

# **Domain Name Service (DNS)**





# Need for Domain Name Service

---

- Natively, a TCP host is identified by its IP address
  - hosts use IP addresses to communicate to each other
  - the whole TCP/IP system software is based on using IP address
- IP address is not user friendly names
  - can not be easily remembered
- User should use meaningful names to human instead of IP address
- Domain name service fills the gap
  - let user use domain names for hosts (human friendly)
  - **map domain names into IP addresses** machines use for TCP/IP application layer



# DNS: Domain Name Service

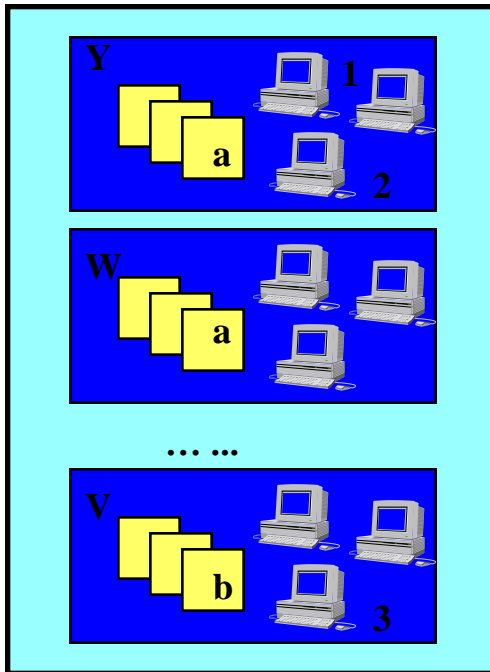
Domain Name Service (DNS) is:

- 1- *distributed database* implemented in hierarchy of many *name servers*
- 2- *application-layer protocol*: host, routers and name servers communicate to *resolve* names (address/name translation). **DNS protocol uses UDP transport protocol and port 53.**
- 3- employed by other application layer protocols (HTTP, SMTP, FTP) to resolve host names.



# Domain Name: a Hierarchy of Domains

Domain X



Domain is

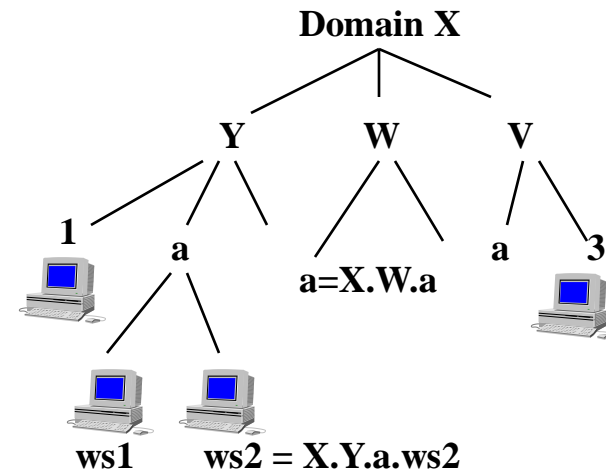
- a collection of hosts and
- a collection of subdomains
- is represented as a tree
  - domain name is the root name
  - a host name is a path from the root to the host
  - a subdomain is a path to the domain



host

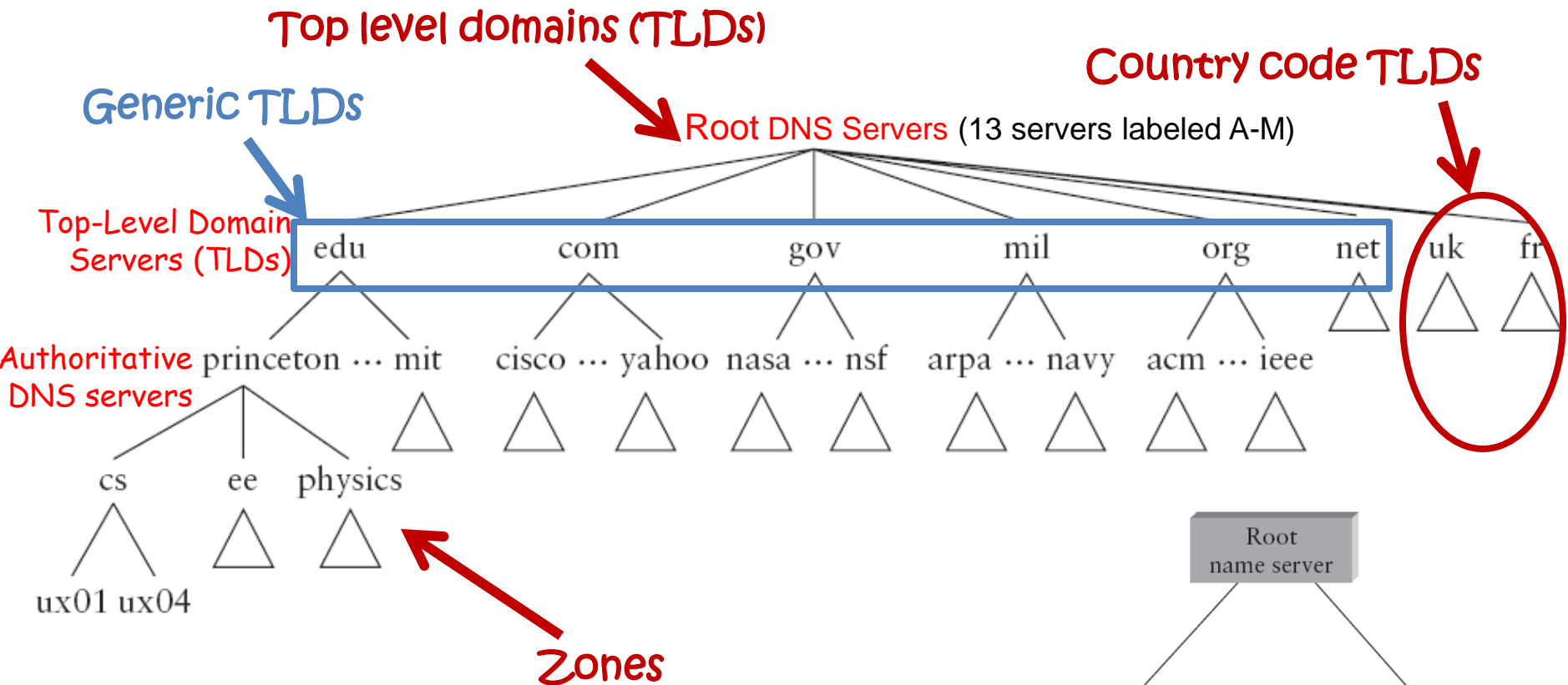


subdomain



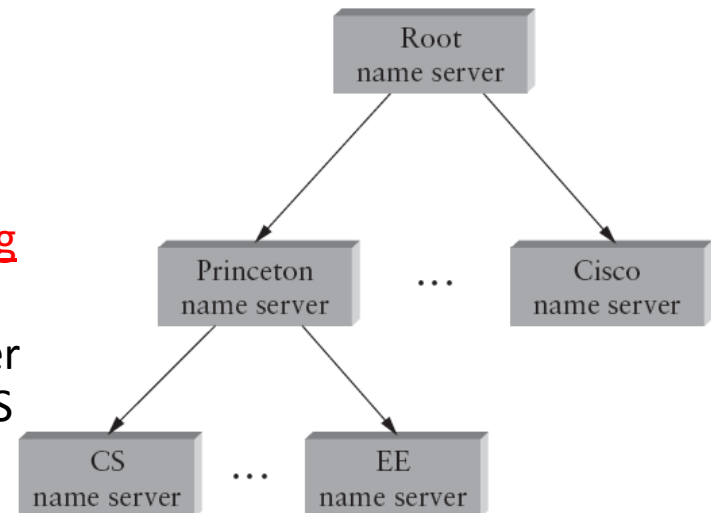


# DNS: Domain Level Hierarchy



Each Client uses a local DNS server that does not belong to the hierarchy:

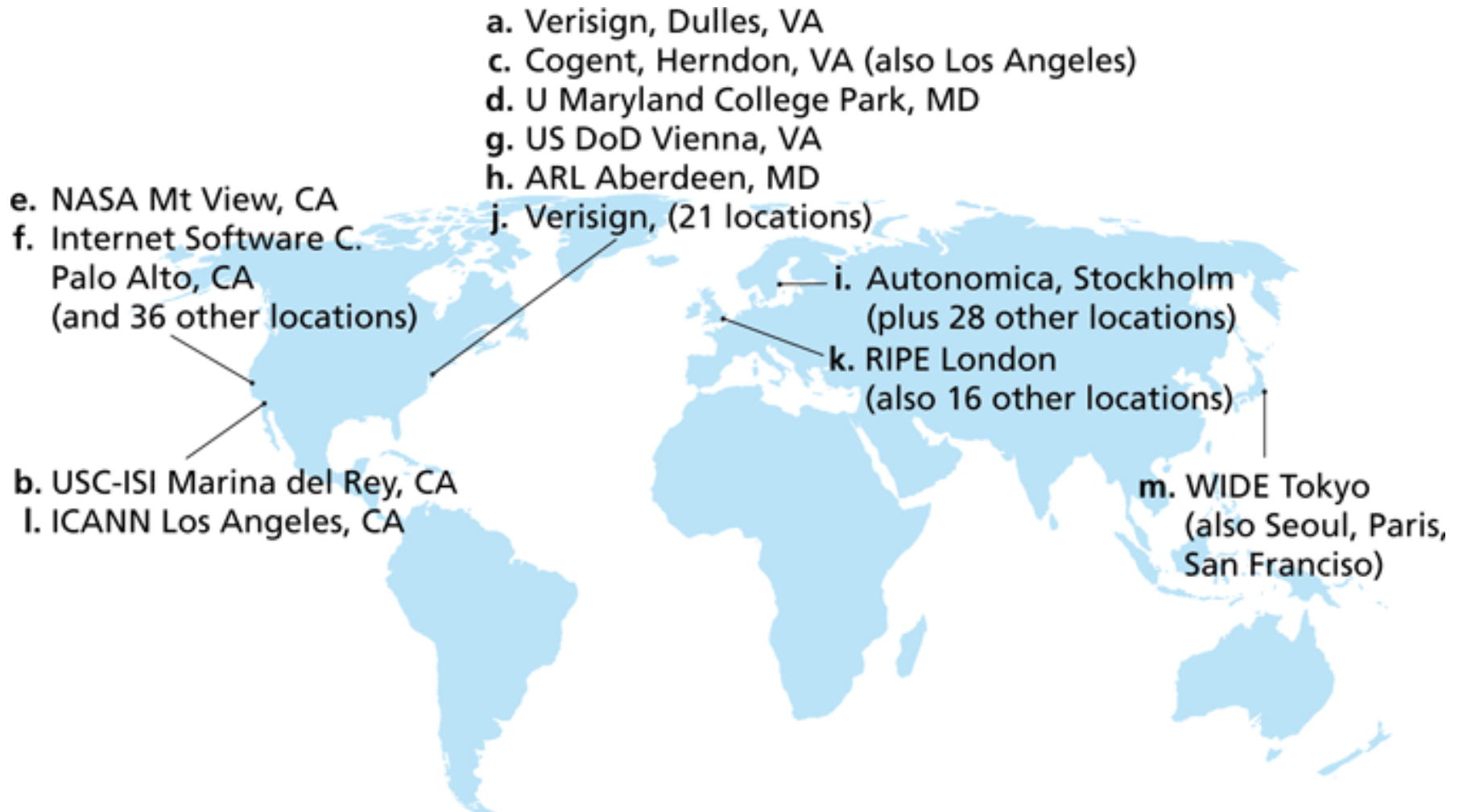
- The local DNS is usually assigned by the DHCP server (run command: "ipconfig /all" to find your local DNS server).





# DNS Root Servers

There are 13 root DNS server world wide that are labeled A-M: map of root DNS



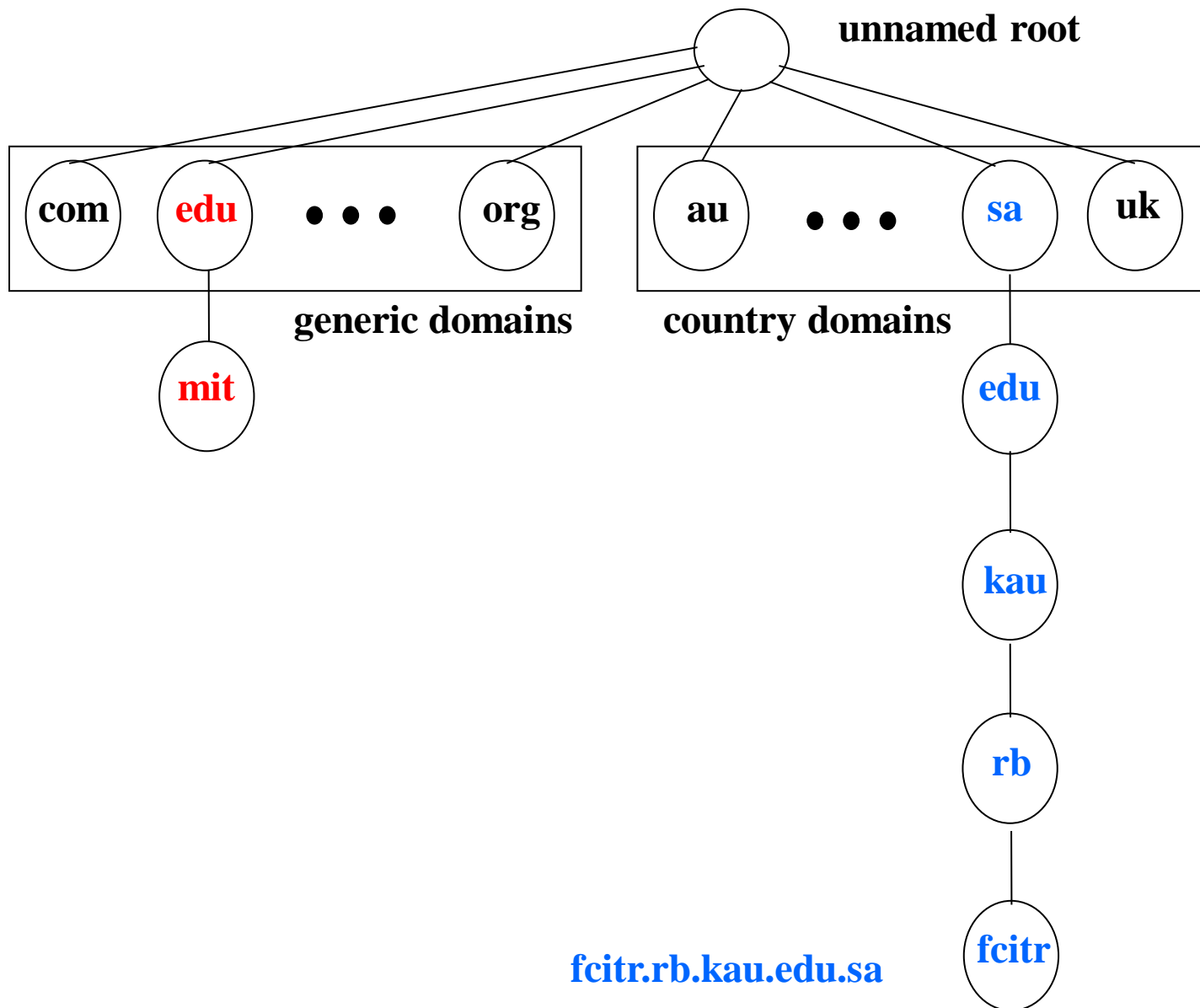


# Country-code Top level domains (cc-TLDs)

Courtesy: [wikimedia.org](https://commons.wikimedia.org/wiki/File:Map_of_North_America_highlighting_top_level_domains)









# TLD and Authoritative Servers

- **Top-level domain (TLD) servers:** responsible for com, org, net, edu, etc, and all country code top-level domains (ccTLD) sa, pk, ca, in, cn, jp.
- **Authoritative DNS servers:** organization's with public names has DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider



# Local Name Server

- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one.
  - Also called “default name server”
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into hierarchy.

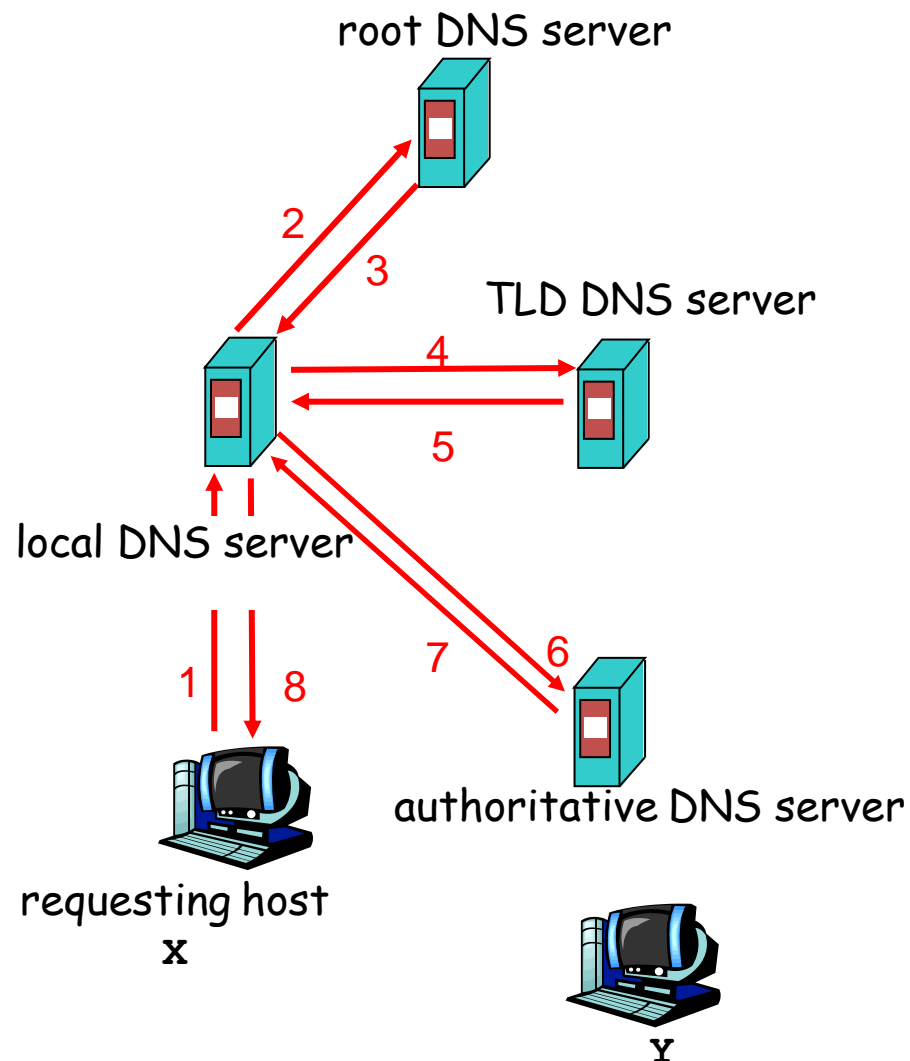


# How DNS Works?

Client X wants IP address for Y

Steps performed:

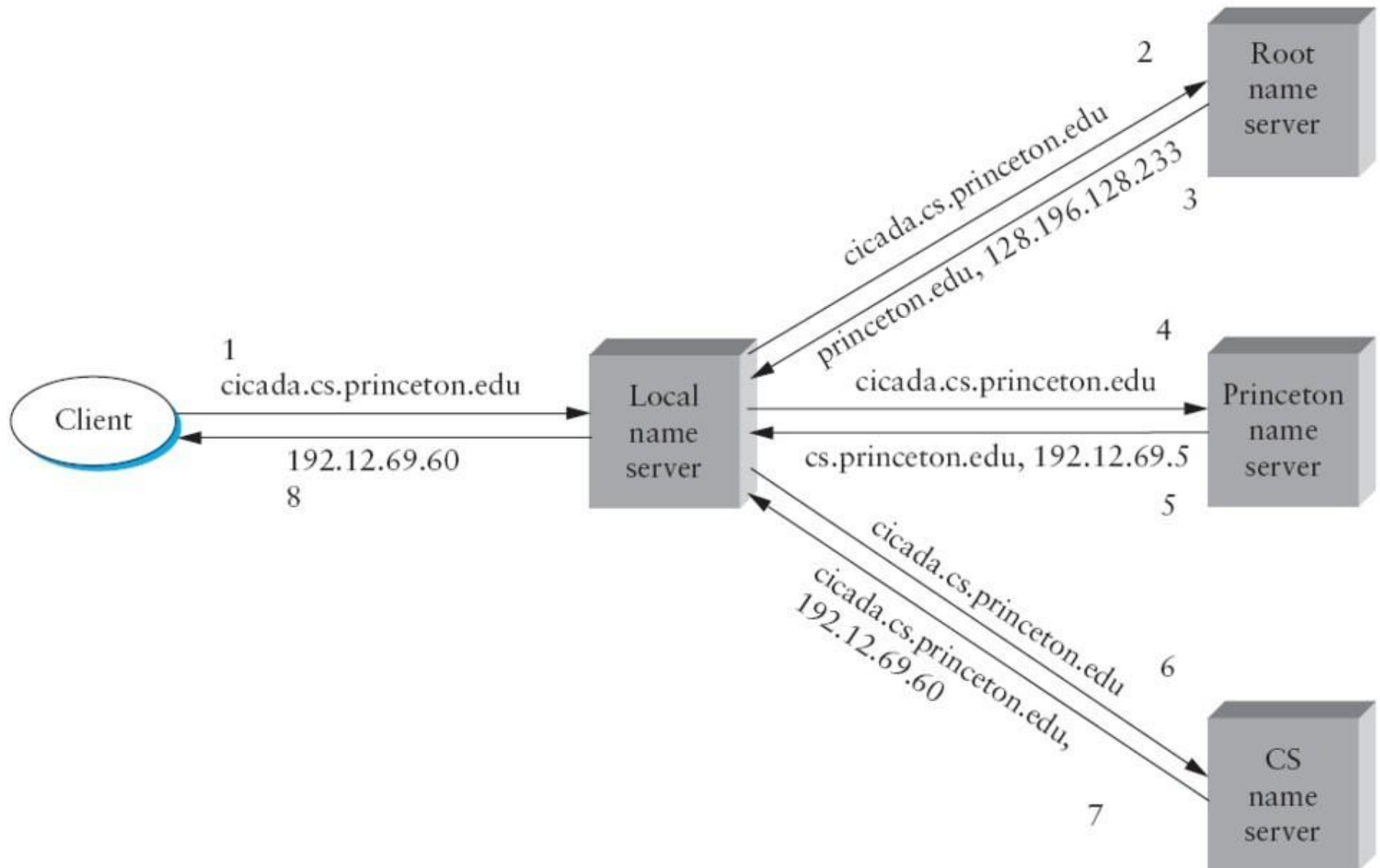
- 1- Client sends DNS request to the local DNS server to search on its behalf
- 2- local DNS contacts one of the root DNSs to resolve hostname Y.
- 3- root DNS returns the TLD DNS IP to local DNS
- 4- local DNS contacts one of the TLDs to get an Authoritative DNS name
- 5- TLD returns IP of authoritative DNS to local DNS
- 6- local DNS contacts authoritative DNS to resolve Y
- 7- authoritative DNS returns IP of Y
- 8- local DNS return IP of Y to X



*Example of how DNS query is typically resolved*



# How DNS resolves hostname





# DNS caching

- once (any) name server learns mapping (website name & its IP Address), it *caches* mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited
  - Client may also cache DNS names in 'hosts file'



# hosts file

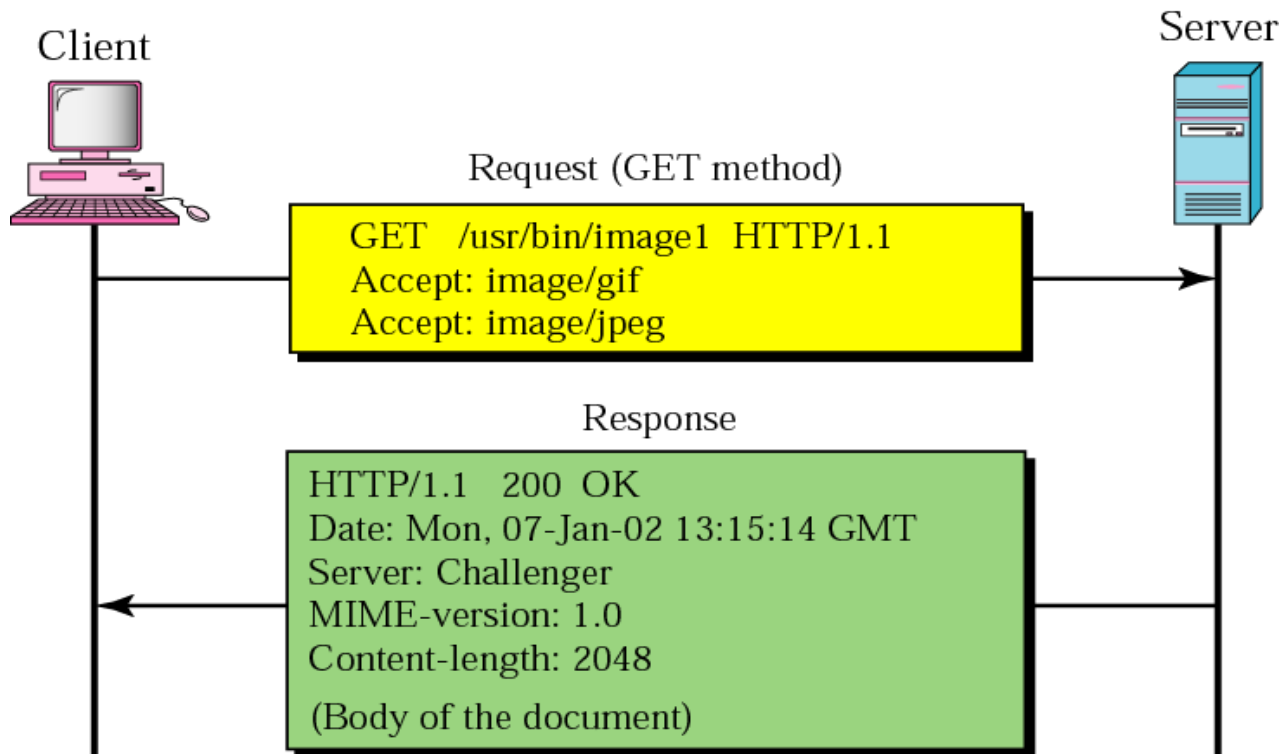
- local file that is checked by the client DNS of the OS before sending a DNS request. It can speed the web access.
- If the requested name is found in the hosts file then its corresponding IP is used.
- Can be used to create custom (name-IP) entries.
- File Location:
  - windows XP C:\WINDOWS\system32\drivers\etc
  - most UNIX and Linux /etc
- File Structure:
  - <IP address><space><name><space><# comment>
  - Example of an entry: 127.0.0.1 localhost #default entry



# **Hypertext Transfer Protocol (HTTP)**

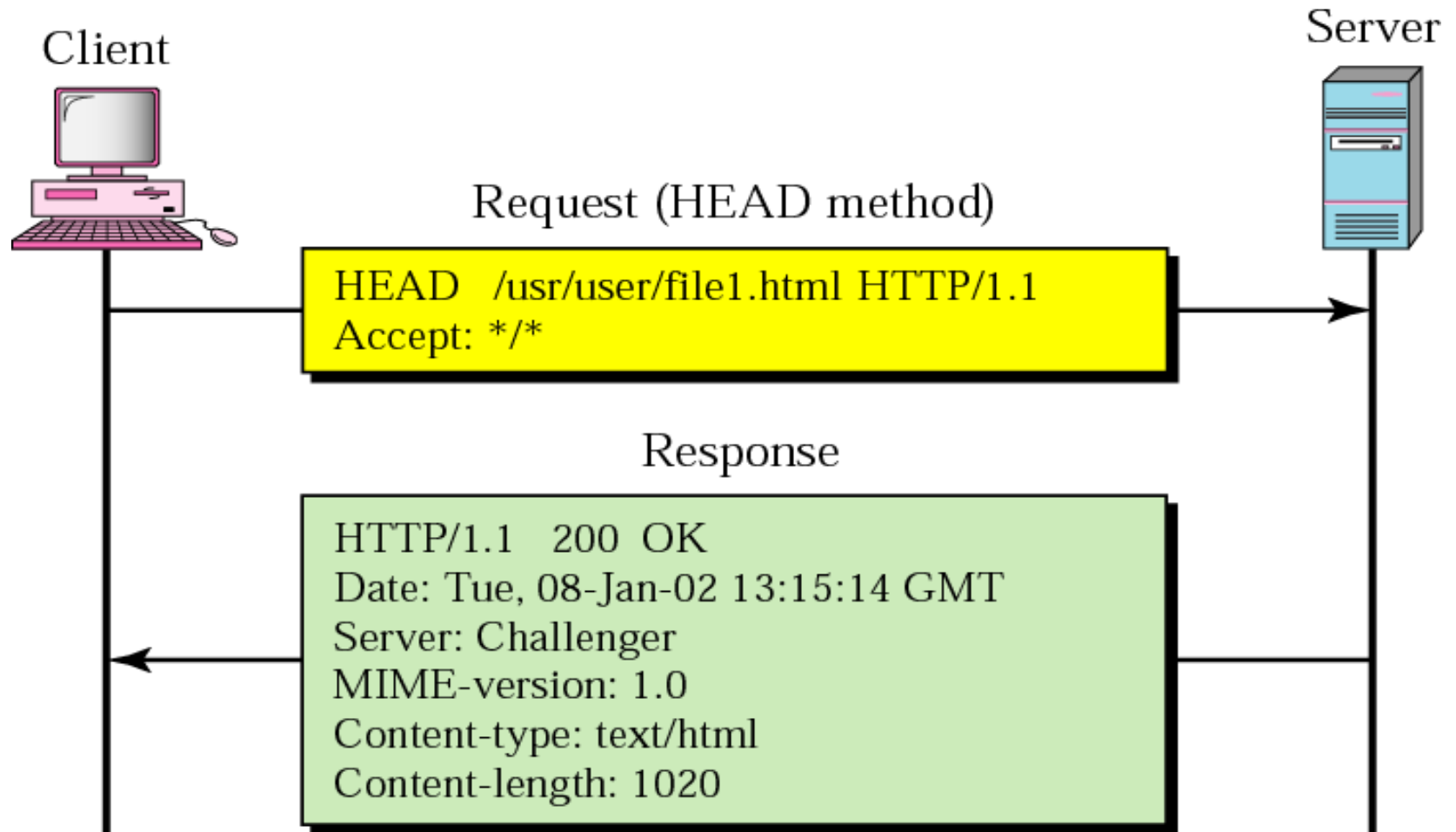


# HTTP GET Request





# HTTP HEAD Request





# HTTP Responses

Code	Type	Example Reasons
1xx	Informational	request received, continuing process
2xx	Success	action successfully received, understood, and accepted
3xx	Redirection	further action must be taken to complete the request
4xx	Client Error	request contains bad syntax or cannot be fulfilled
5xx	Server Error	server failed to fulfill an apparently valid request

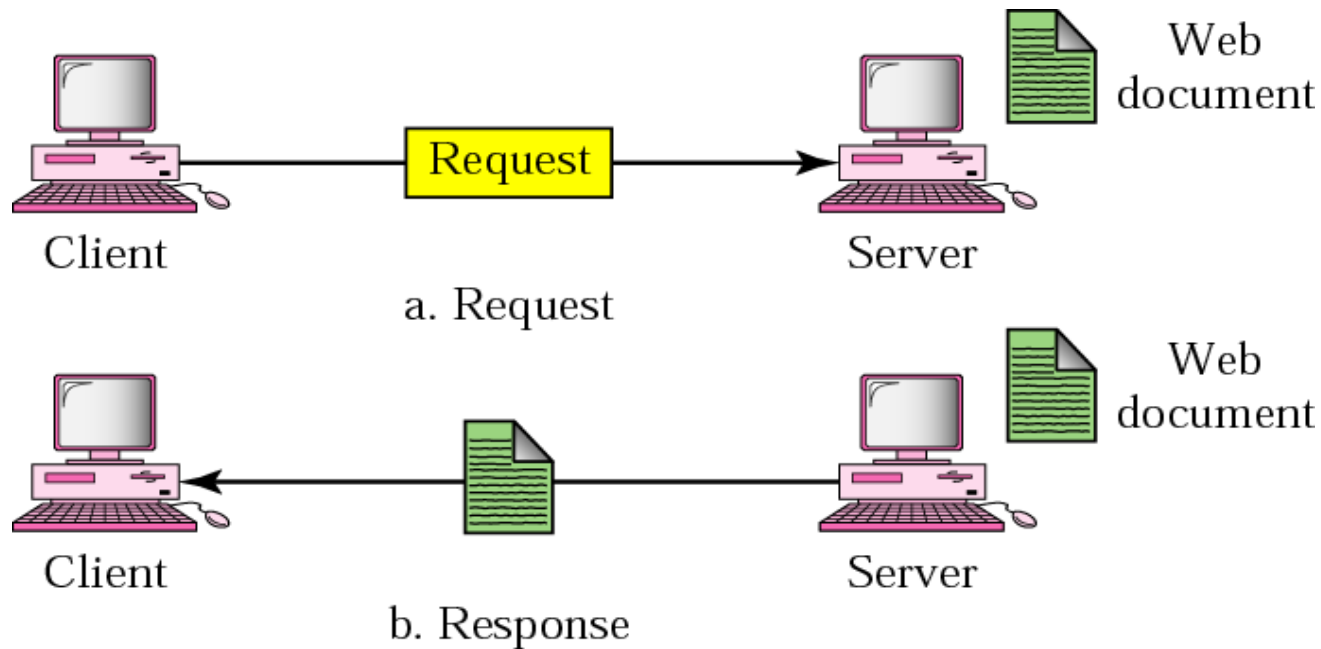


# HTTP Message Headers

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

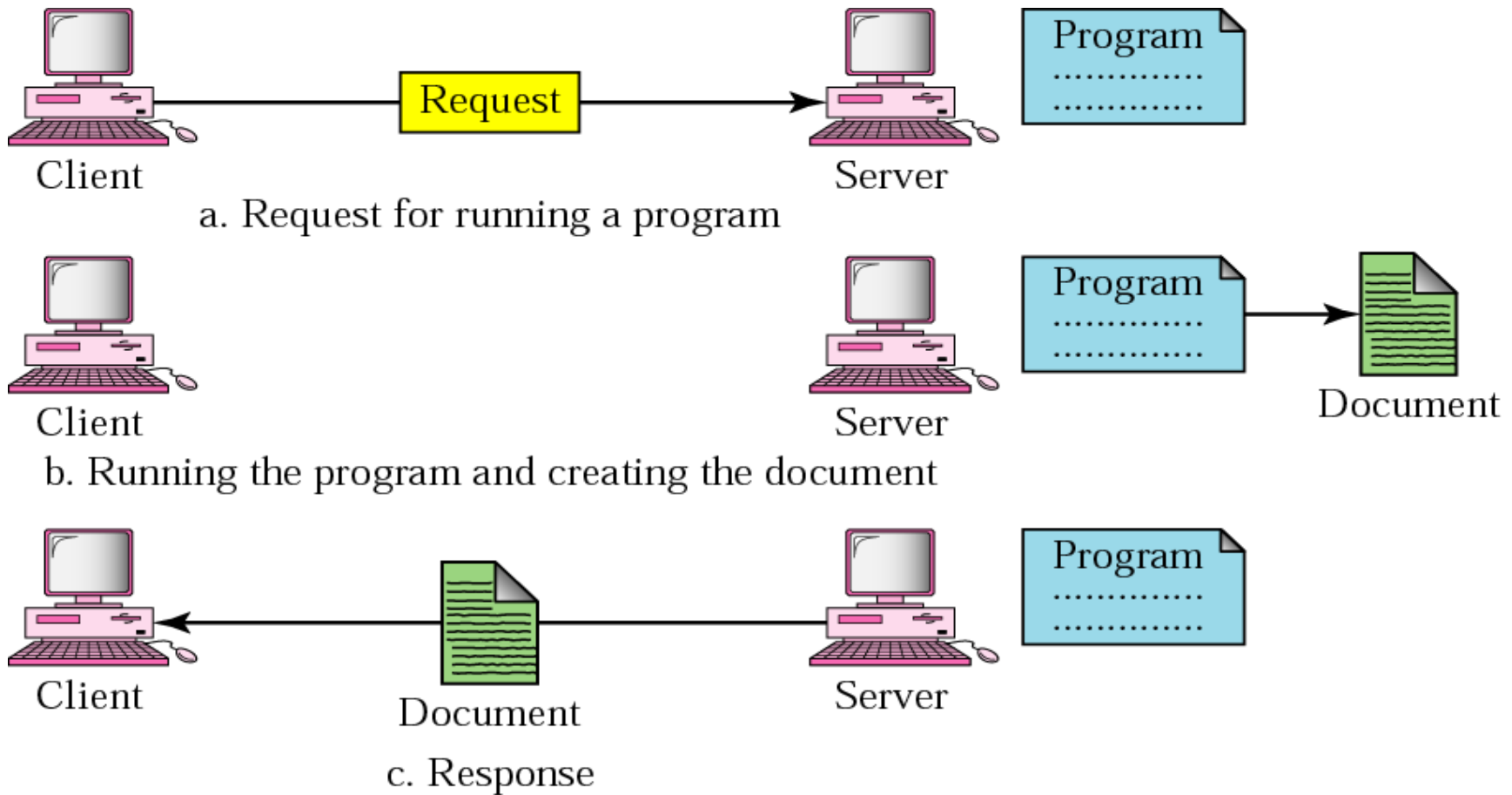


# WWW: Static Document



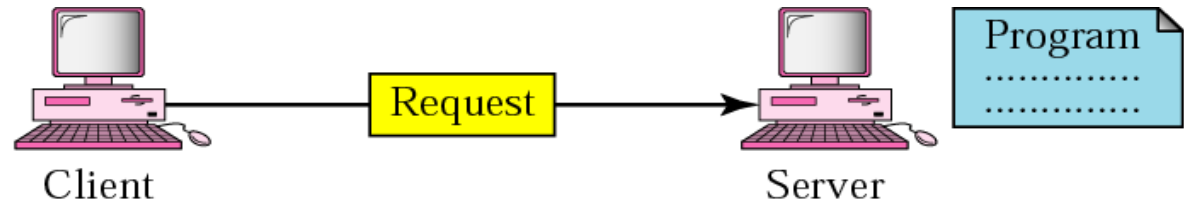


# WWW: Dynamic Document

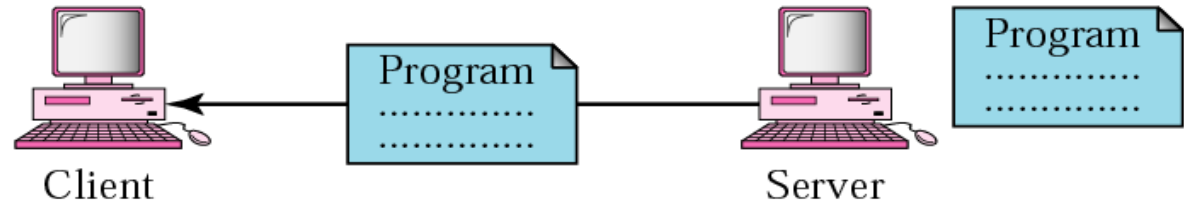




# WWW: Active Document



a. Request for a copy of a program



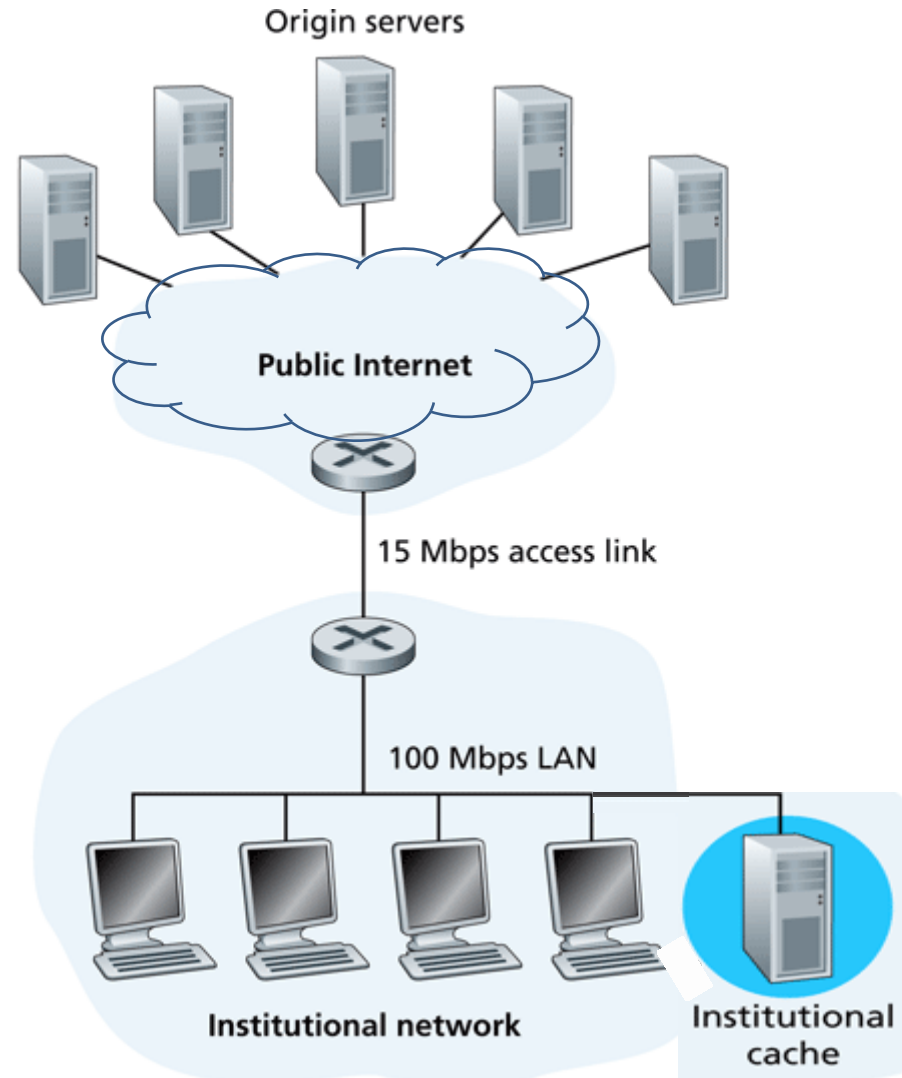
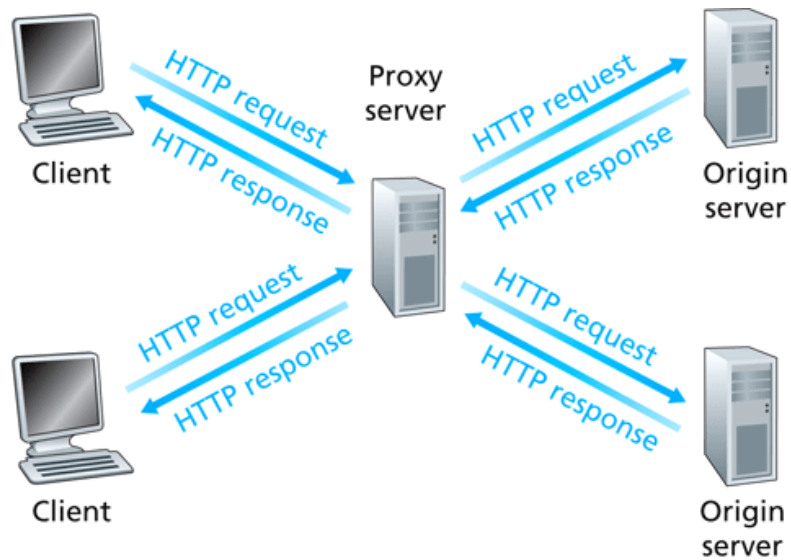
b. Sending a copy of the program



c. Running the program and creating the document



# HTTP Proxy Server





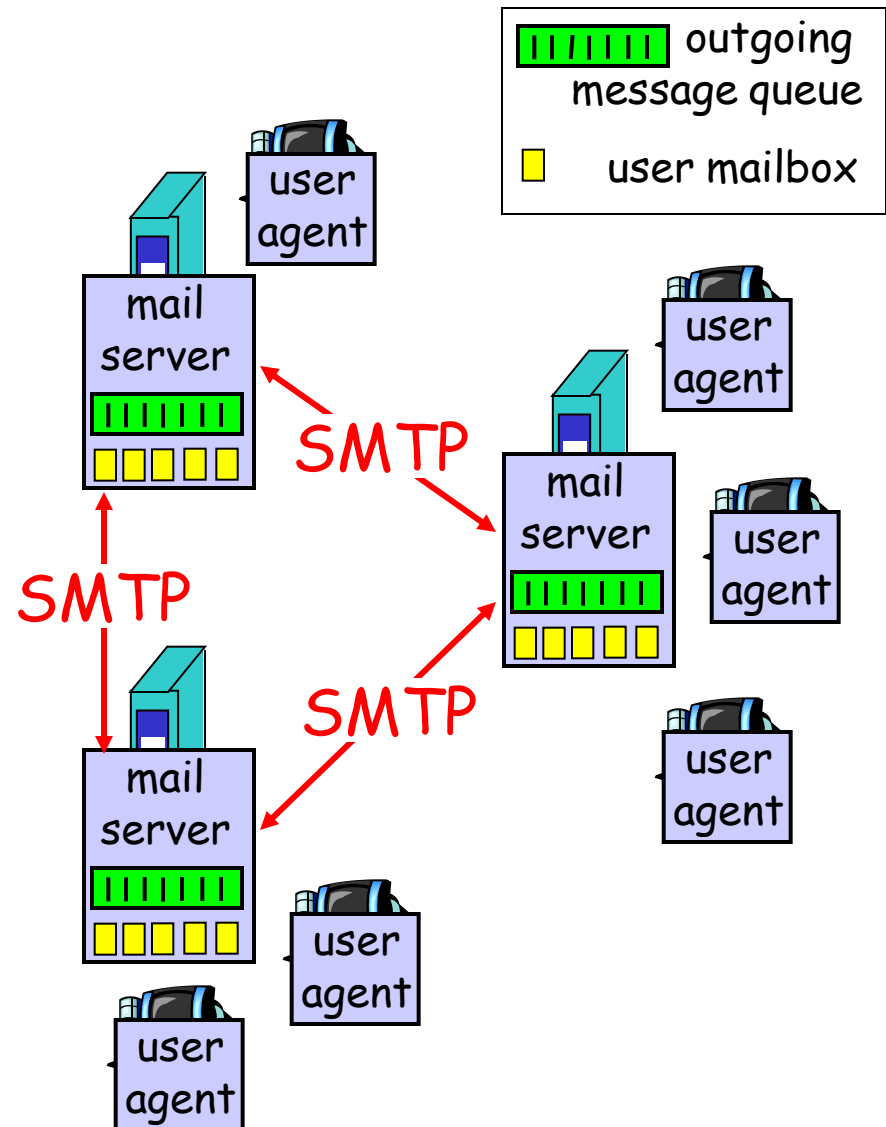
# Electronic Mail

## Three major components:

- ❑ user agents
- ❑ mail servers
- ❑ simple mail transfer protocol: SMTP

## User Agent

- ❑ a.k.a. "mail reader"
- ❑ composing, editing, reading mail messages
- ❑ e.g. Microsoft Outlook, Netscape Messenger
- ❑ outgoing, incoming messages stored on server





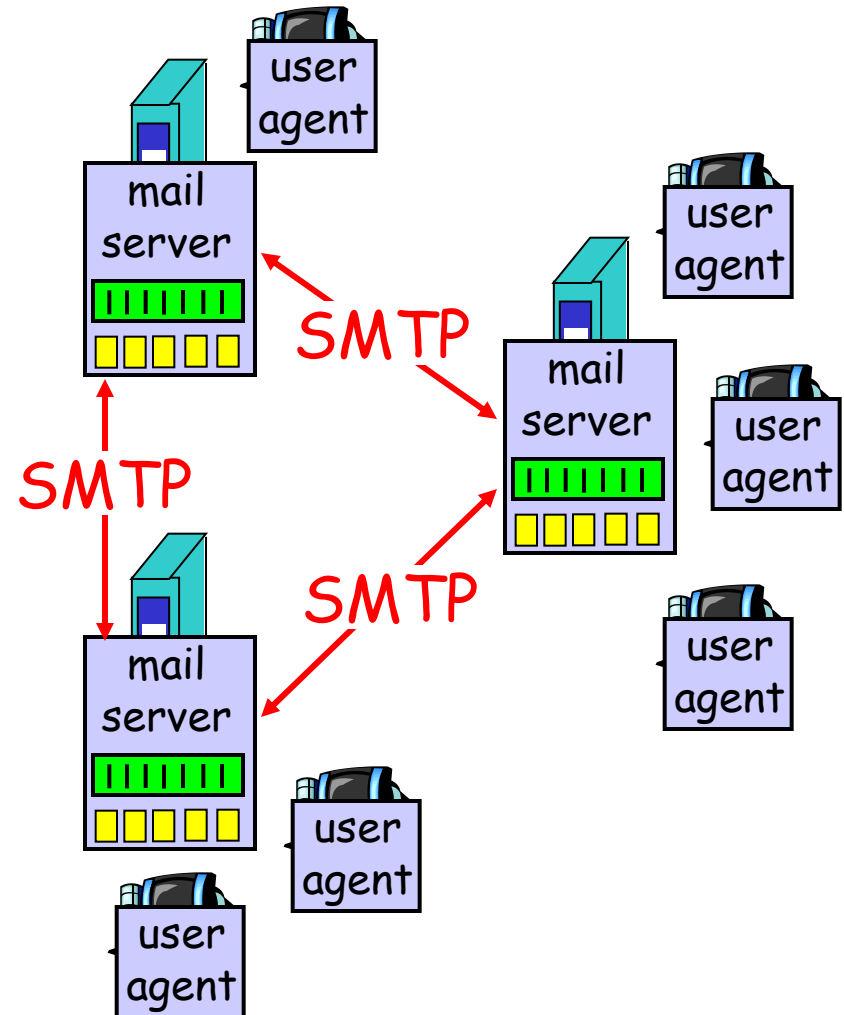
# Mail Servers

## Mail Servers

- ❑ **mailbox** contains incoming messages for user
- ❑ **message queue** of outgoing (to be sent) mail messages

**SMTP protocol** between mail servers to send email messages

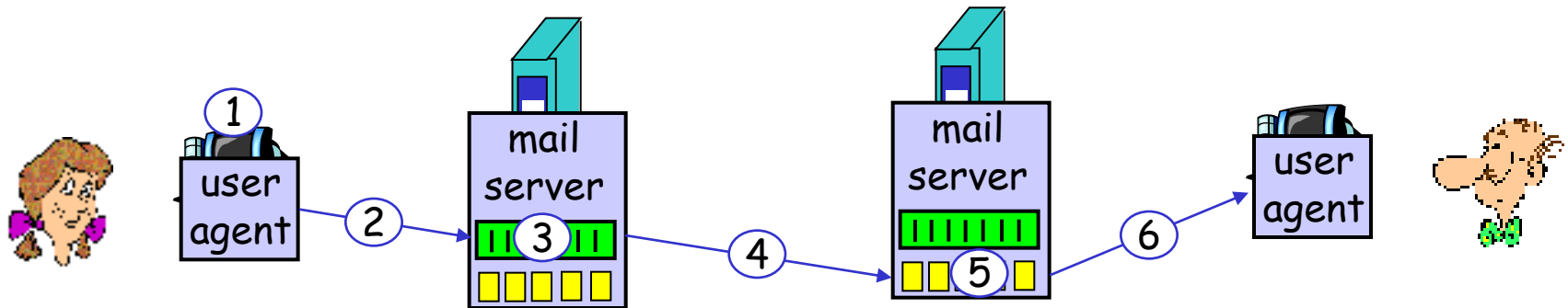
- ❖ client: sending mail server
- ❖ "server": receiving mail server





# Scenario: Alice sends message to Bob

- 1) Alice uses UA to compose message and "to" `bob@wayne.edu`
- 2) Alice's UA sends message to her mail server; message placed in message queue
- 3) Client side of SMTP opens TCP connection with Bob's mail server
- 4) SMTP client sends Alice's message over the TCP connection
- 5) Bob's mail server places the message in Bob's mailbox
- 6) Bob invokes his user agent to read message





# SMTP Interaction

## SMTP Client

telnet smtp.somedomain.com 25

HELO somename

MAIL FROM: someone@somedomain.com

RCPT TO: sarosh@kau.edu.sa

DATA

Subject: some message

From: someone@somedomain.com

To: sarosh@kau.edu.sa

This is a test email

.

QUIT

## SMTP Server

220 mail.server.com

250 HELO somename pleased to meet you!

250 Sender OK

250 Recipient OK

354 End data with "."

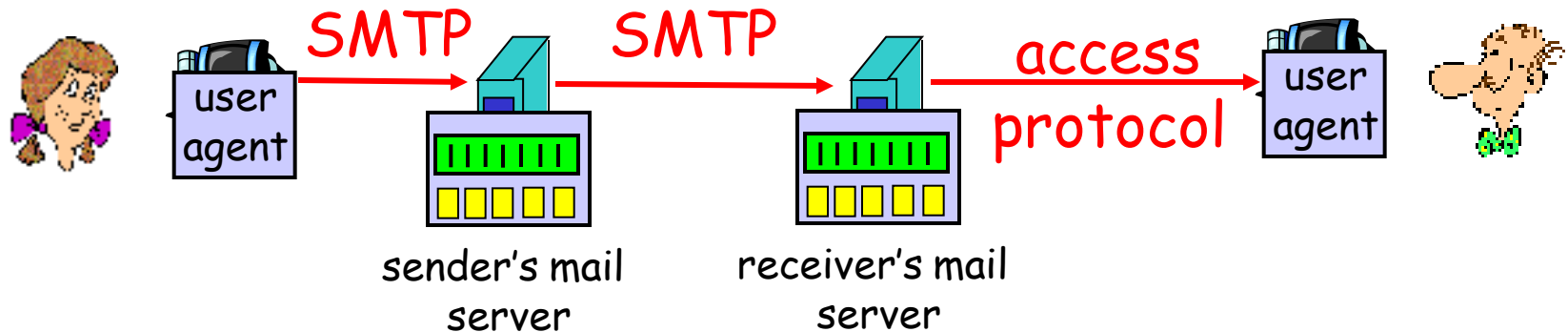
250 Accepted

250 OK: message queued for delivery

221 Bye



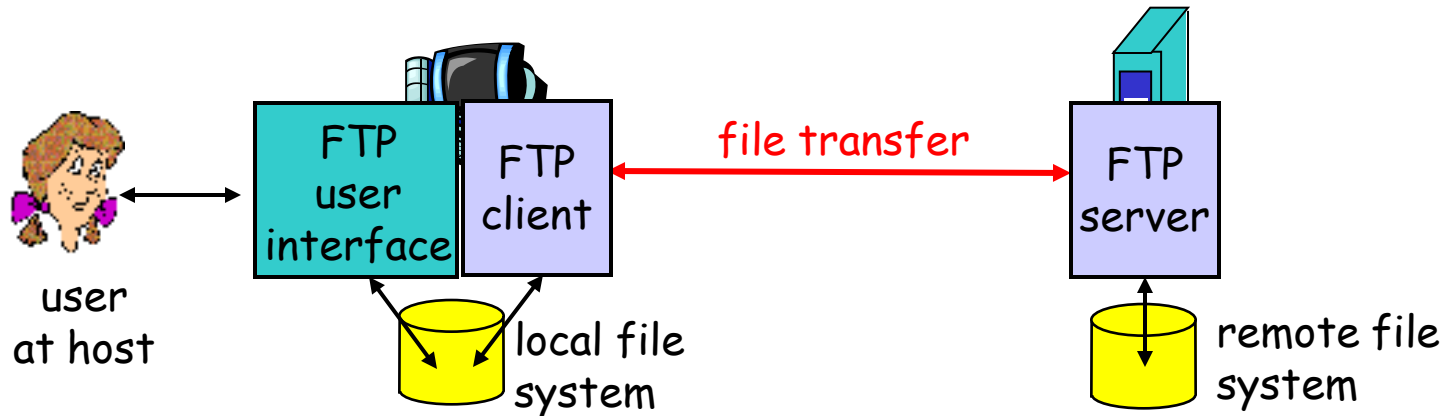
# Mail Access Protocols



- ❑ SMTP: delivery/storage to receiver's server
- ❑ Mail access protocol: retrieval from server
  - ❖ POP3: Post Office uses port 110
    - authorization (agent <-->server) and download
  - ❖ IMAP: Internet Mail Access Protocol
    - more features
    - manipulation of stored msgs on server
  - ❖ HTTP: Hotmail , Yahoo! Mail, etc.



# FTP: the file transfer protocol

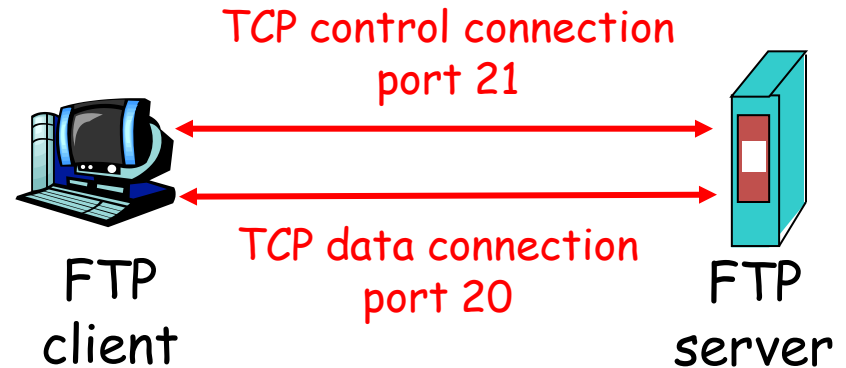


- ❑ transfer file to/from remote host
- ❑ client/server model
  - ❖ *client*: side that initiates transfer (either to/from remote)
  - ❖ *server*: remote host
- ❑ ftp server: port 21 for control, port 20 for data

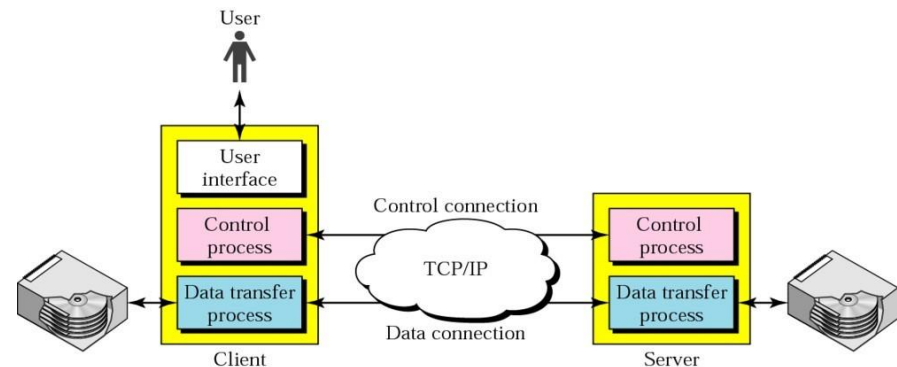


# FTP: separate control, data connections

- FTP client contacts FTP server at port 21, specifying TCP as transport protocol
- Client obtains authorization over control connection
- Client browses remote directory by sending commands over the persistent control connection.
- When server receives a command for a file transfer or directory listing, the server opens a TCP data connection to client
- After transferring one file, server closes connection.



- ❑ Server opens a second TCP data connection to transfer another file.
- ❑ FTP server maintains “state”: current directory, earlier authentication





# Lecture's outline

## Security Attacks

- a. **Malware**---attacks on integrity and privacy

Viruses, Trojan Horses, Spyware and Key-loggers

- b. **Spoofing attacks**---attacks on authenticity

URL, DNS, IP, MAC, Email/ Caller ID spoofing

- c. **Other attacks**--

DoS attack, worms, Non-technical hacking

## Security Mechanisms for Defense

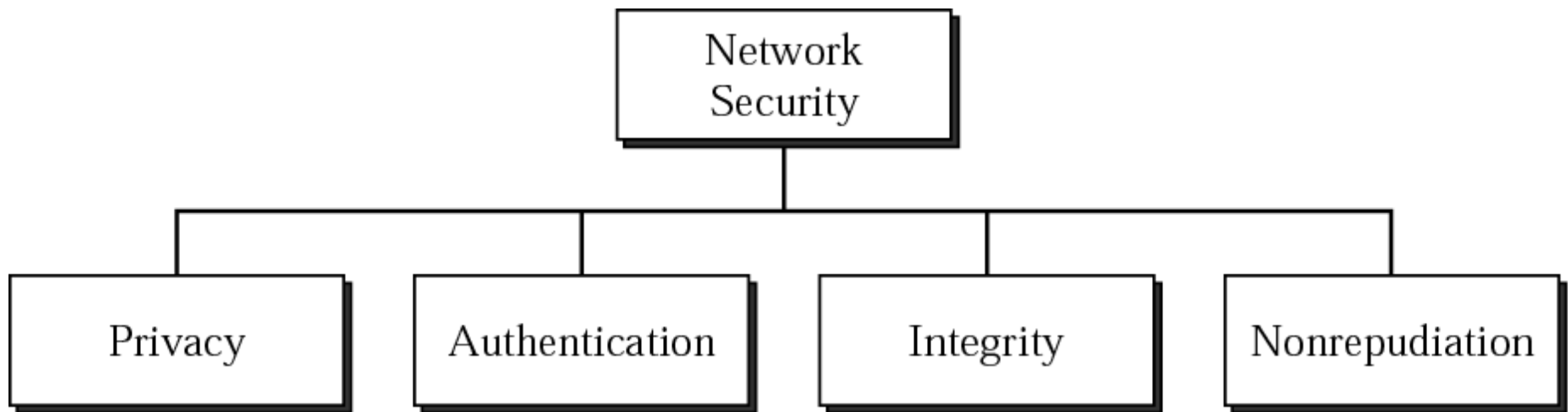
- a. **Encryption** – addresses **privacy** issues

Symmetric key **and** public key cryptography

- b. **Digital Signatures** – addresses **integrity**,  
**authentication** and **non-repudiation** issues



# Core Security Principles





# Core Security Principles

## Confidentiality/Privacy: (السرية / الخصوصية)

- The disclosure of information to unauthorized individuals or systems should be prevented by transforming the **transmitted message such that only the intended receiver can understand it and no one else.**

## Authentication: (المصادقة)

- Authentication means that **the receiver is sure of the sender's identity** and that an imposter has not sent the message.

## Integrity (استقامة):

- Integrity means that the **data must arrive at the receiver exactly as it was sent by the original sender.** There must be no changes in transmission, either accidental or malicious.

## Non-repudiation (عدم إنكار):

- Non-repudiation means that a receiver must be able to prove that a received message came from a specified sender. The **sender must not be able to deny sending a message that it has, in fact, sent.**



# Motivation for security attacks

Adversary	Goal
Student	To have fun snooping on people's e-mail
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by e-mail
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

**Source: "Computer Networks" by Andrew Tanenbaum**





# Malware

The software that is written for malicious purposes

Viruses

Trojan Horses

Spyware

Keyloggers



# Types of Malware

- A **computer virus** is a computer program that can copy itself and infect a computer without the permission or knowledge of the owner.
- A **Trojan horse**, or ***trojan*** for short, is a [malware](#) that appears to the user, to perform a desirable function but in fact, facilitates unauthorized access to the user's computer system.
- **Spyware** is a type of [malware](#) that is installed secretly on [personal computers](#) to collect information about users, their computer or browsing habits without their [informed consent](#).



# Key-loggers and Spyware



	<b>Screen Snapshots</b> 36 Screenshots Logged <input checked="" type="checkbox"/> Enabled
	<b>Clipboard Logs</b> 14 Clips Logged <input checked="" type="checkbox"/> Enabled
	<b>Keystrokes Typed</b> 38 Sessions Logged <input checked="" type="checkbox"/> Enabled
	<b>Passwords</b> 2 Passwords Logged <input checked="" type="checkbox"/> Enabled
	<b>Admin Actions</b> 26 Actions Logged <input checked="" type="checkbox"/> Enabled



# Email Spoofing

- Changing of sender address and other parts of email so that it appears to be coming from a different source other than the original sender. E.g. emails that appear to be sent from a well known organization (e.g. Microsoft), a bank etc.
- Email headers are altered by manually constructing email through manipulating fields in the SMTP protocol



Current Folder: INBOX

[Compose](#) [Addresses](#) [Folders](#) [Options](#) [Search](#) [Help](#)

Viewing Full Header

**Return-Path:** <[support@niit.edu.pk](mailto:support@niit.edu.pk)>

**Received:** from mail.wateen.net (mail.wateen.net [58.27.200.126])  
by webmail.niit.edu.pk (8.13.8/8.13.8/Debian-3) with ESMTP id n1FInmrCO12387  
for <[junaaid.qadir@niit.edu.pk](mailto:junaaid.qadir@niit.edu.pk)>; Sun, 15 Feb 2009 23:49:49 +0500

**Received:** from jq (unknown [10.196.158.31])  
by mail.wateen.net (Postfix) with SMTP id 094B86640CB6  
for <[junaaid.qadir@niit.edu.pk](mailto:junaaid.qadir@niit.edu.pk)>; Sun, 15 Feb 2009 23:43:53 +0500 (PKT)

**Subject:** SEECs mail account verification

**From:** System Support <[support@niit.edu.pk](mailto:support@niit.edu.pk)>

**Reply-To:** [support@niit.edu.pk](mailto:support@niit.edu.pk)

**To:** [junaaid.qadir@gmail.com](mailto:junaaid.qadir@gmail.com)

**Message-Id:** <20090215184400.094B86640CB6@mail.wateen.net>

**Date:** Sun, 15 Feb 2009 23:43:53 +0500 (PKT)

**MIME-Version:** 1.0

**X-wateen\_net-MailScanner-Information:** Please contact the ISP for more information

**X-MailScanner-ID:** 094B86640CB6.0AF15

**X-wateen\_net-MailScanner:** Found to be clean

**X-wateen\_net-MailScanner-From:** [support@niit.edu.pk](mailto:support@niit.edu.pk)

**X-Spam-Status:** No, No

**X-NIIT-MailScanner-Information:** Please contact the ISP for more information

**X-NIIT-MailScanner:** Found to be clean

**X-NIIT-MailScanner-From:** [support@niit.edu.pk](mailto:support@niit.edu.pk)



# URL Spoofing & Phishing Attack

- In URL (Uniform Resource Locator) Spoofing the address displayed in the address bar at the top of a browser is not really of the web page being displayed. For example the user may see `www.citibank.com` in the address location bar but really be on the web page `www.iamgoingtoroby.com`
- URL spoofing is done exploiting **bugs in the browser code** that enable him to write computer code to display whatever URL he wishes in the address location bar.
- The fake website looks and feels like the original website.
- If the fake website requests sensitive information like user name, password or credit card details, it is called a phishing attack. So phishing means acquiring sensitive user information by pretending to be a legitimate entity.



# DNS Spoofing

- Injecting fake entries in the cache of a DNS server resulting in the users being redirected to the fake websites instead of the original ones



# IP/MAC Address Spoofing

- In IP/MAC address spoofing attack the sender replaces the IP or MAC address in the data packets generated by its machine with the IP or MAC address of another machine.
- The addresses are changed to hide real identity of the node sending the data e.g. an attacker can launch a DNS spoofing or phishing attack using the IP address of another node instead of its own address.
- The addresses may also be changed to gain access to resources that allow admission based on some specific IP/MAC address.



# Worms

- Worms are a type of malware that replicate themselves from one computer to another using the network without using any files already present on the infected computer



# Denial of Service (DOS) attack

- In a DOS attack, an attacker (or attackers) repeatedly sends a large number of apparently valid requests to a target machine on the network so that it is unable to receive requests from other legitimate users and becomes inaccessible
- E.g. an attacker sending high volume of DNS queries to a root DNS server so that it becomes inaccessible to other network nodes



# Non-technical **hacking**

Windows Live™

Home

Profile

People

Mail

Photos

More ▾

MSN ▾

Search the web

## Reset your password

Select an option for resetting your password:

☒ Use my location information and secret answer to verify my identity

Country/region:

Question: place of birth

Secret answer:

Five-character minimum; not case sensitive

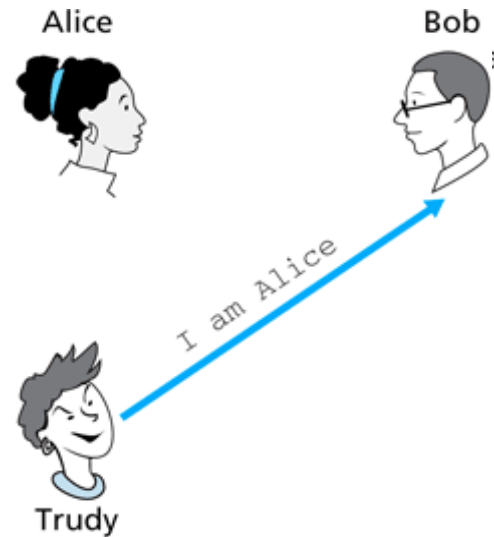
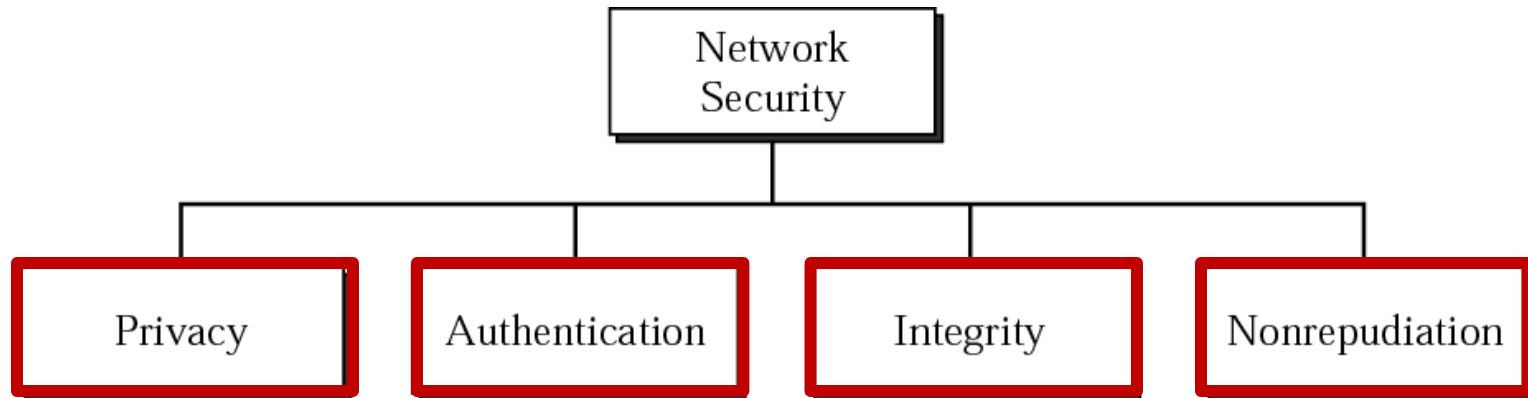
Continue

Cancel

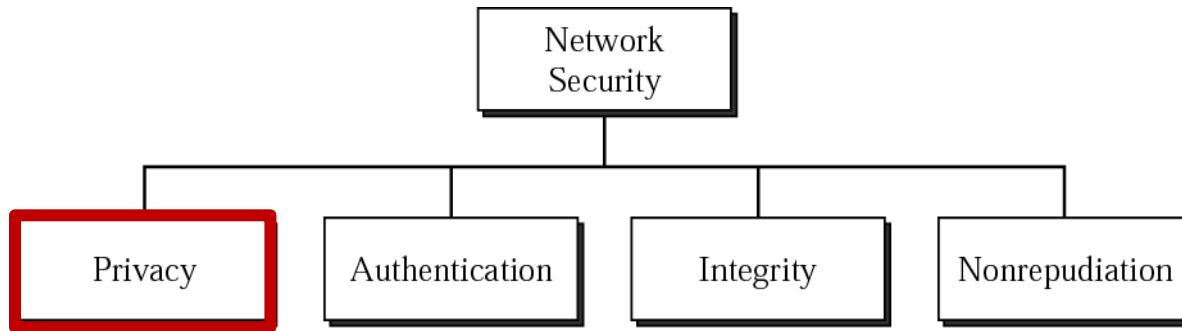
☐ Send password reset instructions to me in e-mail



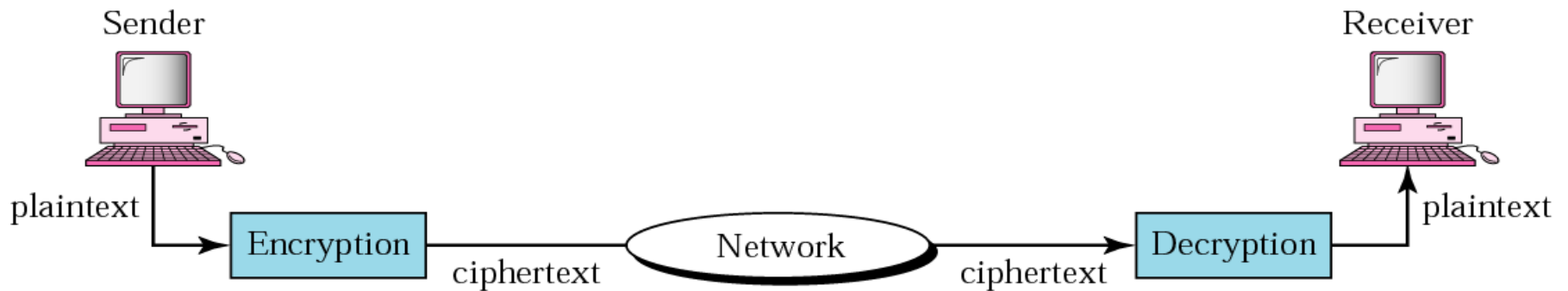
# Security Requirements







# Encryption





# Encryption

- In [cryptography](#), **encryption** is the process of transforming [information](#) using an [algorithm](#) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a [key](#).
- The reverse process, **decryption** can typically also perform decryption), to make the encrypted information readable again

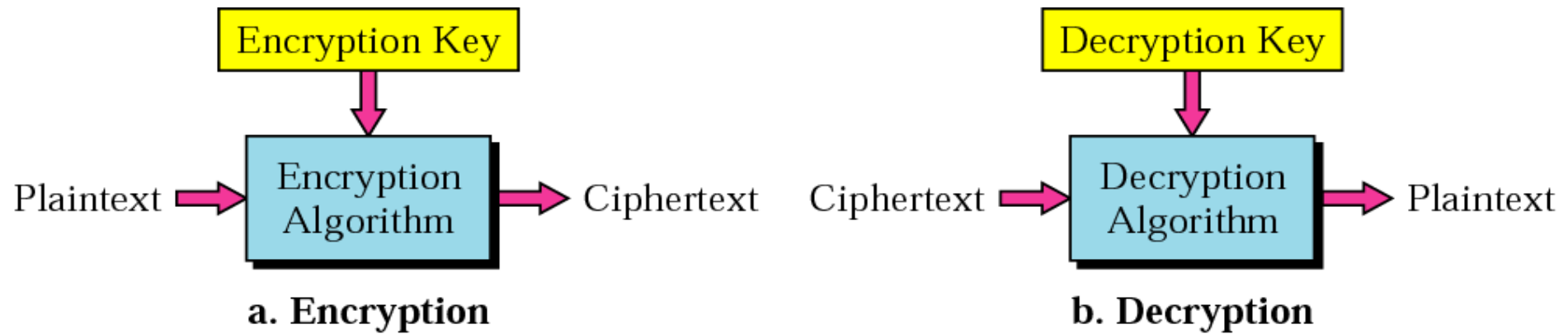


# Uses of Encryption

- Encryption is also used to protect data in transit, for example data being transferred via [networks](#) (e.g. the [Internet](#), [e-commerce](#)), [mobile telephones](#), [wireless microphones](#), [wireless intercom](#) systems, [Bluetooth](#) devices and bank [automatic teller machines](#).



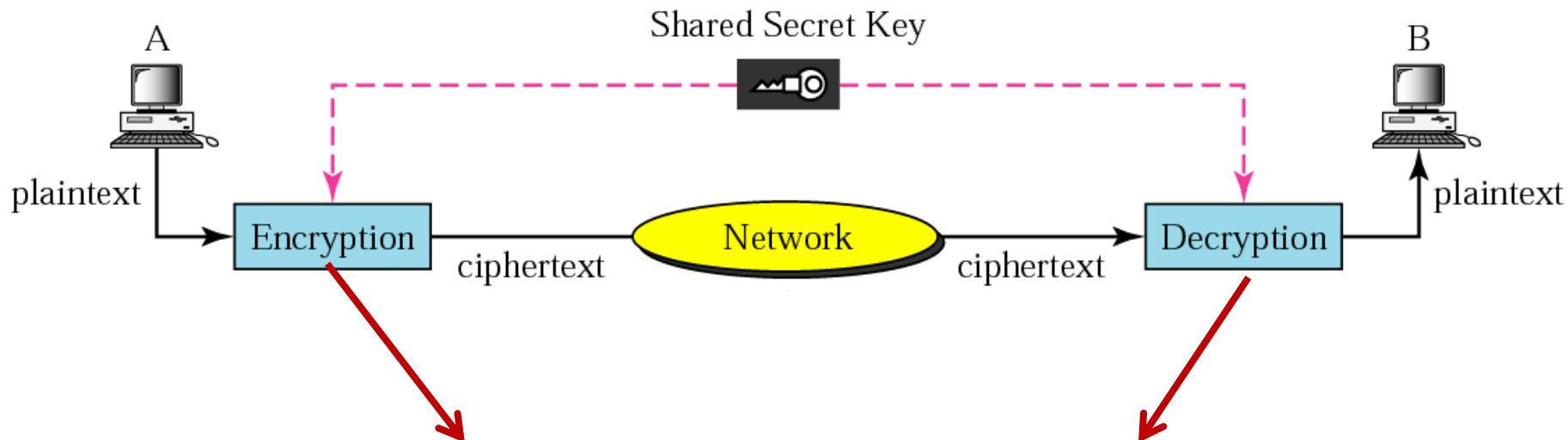
# What is Encryption/Decryption





# Secret Key Cryptography

Also known as symmetric key cryptography uses a single key for both encryption and decryption



**Public algorithms (usually) that are each other's inverse**

that means if the encryption algorithm uses a combination of addition and multiplication, the decrypting algorithm would use a combination of subtraction and division.

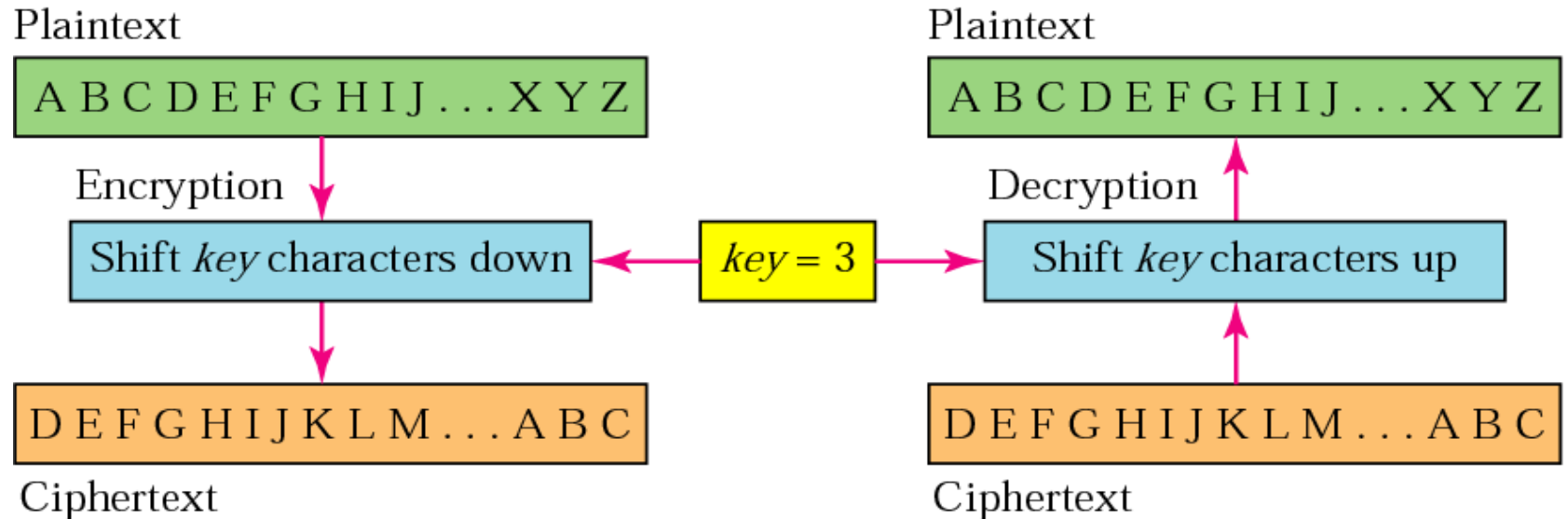
**Advantage:** Relatively quick

**Disadvantage:** Communicating pairs have to share keys



# Example of Secret Key Cryptography

## Caesar's Cipher



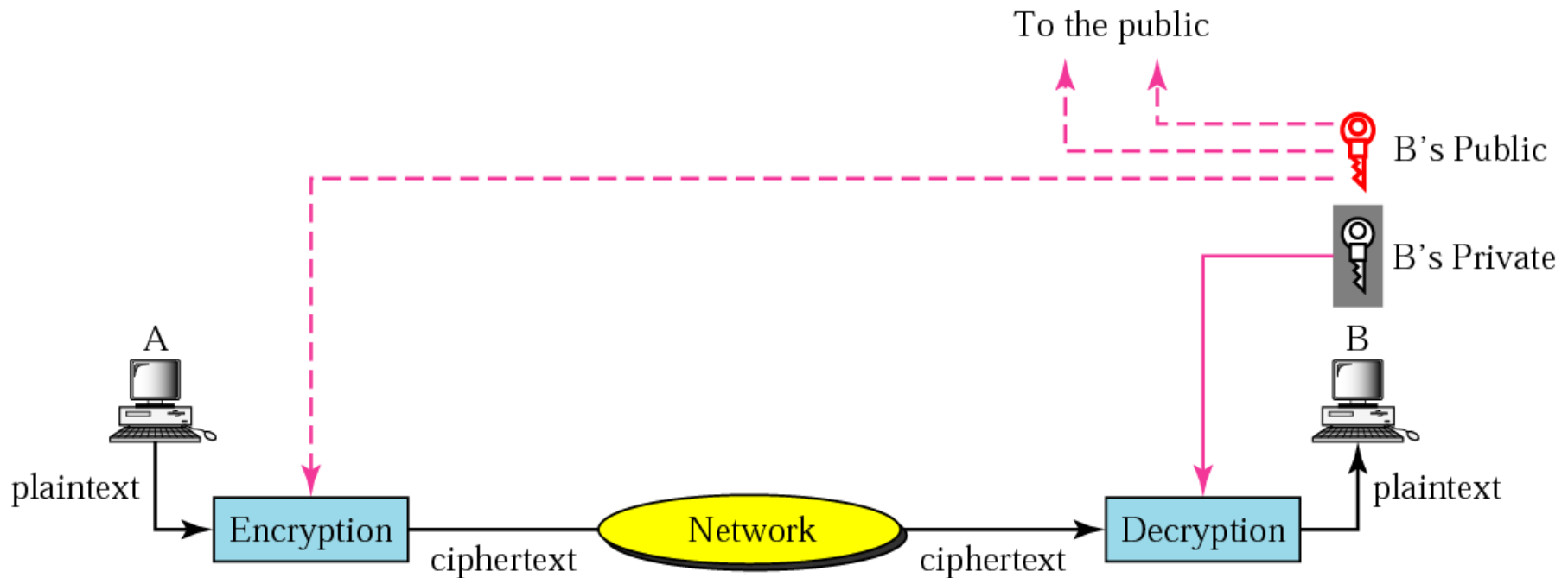


# Public Key Cryptography

- **public-key/two-key/asymmetric cryptography involves the use of two keys:**
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and
  - a **private-key**, known only to the recipient, used to **decrypt messages**,
- **is asymmetric because**
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures



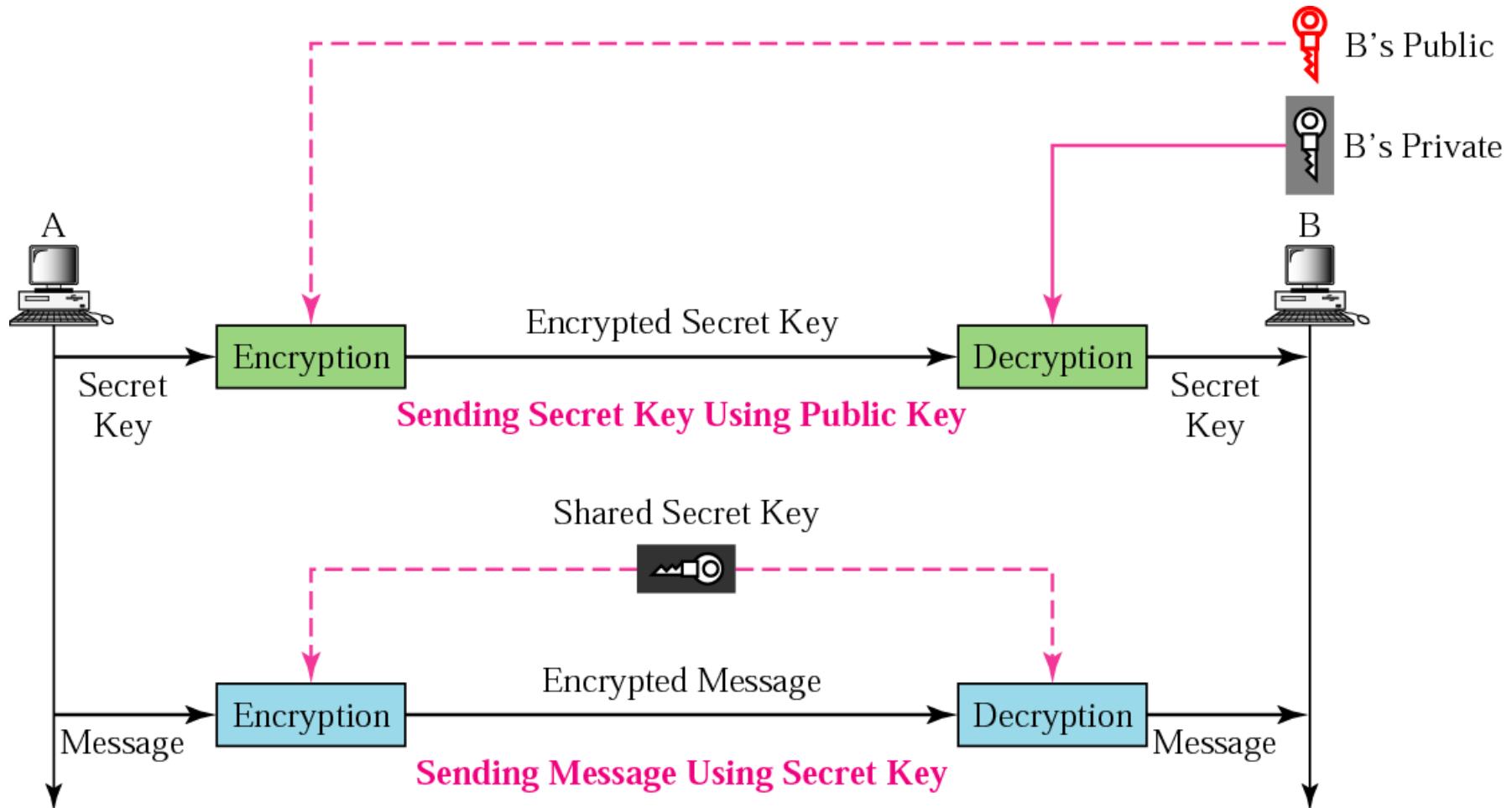
# Public Key Cryptography



**The key to encrypt is different from key that decrypts**

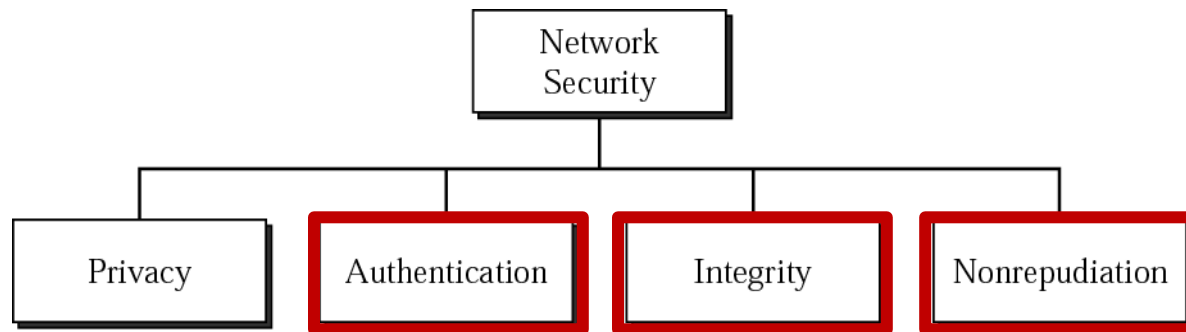


# Hybrid Asymmetric/Symmetric





# Digital Signature





# Digital Signature

- A **digital signature** or **digital signature scheme** is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

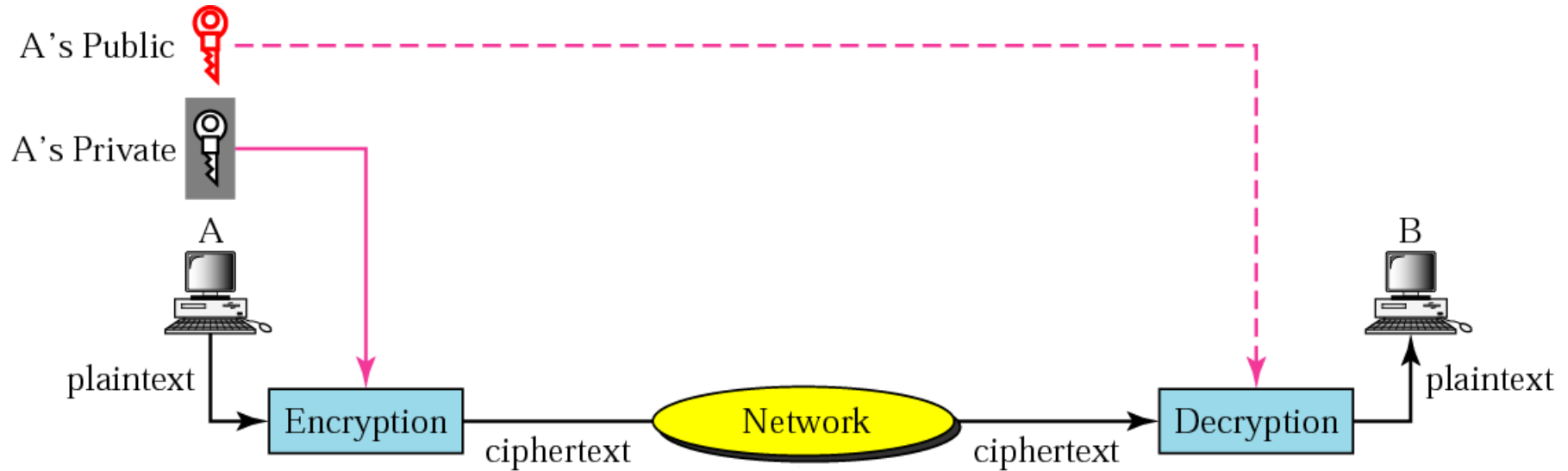


# Digital Signature

- In digital signature, the private key is used to for encryption and the public key for decryption.
- If A encrypted some message with its private key, only the public key for A can be used for decryption. Hence, one is sure that it is A who signed it.

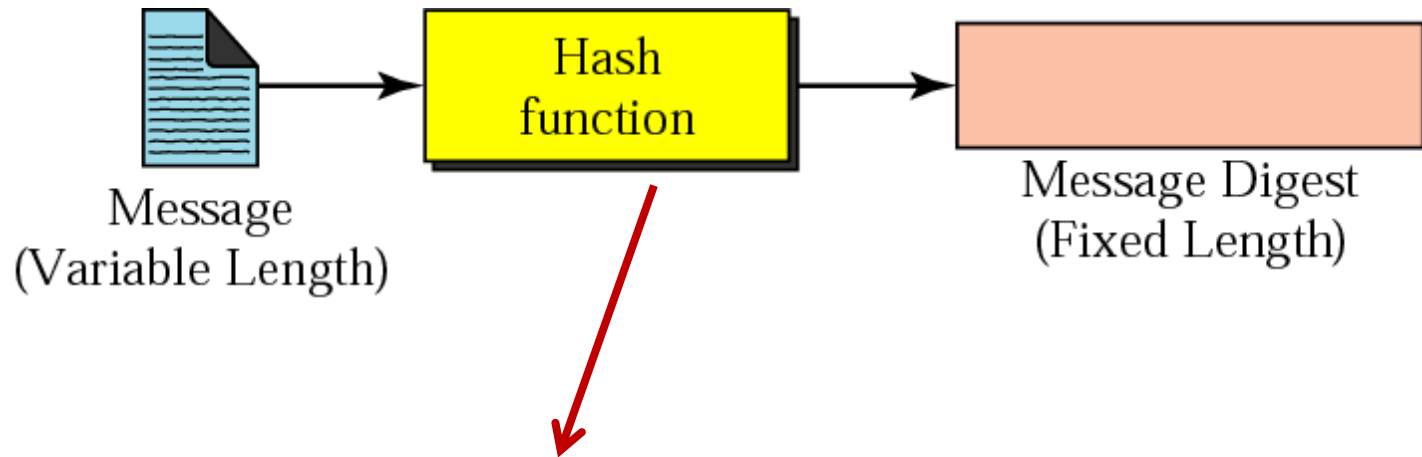


# Signing the whole document





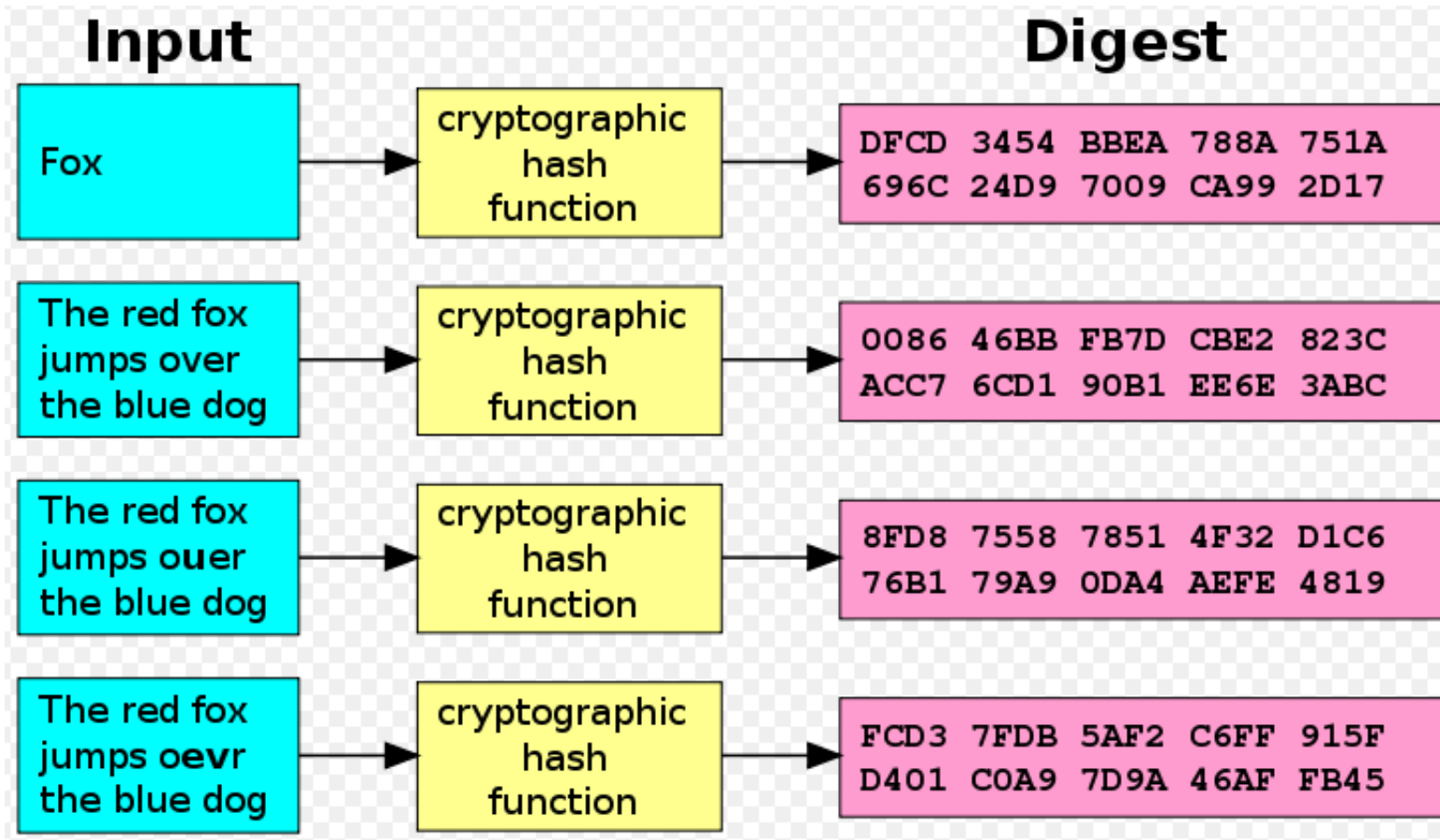
# Signing the digest



Most common hash functions are MD5 and SHA-1



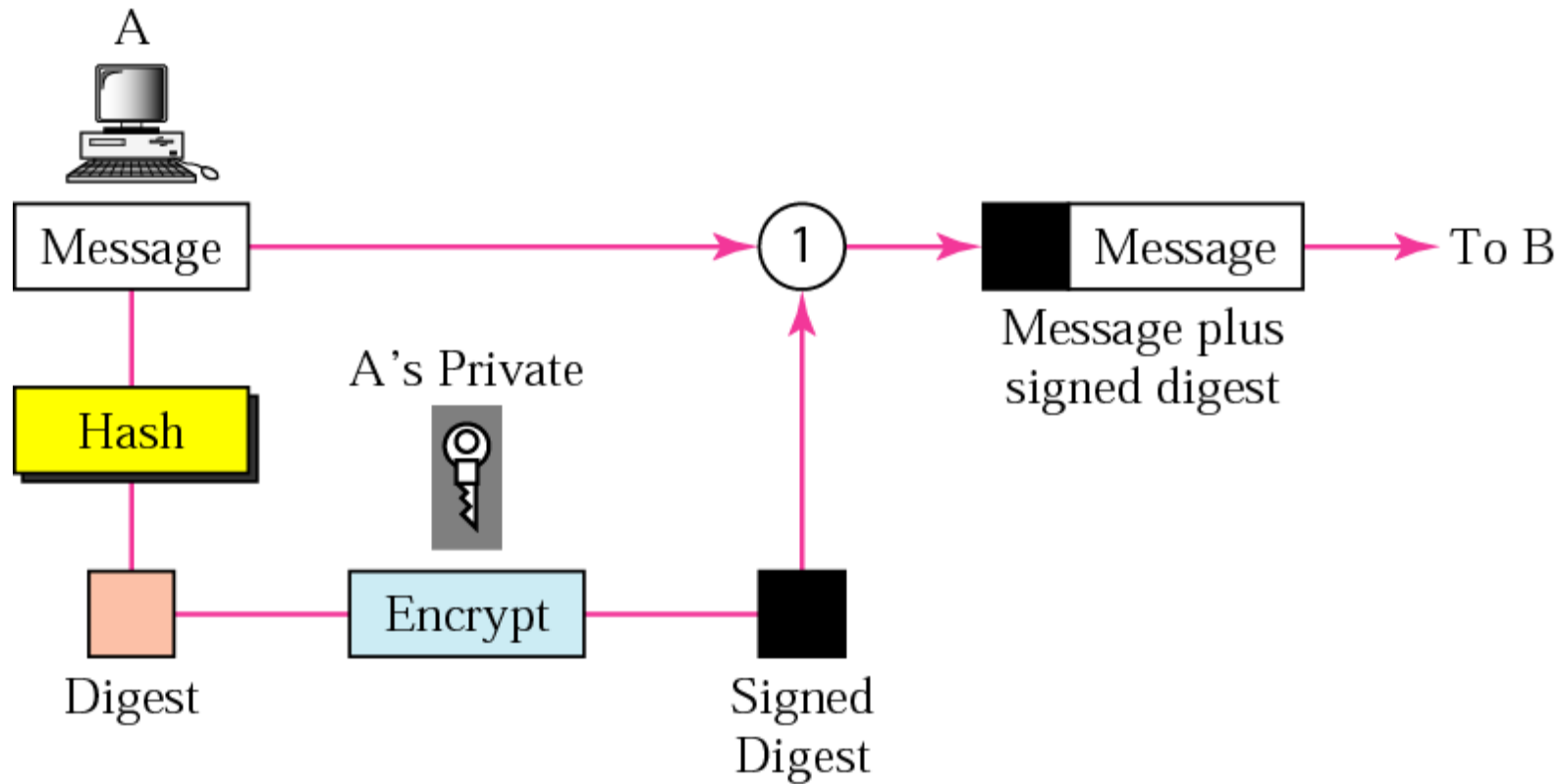
# Signing the digest



Source: [Wikipedia](#) page on Cryptographic Hash Function

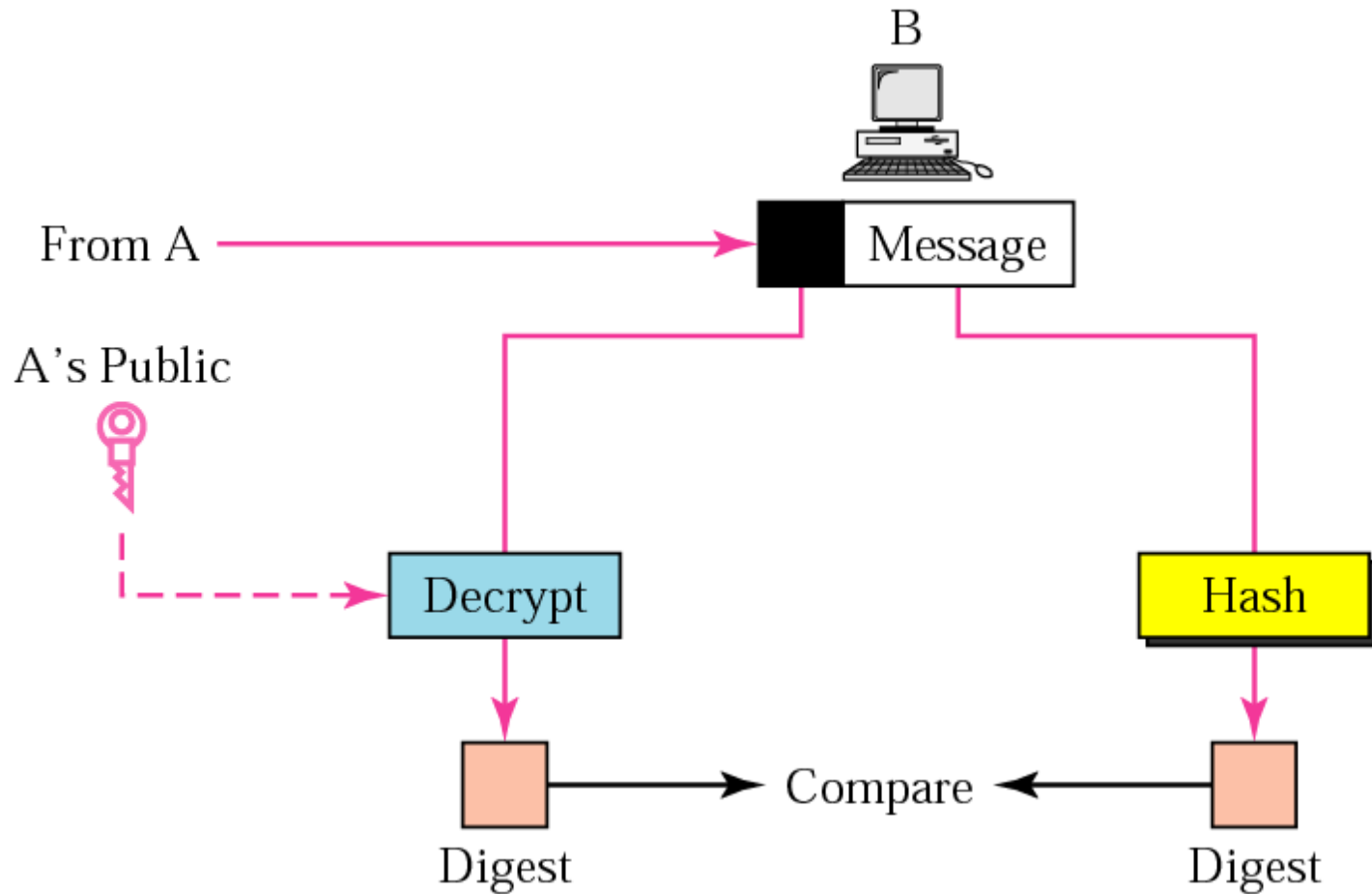


# Sender Site



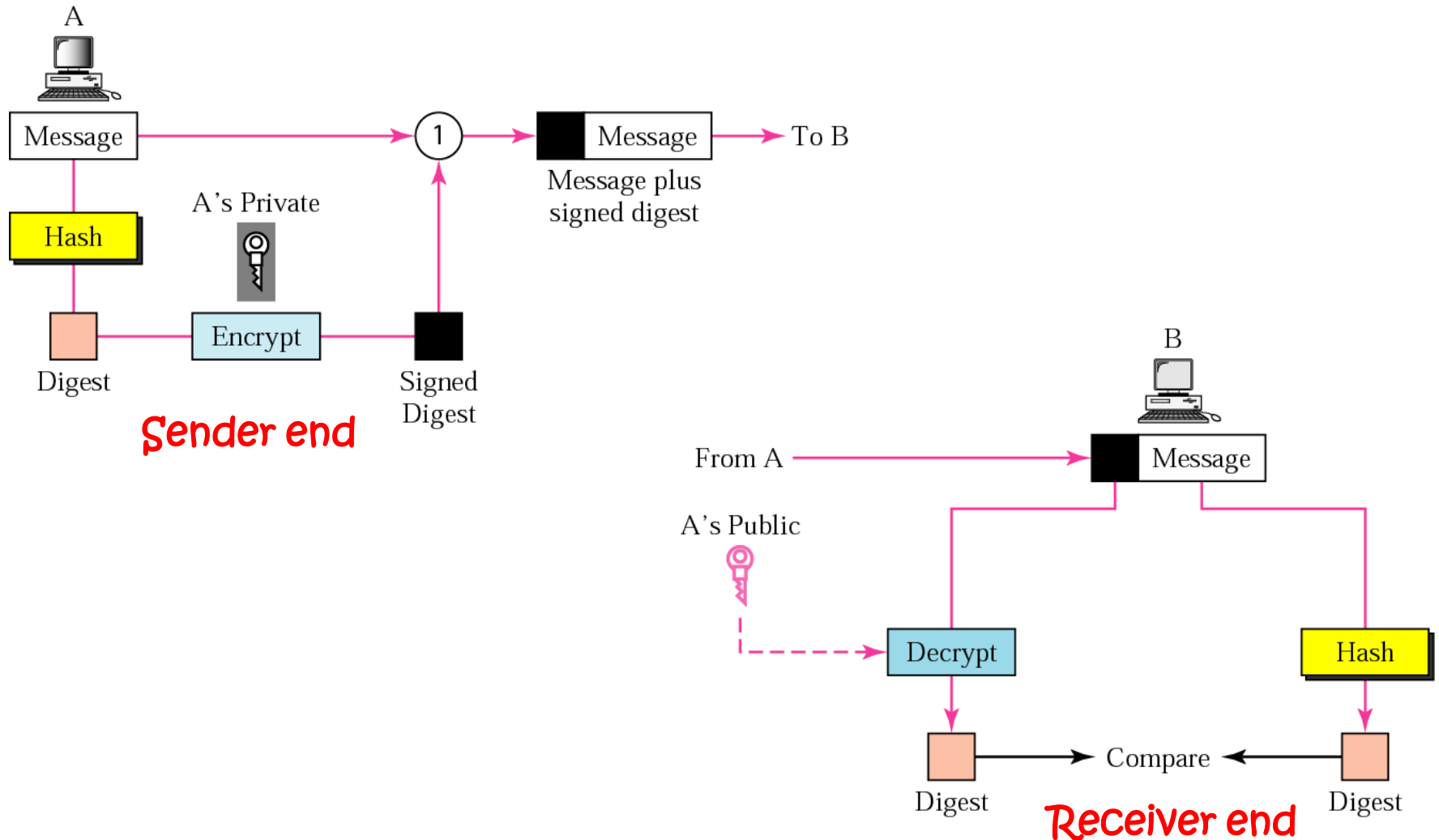


# Receiver Site





# Digital Signatures





# Advantages of Digital Signatures

- In digital signature, private key is used for encryption and public key for decryption. Digital signature can provide integrity, authentication and nonrepudiation:
  - 1) Integrity: If an intruder intercepts and modifies the message, the decrypted message will be unreadable
  - 2) Authentication: Since the plaintext is encrypted with A's private key (whose information is only available with A), if any other public key is used, the resulting text would be garbage. If the right message is decrypted, that means the authentic source (having the right private key) had sent the message.
  - 3) Non repudiation: Similar reasoning as above.



# Firewalls



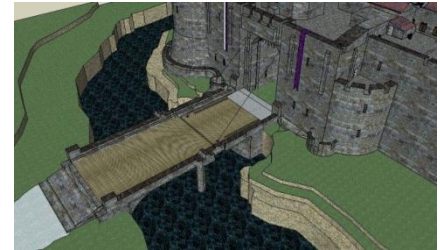
- Basic problem – many network applications and protocols have security problems that are fixed over time
  - Difficult for users to keep up with changes and keep host secure
  - Solution
    - Administrators limit access to end hosts by using a firewall
    - Firewall is kept up-to-date by administrators



# Firewalls

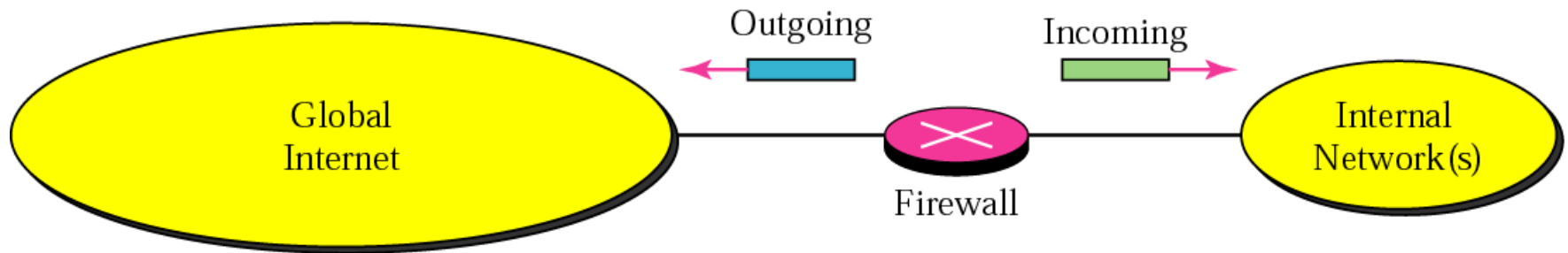


- A firewall is like a castle with a drawbridge
  - Only one point of access into the network
  - This can be good or bad
- Can be hardware or software
  - Ex. Some routers come with firewall functionality
  - ipfw, ipchains, pf on Unix systems, Windows XP and Mac OS X have built in firewalls





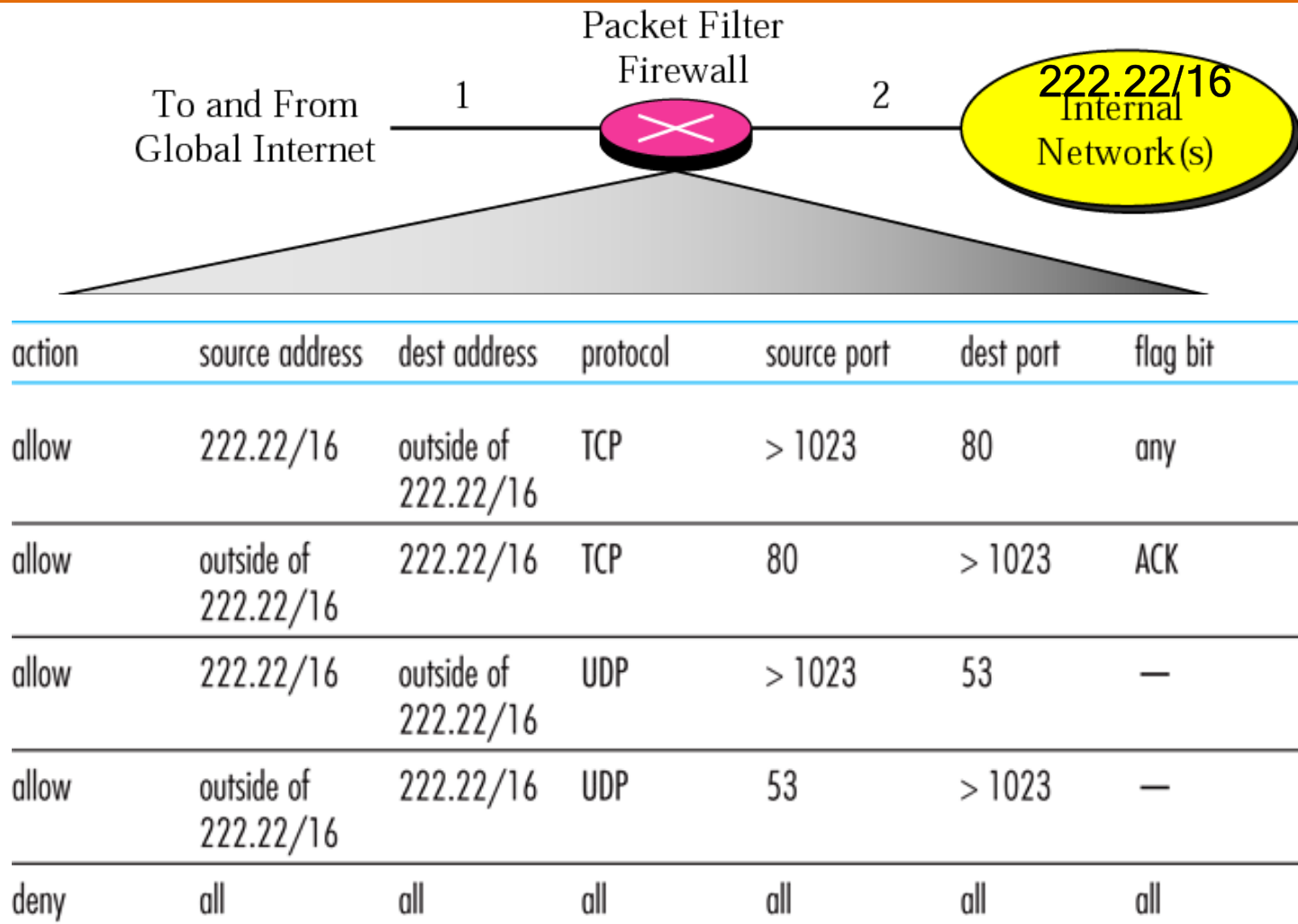
# Firewalls



A firewall is a router installed between the internal network of an organization and the rest of the Internet. It is designed to **forward some packets and filter (not forward) others.**



# Packet Filter Firewall



The access list above defined on a firewall has an implicit deny all and only allows the few packets that match the lines above the last line.