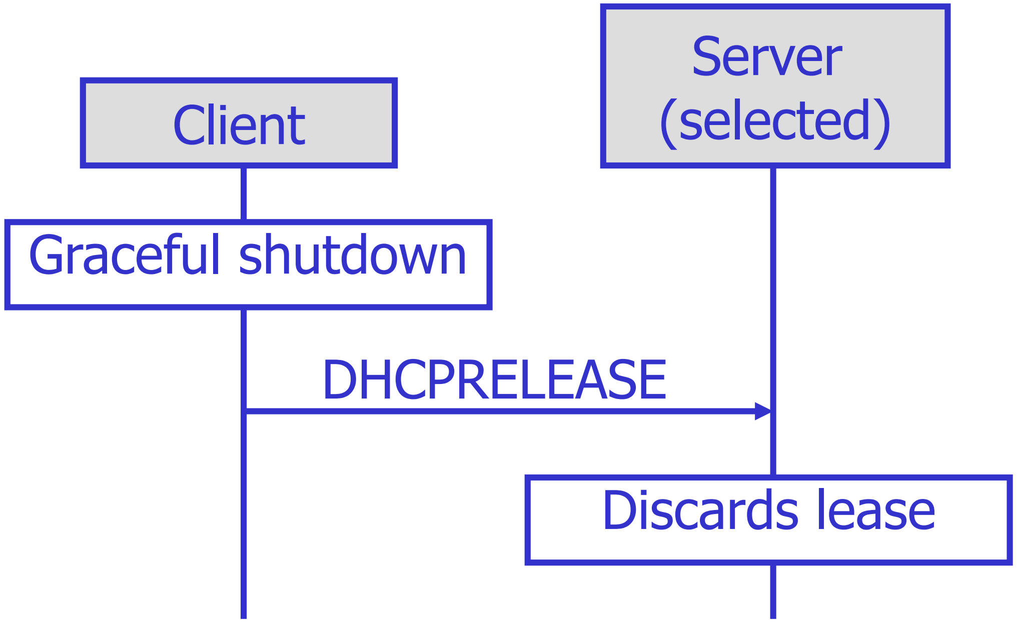
Collects replies & selects configuration

DHCPREQUEST

DHCPREQUEST

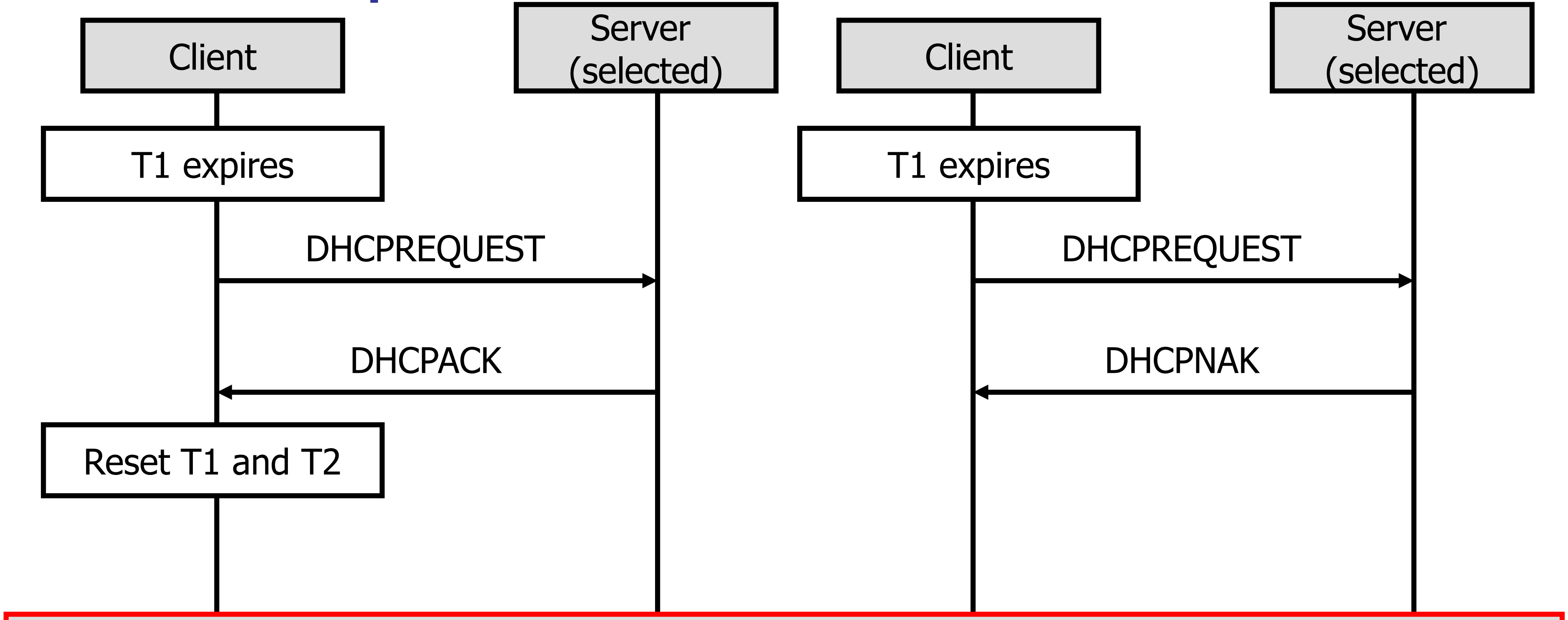
DHCPACK

Early Lease Termination: Procedure



Lease Renewal: Procedure (1)

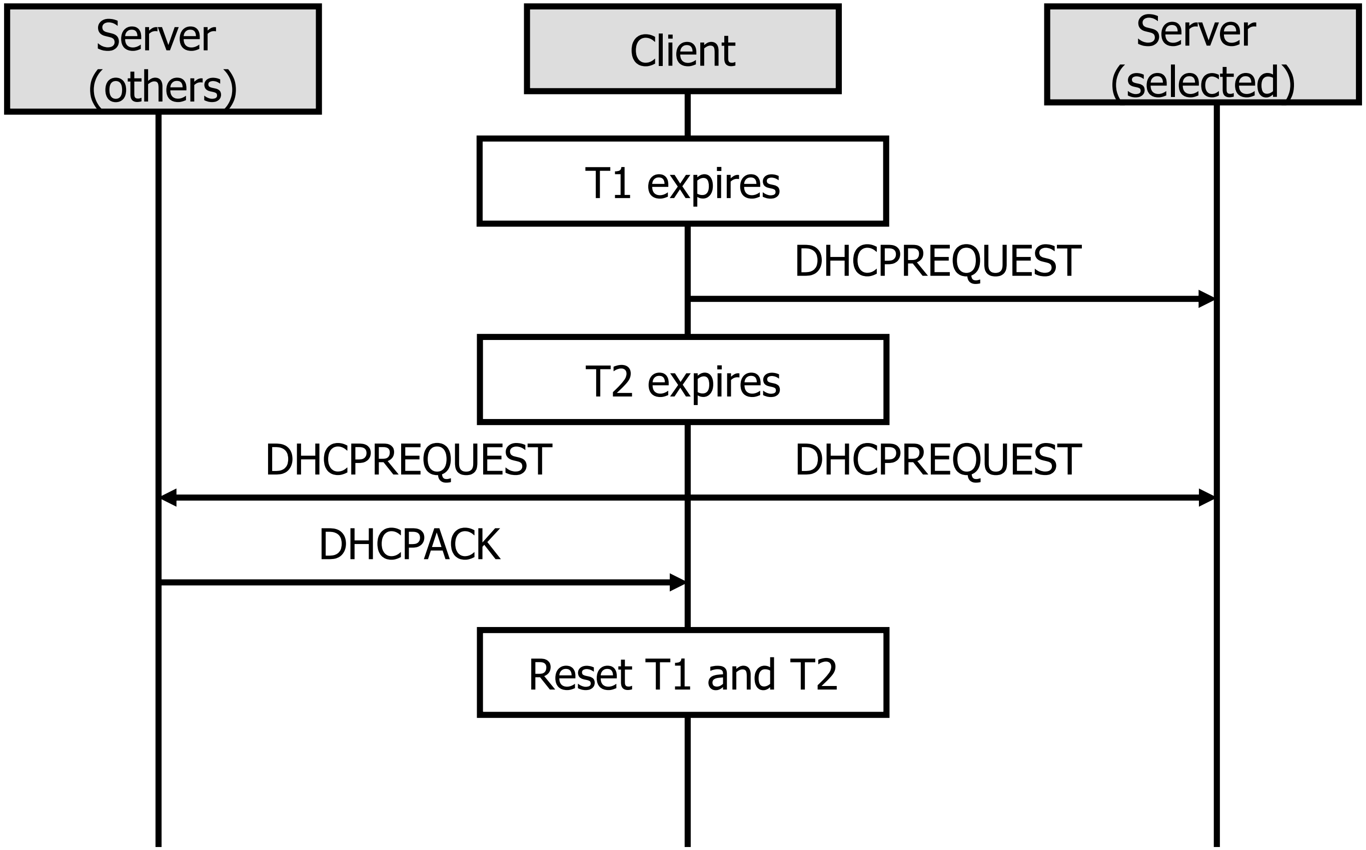
- **T1 expires**



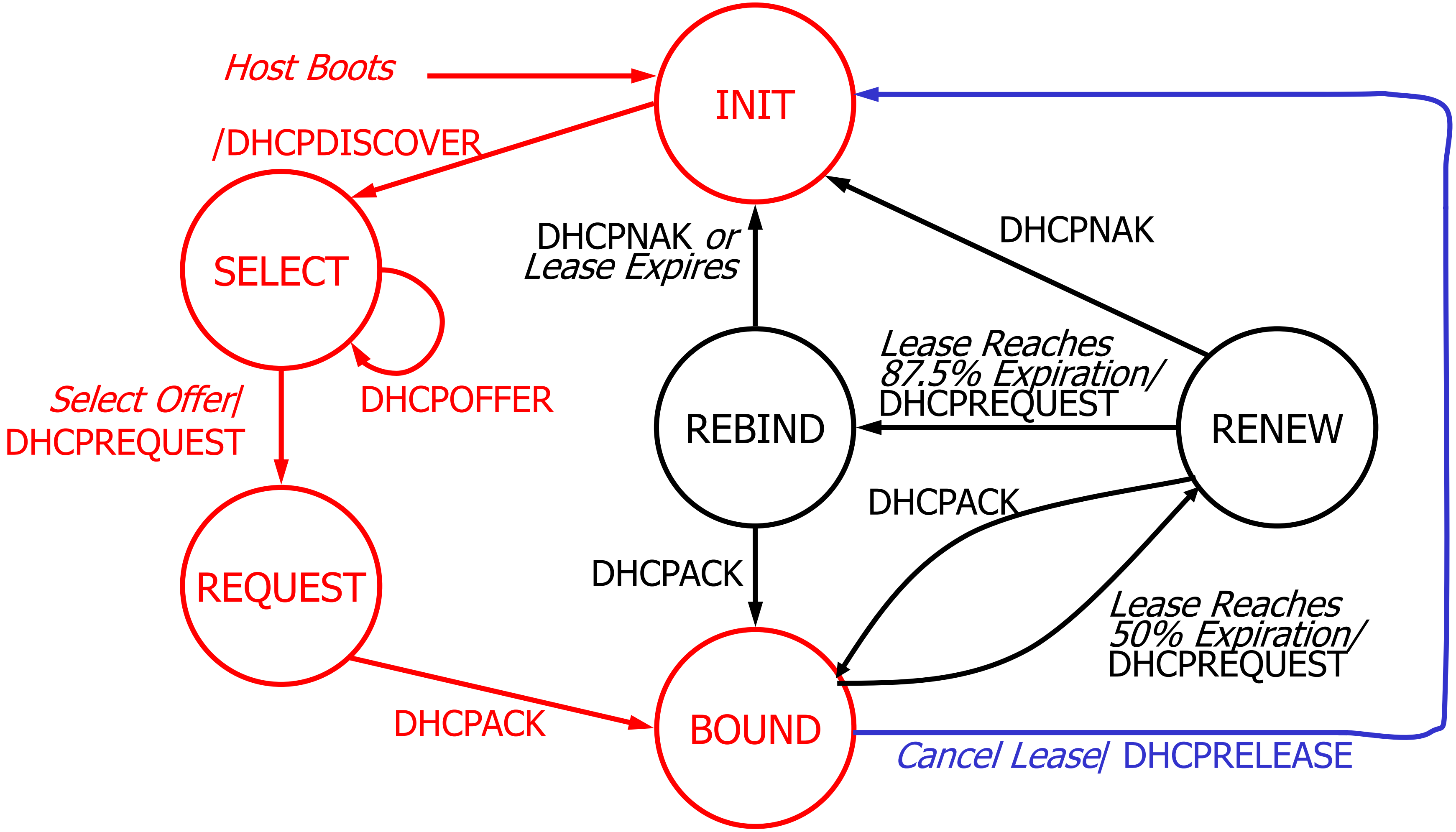
- T1: the time at which the client enters the RENEW state and attempts to contact the server that originally issued the client's network address
  - *0.5 * duration_of_lease*
- T2: the time at which the client enters the REBIND state and attempts to contact any server.
  - *0.875 * duration_of_lease*

Lease Renewal: Procedure (2)

- **Both T1 and T2 expire, address rebound**



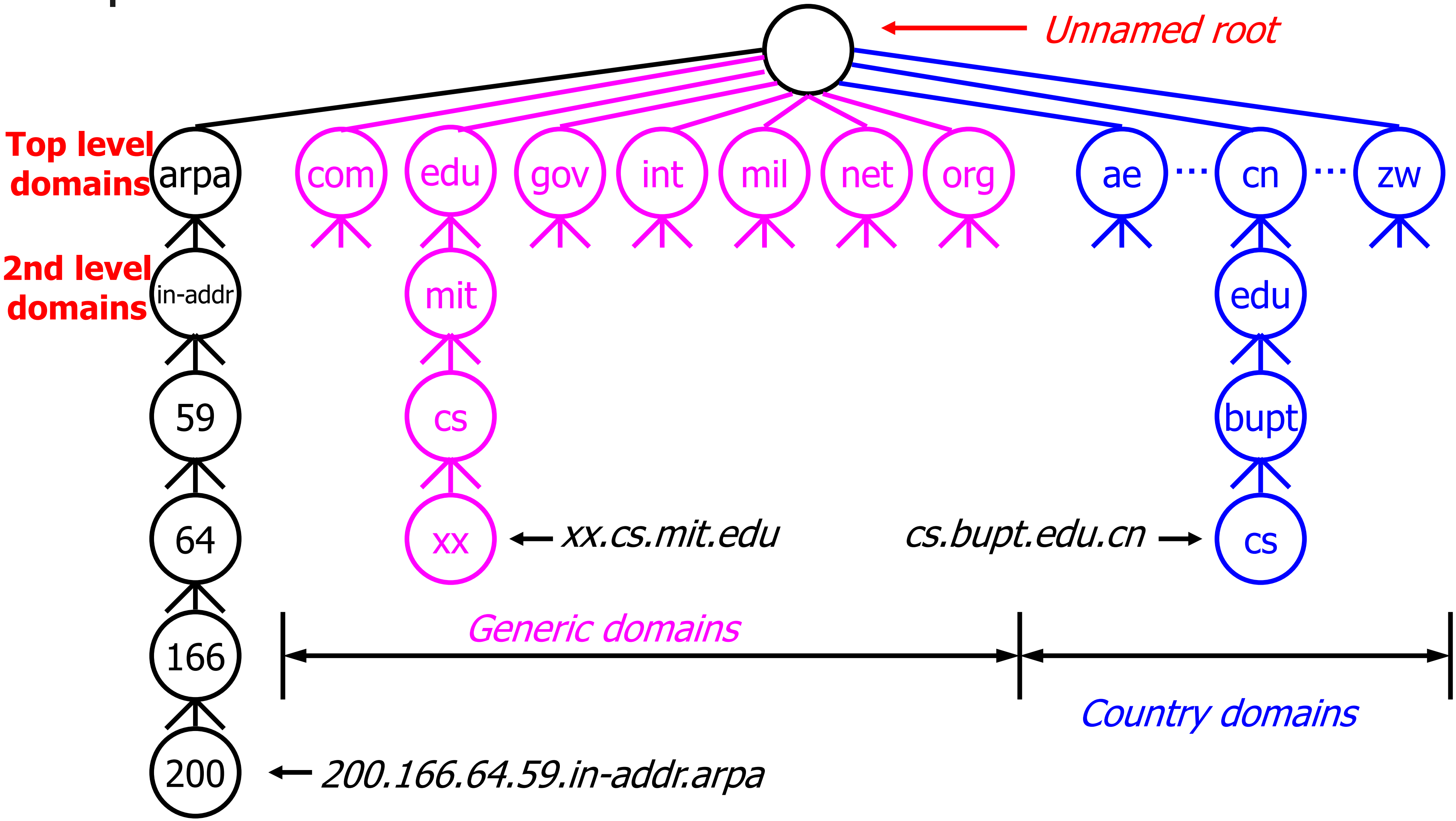
STD of DHCP Client



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

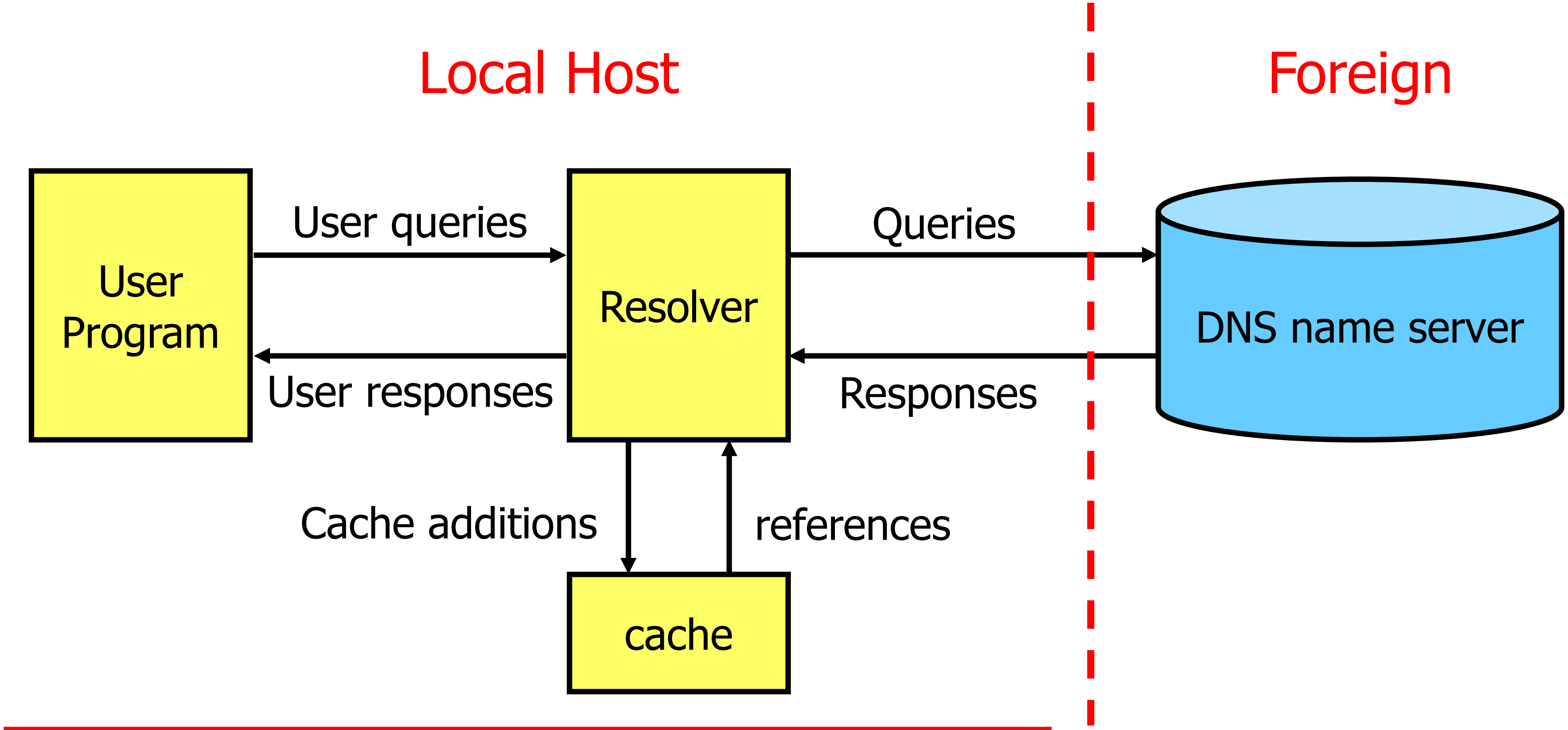
Hierarchical structure



DNS(2)

- Communication model between user program, resolver and name server. 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

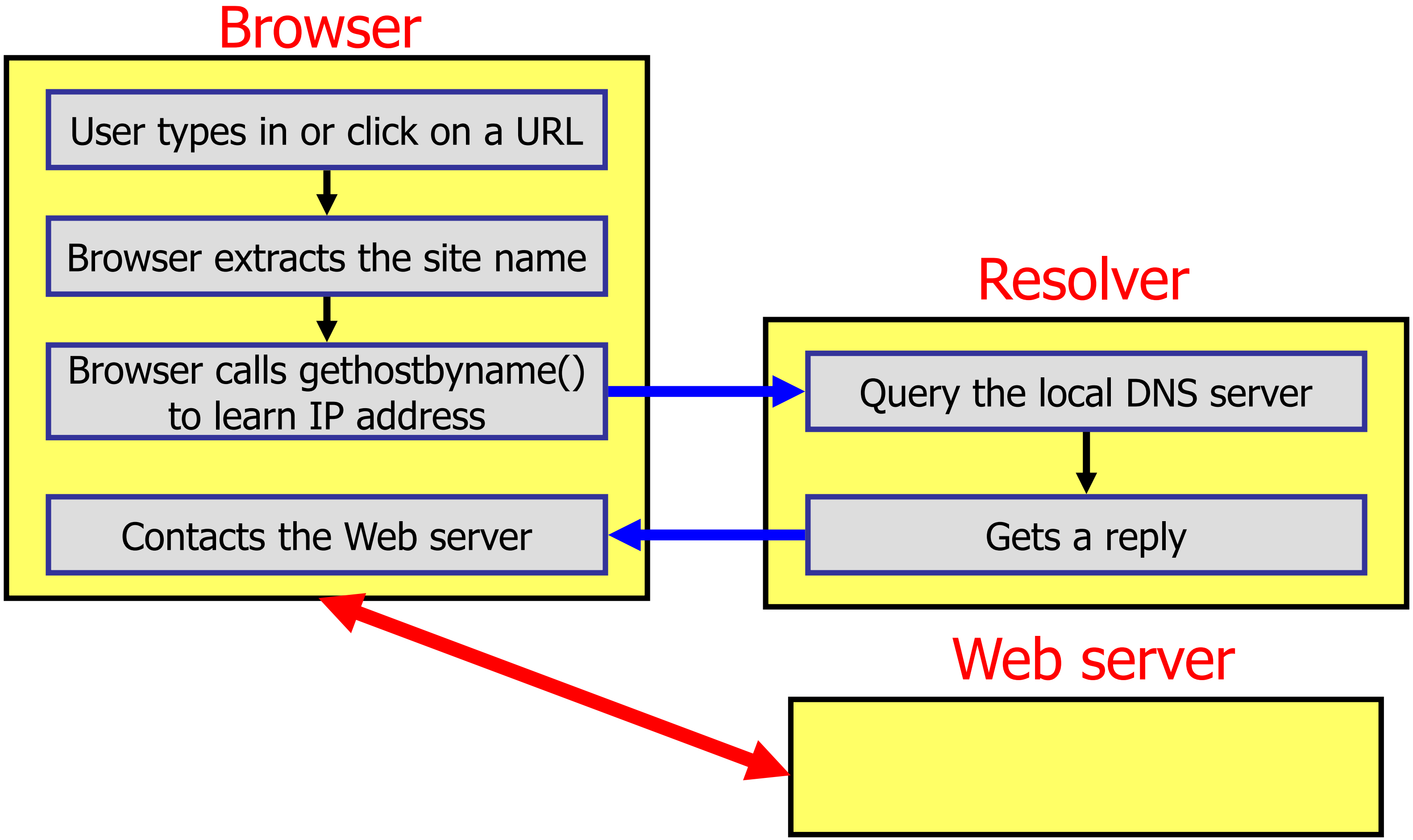


Resolvers are normally implemented in system calls.

E.g. gethostbyname(), gethostbyaddr()

How does DNS work together with the user programs?

- DNS working together with HTTP application



Browser

User types in or click on a URL

Browser extracts the site name

Browser calls gethostbyname() to learn IP address

Contacts the Web server

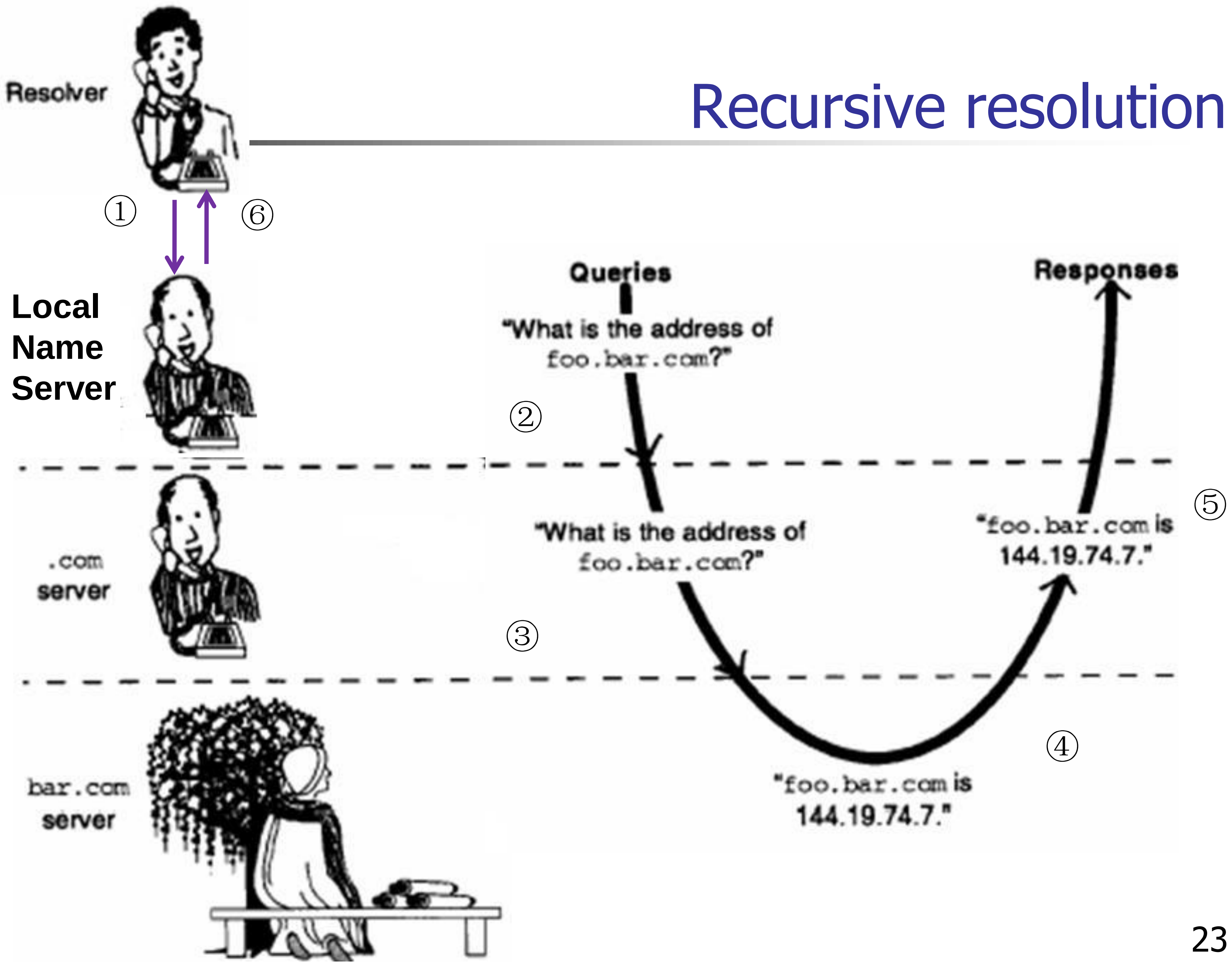
Resolver

Query the local DNS server

Gets a reply

Web server

Recursive resolution



Resolver

①

⑥

Local Name Server

.com server

bar.com server

Queries

"What is the address of foo.bar.com?"

②

"What is the address of foo.bar.com?"

③

"foo.bar.com is 144.19.74.7."

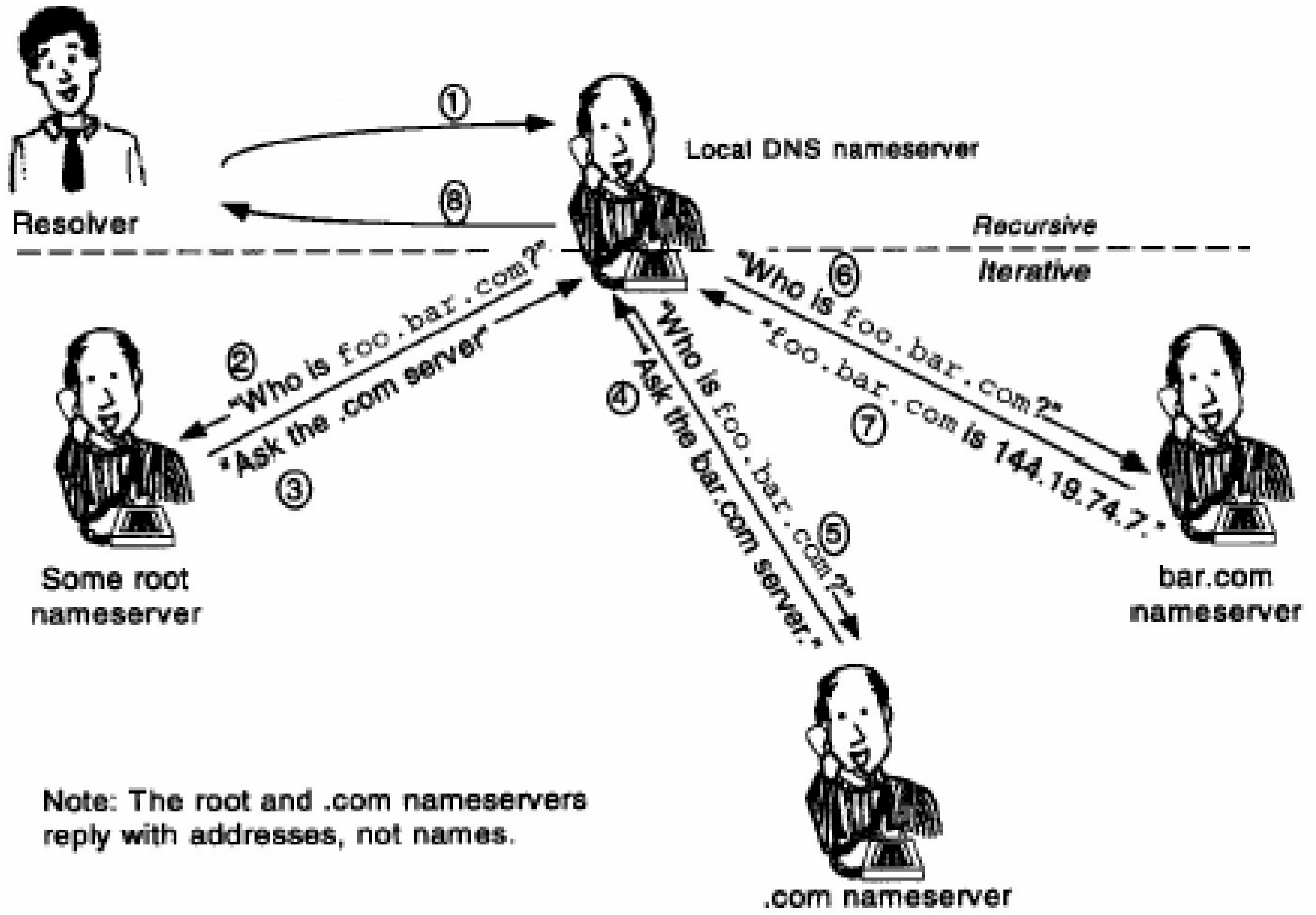
④

Responses

"foo.bar.com is 144.19.74.7."

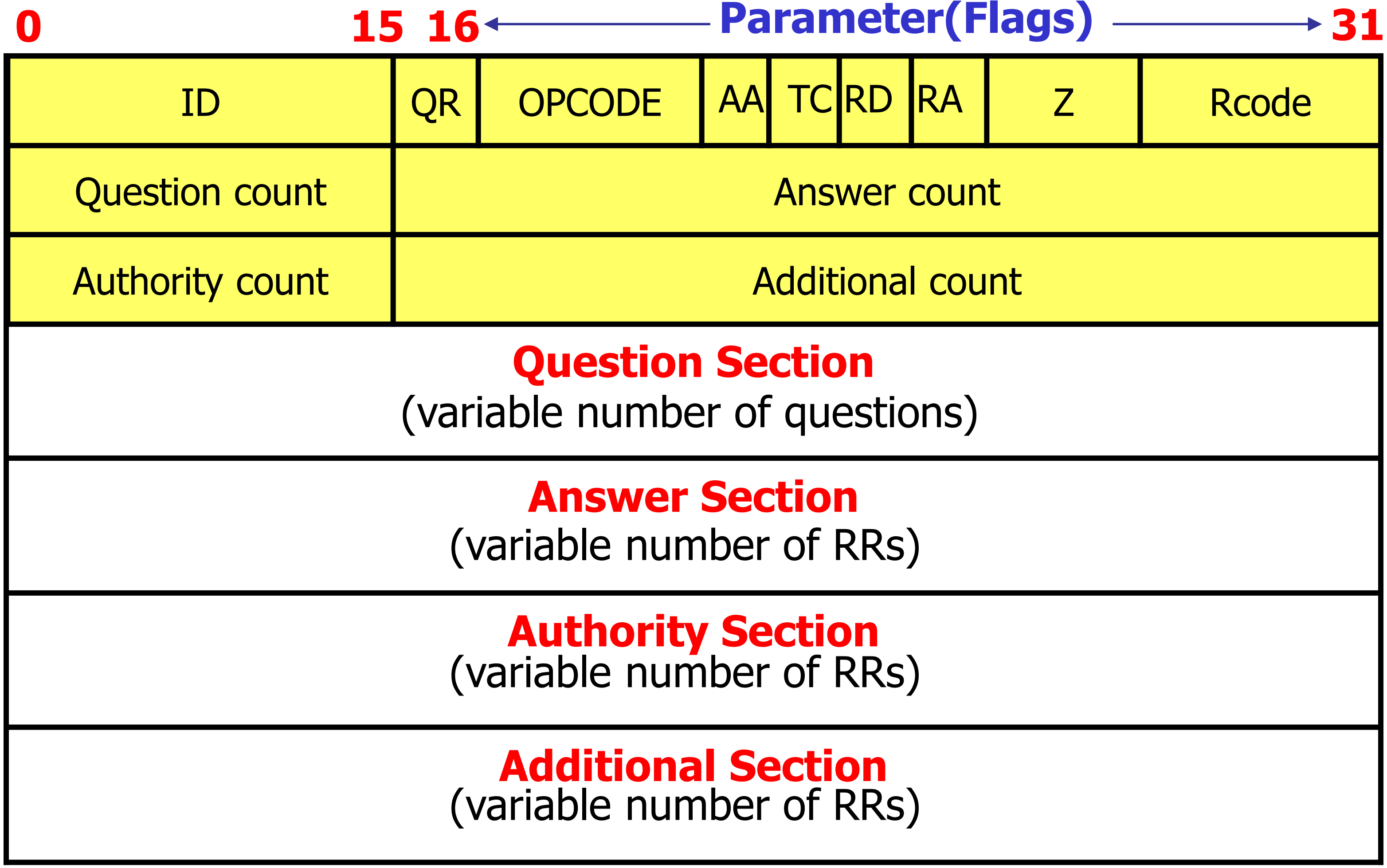
⑤

Iterative resolution



DNS Message Format(from RFC 1035)

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



0 15 16 ← **Parameter(Flags)** → 31

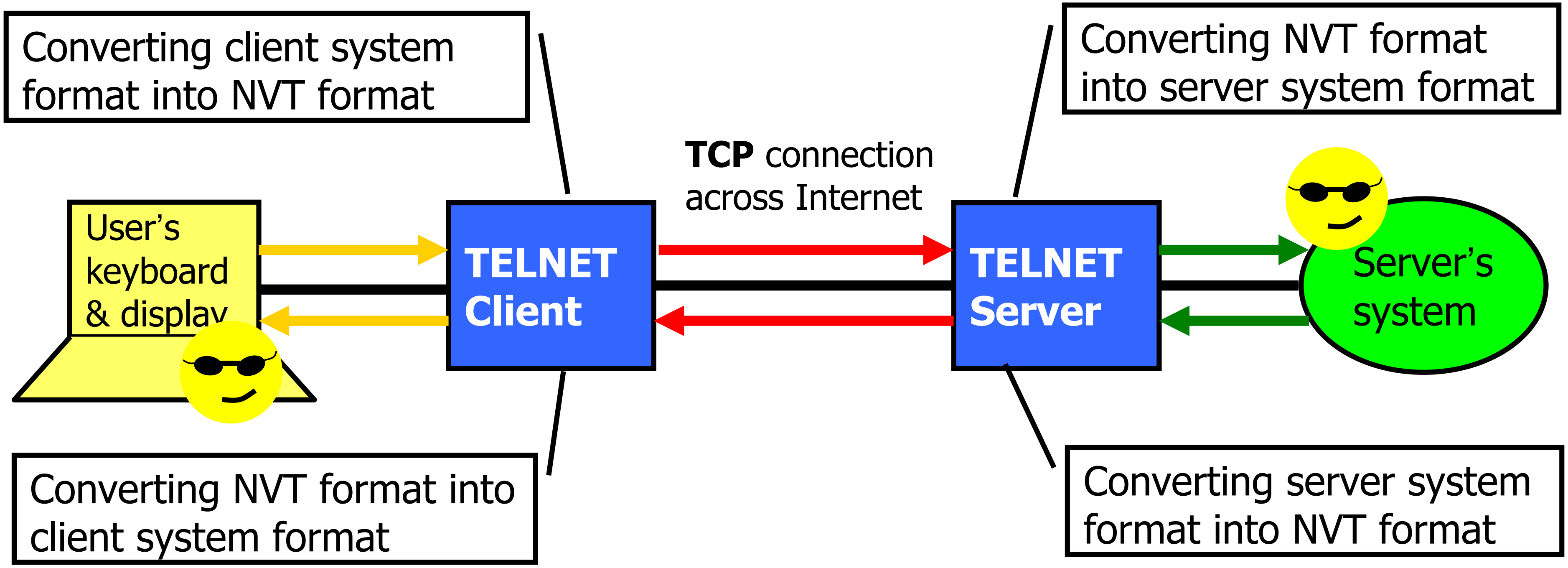
<table>
<tr><td>ID</td><td>QR</td><td>OPCODE</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td>Z</td><td>Rcode</td></tr>
<tr><td>Question count</td><td colspan="8">Answer count</td></tr>
<tr><td>Authority count</td><td colspan="8">Additional count</td></tr>
<tr><td colspan="9"><b>Question Section</b>
(variable number of questions)</td></tr>
<tr><td colspan="9"><b>Answer Section</b>
(variable number of RRs)</td></tr>
<tr><td colspan="9"><b>Authority Section</b>
(variable number of RRs)</td></tr>
<tr><td colspan="9"><b>Additional Section</b>
(variable number of RRs)</td></tr>
</table>

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



Converting client system format into NVT format

Converting NVT format into client system format

Converting NVT format into server system format

Converting server system format into NVT format

🟨 Client System format

🟩 Server System format

🟥 NVT format

TELNET client and server convert between native format and NVT format

Telnet commands

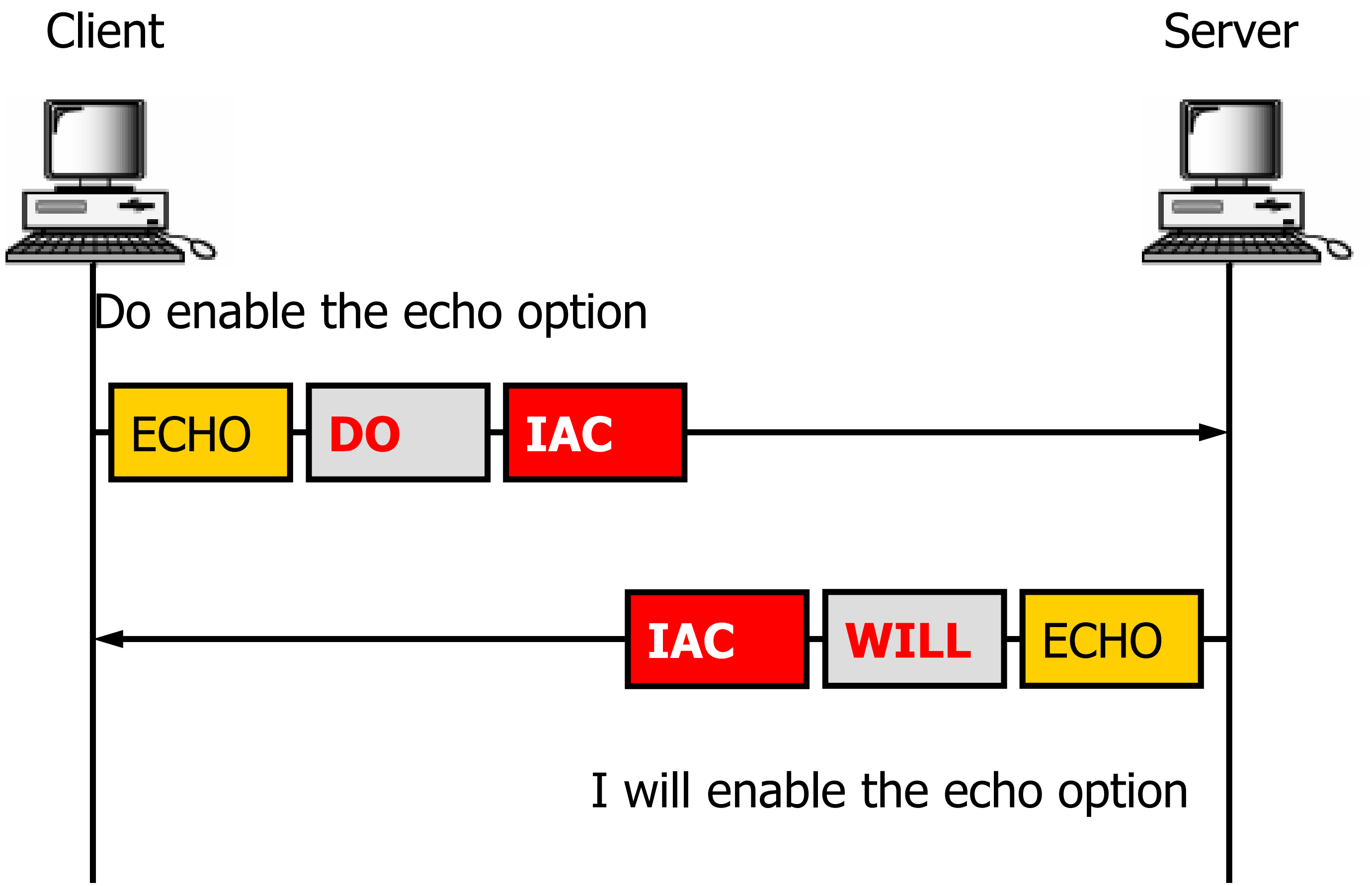
- TELNET control functions:
  - IAC, DO, DONT, WILL, WONT
- TELNET options example
  - Echo modes
  - Binary transmission
  - Line mode vs. character mode
  - Character set
  - Terminal type
- Understand the TELNET session through examples

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


Client

Server

Do enable the echo option

ECHO | DO | IAC

IAC | WILL | ECHO

I will enable the echo option

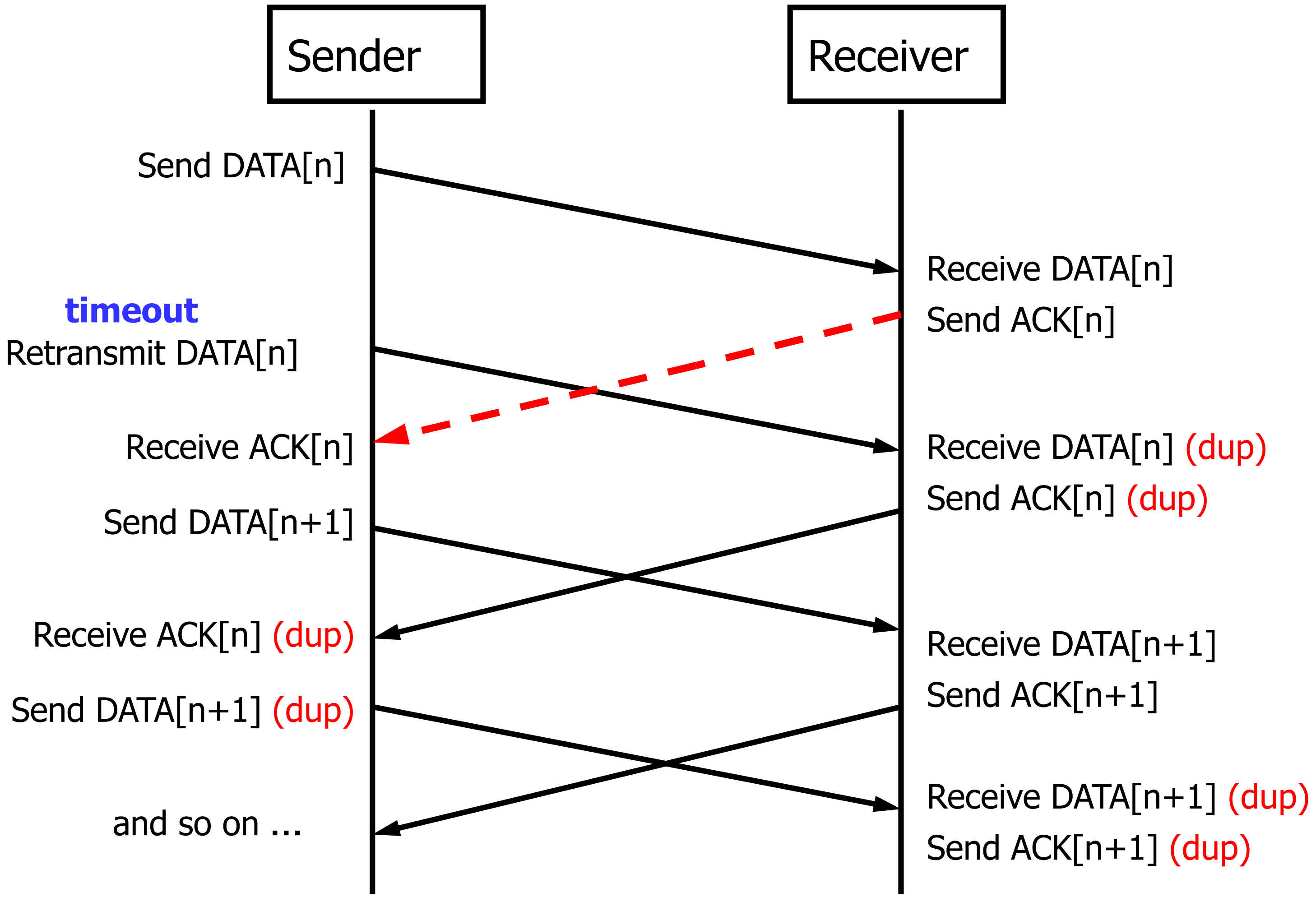
TFTP/FTP(1)

- 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 SAS (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



Sender

Receiver

Send DATA[n]

Receive DATA[n]

Send ACK[n]

timeout

Retransmit DATA[n]

Receive ACK[n]

Receive DATA[n] (dup)

Send ACK[n] (dup)

Send DATA[n+1]

Receive ACK[n] (dup)

Receive DATA[n+1]

Send ACK[n+1]

Send DATA[n+1] (dup)

Receive DATA[n+1] (dup)

Send ACK[n+1] (dup)

and so on ...

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

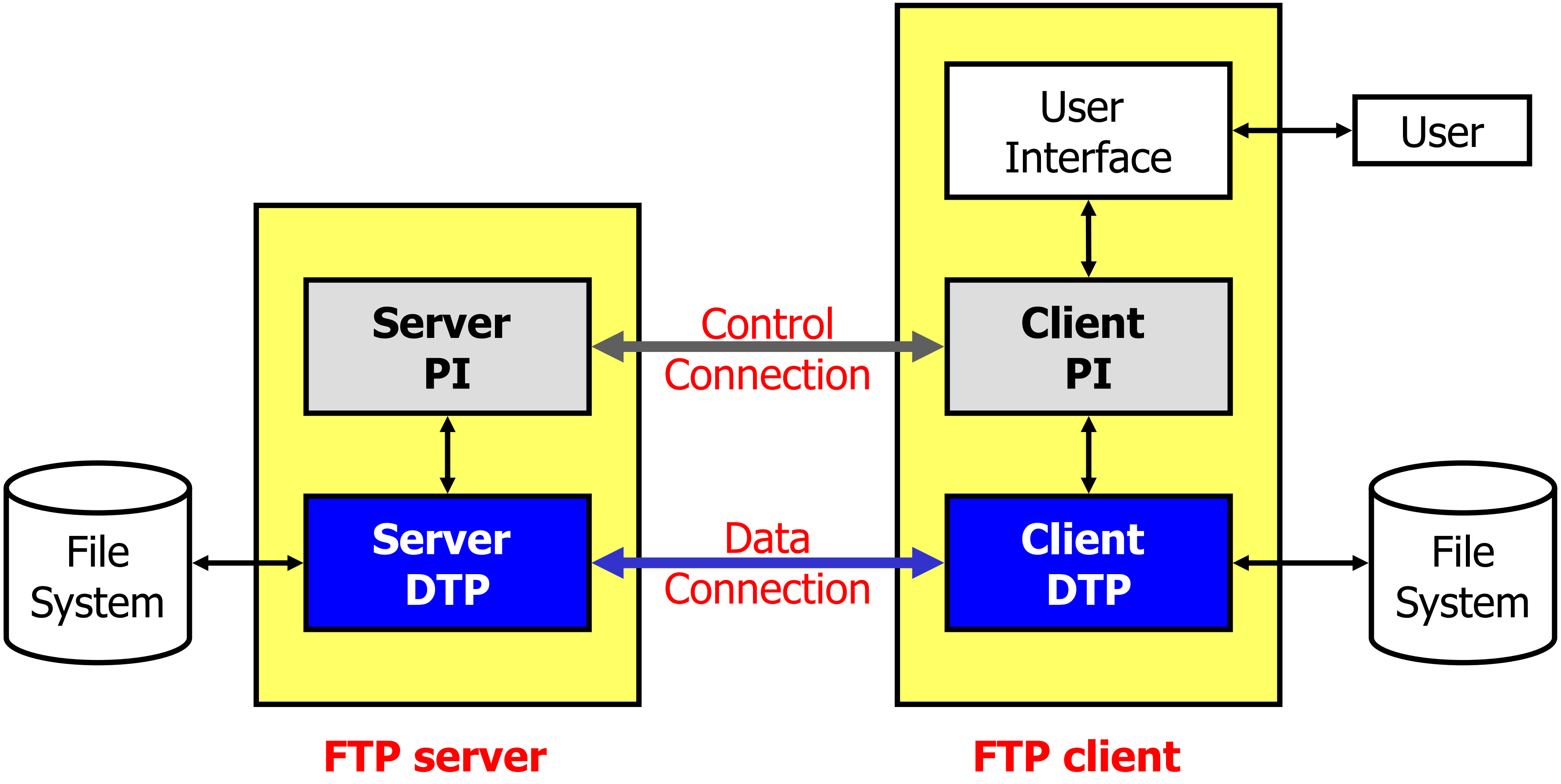
- Leading to excessive retransmissions

- 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 user commands & control commands
- FTP Control Connection & Data Connection
- FTP Active & Passive Mode

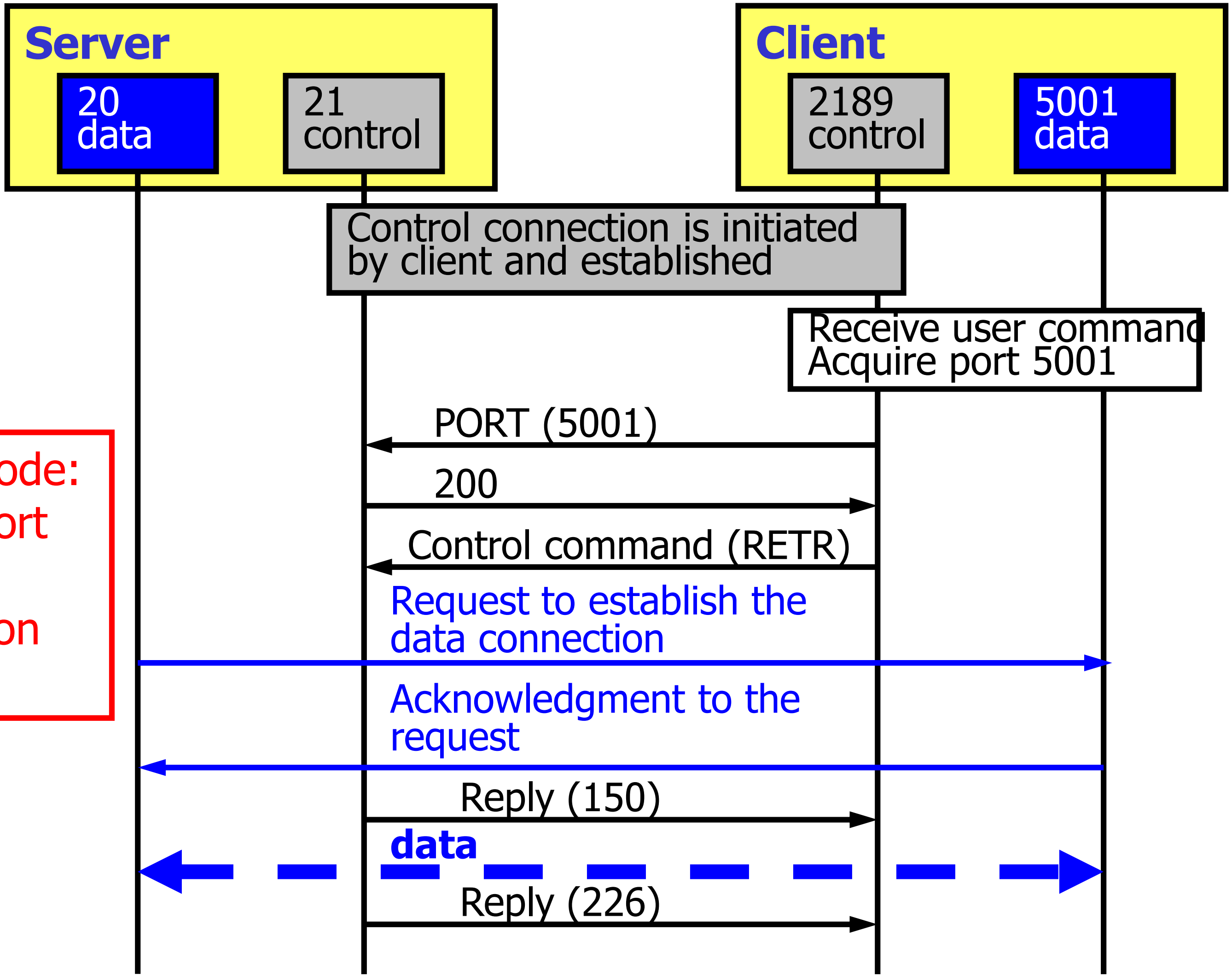
FTP Model


User
Interface
User
Server
PI
Control
Connection
Client
PI
File
System
Server
DTP
Data
Connection
Client
DTP
File
System
FTP server
FTP client

Active and Passive mode

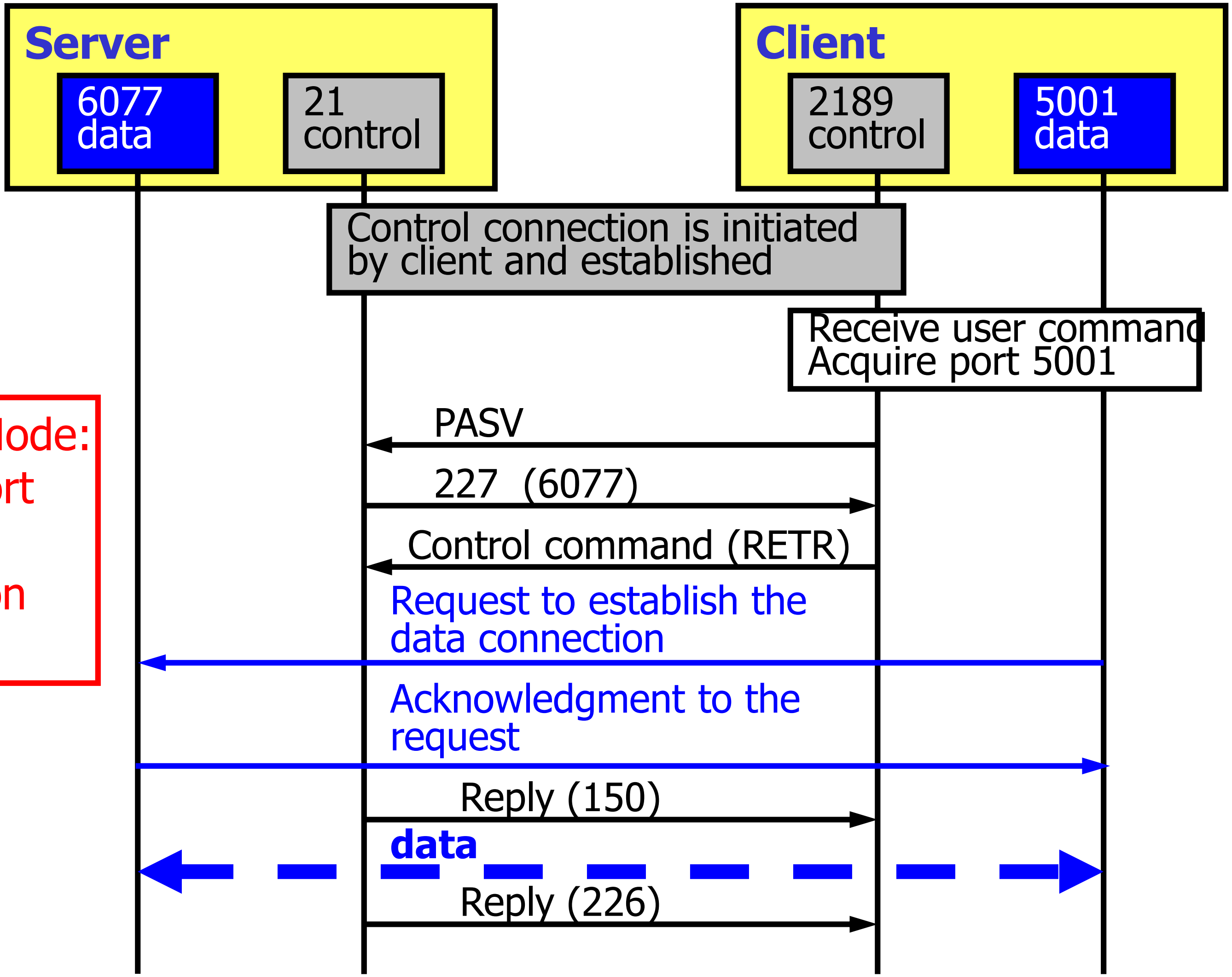
- Typical data connection handling sequence (in active mode)
  - Client sets up to “listen” on a unique port
  - Client uses local socket information to send PORT command to server
  - Server responds with “200” reply to acknowledge the port number
  - Client sends RETR, STOR, or other transfer command
  - 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 port

FTP Control Connection & Data Connection (1)



Active Mode:
Server port
for data
connection
=20

FTP Control Connection & Data Connection (3)

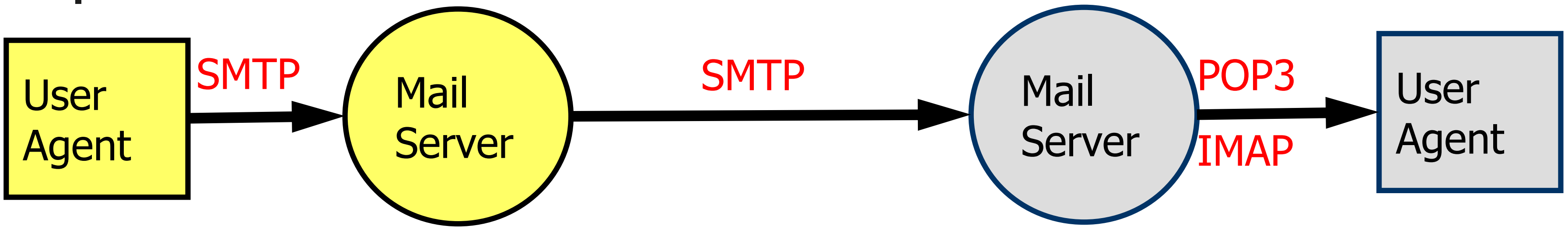


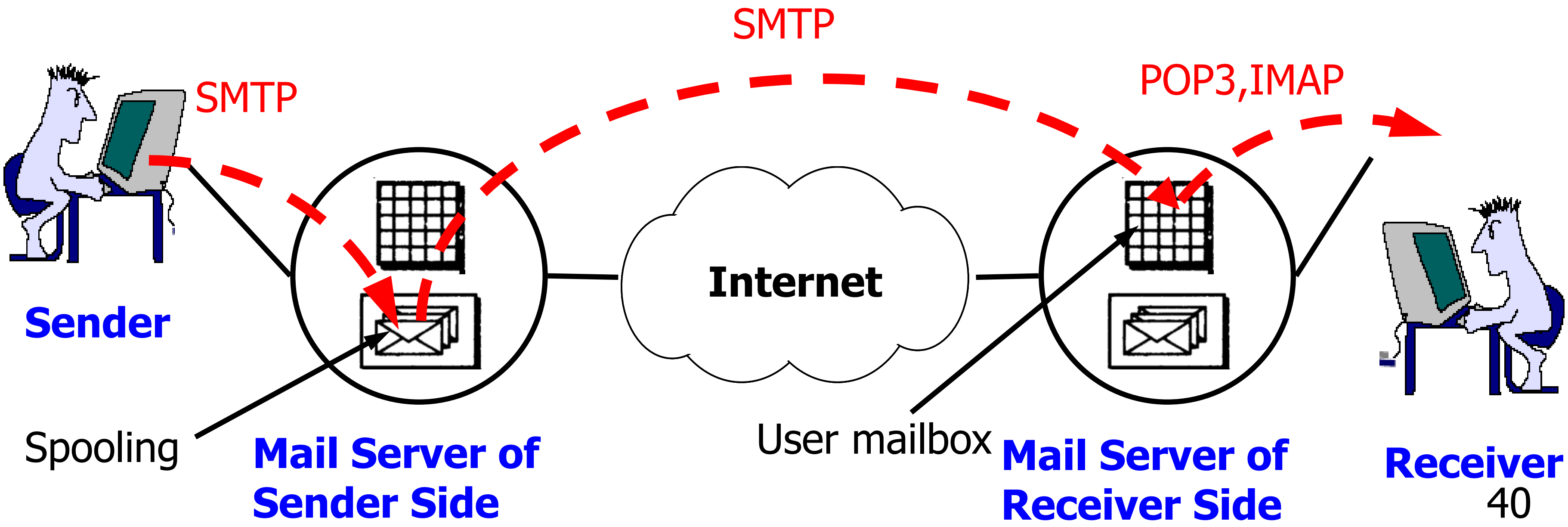
Passive Mode:
Server port for data connection >1024

Email(1)

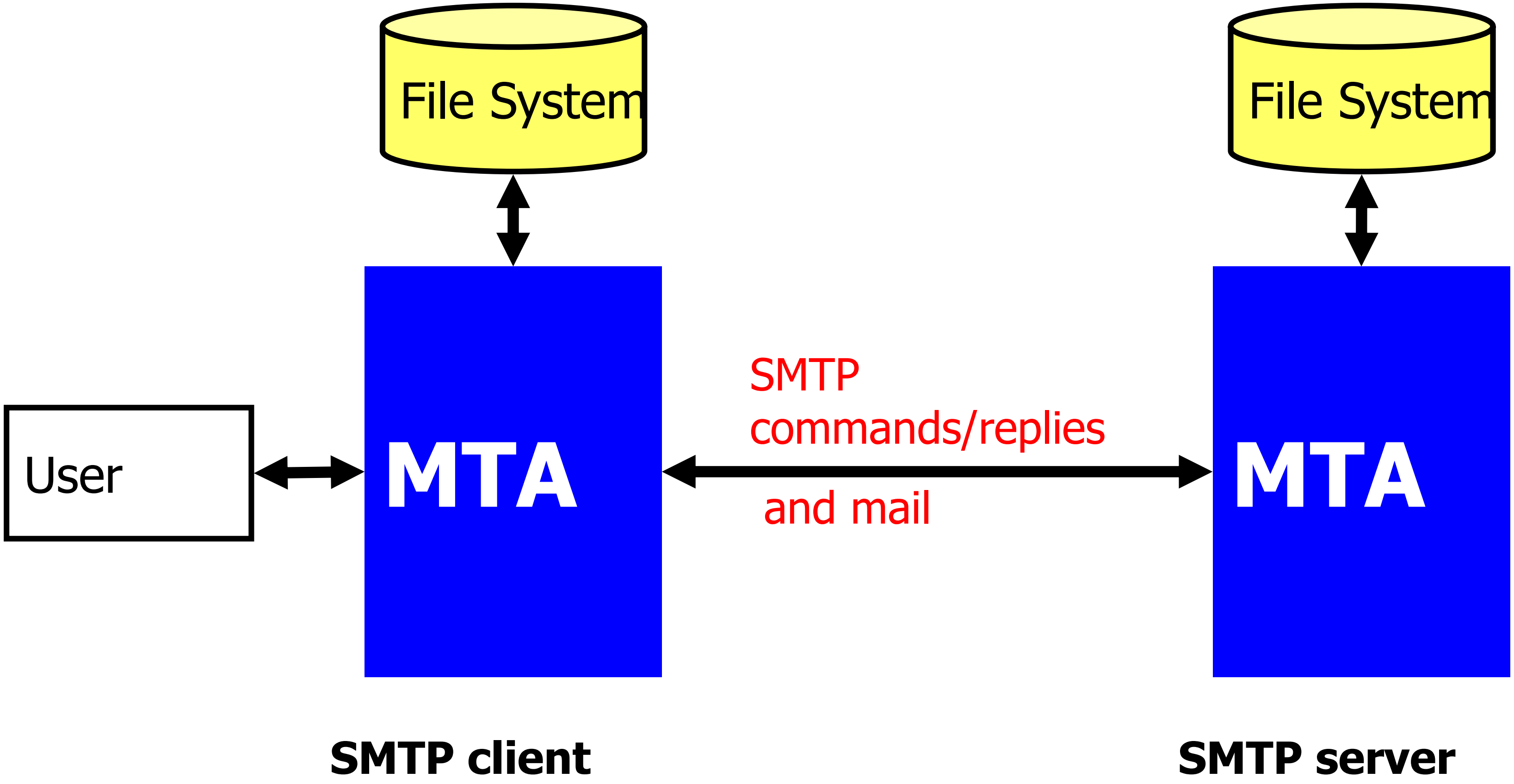
- Email Overview
  - Components of email system
  - Basic functions of email system
    - *Composition, Transfer, Reporting, Displaying, Disposition*
  - Terms: UA, Mail Server, MTA
  - Email address: *mailboxname@domain*
- Message Format
  - Header, blank line, body
- SMTP
  - Basic model
  - Basic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

Components Of Email System (1)



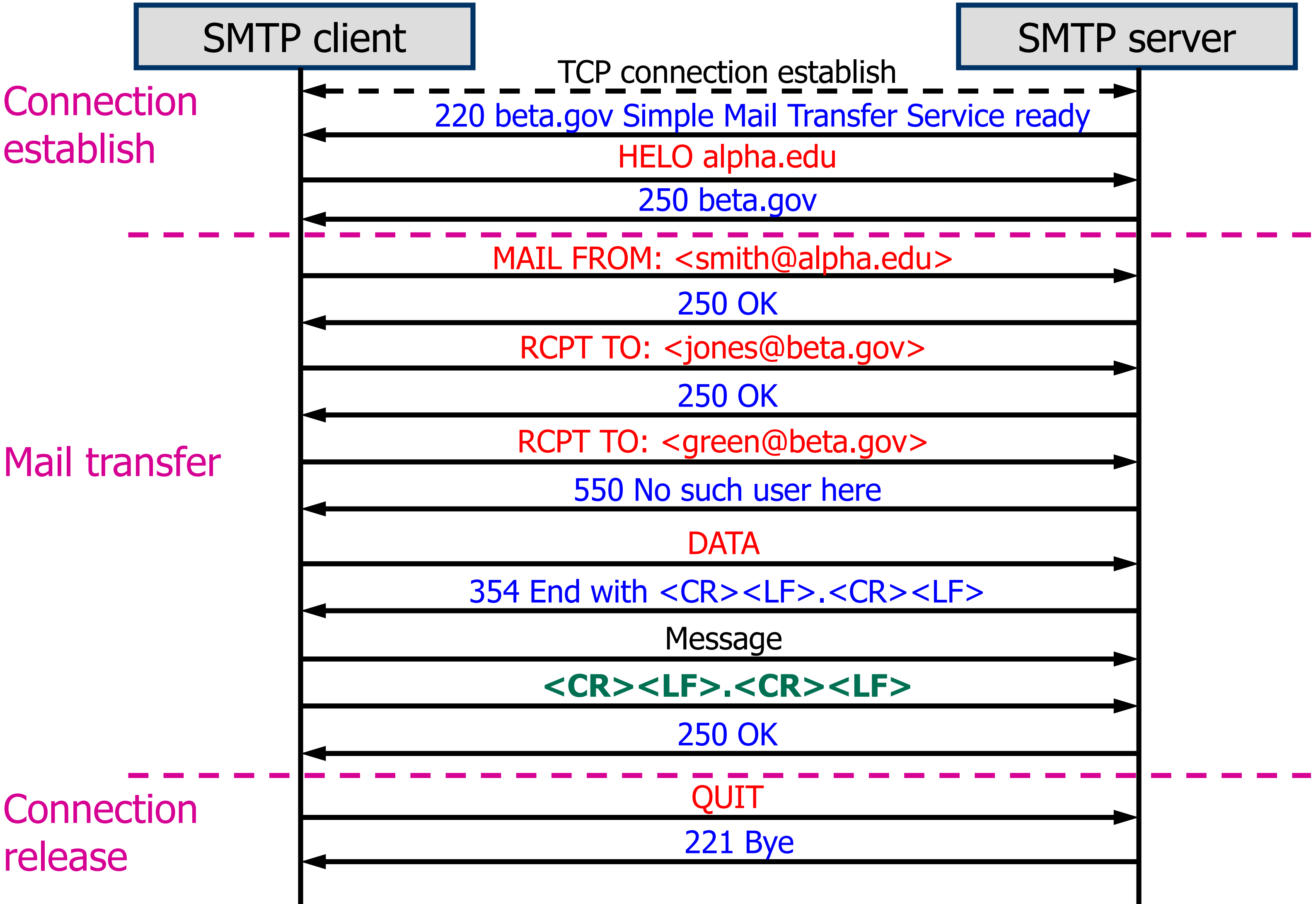


SMTP Basic Model



SMTP Commands and Status codes – example

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



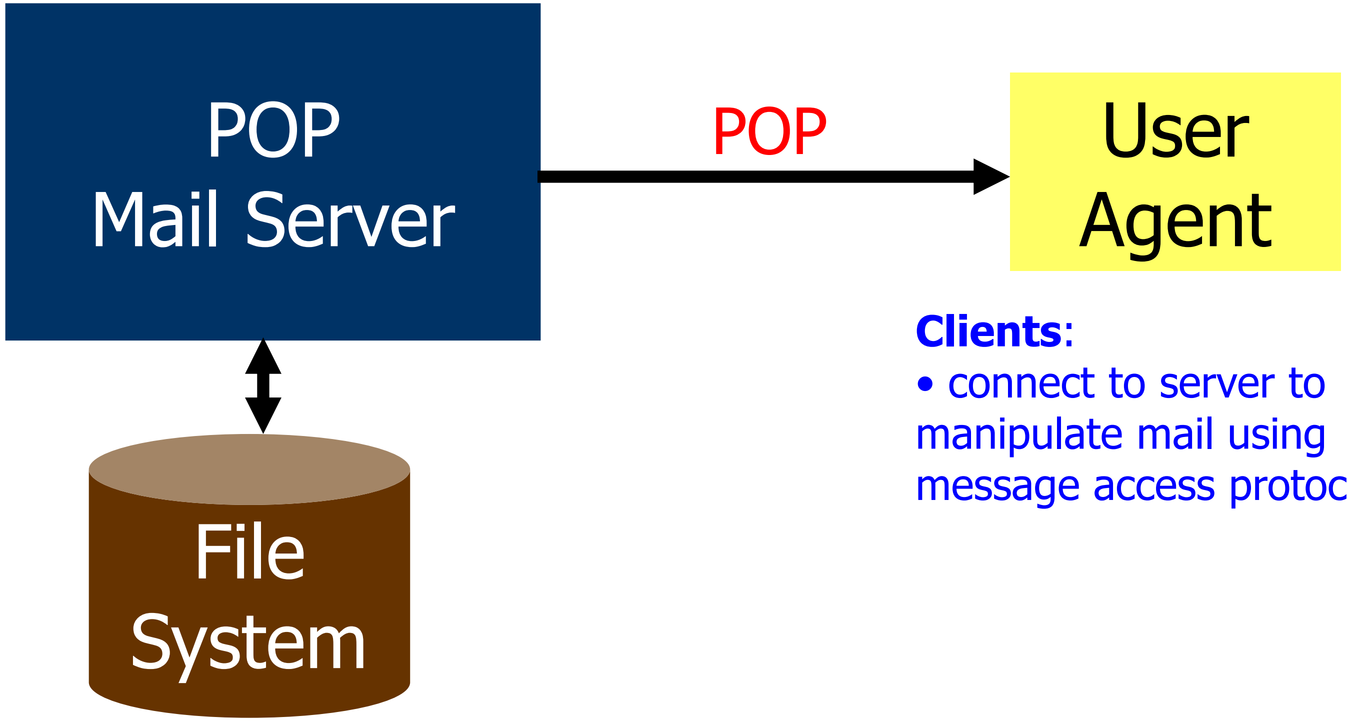
Email(2)

- POP
  - Basic model
  - Basic commands and replies
    - USER, PASS, LIST, RETR, DELE, QUIT, …
- IMAP
  - Features of IMAP
  - Comparison of POP and IMAP
- 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 – Basic Model

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

POP Mail Access Server:
• runs the POP3 service by listening on TCP port 110



Clients:
• connect to server to manipulate mail using message access protocols

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

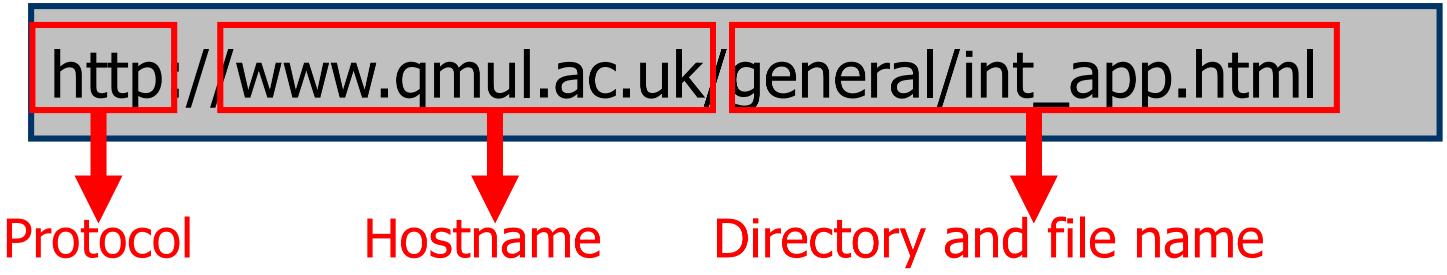
Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is email stored?	User's PC	Server
Where is email read?	Off-line	On-line
Mail Syncing	No	Yes
Direction	One-direction	Bi-directional
Good for mobile users?	No	Yes
Partial message downloads?	No	Yes
Speed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

WWW (1)

- WWW components
  - Client/browser
  - Web server
- 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



http://www.qmul.ac.uk/general/int_app.html

Protocol

Hostname

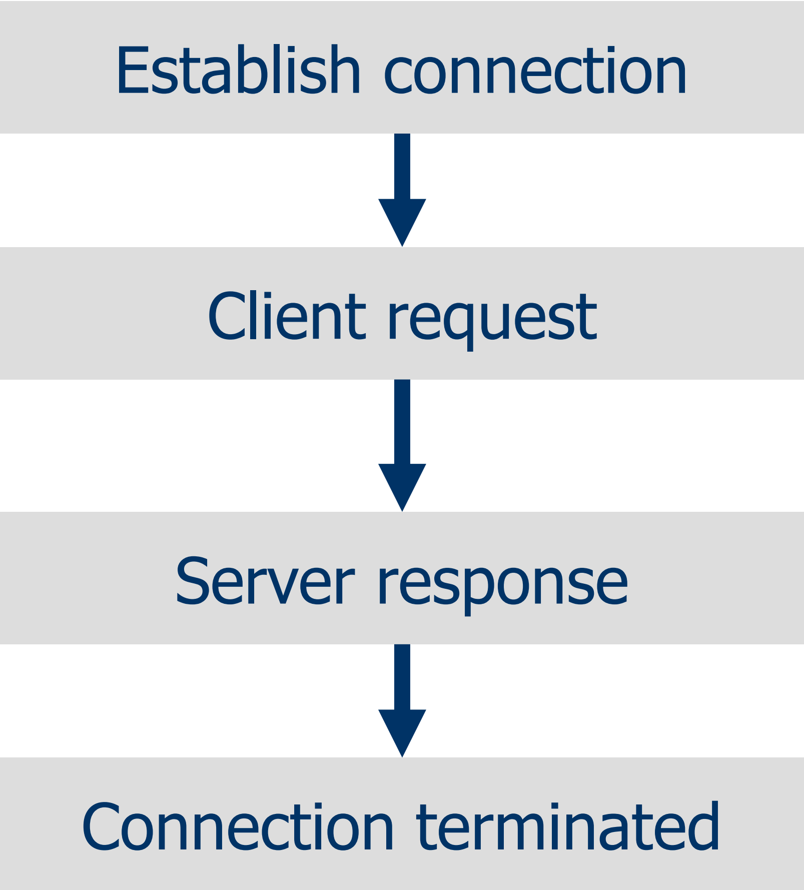
Directory and file name

- 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

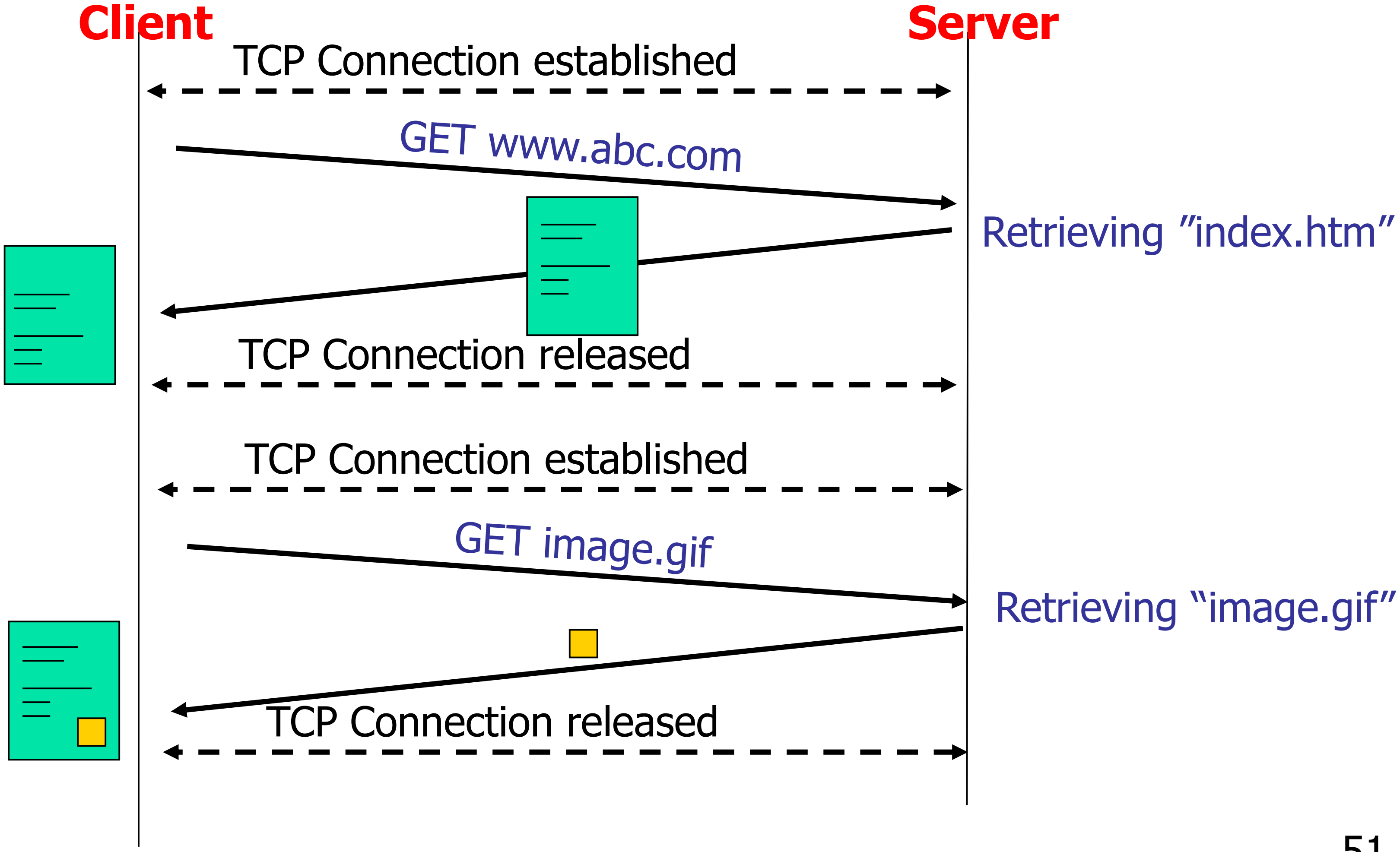
- TCP connection set up
- uses a port number as application reference
- usually port 80

- 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



Client

Server

TCP Connection established

GET www.abc.com

Retrieving ”index.htm”

TCP Connection released

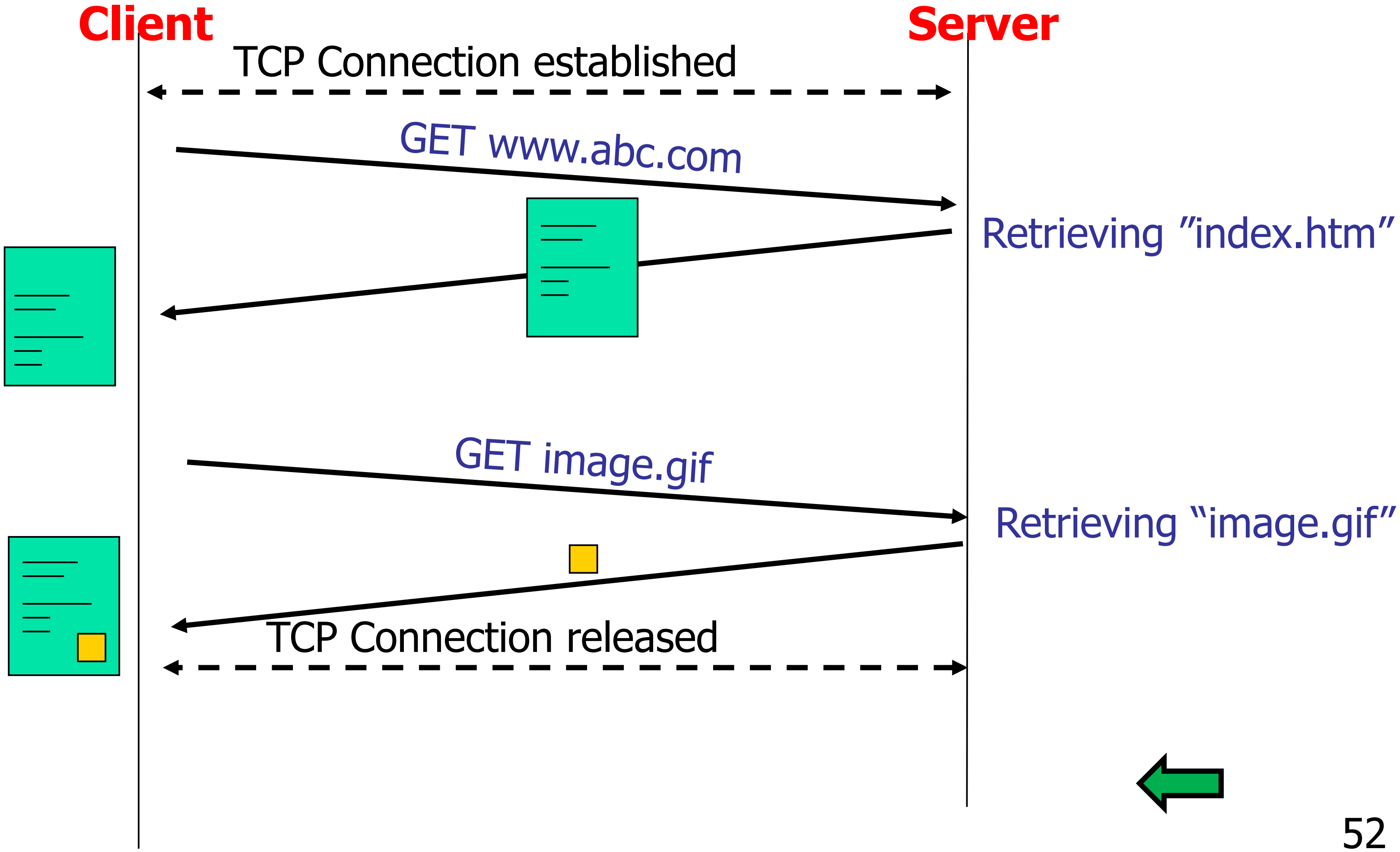
TCP Connection established

GET image.gif

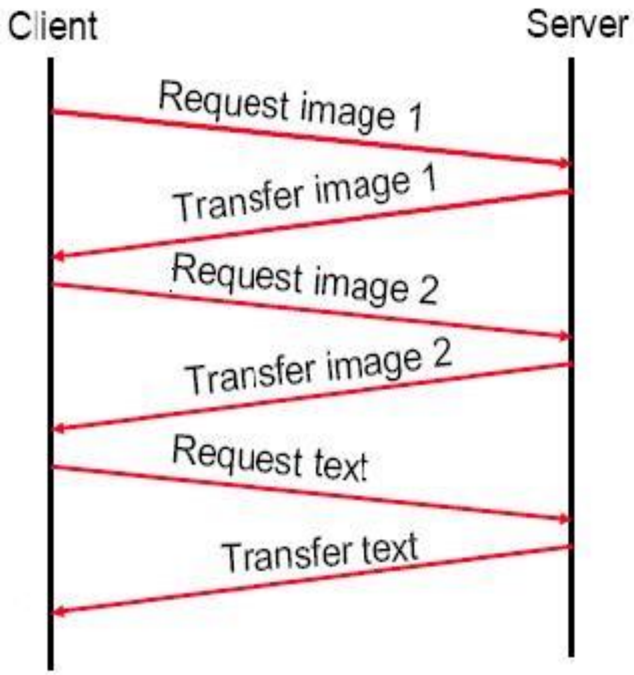
Retrieving “image.gif”

TCP Connection released

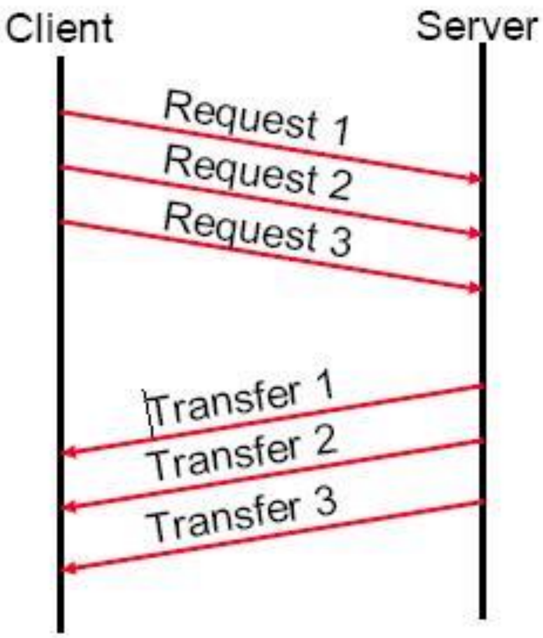
Example of Persistent Connections



Example of Pipelining



Non-pipelining

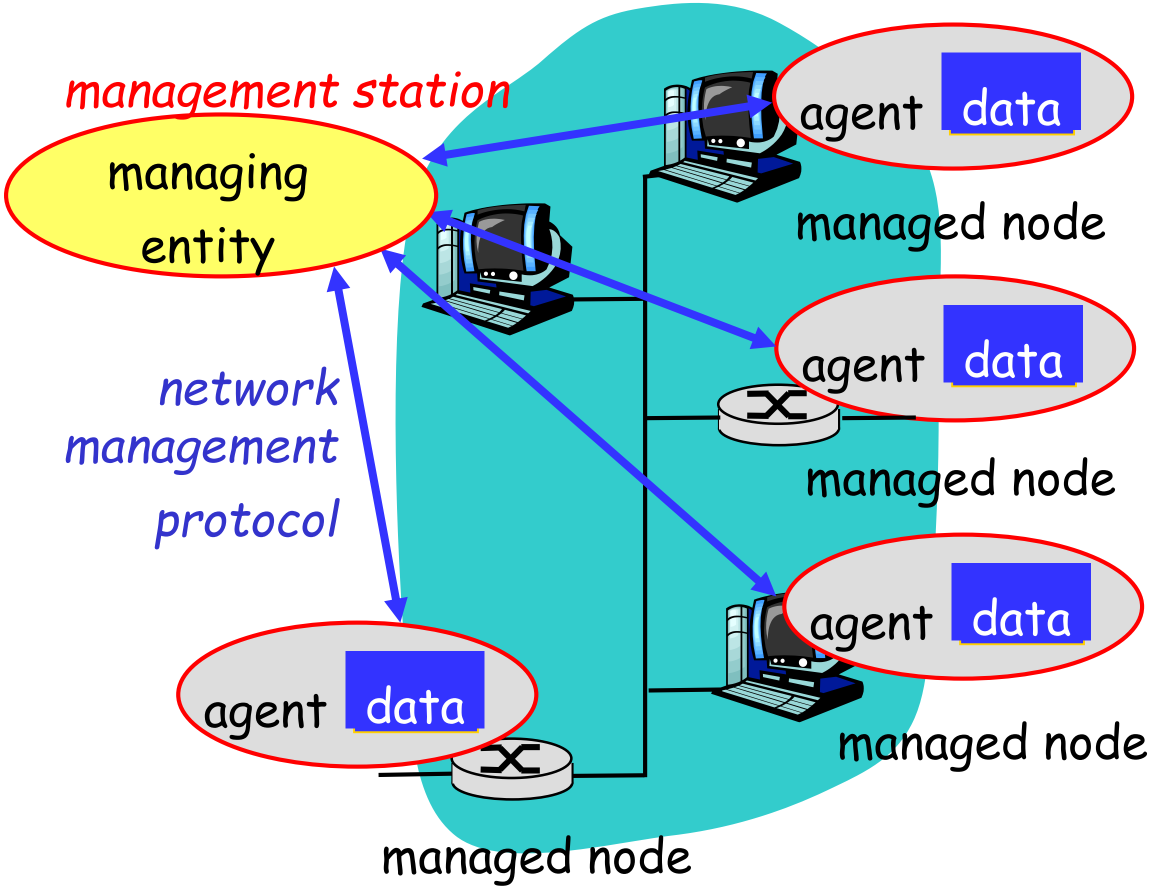


Pipelining

SNMP

- Terminologies: SNMP, MIB, SMI
- Definition, goals and functional areas of network management
  - Functional areas: FCAPS
- SNMP features
  - SNMPv1,v2,v3, RMON1,RMON2
  - SNMP realizes the F-C-P functions of network management
- SNMP model and components
- 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

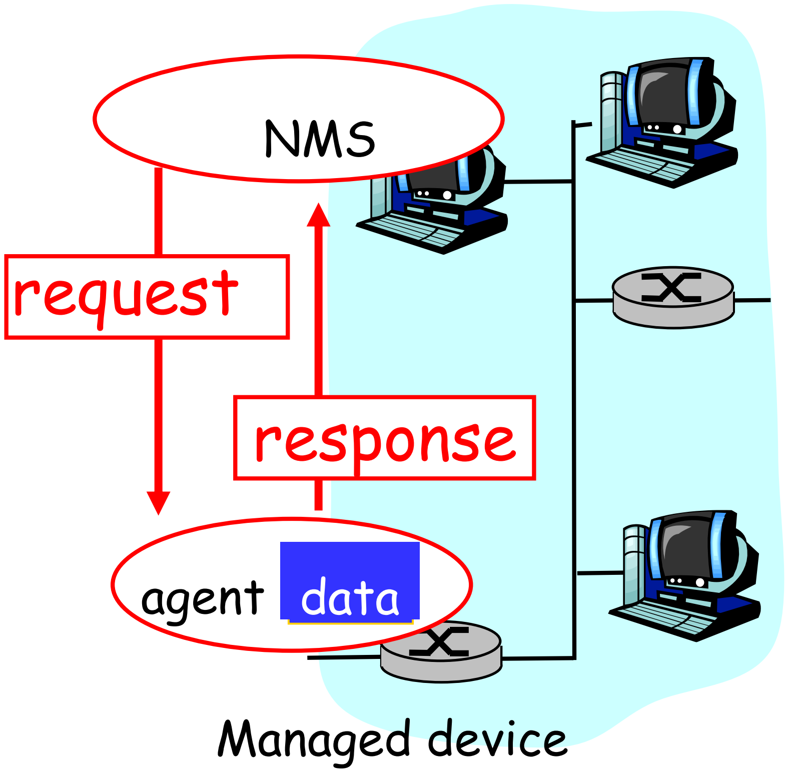
SNMP Model



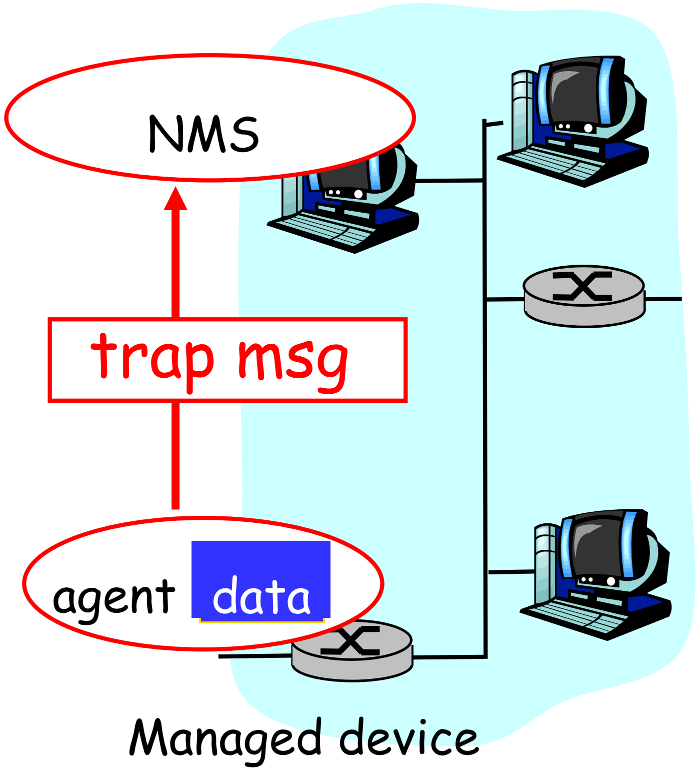
- The SNMP model of a managed network consists of four components:
  - Managed Nodes (Agent)
  - Management Stations (NMS)
  - Management Information (MIB)
  - A Management Protocol (SNMP)

SNMP 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