# Introduction to TCP/IP

**Farooq Ashraf**

farooq@ccse.kfupm.edu.sa

Department of Computer Engineering
King Fahd University of Petroleum and Minerals

# The Internet

- The Internet is the largest and most popular global network.
- It is a network of networks.
- July, 1998: over 36 million networks.
- Jan, 1999: 157 million people online
- Projected to be 327 million by year 2000.

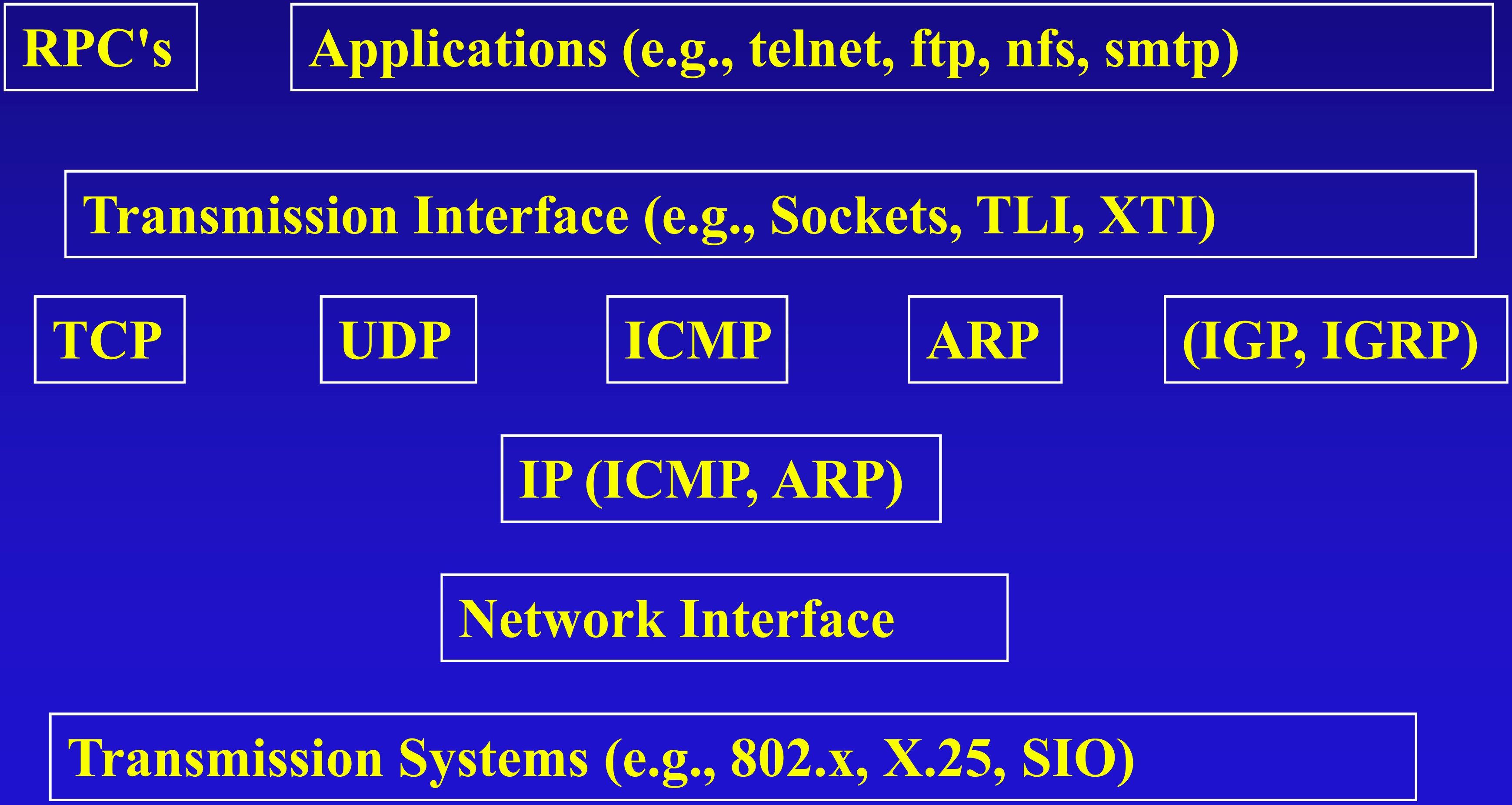
# The Internet (cont.)

- The Internet is connected using dedicated communication links (copper, fiber, satellite)
- Almost all hosts connected to the Internet speak TCP/IP.

# TCP/IP

- TCP/IP is an entire set of data communications protocols
- TCP and IP are two of these protocols
- IP: Internet Protocol.
- TCP: Transmission Control Protocol.
- There are many other protocols in this suite.

# otocols in the TCP/IP Suite



RPC's

Applications (e.g., telnet, ftp, nfs, smtp)

Transmission Interface (e.g., Sockets, TLI, XTI)

TCP

UDP

ICMP

ARP

(IGP, IGRP)

IP (ICMP, ARP)

Network Interface

Transmission Systems (e.g., 802.x, X.25, SIO)

# TCP/IP Features

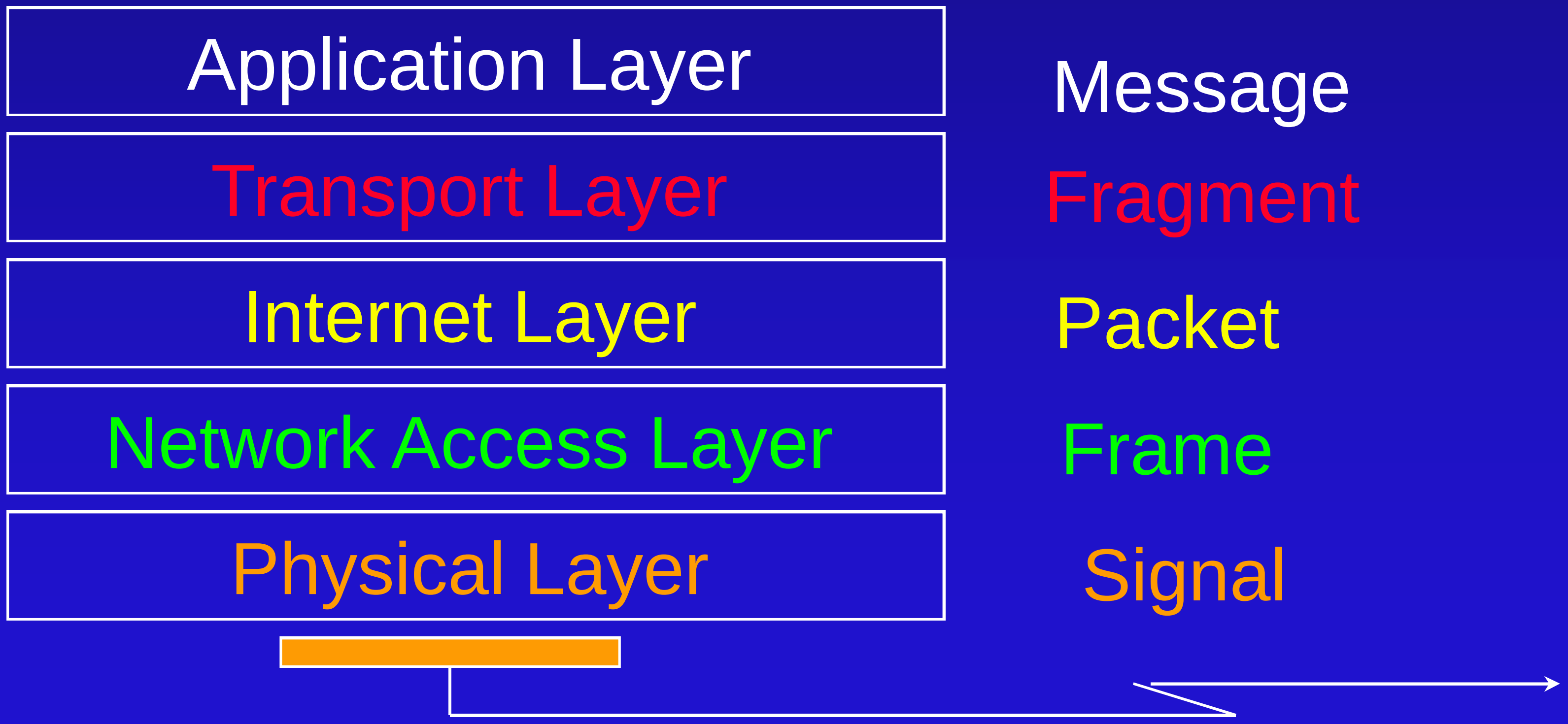
- Popularity of TCP/IP
  - » provides an elegant solution to world wide data communication.
  - » DARPA funding of ARPANET to provide robust communications resulted in TCP/IP
  - » TCP/IP became a defacto standard
- TCP/IP has Open Protocol Standards: freely available, and independent from any hardware platform.

# TCP/IP Features (cont.)

- Independence from specific network hardware
  - TCP/IP allows many types of networks to be integrated (Ethernet, Token Ring, X.25)
  - TCP/IP is used in both LANs/ and WANs
  - Supports dial-up connectivity
- Common addressing scheme
  - Every TCP/IP host has a unique address
- Standardized high-level protocols for world wide available network services

# TCP/IP Protocol Architecture

- **Layered architecture**



| | |
|---|---|
| Application Layer | Message |
| Transport Layer | Fragment |
| Internet Layer | Packet |
| Network Access Layer | Frame |
| Physical Layer | Signal |

# Application Layer

- Includes all software programs that use the Transport Layer protocols to deliver data messages
- Examples of protocols:
  - » Telnet: Network Terminal Protocol
  - » FTP: File Transfer Protocol
  - » SMTP: Simple Mail Transfer Protocol
  - » DNS: Domain Name Service
  - » HTTP: World Wide Web (WWW)

# Transport Layer

- Interface between the Application and Internet layers
- Two main protocols
  - » Transmission Control Protocol (TCP)
    - ï Provides reliable end-to-end data delivery service, connection-oriented
  - » User Datagram Protocol (UDP)
    - ï Provides low overhead connection-less datagram delivery service

# Internet Layer

- Heart of TCP/IP
  - » Provides basic packet delivery service on which TCP/IP networks are built
- Main functions
  - » Defines datagram, basic unit of transmission in the Internet
  - » Provides Internet addressing
  - » Routing of datagrams

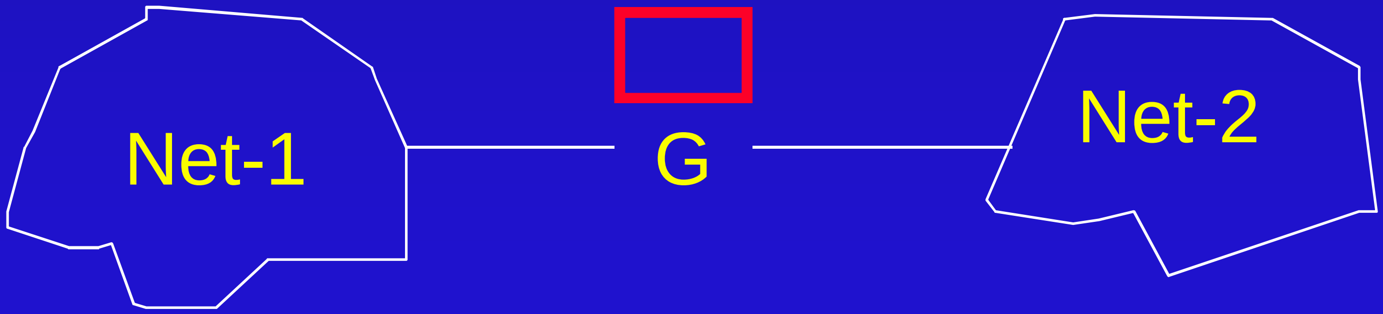
# Internet Layer (cont.)

- » Interfaces the Transport layer and Network Access layer
- » Performs fragmentation and re-assembly of datagrams

- ● IP is an unreliable protocol
  - » no error control

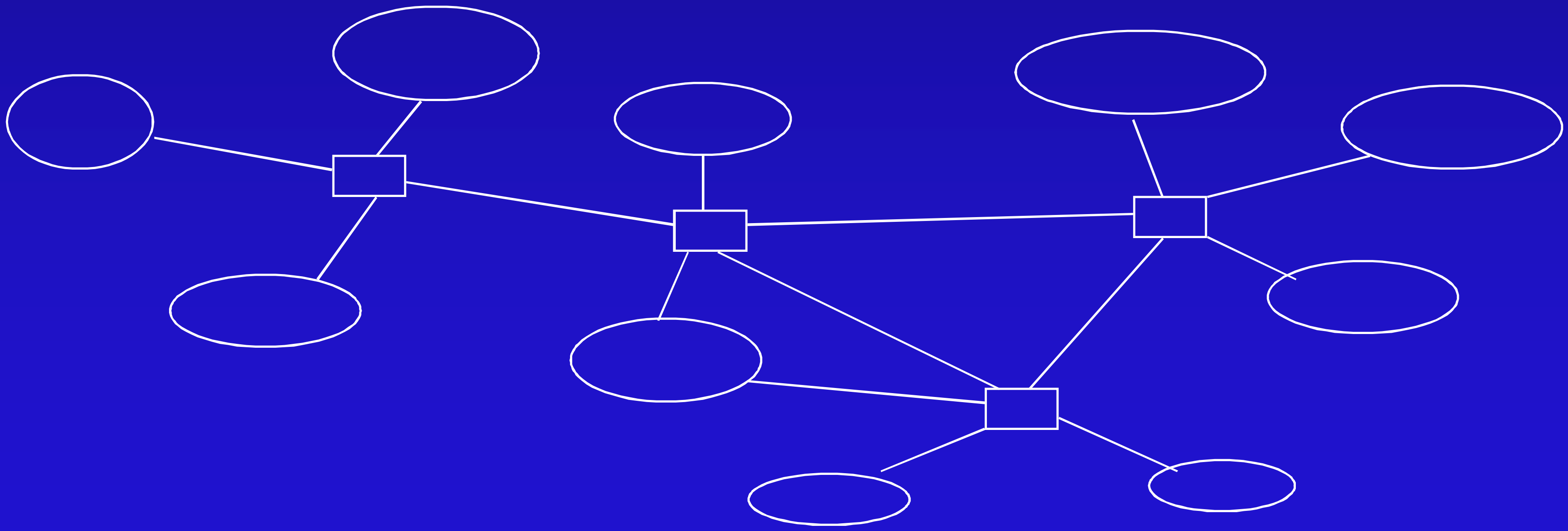
# Internetworking

- Network: Any communication system capable of transferring packets
- Internet Gateways/Routers are used to connect networks together.



# Internetworking (cont.)

- For complex interconnections, gateways must have knowledge of internet topology



# Internetworking (cont.)

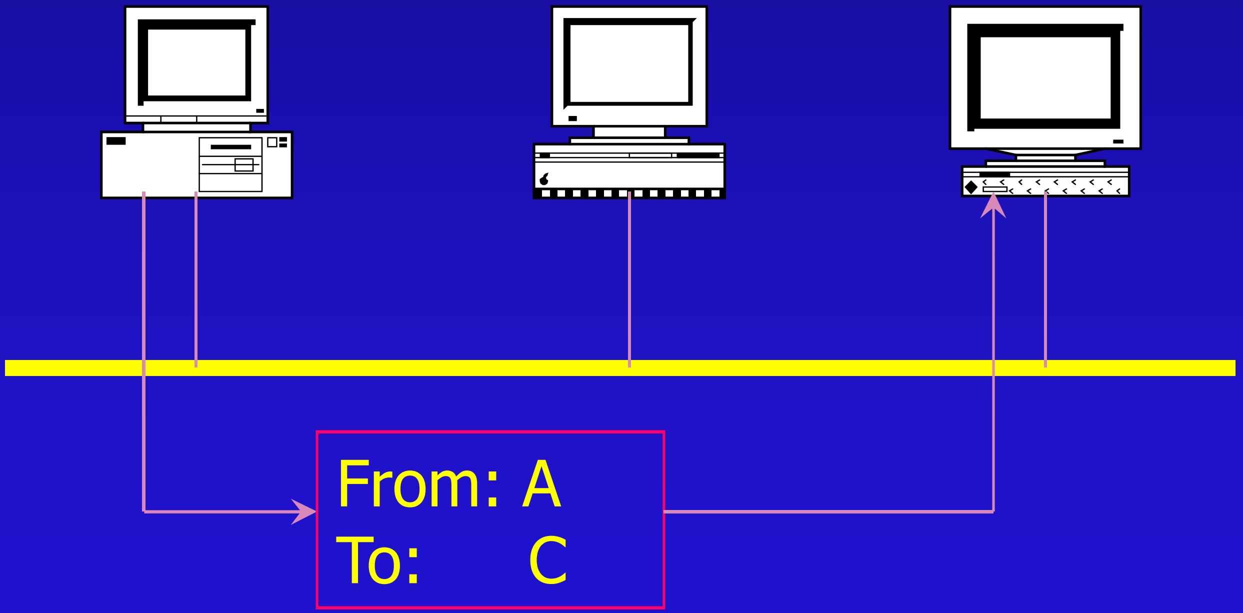
- Gateways route packets based on destination network not on destination host
- Besides the gateways, internet access software is needed on each host to allow application programs to see the internet as a single virtual network
- Application software remains unaffected by changes to the internet

# Important questions

- How are the machines addressed?
- How do internet (IP) addresses relate to physical addresses?
- How do internet gateways learn about routes?

# Simple Addressing

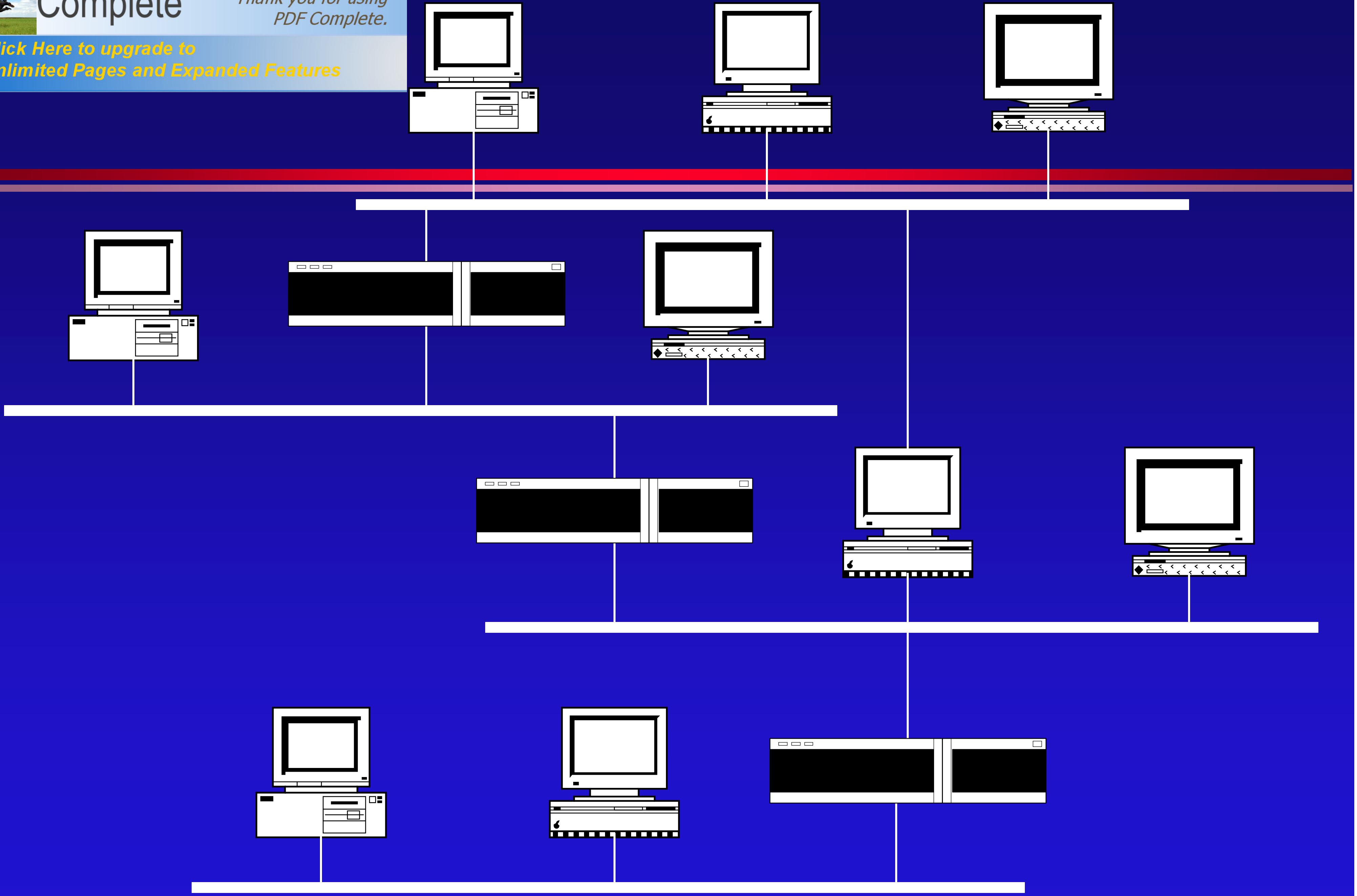
- On simple networks, delivery of messages between devices is quite simple.



- When A wants to send a message to C, A simply adds Côs device address to the message and puts the message on the network.
- If C sees a message that bears its device address, it can retrieve the message.
- However, this is only in the case of very simple, rather trivial networks.
- In complex networks, this mechanism would fail.

# Complex Addressing

- Letôs look at a fairly involved network.
- With this network, a message from A that was addressed to B could take several routes to reach its destination.
- Every place that networks interconnect, devices called **routers** are placed.



- The figure shows a networks of networks, which is commonly referred to as an **internetwork**.
- An internetwork consists of the following elements, in terms of addressing.
  - » A unique address for every device.
  - » Local delivery mechanism.
  - » Message delivery across the internetwork.
  - » Mechanism to determine the best possible path.

# Internetwork Addressing

- Each device on a network or an internetwork is identified by a unique address, often called a **device** or **node address**.
- These addresses are frequently permanently hard-coded into the network hardware.
- Each Ethernet and Token-Ring interface possesses a 48-bit address guaranteed to be unique throughout the world.

- A local delivery mechanism enables devices to place messages on the medium and retrieve messages that are addressed to them.
- This local delivery is performed by using the device address.
- The local delivery is handled by the physical and data link layers.

- A mechanism is also needed for delivering messages that must cross network boundaries and travel through the internetwork.
- Internetworks can be very complex, so there must be a way to find out the best possible path from one node to another across the internetwork.
- This process of finding the best possible paths is referred to as **routing**.

# TCP/IP-based Internetworks

- Where does TCP/IP come into picture?
- It is one of the ways of handling the problems mentioned earlier.
- Itôs of course not the only one, however it provides an excellent and simple approach with the widest acceptance.
- TCP/IP consists of the layers above and including the network layer.

- The lower layers (physical and data link) can be of many types, such as Ethernet, Token-Ring, X.25, Frame Relay, ATM, Serial Line etc.
- TCP/IP was designed explicitly without data link and physical layer specifications because the goal was to make it adapt to most types of physical media.
- TCP/IP relies on the physical layer to deliver messages on the local network.

- For delivering messages across network boundaries, TCP/IP has its own addressing mechanism.
- This mechanism works at the network layer, and is handled by the IP (Internet Protocol) software.
- In TCP/IP terminology, any device that is connected to the network is referred to as a **host**.
- A host may be, a computer, router, network printer, etc.

# Local Message Delivery

- When IP sends a message that is directed to a device on the local network, it hands the message over to the physical layer software which tags the message with the physical address of the recipient, and sends.
- The device that matches the physical address retrieves the message.

# Message Routing

- When a message is not destined for a device on the local network, it must be routed.
- TCP/IP assigns an address to each host and to each network.
- Each host is configured with a default router to which it sends messages that must be sent to a remote network.

- The responsibility of determining how messages should be addressed is one of the tasks of the IP layer.
- IP identifies whether a message is destined for a host on the local network or it should be sent to the default router.
- It makes use of addresses called **IP addresses** to logically identify networks and hosts.

- The physical address of either the local host or the default router is added by the physical layer software to each message that is sent.
- IP receives data from the higher level protocols, and attaches to each data segment a header containing addressing information.
- The combination of data from higher layers with the IP header is referred to as a **packet**.

- Determining routing paths between routers is usually the responsibility of one of the following two protocols.
  - » Routing Information Protocol (RIP)
  - » Open Shortest Path First (OSPF) IP receives data from the higher level protocols, and attaches to each data segment a header containing addressing information.

# IP Addresses

- IP addresses, unlike hardware address, are not hard-coded into hosts.
- Assigned by network administrators to each network interface and configured into software running on networked hosts.
- Independent of the physical layer.
- A host can retain its IP addresses, even though its physical address changes.

# IP Address Format

- IP addresses are 32-bit integers containing both the network address and a host address.
- An example IP address is:

```
11000001000010100001111000000010
```

- This is not easy to read or remember.
- It is even hard to identify differences in two such numbers.

- To make IP address easier to work with, the 32-bit addresses are typically divided into four parts called **octets**.

  11000001 00001010 00011110 00000010

- Each of these octets can be translated into a decimal number in the range of 0 to 255.

- This leads to the more human-readable representation of an IP address.

  193.10.30.2

- This format is known as the **dotted-decimal notation**.
- This is simply for the ease of human users.
- The hosts still convert these octets into the binary form seen earlier.

# IP Address Classes

- Each IP Address consists of two fields.
  - » A **network id** field, which is the logical network address to which the host belongs.
  - » A **host id** field, which is the logical address that uniquely identifies each host on a network.
- Together, the network id and the host id, provide each host on and internetwork with a unique IP address.

- When TCP/IP was originally designed, it was thought that computer networks would fall into one of three categories.
  - » A small number of networks that had a large number of hosts.
  - » Some networks with an intermediate number of hosts.
  - » A large number of networks that had a small number of hosts.

- Because of this, IP addresses were organized into **classes**.
- The class of an IP address would be identified by looking at its first octet.
  - » If the first octet has a value between 0 and 127, it is a **class A** address.
  - » If the first octet has a value between 128 and 191, it is a **class B** address.
  - » If the first octet has a value between 192 and 223, it is a **class C** address.

- In class A, 0 and 127 in the first octet have special uses, so only values between 1 and 126 can be used.
- The number of hosts that a class can support depends on the way the class allocates octets to subnet ids and host ids.

| | | | | |
|---|---|---|---|---|
| Class A | NNNNNNNN | HHHHHHHH | HHHHHHHH | HHHHHHHH |
| Class B | NNNNNNNN | NNNNNNNN | HHHHHHHH | HHHHHHHH |
| Class C | NNNNNNNN | NNNNNNNN | NNNNNNNN | HHHHHHHH |

- Class A can support up to 16,777,214 hosts and 254 networks.
- Class B can support up to 65,534 hosts and 65,536 networks.
- Class C can support up to 254 hosts and 16,777,214 networks.

- Technically, the class of an address is defined by the leftmost bits in the first octet.
  - If the first bit is a 0, the address is class A.
  - If the first two bits are 10, the address is class B.
  - If the first three bits are 110, the address is class C.
  - If the first four bits are 1110, the address is class D.
  - If the first four bits are 1111, the address is class E.

- Classes D and E are not available for standard network addressing.

# Special IP Addresses

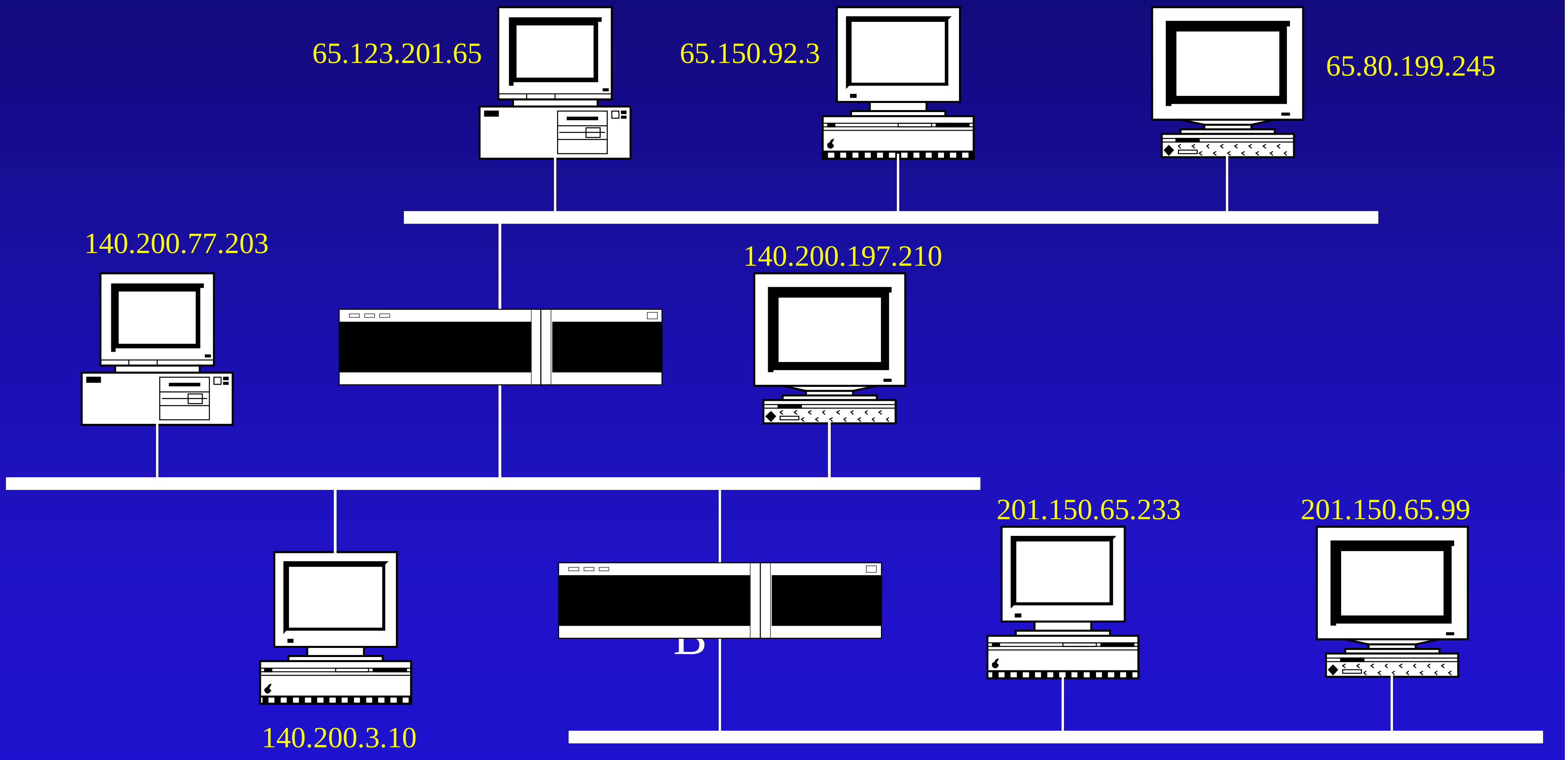
- There are several IP addresses that are reserved for special purposes and are not available for assignment to hosts.
- Any address with a first octet value of 127 is a **loopback** address.
- A loopback address is used by a host to communicate with itself through TCP/IP.
- It is also used for testing and diagnostics.

- 255 in the last octet of either the host id or the network id designates a broadcast or multicast.
- A message sent to 255.255.255.255 is broadcast to every host on the local network.
- A message sent to 196.1.64.255 is multicast to every host on network 196.2.64.

- The first octet cannot be greater than 223.
- The last octet of a host id cannot be 0 or 255.

# Networks and Network IDs

- Every host on a TCP/IP network must be configured with the same network id.
- It is a requirement to facilitate routing and message delivery.
- An example network consisting of local networks, each assigned IP addresses from a different class.


65.123.201.65
65.150.92.3
65.80.199.245
140.200.77.203
140.200.197.210
201.150.65.233
201.150.65.99
B
140.200.3.10

# Subnet Mask

- A **subnet mask** is a bit pattern that defines which portion of the IP address represents a network address.
- Consider the class B address 170.203.93.5.

- The binary representation for this address is

  `10101010 11001011 1011101 00000101`

- The default subnet mask for a class B is

  `11111111 11111111 00000000 00000000`

- The subnet mask has **1** in each bit position that corresponds to a bit in the network id component in the address.
- When a **1** appears in the subnet mast, the corresponding bit in the IP address is part of the network id of the network.
- The network id for the example IP address is

```
        10101010 11001011
```

- A `0` in a subnet mask indicates that the corresponding bit in the IP address is part of the host id.
- Like IP addresses, subnet masks are also represented in dotted decimal notation, e.g., **`255.255.0.0`**.

- Subnet masks make it easier and faster for IP to identify the network id portion of the IP address.
- They also allow further suballocation of network ids.

# Subnet Addressing

- Under TCP/IP all hosts are required to support a feature called **subnet addressing**.
- In subnet addressing, instead of considering an IP address as just a network id and host id, the host id portion is divided into a **subnet id** and a host id.

- This capability is important when your network is connected to the Internet, because you will be assigned only few IP addresses.

- You may not be able to obtain enough IP addresses for each of your local networks.
- Even if you are able to obtain enough for the time being, but in future you may need to further subdivide your network into more segments.

- There is another rationale for subnetting as well.
- Class A and B addresses have too may bits allocated for host id, $2^{24}$ - 2 and $2^{16}$ - 2, respectively.

- People donôt attach that many hosts to a single network.
- In a number of cases, the natural 8-bit boundary is used in the 16 bits of a class B host id as the subnet boundary.
- However, this is not a requirement.

|  | 16 bits | 8 bits | 8 bits |
|---|---|---|---|
| Class B | net id = 140.252 | subnet id | host id |

- Subnetting hides the details of internal network organization from external routers.
- To performing subnetting, subnet masks come into play.
- Consider the class C network address 205.101.55.
- The default subnet mask would be 255.255.255.0.

- To do subnetting, the subnet mask is extended into the 4th octet.
- The binary representation of that is

  `11111111 11111111 11111111 11100000`
- In dotted decimal notation this would be `255.255.255.224`.
- This subnet mask designates the first three bits of the 4th octet of the IP address to the subnet id.

- Under this scheme, consider an IP address 205.101.55.91 that has the following binary representation.

  11001101 01100101 00110111 01011011

- After applying the subnet mast, the network is for the subnet is

  11001101 01100101 00110111 $\underline{010}$00000

- The host id consists of 5 bits, corresponding to 27.

- The first three bits of the 4th octet of the IP address can have values ranging from 001 through 110.
- Because 000 and 111 are not valid subnet ids, a total of 6 subnets are made available by a subnet id of 111.
- The number of host ids allowed within each subnet would be from 00001 to 11110, i.e., 1 through 30.

- The 6 subnets designated by a subnet mask of 255.255.255.224 would be associated with the following ranges of values in the 4th octet of the IP address.
  - 00100001 through 00111110 (33 - 62)
  - 01000001 through 01011110 (65 - 94)
  - 01100001 through 01111110 (97 - 126)
  - 10000001 through 10011110 (129 - 158)
  - 10100001 through 10111110 (161 - 190)
  - 11000001 through 11011110 (193 - 222)

- The use of subnetting makes a considerable number of possible values unavailable.
- The benefit of creating multiple subnets with a single class C address must be weighed against the cost in terms of unavailable addresses.

# IP Address to Physical Address

- How does a machine map its IP address to its physical network address?
  - » Example:
    - ï Machines A and B connected to the same network, with IP addresses IA and IB and physical addresses PA and PB.
    - ï Suppose A has has only Bôs IP address, then how does A map IB to PB?

# Address Resolution

- Some protocol suites adopt one of the following:
  - » Keep mapping tables in each machine
  - » Hardware (physical) addresses are encoded in the high level addresses

- Both are ad-hoc, awkward solutions

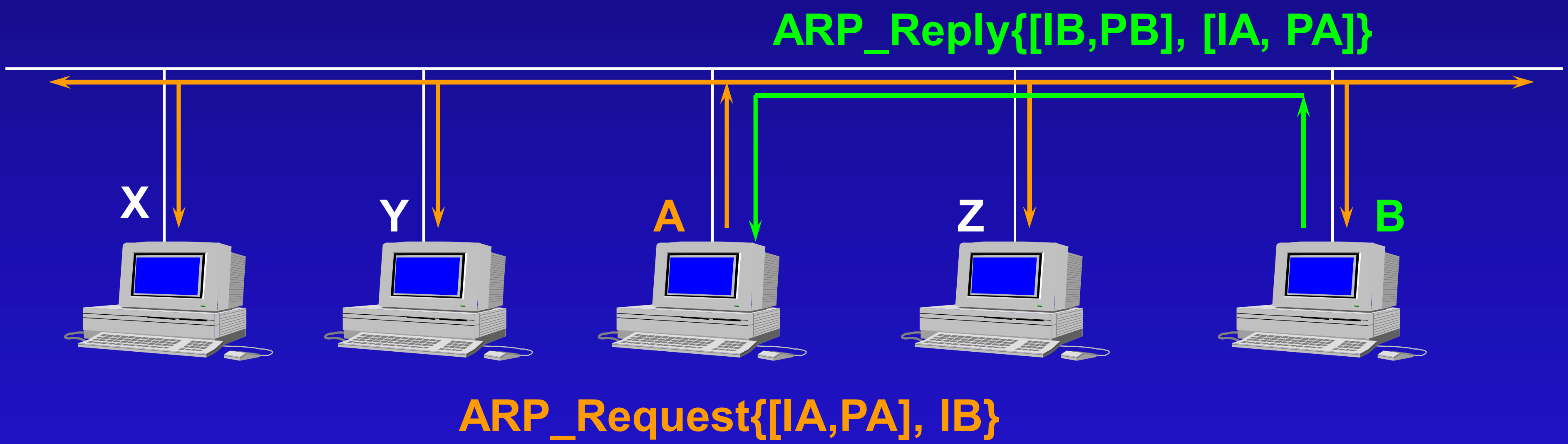
# on Through Dynamic Binding (ARP)

- Ethernet uses 48-bit physical addresses
  - » Addresses assigned by manufacturers
  - » Replacing a faulty interface card meant a change to the machine physical address
- Canôt encode 48-bit long address into a 32-bit long IP address
- TCP/IP solution: Address Resolution Protocol (ARP)

# ARP

- Exploits broadcast capability of Ethernet
- Allows a host to find the Ethernet address of a target host on the same network, given the targetôs IP address
- Allows new machines to be added with no code recompilation
- Builds and maintains dynamically a table to translate IP addresses into Ethernet physical addresses

# ARP (cont.)



# ARP (cont.)

- Hosts that use ARP maintain a small cache of recently acquired (IP,P) address bindings
- Cache is updated dynamically
  - Timer for each entry
  - Whenever a new binding is received, update the corresponding table entry and reset the associated timer

# ARP (cont.)

- ARP is a low level protocol that hides the underlying network physical addressing, permitting us to assign IP addresses of our choosing to every machine
- We think of it as part of the physical network and not as part of the internet protocols

# ning an IP Address at Startup

- Diskless machines use IP addresses to communicate with the file server
- Also, many diskless machines use TCP/IP TFTP protocols to obtain their initial boot image, thus requiring that they obtain and use IP addresses
- Designers keep both the bootstrap code and initial OS images free from specific IP addresses for portability

# ng an IP Address at Startup (cont.)

- How does a diskless machine determine its IP address?
- When bootsrap code starts execution on a diskless machine, it uses the network to contact a RARP server to obtain the machineôs IP address
- Usually, a machineôs IP address is kept in a database where the OS finds it at startup

# e Address Resolution Protocol

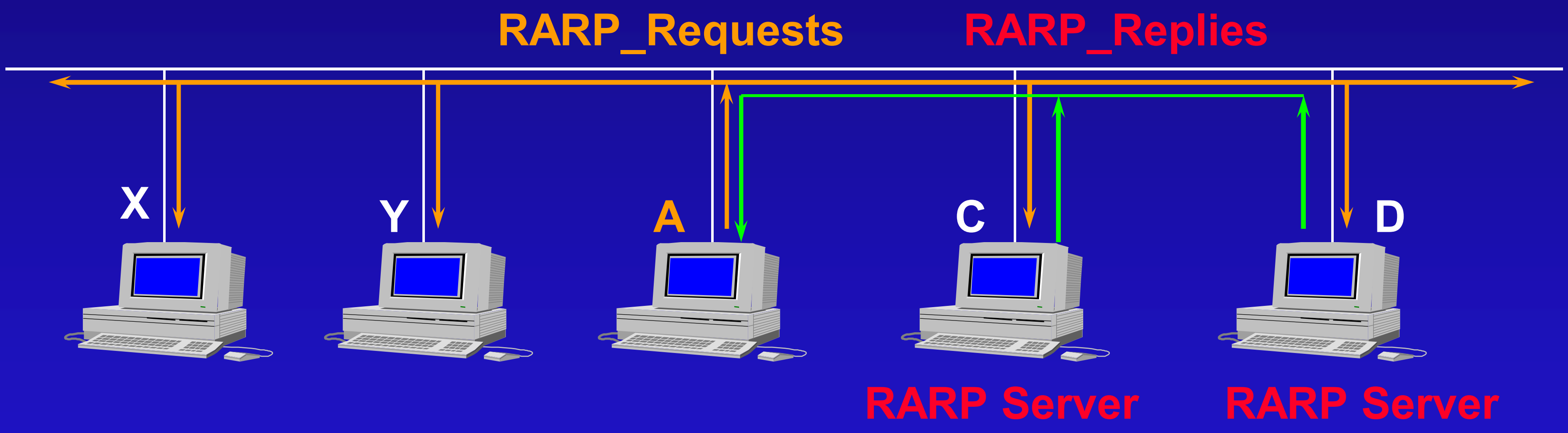
- RARP is the protocol used to solve the reverse problem solved by ARP
  - » Given a physical address, get the corresponding IP address
- RARP uses the same message format as ARP
- RARP messages are sent encapsulated in Ethernet frames

# RARP (cont.)

  - » The frame type field contains the value &8035 to identify the contents of the frame as a RARP message
  - » The data portion of the frame contains the 28-octet RARP message

- ● RARP allows a host to ask about an arbitrary target
    - ï The sender supplies its HA separate from the target HA, and the server is careful to reply to the senderôs HA

# RARP (cont.)



# Internet Protocol (IP)

- Connectionless Protocol
  - » does **not** exchange control information to establish end-to-end connection before exchanging data
  - » no **handshaking**
  - » contrast with connection-oriented protocols
- IP relies on protocols in other layers to establish a connection if they require connection oriented service
- IP is an **unreliable** protocol
  - » no error detection and recovery code
  - » protocols in other layers provide this checking when required

# Routing Datagrams

- Header contains destination address
  - » 32 bit IP address identifies destination network and specific host on it
  - » If destination addr is that of a host on the local network
    - ï packet is delivered directly
  - » If destination addr is not on the local network
    - ï packet is passed to a gateway for delivery
- Gateways are devices that switch packets between the different physical networks
  - » IP makes the **routing** decision for each packet

# Routing Datagrams

- Internet gateways are called IP routers
- Two types of network devices
  - Hosts
  - Gateways
- Multi-homed hosts act as gateways
- Hosts (end-systems) process packets through all four TCP/IP protocol layers
- Gateways (intermediate systems) process the packets only up to the Internet layer where routing decisions are made
- Routing is done at IP level
  - a datagram may travel through several different types of physical networks

# Fragmenting Datagrams

- Each network type has an **MTU**
  - » Maximum Transmission Unit
  - » largest packet that network can transfer
- If gateway connects dissimilar networks
  - » MTU may be different
  - » if datagram recvôd from one network is longer than other networkôs MTU divide datagram into smaller fragments for transmission
    - ï **fragmentation**
- Re-assembly of datagram occurs at internet layer of final destination
- Information about fragmentation is kept in the datagram header

# Passing Datagrams Up

- If datagram is for local host
  - » IP strips header and passes data portion to the correct Transport Layer protocol
- Which protocol to pass up to?
  - » each Transport Layer protocol has a unique protocol number
  - » Information is kept in Protocol field of datagram header

# Delivering the Data

- To deliver data
  - » get it to correct host
  - » within the host get it to the correct user or application
- Addressing
  - » IP addresses uniquely identify each host
- Routing
  - » Gateways deliver data to correct network
- Multiplexing
  - » Protocol and port numbers deliver data to correct software module within the host

# Internet Routing Architecture

- Core Gateways
  - backbone of the Internet
  - Exchange routing information using GGP
    - Gateway to Gateway Protocol
- Autonomous Systems
  - groups of networks outside core
  - Reachability information using EGP
    - Exterior Gateway Protocol
- Routing Domains
  - Border gateway Protocol (BGP)

# Routing

- Both hosts and gateways make routing decisions
- For most hosts
  - » if dest host is on local network
    - ï direct delivery
  - » if dest host is on a remote network
    - ï forward to local gateway
- Routing is network oriented
  - » IP computes network portion of IP address
  - » Network is looked up in local routing table

# Routing Tables

- Pairs of Destination & Gateway
  - » Specify gateways for particular destination networks
  - » e.g. for net 196.1.67 use gateway 196.1.65.250
- Default Route
  - » default gateway
- Loopback route for local host
- All gateways in routing table are on networks directly connected to local system
- Routing table does not contain end-to-end routes it only points to the next hop

# ICMP

- Internet Control Message Protocol
  - » part of Internet Layer
- Flow Control
- Detecting unreachable destinations
- Redirecting routes
- Checking remote hosts

# Transport Layer

Between Application and Internet Layers

<u>Two important protocols :</u>

- Transmission Control Protocol (TCP)
  - provides reliable data delivery service with end-to-end error detection and correction
- User Datagram Protocol (UDP)
  - provides low-overhead connectionless datagram delivery service

Application programs can choose appropriate service

# User Datagram Protocol (UDP)

- Gives application programs direct access to a datagram delivery service
- Unreliable, connectionless protocol
- UDP uses 16-bit port number to deliver data to the correct application process
  - » Source Port
  - » Destination Port

# UDP

- Why use UDP?
  - » low overhead
  - » if amount of data is small
  - » query-response model
  - » application provides own technique for reliable data delivery

# ssion Control Protocol (TCP)

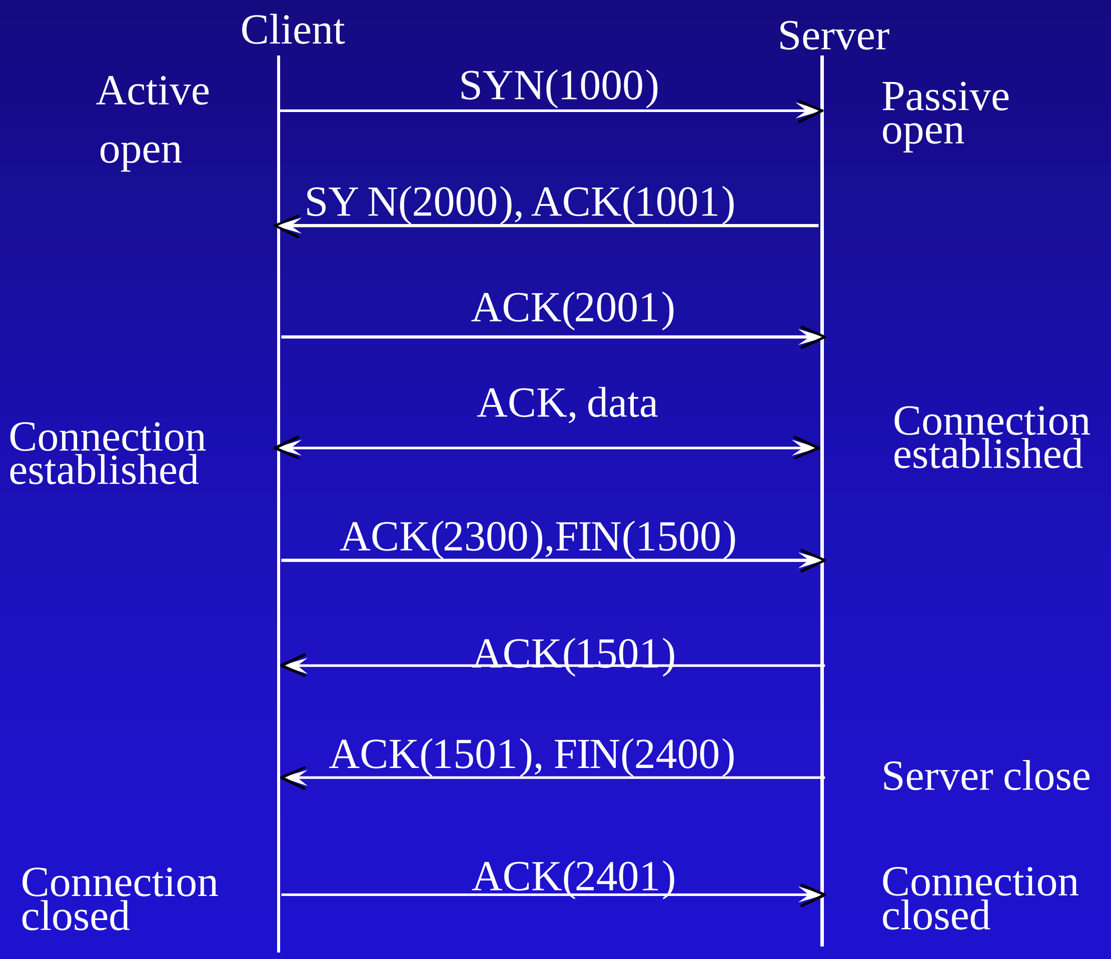
- TCP verifies data is delivered accurately and in sequence
- TCP is a reliable, connection-oriented, byte-stream protocol

# TCP's Virtual Circuit

- Uses a sliding window protocol
- Reliability
  - » positive acknowledgment with re-transmission (PAR)
  - » each TCP segment has checksum
  - » if received undamaged, receiver sends positive acknowledgment
  - » after appropriate time-out sender will re-transmit packets for which no positive ack has been received

# P Connection Estab. and Term.

- Connection-Oriented
  - » TCP establishes logical end-to-end connection between two hosts
- 3-way handshake
- At end of xfer another 3-way handshake
  - » FIN (no more data)



# TCP : Data Flow

- TCP views data as a stream of bytes, not as independent packets
  - » maintains sequence of bytes
  - » Sequence Number and Acknowledgment Number fields in TCP header keep track of bytes

- Acknowledgment Segment
  - » positive acknowledgment - tells sender how much data has been recvôd
  - » flow control - **window** field tells sender how much more data the remote end is willing to accept
    - ï sliding window
- TCP xfers data to correct application
  - » uses port numbers

# TCP Segment

0 8 16 31

<table>
<tr><td colspan="3">SOURCE PORT</td><td colspan="2">DESTINATION PORT</td></tr>
<tr><td colspan="5">SEQUENCE NUMBER</td></tr>
<tr><td colspan="5">ACKNOWLEDGMENT NUMBER</td></tr>
<tr><td>OFF.</td><td>RES.</td><td>CODE</td><td colspan="2">WINDOW</td></tr>
<tr><td colspan="3">CHECKSUM</td><td colspan="2">URGENT POINTER</td></tr>
<tr><td colspan="4">OPTIONS</td><td>PADDING</td></tr>
<tr><td colspan="5">DATA</td></tr>
<tr><td colspan="5">* * *</td></tr>
</table>

# Client Server Model

- Client-Server paradigm is the primary pattern of interactions among cooperating applications.
- This model constitutes the foundation on which distributed algorithms are built.

# Client Server Model (cont.)

- Server: Any program that offers a service reachable over the network
  - » If a machineôs primary purpose is to support a particular server program, the term server is usually applied to both, the machine and the server program.
- Client: An executing program becomes a client when it sends a request to a server and waits for a response.

# Client Server Model (cont.)

- Servers accept requests arriving over the network, perform the requested services, and return the results to the requesters
- Simplest service
  - » Request arrives in a single IP datagram
  - » Server responds in another IP datagram

# Multiplexing

- Data on destination must be delivered to the correct user or process or server
- Data moves up and down TCP/IP layers
  - » mechanism to deliver it to correct protocols in each layer
- Multiplexing
  - » System combines data from several applications into a few transport protocols

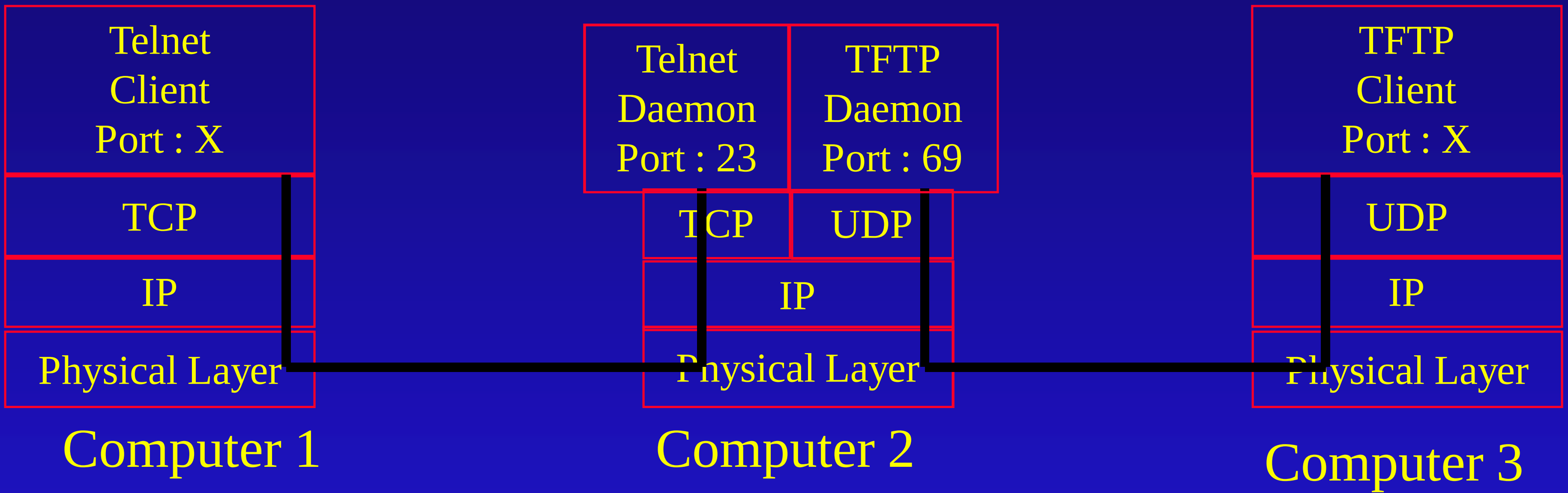
- Multiplexing
  - » System combines data from several applications into a few transport protocols
- Data arriving from network must be **demultiplexed**
  - » TCP/IP uses protocol numbers and port numbers for this

# Demultiplexing

- Protocol Numbers
  - » byte in datagram header
  - » when datagram arrives at dest., IP layer has to forward it to one of the transport protocols above it
  - » decided using datagramôs protocol number
    - ï e.g. 6 (TCP), 17 (UDP)

- Port Numbers
  - » helps transport protocol determine which application layer protocol to forward data to
  - » Source and Destination Port Numbers
  - » Defined numbers for well-known services
  - » Dynamically assigned ports

# ultiplexing and Demultiplexing



- TCP : Connection oriented service

A connection is defined by the four tuple:

(Src IP Addr, Src Port #) (Dest IP Addr, Dest Port #)

- UDP : Datagram service

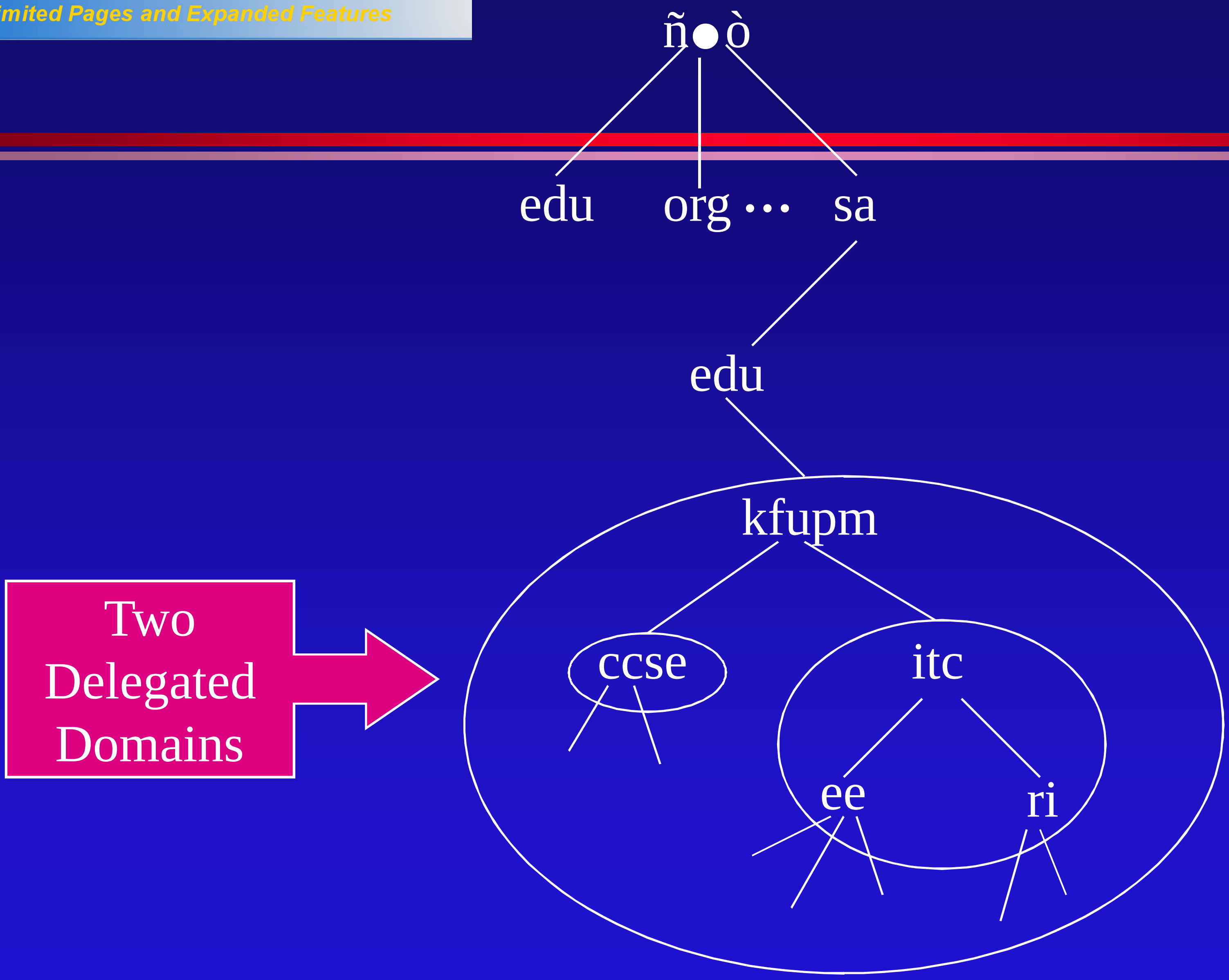
# [illegible] C o

- The DNS has four major components:
  - » Domain name space
  - » Resource Records (RRs)
  - Name Servers
  - Resolvers

# Delegation

- One of the main goals of the design of the Domain Name System was decentralizing administration.
- This is achieved through delegation.
- It works a lot like delegating tasks at work.
- An organization can divide its domain into sub-domains, each of which is delegated to other organizations.

- This means that the organization delegated to becomes responsible for it.
- They can freely change the data, and can even divide their sub-domain into further sub-domains and delegate those to other organizations.
- The parent domain contains only pointers to sources of the sub-domainôs data, so that it can refer queries there.



ñ●ò

edu

org …

sa

edu

kfupm

ccse

itc

ee

ri

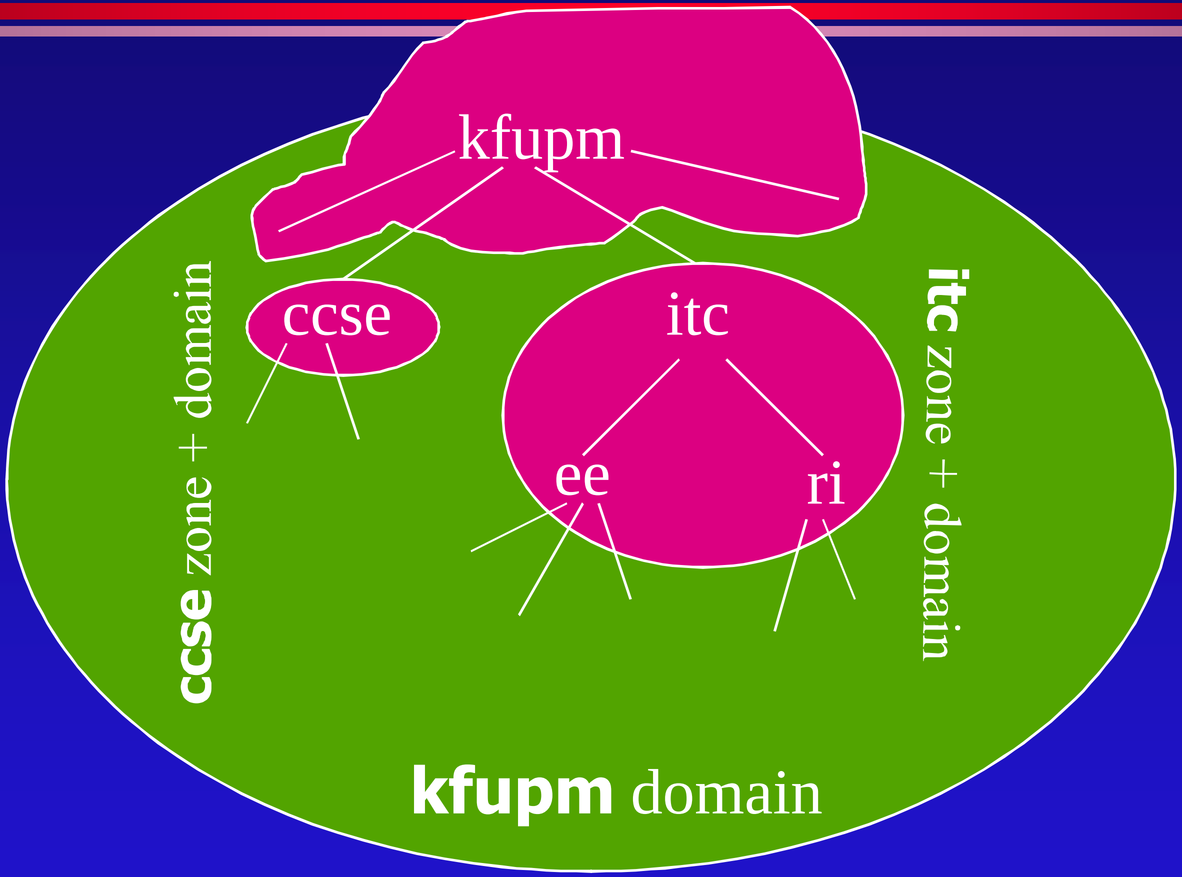
Two Delegated Domains

# Name Servers

- The programs that store the information about the domain name space are called name servers.
- The domain database is divided up into parts called zones, which are distributed among various name servers.
- The name server that handles a particular zone is said to have authority over that zone.

- A name server can be **authoritative** over multiple zones as well.
- A zone contains the domain names and data that a domain contains, except for domain names and data that are delegated elsewhere.

# **kfupm** zone


kfupm
ccse
itc
ee
ri
**ccse** zone + domain
**itc** zone + domain
**kfupm** domain

# Types of Name Servers

- The DNS specs define two types of name servers.
  - » Primary master
  - » Secondary master
- A primary master name server gets the data for the zones itôs authoritative for from files on the host it runs on.

- A secondary master name server gets its zone data from another name server authoritative for the zone.
- Once setup the secondaries will periodically query the primary to keep the zone data up-to-date.

- It is important to set up more than one name server for a given zone, for load balancing, redundancy and reduced network traffic.
- A given name server can support one or more zones.

- Similarly, a name server can be primary master for one zone and secondary master for another.
- It may also have cached non-authoritative data about other zones, which it marks in the response to a query as non-authoritative.

# Resolvers

- Resolvers are clients that access name servers, and interface user programs to the DNS.
- Programs running on a host that need information from the domain name space use the resolver.
- The resolver is located on the same host as the program that requests the resolverôs services.

- In the simplest case, a resolver receives a request from a user program (e.g., mail programs, TELNET, FTP) in the form of a subroutine call, system call etc., and returns the desired information in a form compatible with the local hostôs data formats.

- The resolver handles:
  - » Querying a name server
  - » Interpreting responses
  - » Returning the information to the requesting program.

# Name Resolution

- The name servers not only provide data about zones they are authoritative for, but can also search for data belonging to zones for which they are not authoritative.
- This is called name resolution or simply resolution.

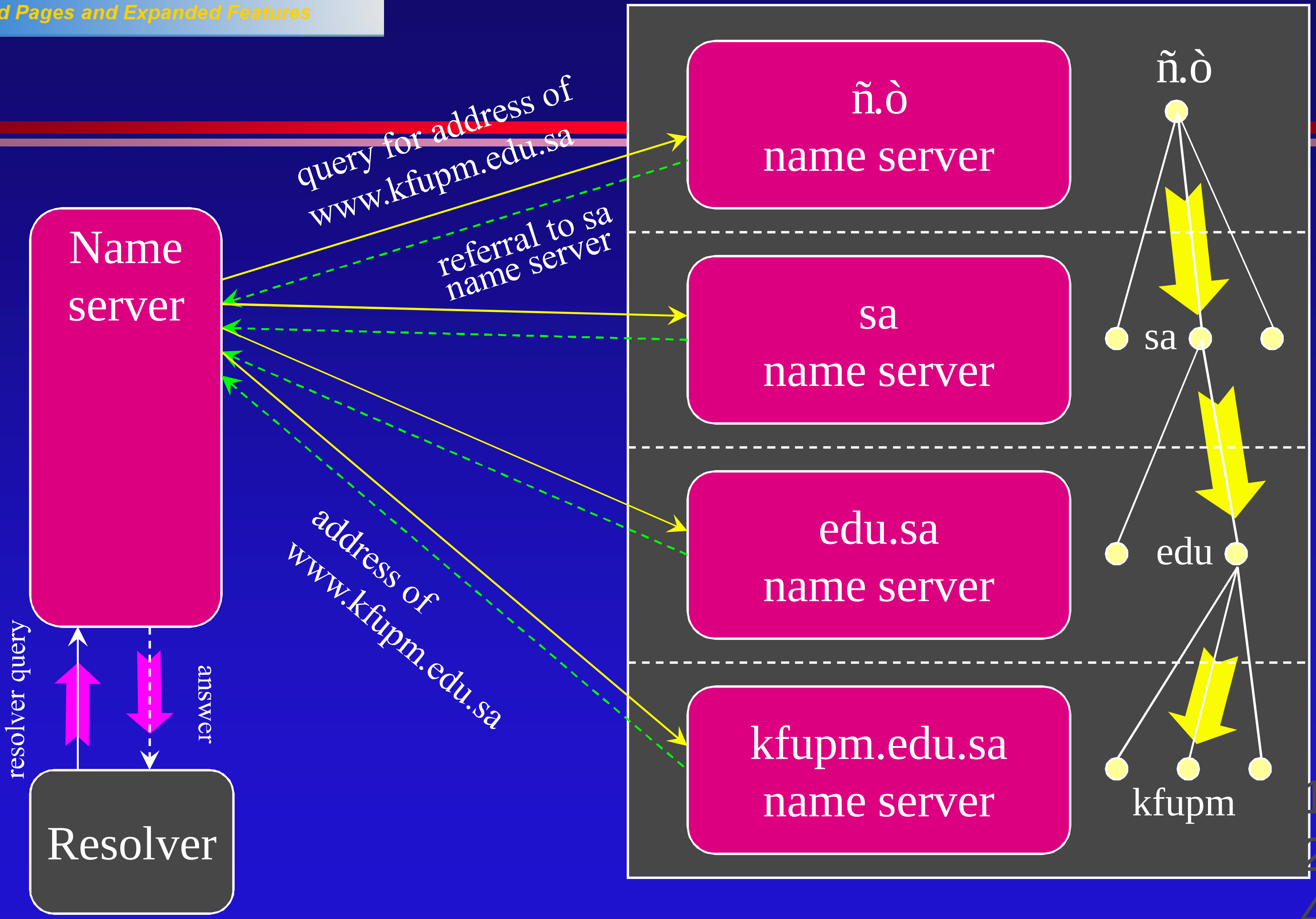
- Because the entire name space is structured as an inverted tree, a name server only needs one piece of information to find its way to any point in the tree, i.e., the name and address of the root name servers.
- A name server can issue a query to a root name server for any name in the domain name space, and the root name server will ultimately find it.

# Root Name Servers

- The root name servers know where name servers authoritative for all the top-level domains are.
- Given a query about any domain name, the root name servers can at least provide the names and addresses of the name servers authoritative for the top-level domain of which the required domain is a part.

- The top-level domains in turn can provide the list of name servers authoritative for the second-level domain which contains the required domain.
- Each name server queried gives the querier information about how to get ñcloserò to the answer itôs seeking, or provide the answer itself.

- The root name servers are very important to the resolution process.
- If all the root name servers were unreachable for an extended period, all resolution would fail.



query for address of www.kfupm.edu.sa

referral to sa name server

address of www.kfupm.edu.sa

Name server

resolver query

answer

Resolver

ñ.ò name server

sa name server

edu.sa name server

kfupm.edu.sa name server

ñ.ò

sa

edu

kfupm

# Recursive Queries

- Queries issued by resolvers are of two types.
  - » Recursive
  - » Iterative

- Recursive queries place most of the burden of resolution on a single name server.
- Recursive resolution denotes the process that the name server follows when it receives recursive queries.

- In recursive resolution, a resolver sends a recursive query to a name server for information about a particular domain name.
- The queried name server is then obliged to respond with the requested data, or with an error stating that data of the requested type doesnôt exist or that the domain name specified doesnôt exist.
- The name server canôt just refer the querier to a different name server.

- If the queried name server isnôt authoritative for the data requested, it will have to query other name servers to find the answer.
- It could send recursive queries to those name servers, thereby obliging them to find the answer and return it.
- Or it could send iterative queries, and possibly be referred to other name servers ñcloserò to the domain name itôs looking for.

# Iterative Queries

- Iterative resolution refers to the resolution process used by a name server when it receives iterative queries.
- In iterative resolution, a name server simply gives the best answer that it already knows back to the querier.
- It consults its local database, including its cache for the data requested.

- If it doesnôt find the data there, it makes its best attempt to give the querier data that will help it continue the resolution process.
- Usually this includes names and addresses of name servers ñcloserò to the data it is seeking.

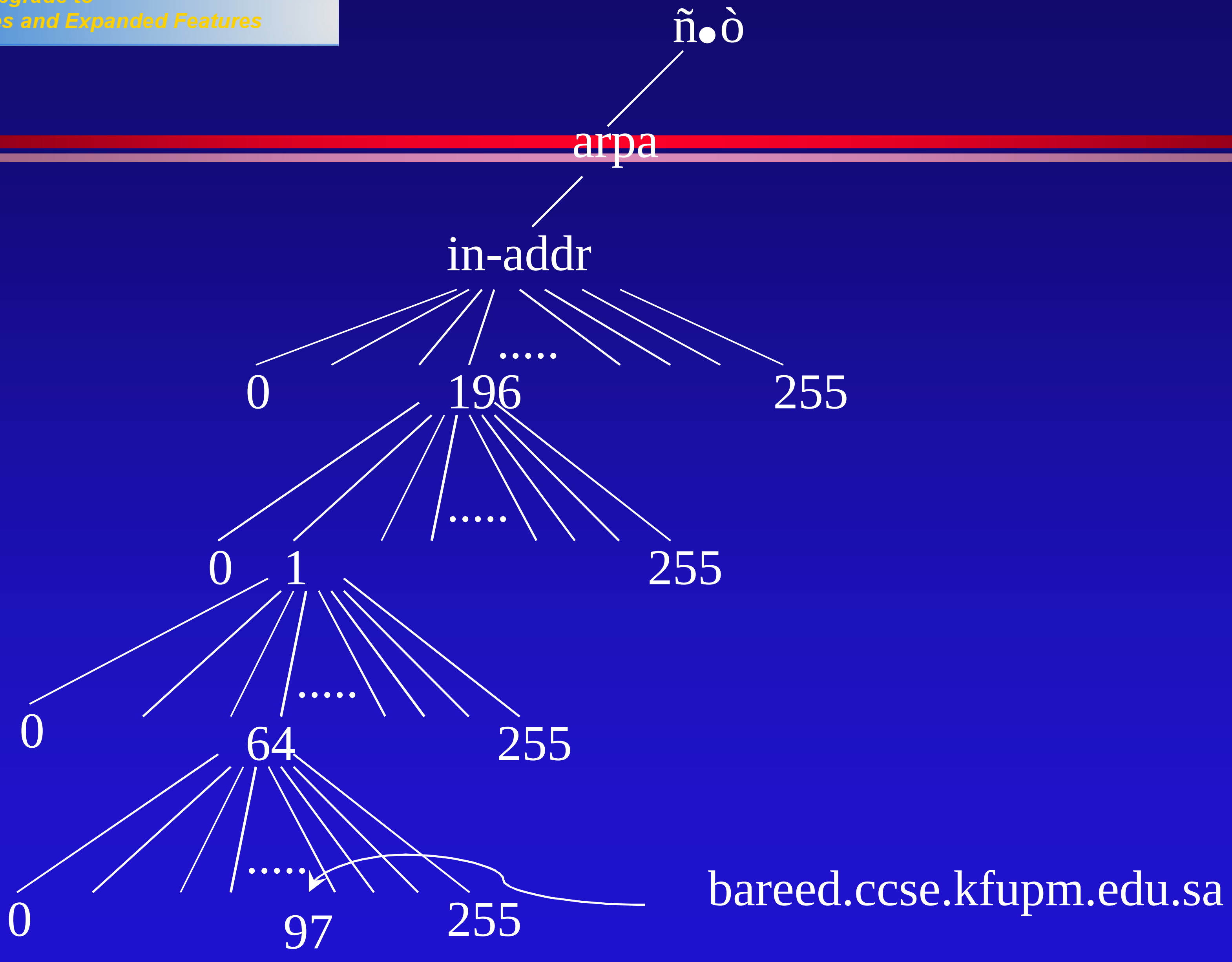
# Mapping Addresses to Names

- Address-to-name mapping is used to produce output that is easier for humans to read and interpret, e.g., in log files, etc.
- It is also used in some authorization checks.
- In DNS, address-to-name mapping isnôt simple, because the data, including addresses, in the domain name space are indexed by name.

- Finding an address given a domain name is relatively easy.
- However, finding the domain name that maps to a given address would seem to require an exhaustive search of every domain name in the tree.

- There is a very effective and clever solution to this problem.
- Since it is easy to find data once the name that indexes the data is given, in a part of the domain name space, addresses can be used as names.

- This part of the name space is called the **in-addr.arpa** domain.
- Nodes in the **in-addr.arpa** domain are named after the numbers in the dotted-octet representation of IP addresses.
- The **in-addr.arpa** domain domain could have up to 256 sub-domains, one corresponding to each possible value in the first octet of an IP address, and similarly the further sub-domains.


ñ●ò
arpa
in-addr
0
.....
196
255
0
1
.....
255
0
.....
64
255
0
.....
97
255
bareed.ccse.kfupm.edu.sa

- When read as a domain name, the IP address appears backwards, since the name is read leaf-to-root.
- bareed.ccse.kfupm.edu.sa has the IP address 196.1.64.97.
- The corresponding **in-addr.arpa** sub-domain is 97.64.1.196.in-addr.arpa, which maps back to the domain name bareed.ccse.kfupm.edu.sa.

# Application Level Protocols

Internet services are provided through application level programs

- ***Telnet*** is a terminal emulation application program.
  - » Allows a user to remote-login on to another computer.
- ***FTP*** is the major TCP/IP file transfer protocol
  - » A facility to access files on remote machines
  - » File transfer is among the most frequently used TCP/IP applications
  - » Anonymous downloading of files.

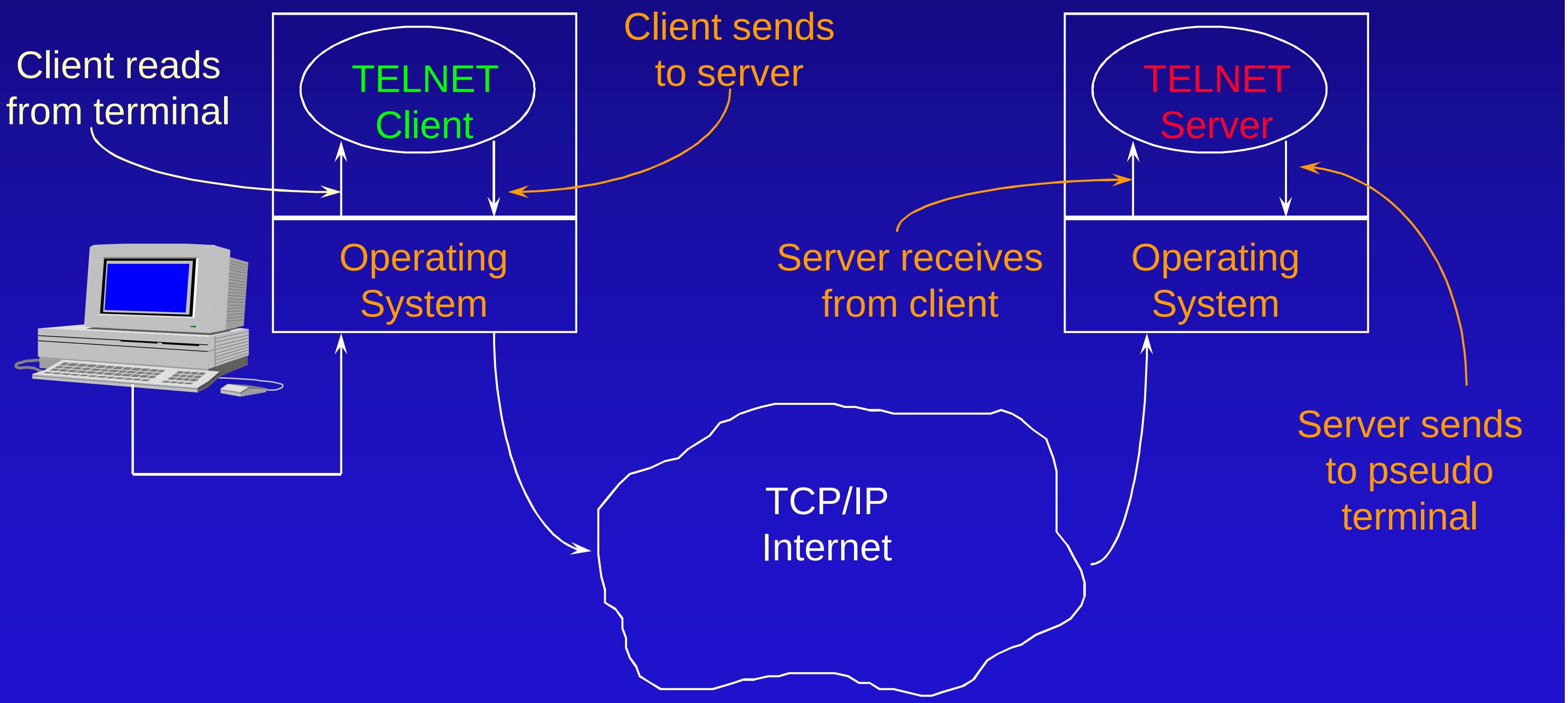
# TELNET (cont.)

- **TELNET**
  - Allows a user at one site to establish a TCP connection to a login server at another
    - TELNET client software allows the user to specify a remote machine by giving its domain name or IP address
  - Passes keystrokes from the user terminal (client site) to the remote machine (server)
  - Carries output from the remote machine back to the userǒs terminal

# TELNET (cont.)

- TELNET offers three basic services
  - » It defines a Network Virtual Terminal (NVT) that provides a standard interface to remote systems
  - » It includes a mechanism that allows the client and server to negotiate options, and it provides a set of standard options
  - » It treats both ends symmetrically (either end can negotiate options)

# TELNET (cont.)



Client reads from terminal

TELNET Client

Operating System

Client sends to server

TCP/IP Internet

Server receives from client

TELNET Server

Operating System

Server sends to pseudo terminal

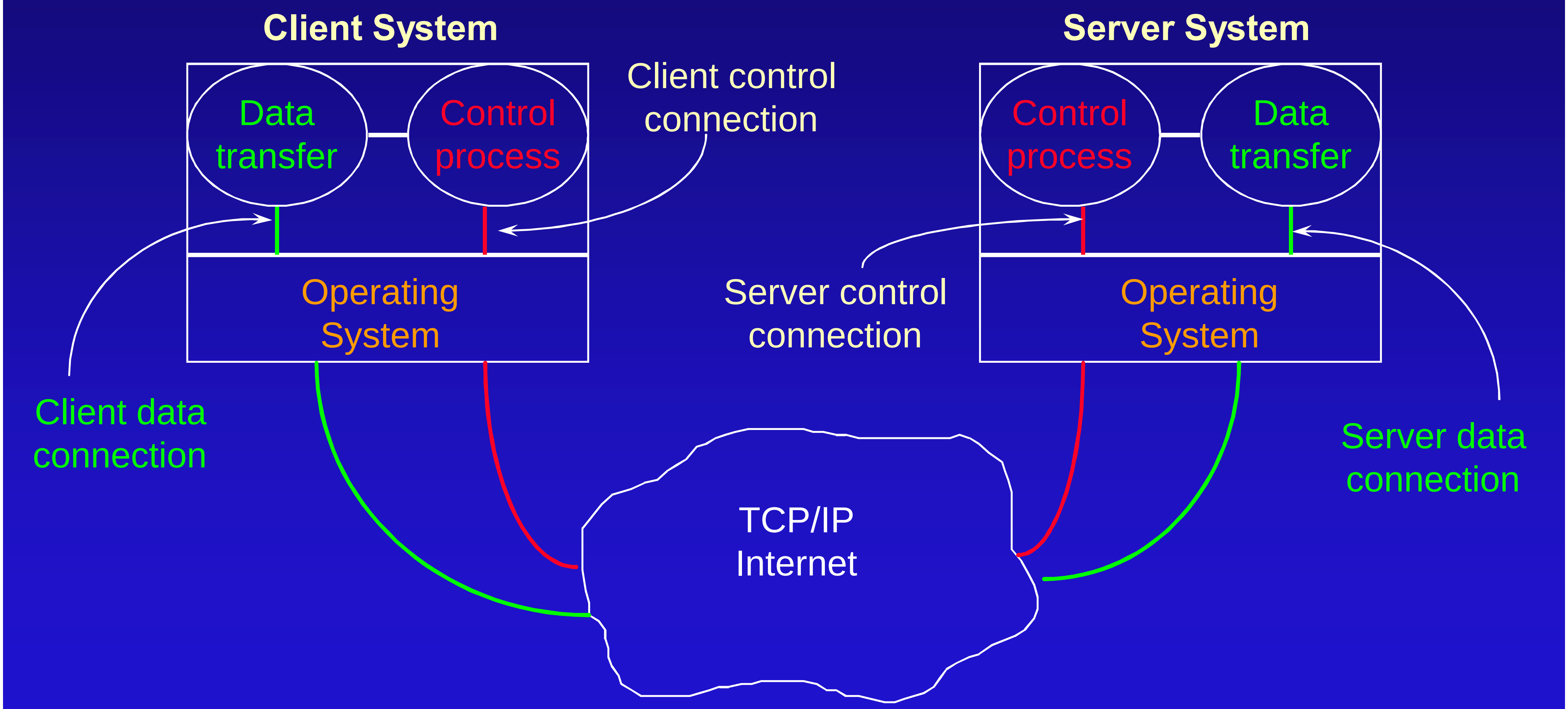
# File Transfer Protocol

- Clients use TCP to connect to the server
- FTP uses two different connections for file transfer. One for data and one for control information
  - ï Control connection carries commands telling the server which file to transfer
  - ï Data transfer connection carries data transfers
- A single master server process awaits connections and creates a slave process to handle each connection

# File Access Model

- Control connection is used to
  - » pass user commands to the server
  - » allow client and server control processes to coordinate their use of dynamically assigned TCP ports and the creation of data transfer processes that use those ports
- The format used by FTP for passing data across the control connection is the NVT format

# File Access Model (cont.)



# File Access Model (cont.)

- Data transfer connections and the data transfer processes that use them are created dynamically, but the control connection persists throughout a session
- Once the control connection disappears, the session is terminated, and software at both ends terminates all data transfer processes

# Email

- Email is the first encounter of users with computer networks
- Millions connected to the Internet use it.
- Low cost and fast communication.
- Encourages collaboration.
- "A person ... can say HELP to 10,000 people ... The next morning he may have 15 answers to his problem."

# Email (cont.)

- E-mail is delivered in few minutes.
- E-mail costs half that of regular postal mail (SNAIL MAIL) and ONLY 15% that of Fax.
- In 1992, responsible for 20% of traffic.

# Email (cont.)

farooq@ccse.kfupm.edu.sa

**farooq** : User name

@ : Connects the who to where

**ccse** : subdomain name

**kfupm** : domain

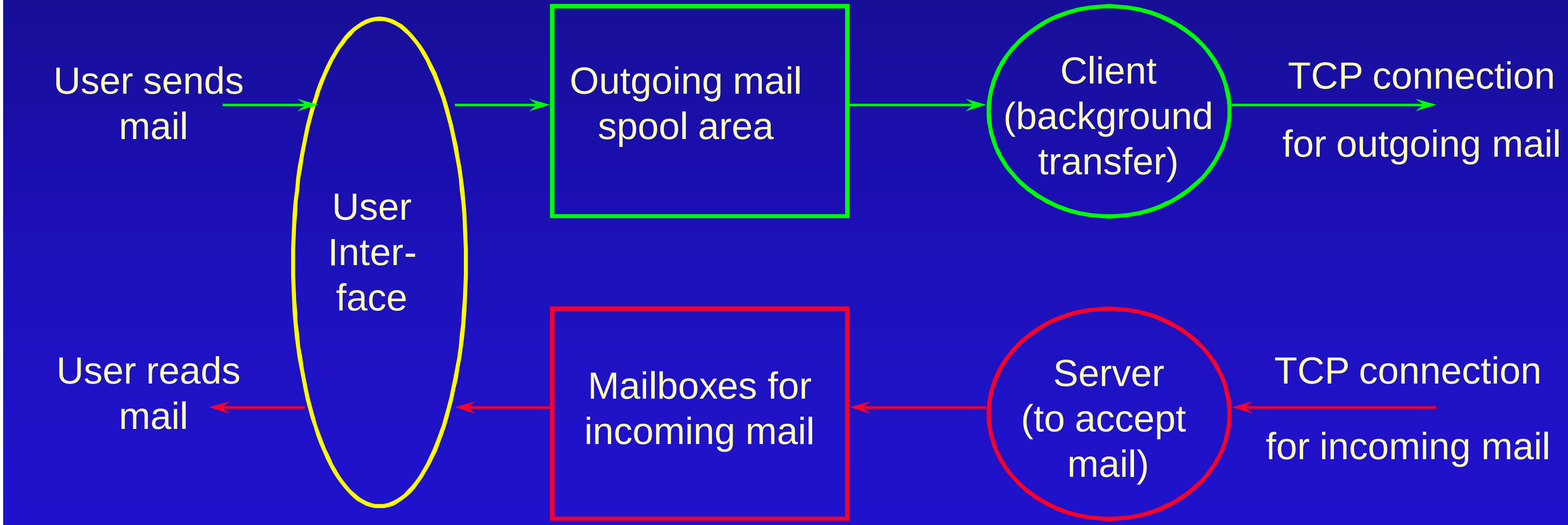
**edu** : segment type

**sa** : final where segment (sa=Saudi Arabia, tn= Tunisia, ca: Canada)

# Email (cont.)

- Mail systems use Spooling technique to handle delayed delivery
  - » When a user sends a message, the system places a copy in its private storage (spool) area along with the identification of sender, recipient, dest machine, and time of deposit
  - » The transfer is initiated in the background, allowing the sender to proceed with other activities

# eptual Components of an Email System


User sends mail
User Inter-face
Outgoing mail spool area
Client (background transfer)
TCP connection for outgoing mail
User reads mail
Mailboxes for incoming mail
Server (to accept mail)
TCP connection for incoming mail

# Email concepts (cont.)

- The background mail transfer process becomes a client
  - It maps the dest machine name to an IP address
  - It forms a TCP connection to the mail server on dest machine
  - It passes a copy of the message to the remote server, which stores a copy in the remoteôs system spool area

# Email concepts (cont.)

- Once the client and server agree that the copy has been accepted and stored, the client removes the local copy

- If TCP connection fails, the transfer process records the time it tried delivery and terminates

# Email concepts (cont.)

» The background transfer process sweeps through the spool area periodically

For each undelivered or new outgoing mail

ï It attempts delivery again

ï If a mail message cannot be delivered after an extended time (3 days), it returns the mail message to the sender

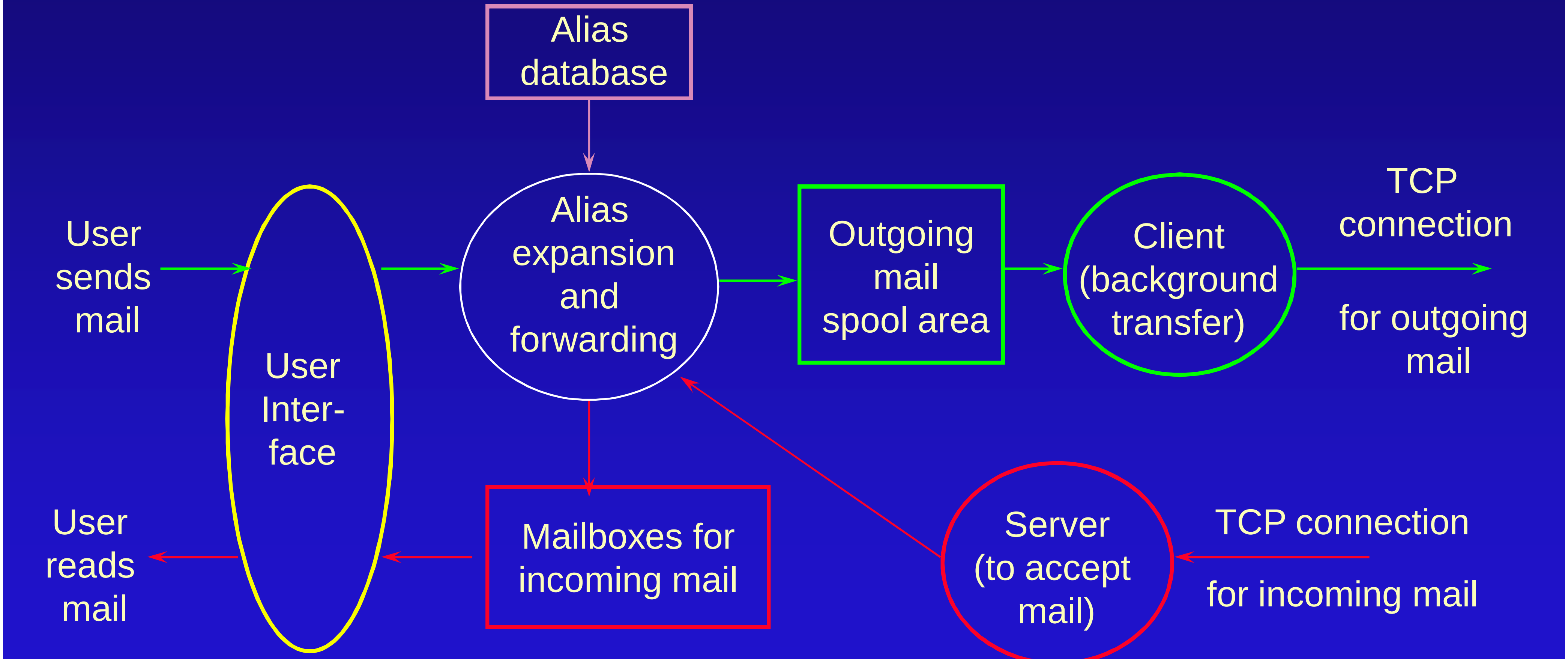
# Mailbox names and Aliases

- Users specify
  - » the mail destination machine (usually the machineôs domain name)
  - » a mailbox at that machine (usually the userôs login Id)
- Most systems provide mail forwarding software that includes alias expansion mechanism

# Expansion and Mail Forwarding

- A mail forwarder allows the local site to map Ids used in mail addresses to a set of one or more new mail addresses
- After a user composes a message and names a recipient
  - » the mail interface consults the local aliases to perform necessary mappings before passing the message to the delivery system

# Conceptual Model of a Mail System



# TCP/IP Standard for Email Service

- TCP/IP divides its mail standard into two sets
  - » One standard specifies the format for mail messages (RFC 822)
  - » The other specifies the details of electronic mail exchange between two computers
- This division makes it possible to build mail gateways to non TCP/IP networks while still using the same format

# Standard Format

- Headers contain readable text, divided into lines that consist of
  - a keyword
  - a colon ñò
  - a value
- Some keywords are required, others are optional, and the rest are un-interpreted

# Electronic Mail Addresses

- Email addresses have a simple, easy to remember form

local-part@domain-name

domain-name: mail exchanger of the mail destination

local-part: address of a mailbox on that machine

farooq@ccse.kfupm.edu.sa

# Mail Transfer Protocol (SMTP)

- SMTP is the standard mail transfer protocol of TCP/IP
- SMTP focuses on how the underlying mail delivery system passes messages across a link from one machine to another
- SMTP is simple.

# SMTP (cont.)

- Communication between a client and a server consists of readable text
- Initially, the client establishes a reliable stream connection to the server
- It then waits for the server to send the message ñ220 READY FOR MAILò
- Upon receipt of the 220 message, the client sends a ñHELOòcommand

(End of line marks the end of a command)

# SMTP (cont.)

- The server responds  by identifying itself
- Then the sender can transmit one or more mail messages, terminate the connection or request the server to exchange the roles of sender & receiver
- The receiver must ACK each message. It can also abort the entire connection or abort the current message transfer

# ic Host Configuration Protocol (DHCP)

- DHCP automatically sets up a host during boot up on a TCP/IP network and can also change settings while the host is attached to the network.
- When properly configured, it reduces a large portion of daily network maintenance.

- It eliminates errors due to improperly configured TCP/IP hosts on the network.
- The hosts do not need to have the IP addresses configured manually.
- IT ensures that the IP addresses are used efficiently by the network hosts when there is a limited number of IP addresses available.

- A host receives an IP address on boot up.
- Later it can notify the DHCP server when it powers down or logs off.
- After an IP address has not been used for a specified period of time, the DHCP server can make that available to other DHCP clients.

- TCP/IP configuration parameters that are commonly stored in the DHCP database include:
  - » The range(s) of valid IP addresses and subnet masks for the local network
  - » IP addresses reserved for certain hosts.
  - » The WINS servers that will be used.
  - » Any other values in the TCP/IP configuration dialog boxes.

- To setup a host on a TCP/IP network, the correct values for the following parameters must be known:
  - » host IP address
  - » subnet mask
  - » default gateway
- Optional parameters include the DNS and WINS server addresses.

- An incorrect entry for one of the required parameters can result in:
  - » failure to communicate
  - » failure to initialize
  - » causing other network hosts to hang or misbehave.
- If properly used, DHCP can eliminate such problems.

- Using DHCP, the administrator enters the valid IP addresses or range of IP addresses (called a **scope**) in the DHCP server database, which then assigns (or **leases**) the IP addresses to the DHCP clients.
- Since TCP/IP settings are entered in one place in the DHCP database, chances of typing mistakes are reduced.

- An IP address is normally leased for a limited amount of time.
- The client must renew this lease periodically before the lease expires.
- If the host is no longer using the IP address, the lease will expire and can be assigned to another client.

- If the host is physically moved to a different subnet, the DHCP server on that subnet will automatically reconfigure the host with the proper TCP/IP settings.

# Limitations of DHCP

- It does not detect IP addresses already in use on a network by non-DHCP clients, so such addresses should be excluded from any scopes configured on the DHCP server.
- A DHCP server does not communicate with other DHCP servers and cannot detect addresses leased by other servers.

- DHCP servers cannot communicate with clients across routers unless BOOTP forwarding is enabled on the router.
- Incorrect values configured for a DHCP scope can cause unexpected and potentially disastrous results.
- Except for the IP address and subnet mask, any values manually configured on the client will override the DHCP server scope setting.

# DHCP Server Requirements

- The DHCP server software (service) must be running.
- The server must have a manually configured IP address.
- A DHCP server must be located on the same subnet as the DHCP clients, or the routers connecting the two subnets must be able to forward the DHCP datagrams.

- A pool of IP addresses, known as **scopes**, must be configured on the DHCP server.
- WINS, DNS, and DHCP servers must always be assigned static IP addresses.
- IP addresses must not be duplicated on another DHCP server.

# DHCP Scope

- When a DHCP client initializes, it requests an IP address and subnet mask from the DHCP server.
- Before the client can obtain adresses from DHCP, one or more scopes must be created on the DHCP server.
- A **scope** is a range of IP addresses that can be leased by clients.

- The scope contains a collection of information including:
  - » a range of valid IP addresses
  - » the subnet mask for the network segment.
  - » Optional DHCP client information, called **scope options**, such as default gateway. If any of these options are set on the DHCP server, they are sent to the DHCP client with the leased IP address and the subnet mask.

- The following scope options are supported by the Microsoft DHCP clients:
  - Subnet Mask
  - DNS Server
  - WINS Server
  - NetBIOS Scope ID
  - Default Router
  - DNS Domain Name
  - NetBIOS Node Type
- There are other options available on the server, however, the clients only use these and ignore others.

- Two types of DHCP scope options are available:
  - » Global options, which are set for all the scopes maintained by the DHCP server.
  - » Scope options, which are set for a selected scope.

# DHCP Operation

- When the DHCP client initializes, it broadcasts a request for an IP lease from a DHCP server called a **DHCPDISCOVER**.
- Because the client does not yet have an IP address, it lists 0.0.0.0 as the source IP address and includes its hardware address and NetBIOS computer name.

- All DHCP servers that receive the IP lease request respond to the DHCP client with an IP lease offer known as a **DHCPOFFER**.
- The offer is ignored by all other hosts and the only host that receives the offer is the one that sent the **DHCPDISCOVER**.

- In addition to the hardware address, the **DHCPOFFER** message also includes the IP address and subnet mask of the DHCP server sending the offer and the duration of the lease being offered.
- If no **DHCPOFFER** is received by the client, it keeps rebroadcasting the message time and again with a timeout until it gets one.

- The client selects the first offer it receives and broadcasts an IP lease selection message specifying the IP address it has selected, known as **DHCPREQUEST**.
- This request also contains the clientôs hardware address, so that the server can acknowledge this request.

- The DHCP server that offered the selected lease responds with a DHCP lease acknowledgement message known as **DHCPACK**.
- The DHCPACK contains the same information as the **DHCPOFFER** that was sent, plus optional DHCP information that has been configured for that scope.

- If the requested lease is no longer available, the DHCP server broadcasts a DHCP negative acknowledgement (**DHCPNACK**).
- When the client receives the **DHCPNACK**, it must start the lease request process over again with a **DHCPDISCOVER** message.

- After having sent the **DHCPACK**, the server updates its DHCP database so that the lease is no longer available.
- After receiving the **DHCPACK**, the client continues to initialize TCP/IP.
- During this start-up phase, the client updates its registry so that it includes the IP addressing information with the lease.

- To renew a lease, the client sends a **DHCPREQUEST** to the server, and the whole process repeats.

# DHCP Address Reservation

- There are situations in which a particular client must always have the same IP address, e.g.:
  - » hosts on  a network with non-WINS clients. If the host always has a new IP address, the non-WINS clients will not be able to reach them.
- So, DHCP allows to reserve an IP address for a host, if need be.

# [illegible] i p

- ipconfig
- ipconfig /all
- ipconfig /renew <adapter>
- ipconfig /release <adapter>
- ipconfig /?

# Name Resolution Process

- To resolve a destination host name to an IP address, source host needs access to an IP address mapping for the required host name.
- The source host checks the following locations in order:
  - » Its own host name
  - » The local HOSTS file

» If configured to one or more DNS servers, it checks for a mapping entry on the DNS servers.

» Its own NetBIOS name cache for a NetBIOS name that is the same as the required destination host name.

» If configured to use one or two WINS servers, then it attempts to contact the WINS server(s) to see if it contains a NetBIOS name that is required.

- It then tries up to 3 b-node broadcasts to see if the destination host name responds with its IP address.
- It then consults its local LMHOSTS file for a NetBIOS name that is the same as the required destination.

# NetBIOS Host Name Resolution

- A **NetBIOS name** is the name used by the NetBIOS Application Programming Interface (API) of the Microsoft networking architecture.
- Computer names, Windows NT domain names, printer names, and share names are all examples of NetBIOS names.

- Most of the time NetBIOS names are only used for those network functions supporting NetBIOS, e.g., Microsoft network.
- NetBIOS applications use NetBIOS names when talking about physical and virtual network devices such as computer names and share names.

# Name Registration

- When a TCP/IP host starts up and logs on to the network, it sends out a little note called a **NetBIOS name registration request**.
- If the host is configured to register its NetBIOS name with a WINS server, the registration request is sent directly to the server.

- The alternative is to broadcast the announcement on the local network - called a UDP NetBIOS name registration request.
- If a registration request is broadcast and another host is already using the NetBIOS name, the host sends a negative name registration response to the sender.

- If a WINS server receives a name registration request for a name already in the database, it compares the IP addresses to see if the name request came from the host that already has the name.
- If the addresses differ, it queries the currently registered host and waits for a response.

- If WINS gets no response to the query, it accepts the new registration.
- By default a Windows NT client will first use WINS, then broadcast for name registration and resolution if WINS is not enabled.

# NetBT Name Resolution Modes

- The NetBIOS **name resolution mode** (also called the NetBIOS **node type**) sets the order a client must follow when trying different methods to resolve a NetBIOS name.
- These modes are:
  - » enhanced b-node
  - » p-node

  - m-node
  - h-node

- By default, a Windows NT computer uses enhanced b-node, unless it configures with a WINS server address, in which case it uses h-node.

# B-Node

- The mode b-node uses NetBIOS name query broadcasts to resolve and register NetBIOS names.
- The broadcasts are limited to the local network and to any networks that are attached by routers set up to forward b-node broadcasts.
- This is not the preferred mode as it causes excessive broadcasts.

# Enhanced B-node

- Enhanced b-node attempts to look up the NetBIOS name in a local LMHOSTS file if the broadcasts fail to resolve the name.
- To enable enhanced b-node, LMHOSTS look up should be enabled.

# P-node

- P-node mode uses a WINS server to resolve names, and broadcasts are not used.
- In p-node resolution mode, a host sends the name resolution query directly to the IP address of the WINS server.
- P-node queries can also be sent across networks.

# M-node

- The m-node mode first tries to use b-mode broadcasts.
- If this fails, a p-node name query is sent directly to the WINS server.

# H-Node

- H-node is the default name resolution mode when a host is configured with the IP address of a primary WINS server.
- In this mode, a host first checks the local NetBIOS name cache for an IP address mapping for the NetBIOS name.
- If this does not produce anything, the host sends name queries directly to the WINS server.

- The queries will automatically go to a secondary WINS server if the primary is unavailable and if a secondary WINS server is configured.
- If neither WINS server responds, the host sends up to 3 b-node broadcasts to the local network to resolve the name.

- If any of these modes does not get the proper response, then the following methods are tried:
  - » local LMHOSTS file
  - » DNS server
  - » local HOSTS file

# nbtstat

- nbtstat -n
- nbtstat -c
- nbtstat -r
- nbtstat /?

# Computer Browser Service

- An advantage of a networked computer system is the ability to share resources.
- In order for these resources to be used efficiently, there needs to be a way to determine what resources are available.
- To accomplish this, Windows NT uses the Computer Browser service to identify and list the available network resources.

- The Windows NT Computer Browser service assigns specific computers on the network to be browsers.
- A **browser** maintains a centralized list of available network servers, thereby eliminating the need for all computers to maintain a list of shared network resources.

- By assigning the browser role to specific computers, the Computer Browser service lowers the amount of network traffic required to build and maintain a list of shared resources on the network.

# Browser Types

- **Domain Master Browser**: It is the computer that collects and maintains the master list of available network servers. It also distributes this list to the master browser of each subnet in the domain.
- **Master Browser**: It is the computer that maintains the master list of available network servers. It also distributes this list, called **browse list**, to backup browsers.

- **Backup Browsers**: It is a computer that receives a copy of the browse list from the master browser. It then distributes the list to the browser clients upon request.
- **Potential Browser**: It is a computer that is capable of becoming a browser if instructed to do so by a master browser.

# Browse Process

- After startup, all computers that are running the server service announce their presence to the master browser in their workgroup or domain subnet.
- The first time a client computer attempts to locate an available network server, it contacts the master browser of the domain subnet or workgroup for a list of backup browsers.

- The client then requests the network server list from a backup browser, which responds to the requesting client with a list of domains and workgroups and the lists of servers local to the clientôs domain or workgroup.

# Browser Election

- When a client computer cannot locate a master browser, or when a backup browser attempts to update its network resource list and cannot locate the master browser, a new master browser must be selected.
- This selection process is called a **browser election**.

- The election process ensures that only one master browser exists per workgroup or segment in a domain.
- A browser is elected based on the type of operating system it is running, the version of the operating system and its current role in browsing.

- The order is as follows:
  - » Windows NT Server
  - » Windows NT Workstation
  - » Windows 95/98
  - » Windows for Workgroups
  - » OS versions, in order or preference, are: 4.0, 3.51, 3.5, or 3.1
  - » The order of preference in the current browsing role is: master, backup, potential.

- Browse lists are maintained and exchanged using local broadcasts.
- If the domain spans routers, however, one of the following conditions must be met before browse lists can be distributed properly:
  - a router must be configured to forward NetBT broadcasts.
  - WINS must be used.

# Internet Name Service (WINS)

- The WINS is a database server where hosts can register their NetBIOS names and IP addresses instead of broadcasting their name registrations and queries.
- When a host (WINS client) needs to communicate with another host by its NetBIOS name, the WINS client asks the WINS server for the recipientôs IP address.

- The WINS server will send back the correct IP address if the recipient registered with the WINS server.
- The major benefit of using WINS is that NetBIOS name query broadcasts are reduced or eliminated because clients send name queries directly to the WINS server, even if it is on another network.

- The major drawback of WINS is that it is a Microsoft-only protocol.
- WINS will not detect NetBIOS names already in use on a network by non-WINS clients.

# WINS Operation

- WINS uses a four-step process to resolv NetBIOS names to IP addresses:
  - » NetBIOS name registration
  - » NetBIOS name registration
  - » NetBIOS name release
  - » NetBIOS name query and resolution

# Name Registration

- When a WINS client initializes, NetBIOS over TCP/IP (NetBT) sends a **name registration query** (called a **NAMEREGISTRATIONREQUEST**) message directly to the primary WINS server for that client.
- This query includes the source IP address of the client, destination IP address of the server, and the name to be registered.

- If the WINS server is available and the name is not already registered, the server replies to the client with a **positive name registration response** message (also called a **NAMEREGISTRATIONRESPONSE**).
- The registration has a time out value after which the name is removed from the database unless renewed.

# Name Renewal

- To continue using a registered name, a WINS client must periodically renew its registration with the WINS server.
- If it is not renewed, the registration expires.

# Name Release

- When a WINS client begins to shutdown, it sends a name release request to the WINS server for each of its registered names.
- When the WINS server receives this request, it checks the local WINS database.

- If the name exists, the WINS server marks its entry as **released** and sends a **positive name release** response to the client.

# [illegible] W I

- Clients that do not support WINS usually use b-node broadcasts to resolve NetBIOS names to IP addresses.
- To catch these broadcasts, a WINS proxy agent can be setup which will listen for NetBIOS name registration and query broadcasts and then forwards these requests directly to the WINS server.

# WINS Database Replication

- WINS servers on different subnets can send database updates to each other regularly to keep NetBIOS name information current.
- This is called **replication** of WINS database.

- To setup replication, each server is configured as either a **push partner** or a **pull partner**.
- A **push** means to send database changes to another WINS; a **pull** requests database changes from another WINS server.

# DNS and WINS

- The DNS sevrer on Microsoft Windows NT can be configured to use WINS to resolve names that do not appear in the zone database.
- This can be really helpful where hosts on the network are assigned IP addresses (and names) using DHCP which are then registered with WINS.

- Now if the network consists of non-WINS (e.g., Unix) clients which can only do DNS lookup, then the DNS server can do WINS lookup on behalf of the non-WINS client.
- To configure Microsoft DNS to use WINS, at least one WINS server must be operating, and WINS lookup must be enabled in the zone database.