برنامه نویسی پیشرفته

رفع اشکال: جلسه ۹

Socket Programming

Basics of TCP/UDP sockets.

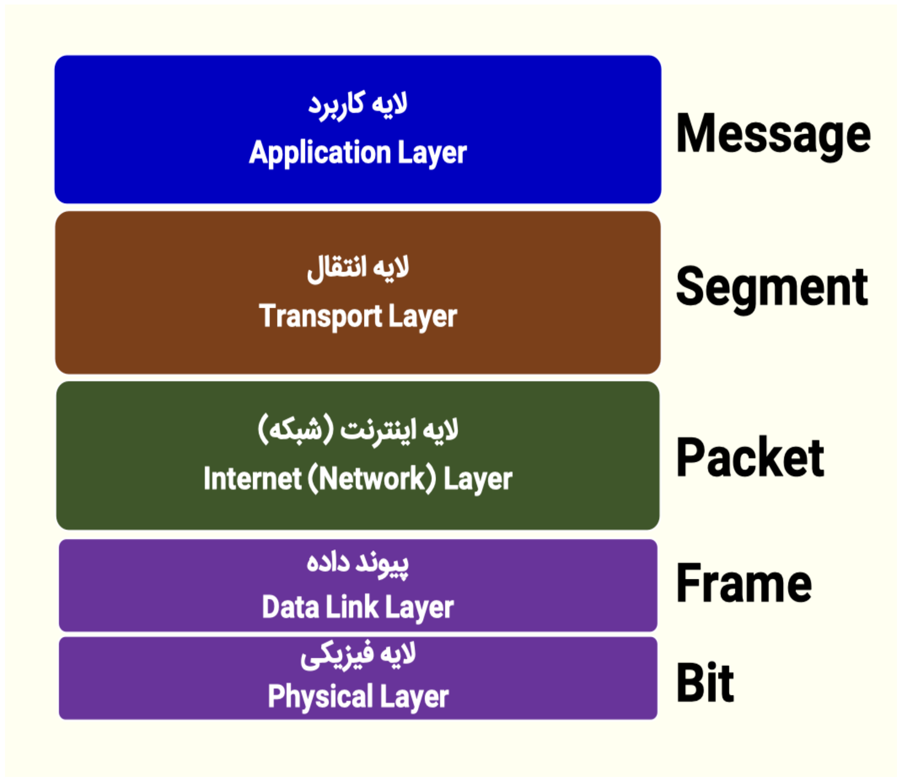
Building a simple client-server application.

Introduction to Http

NETWORK ARCHITECTURE

A network architecture is a set of layers and protocols used to reduce network design complexity.

- Each layer is connected only to the two layers above and below it.
- The service of each layer to the higher layer is data transfer.
- Each layer has a specific task routing, error control, or …
- Each layer in the source node has a protocol with its corresponding layer in the destination node to perform its tasks.



TCP/IP MODEL

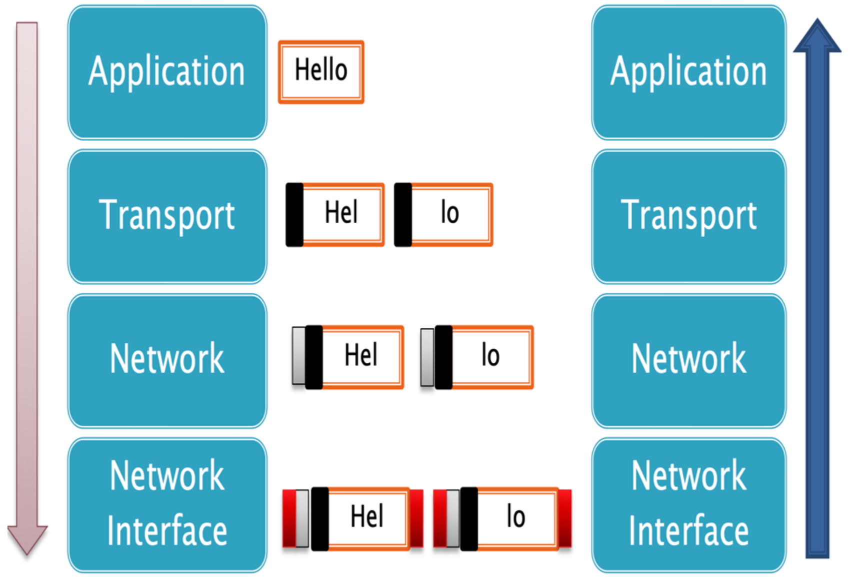
Application Layer : Where user applications such as browsers, chat clients, or Java applications work with data.

Transport Layer : This layer breaks data into smaller pieces called segments and is responsible for ensuring that they are sent correctly and in the correct order.

Internet/Network Layer : Adding the destination and source IP addresses so the packet knows where to go.

Data Link Layer : Responsible for packet delivery between two direct nodes on the network (e.g. two devices on a LAN) via MAC address.

Physical Layer : At this layer, data is converted into bits (0 and 1) and transmitted as electrical, optical, or wave signals over a cable or Wi-Fi.



CLIENT-SERVER MODEL

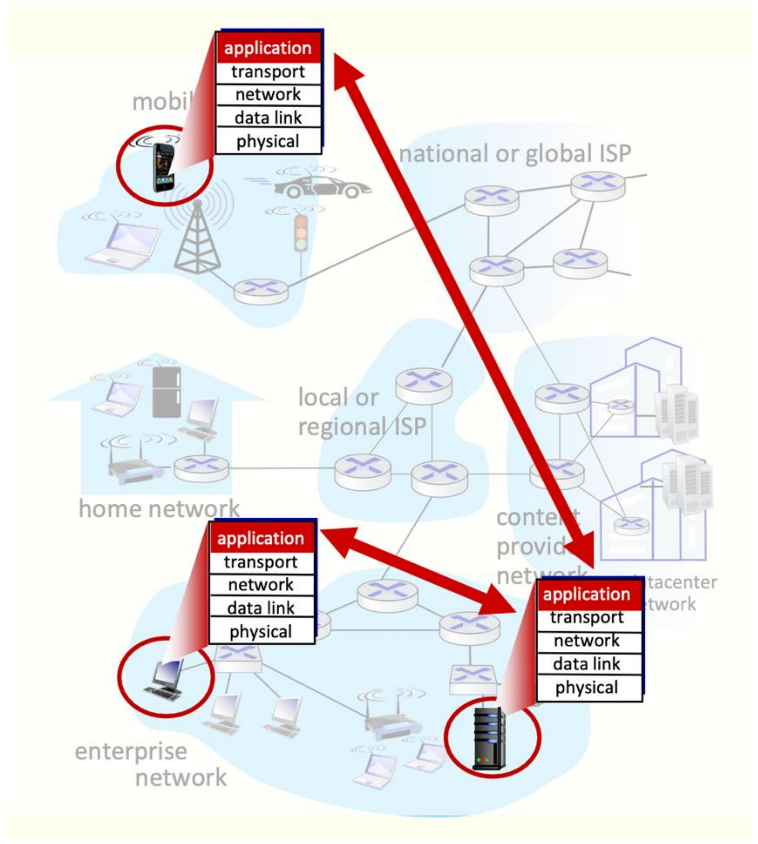
Applications have only one of two roles: server or client.

Servers :

- Are always running and ready to serve.
- Static IP address
- Has a permanent connection to the Internet.
- Often in data centers for scalability and to reduce running costs

Clients

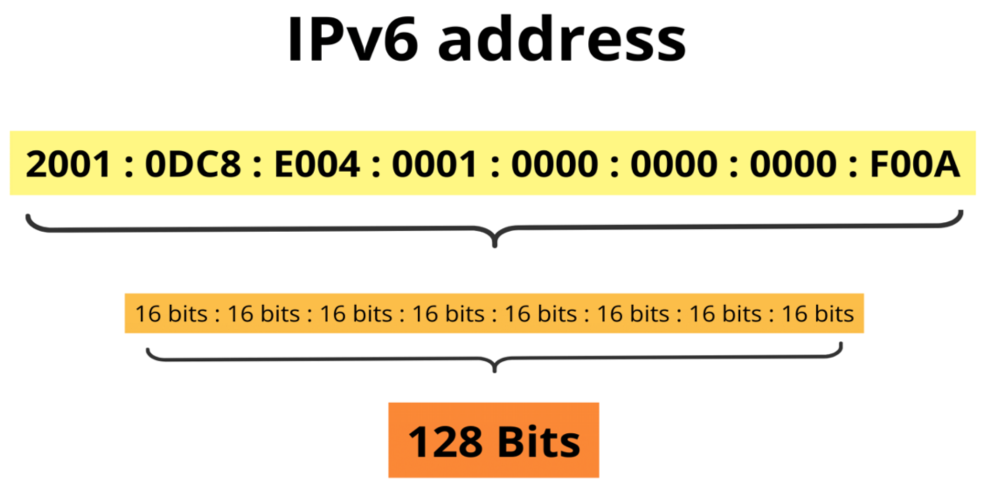
- Must already have the IP address and port number of the server
- By sending a message to the server, they announce their request to receive the service.
- May not have a static IP address.
- May not have a permanent connection to the Internet.

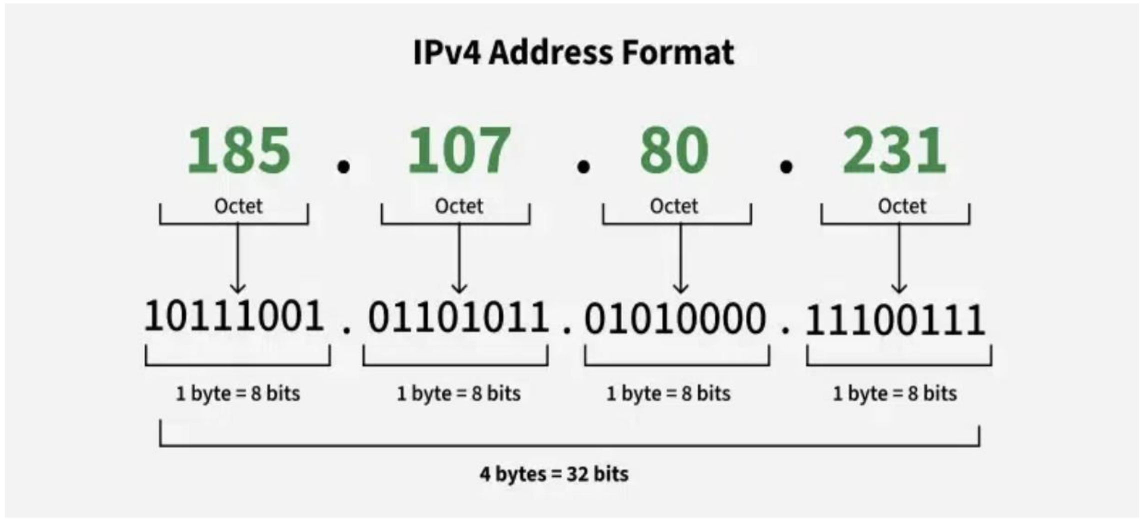


IP ADDRESS

An IP address is a unique numerical identifier assigned to each device on a network to allow communication between nodes.

Like a postal address, an IP address helps systems send data to the correct destination.


IPv6 address
2001 : 0DC8 : E004 : 0001 : 0000 : 0000 : 0000 : F00A
16 bits : 16 bits : 16 bits : 16 bits : 16 bits : 16 bits : 16 bits : 16 bits
128 Bits



PORT

A port is a 16-bit number (between 0 and 65535) used as an identifier for a service or process on a device

When data arrives at a device, the IP specifies which device, and the port specifies which program receives it.

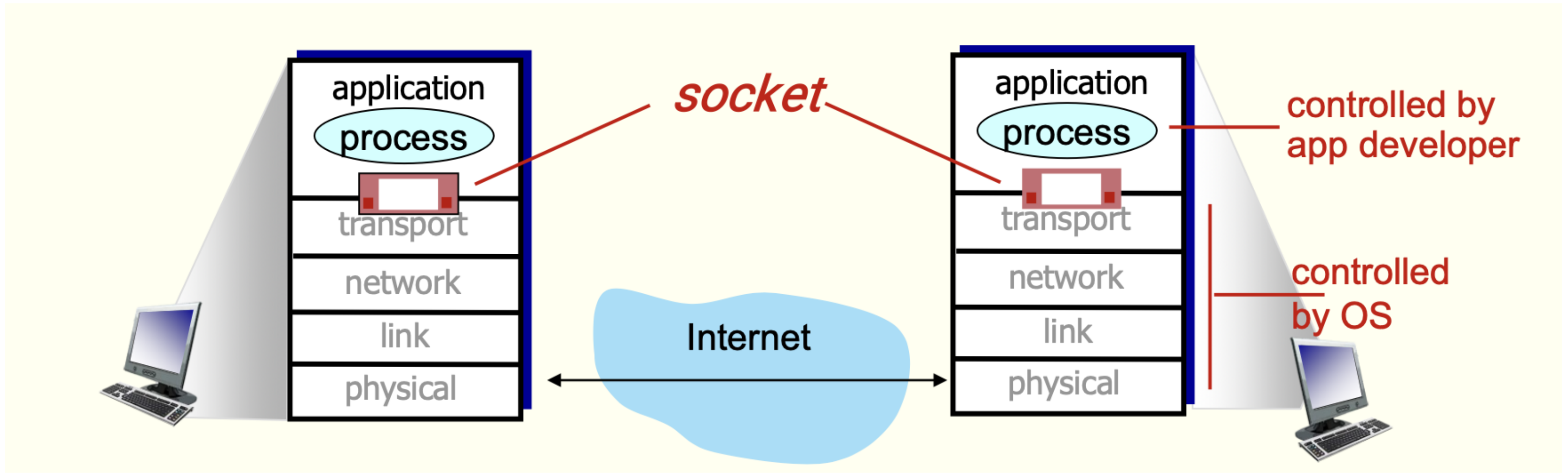
Well-Known Ports : Range 0 to 1023
Reserved by IANA for basic and universal services .
Protocols and services that must be available on all systems and networks operate on this range of ports.

Dynamic / Private Ports (Ephemeral) : Range: 49152 to 65535

Allocator: Automatically by the operating system
When a client wants to establish a temporary connection, the OS itself selects a free port.
These ports are used only for the duration of the connection.

SOCKET

A socket is a software endpoint between two processes that communicate over a network using a transport protocol (such as TCP or UDP) on two different machines or on the same machine.



In fact, a socket is a point in the Application Layer through which data is input/output, and the operating system is responsible for transferring it through the lower layers (Transport, Network, Link, Physical).

NOTE

In Java, when you use Socket, you are essentially saying to the OS:

"Please open a communication port for me to talk to another program on another system."

The OS:

- Manages buffers
- Does the packaging and addressing
- Provides a stable and secure TCP connection or a fast and simple UDP connection

A socket is a logical gateway between your application and the network; not a physical connection

PROTOCOL

A protocol is a set of rules for transferring data. In socket writing, it is usually chosen between TCP and UDP.

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection-oriented	Yes (connect(), handshake required)	No (connectionless)
Delivery guarantee	Guaranteed	Not guaranteed
Order of data	Ordered delivery	May arrive out of order
Speed	Slower (due to reliability checks)	Faster (no delivery assurance)
Overhead	Higher	Lower
Java Usage	Socket, ServerSocket	DatagramSocket, DatagramPacket

TCP

Server

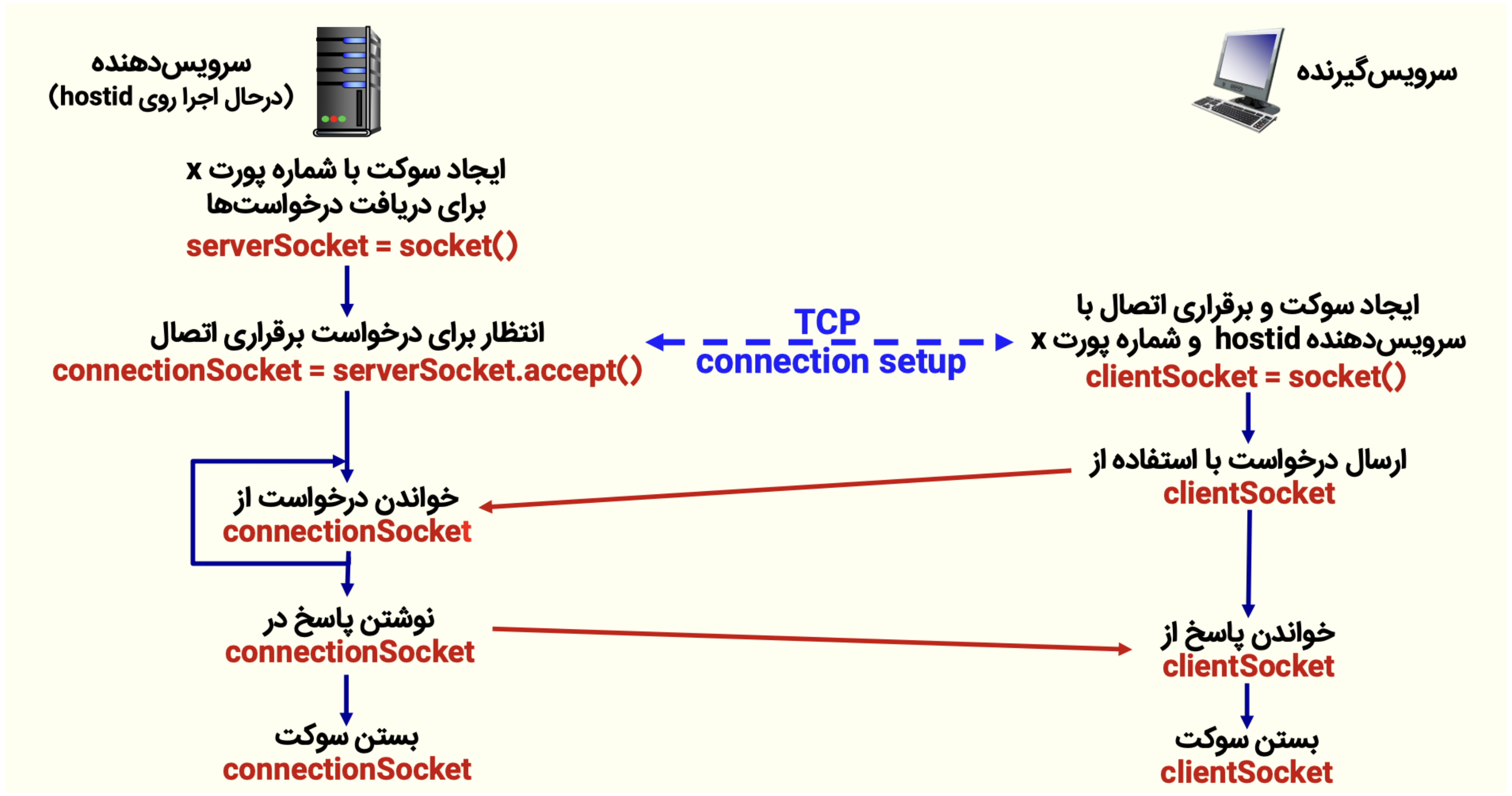
```
public static void main(String[] args) {

    try {

        ServerSocket serverSocket = new
        ServerSocket(6000);

        System.out.println("Server is listening...");

        Socket client = serverSocket.accept();

        System.out.println("connected to a client");

        System.out.println(client.getPort());

    } catch (IOException exception) {

        exception.printStackTrace();

    }

}
```

Client

```
public static void main(String[] args) {
        try {
                Socket socket = new Socket("127.0.0.1",
                6000);
                System.out.println(socket.getInetAddress()
                );
                } catch (IOException exception) {
                exception.printStackTrace();
        }
}
```

* The accept method is a blocking method and the following lines will not be executed until the client connects to the server

TCP DIAGRAM


سرویس‌دهنده
(درحال اجرا روی hostid)
سرویس‌گیرنده
ایجاد سوکت با شماره پورت x
برای دریافت درخواست‌ها
serverSocket = socket()
انتظار برای درخواست برقراری اتصال
connectionSocket = serverSocket.accept()
TCP
connection setup
ایجاد سوکت و برقراری اتصال با
سرویس‌دهنده hostid و شماره پورت x
clientSocket = socket()
ارسال درخواست با استفاده از
clientSocket
خواندن درخواست از
connectionSocket
نوشتن پاسخ در
connectionSocket
خواندن پاسخ از
clientSocket
بستن سوکت
connectionSocket
بستن سوکت
clientSocket

UDP

USAGE :Very high speed and tolerance for some packet loss For VoIP & online gaming

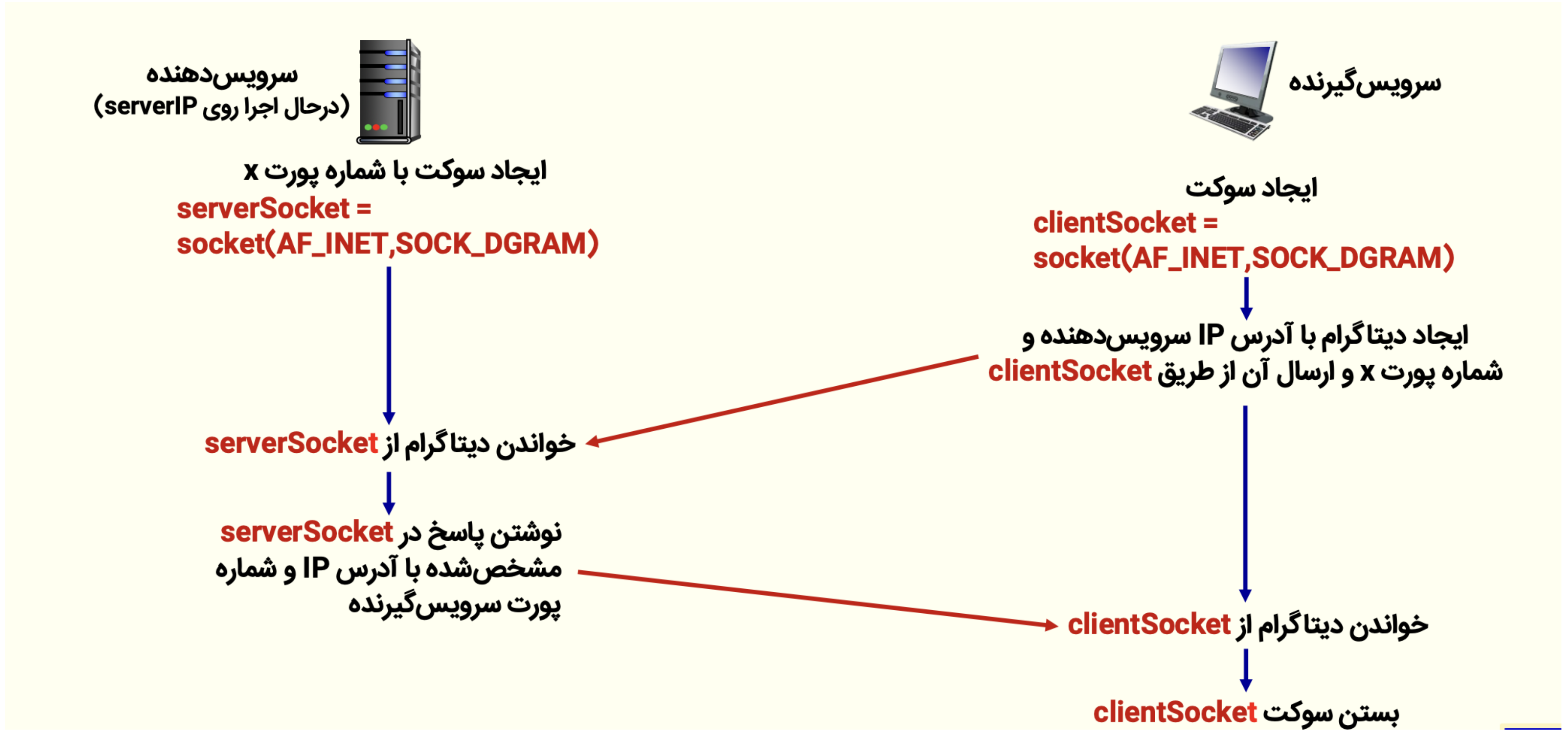
Client

```
DatagramSocket socket = new DatagramSocket();
String msg = "Hello UDP";
DatagramPacket packet = new
DatagramPacket(msg.getBytes(),msg.length(),InetAddress.getByName("localhost"), 6000);
socket.send(packet);
```

Server

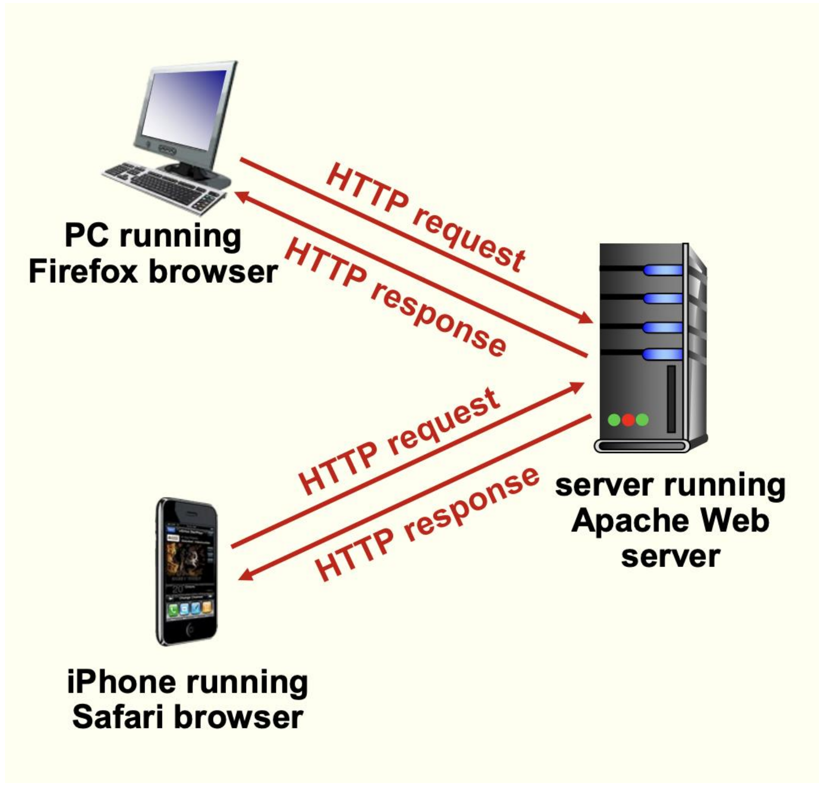
```
DatagramSocket serverSocket = new DatagramSocket(6000);
byte[] buffer = new byte[1024];
DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
serverSocket.receive(packet);
System.out.println("Received: " + new String(packet.getData()));
```

UDP DIAGRAM


سرویس‌دهنده
(درحال اجرا روی serverIP)
ایجاد سوکت با شماره پورت x
serverSocket =
socket(AF_INET,SOCK_DGRAM)
خواندن دیتاگرام از serverSocket
نوشتن پاسخ در serverSocket
مشخص‌شده با آدرس IP و شماره
پورت سرویس‌گیرنده
سرویس‌گیرنده
ایجاد سوکت
clientSocket =
socket(AF_INET,SOCK_DGRAM)
ایجاد دیتاگرام با آدرس IP سرویس‌دهنده و
شماره پورت x و ارسال آن از طریق clientSocket
خواندن دیتاگرام از clientSocket
بستن سوکت clientSocket

INTRODUCTION TO HTTP

(HyperText Transfer Protocol) is an application-layer protocol used to transfer hypertext (like HTML) over the web between clients and servers


HTTP request
HTTP response
PC running Firefox browser
HTTP request
HTTP response
server running Apache Web server
iPhone running Safari browser

Client
The web browser uses the HTTP protocol to request web objects.
The web browser displays the received web pages in response to its requests.

Server
The web server uses the HTTP protocol to send web pages in response to the client's requests.

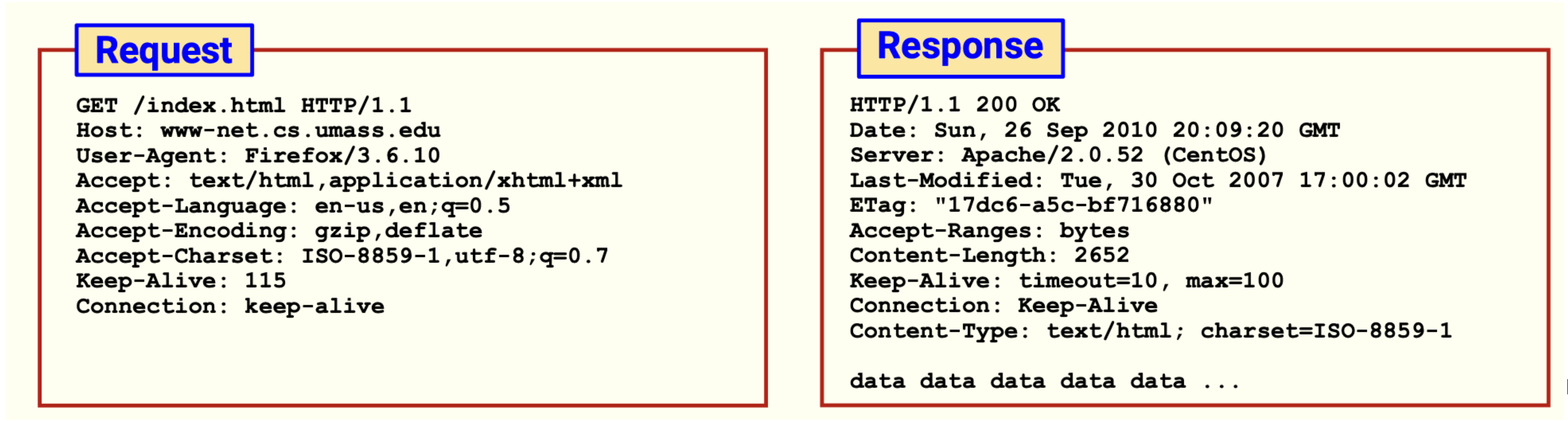
HTTP is just data sent in a specific format over TCP.
And just the appearance of the exchange, but its transport is the responsibility of TCP and sockets.

HTTP METHODS

Method	Purpose
GET	Retrieve data from server (read-only)
POST	Submit data (e.g., form submission)
PUT	Update or replace a resource
DELETE	Remove a resource

HTTP is stateless !

Because each request is independent and the server does not remember previous requests unless you use mechanisms like cookies or sessions.



Request

```
GET /index.html HTTP/1.1
Host: www-net.cs.umass.edu
User-Agent: Firefox/3.6.10
Accept: text/html,application/xhtml+xml
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 26 Sep 2010 20:09:20 GMT
Server: Apache/2.0.52 (CentOS)
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT
ETag: "17dc6-a5c-bf716880"
Accept-Ranges: bytes
Content-Length: 2652
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

data data data data data ...
```

QUESTION

Choose the correct statements

1) By executing the accept method in the Socket class, the server's desired conditions are agreed upon and the connection is established.

2) The length of the IP address (version (IPv4)) is exactly 32 bits.

3) In a single-threaded Java program, it is not possible to easily send or receive messages to multiple users simultaneously.

4) To create a socket and connect to the server, it is enough to have the port of that server.

ANSWER

1) False The Socket class is for the client. The accept() method is for the ServerSocket class. So this option is technically wrong.

2) True IPv4 addresses are made up of 4 8-bit numbers (such as 192.168.1.1), totaling 32 bits.

3)True In single-threaded applications, it is difficult to accept and respond to simultaneous requests from multiple users because there is only one execution path.

4) False To connect to a server, we need both the IP address and the port number. The port alone is not enough.

پایان