CHAPTER 4

■ ■ ■

Network Programming

QUESTIONS AND EXERCISES

1. What is network programming in Java?

Answer:

Writing Java programs that facilitate the exchange of information between processes running on different computers on the network is called network programming.

2. What are the network types: LAN, CAN, MAN, and WAN?

Answer:

A network can be categorized based on the geographical area it is spread over as follows:

- *Local Area Network (LAN)*: It covers a small area such as a building or a block of buildings.
- *Campus Area Network (CAN)*: It covers a campus such as a university campus, interconnecting multiple LANs within that campus.
- *Metropolitan Area Network (MAN)*: It covers more geographical area than a LAN. Usually, it covers a city.
- *Wide Area Network (WAN)*: It covers a larger geographical area such as a region of a country or multiple regions in different countries in the world.

3. What is a network protocol?

Answer:

Rules defining communication between network devices is called a network protocol.

4. What is an IP address? Can a computer have more than one IP address?

Answer:

An IP address uniquely identifies a connection between a computer and a router.

Yes. A computer can have more than one IP address. A computer can be connected to multiple networks using multiple routers and each connection between the computer and the router will have a unique IP address. In such cases, the computer will be assigned multiple IP addresses and the computer is known as *multi-homed*. Multi-homing increases the availability of the network connection to a computer. If one network connection fails, the computer can use other available network connections.

5. How many bytes are used to represent an IP address in IPv4 and IPv6? Describe the textual format of representing IP addresses in IPv4 and IPv6 formats.

Answer:

IPv4 and IPv6 use 32 bits and 128 bits, respectively, to represent IP addresses.

IPv4 allows you to specify IP addresses using four decimal numbers. Each decimal number is in the range from 0 to 255. Programs take care of converting decimal numbers into a 4-byte binary number that will be used by the computer. The decimal number format of IPv4 is called dotted decimal format because a dot is used to separate two decimal numbers. Each decimal number represents the value contained in 8 bits of the 32-bit number. For example, an IPv4 address of `11000000101010000000000111100111` in the binary format can be represented as `192.168.1.231` in the dotted decimal format.

IPv6 uses an easy-to-understand notation to represent an IP address in a textual form. The 128 bits are divided into 8 fields of 16 bits each. Each field is written in hexadecimal form and separated by a colon. The following are some examples of IPv6 addresses:

- `F6DC:0:0:4015:0:BA98:C0A8:1E7`
- `F6DC:0:0:7678:0:0:0:A21D`
- `F6DC:0:0:0:0:0:0:A21D`
- `0:0:0:0:0:0:0:1`

It is common to have many fields in an IPv6 address with zero values, especially for all IPv4 addresses. The IPv6 address notation lets you compress contiguous fields of zero values by using two consecutive colons. You can use two colons to suppress contiguous zero value fields only once in an address. The above IPv6 address may be rewritten using the zero-compression technique:

- `F6DC::4015:0:BA98:C0A8:1E7`

- F6DC:0:0:7678::A21D
- F6DC::A21D
- ::1

Note that we could suppress only one of the two sets of contiguous zero fields in the second address, F6DC:0:0:7678::A21D. Rewriting it as F6DC::7678::A21D would be invalid because it uses two colons more than once. You can use two colons to suppress contiguous zero fields, which may occur in the beginning, middle, or end of the address string. If an address contains all zeros in it, you can represent it simply as ::.

You can also mix hexadecimal and decimal formats in an IPv6 address. The notation is useful when you have an IPv4 address and want to write it in IPv6 format. You can write the first six 16-bit fields using a hexadecimal notation as described above and use dotted decimal notation for IPv4 for the last two 16-bit fields. The mixed notation takes the form X:X:X:X:X:X:D.D.D.D, where an X is a hexadecimal number and a D is a decimal number. You can rewrite the above IPv6 addresses using this notation as follows:

- F6DC::4015:0:BA98:192.168.1.231
- F6DC:0:0:7678::0.0.162.29
- F6DC::0.0.162.29
- ::0.0.0.1

6. You have an IP address of 0.0.0.0, which is in IPv4 format. How will you rewrite this IP address in IPv6 format?

Answer:

You can the IP address 0.0.0.0 in represent IPv4 as :: in IPv6.

7. Describe the use of the following address types: Loopback IP Address, Unicast IP Address, Multicast IP Address, Anycast IP Address, Broadcast IP Address, and Unspecified IP Address

Answer:

Loopback IP Address

You need at least two computers connected via a network to test or run a network program. Sometimes it may not be feasible or desirable to set up a network when you want to test your network program during the development phase of your project. There is a

provision in the IP addressing scheme to treat an IP address as a loopback address to facilitate testing of network programs using only one computer. When the Internet layer in the protocol suite detects a loopback IP address as the destination for an IP datagram, it does not pass over the packet to the protocol layer below it (that is network interface layer). Rather, it turns around (or loops back, hence the name *loopback address*) and routes the packet back to the transport layer on the same computer. The transport layer will deliver the packet to the destination process on the same host as it would have done had the packet come from a remote host. A loopback IP address makes testing of a network program using one computer possible.

Loopback IP addresses are reserved addresses and the IP is required not to forward a packet with a loopback IP address as its destination address to the network interface layer.

In the IPv4 addressing scheme, `127.X.X.X` block is reserved for loopback addresses, where `X` is a decimal number between 0 and 255. Typically, `127.0.0.1` is used as a loopback address in IPv4. However, you are not limited to using only `127.0.0.1` as the only loopback address. If you wish, you can also use `127.0.0.2` or `127.3.5.11` as a valid loopback address. Typically, the name `localhost` is mapped to a loopback address of `127.0.0.1` on a computer.

In an IPv6 addressing scheme, there is only one loopback address, which is sufficient to perform any local testing for a network program. It is `0:0:0:0:0:0:0:1` or simply `::1`.

Unicast IP Address

Unicast is one-to-one communication between two computers on a network in which an IP packet is delivered to a single remote host. A unicast IP address identifies a unique host on a network. IPv4 and IPv6 support unicast IP addresses.

Multicast IP Address

Multicast is a one-to-many communication where one computer sends an IP packet that is delivered to multiple remote computers. Multicasting lets you implement the concept of group interaction such as audio or video conferencing, where one computer sends information to all computers in the group. The benefit of using multicasting in place of multiple unicasts is that the sender sends only one copy of the packet. One copy of the packet travels along the network as long it can. If receivers of the packet are on multiple networks, a copy of the packet is made when needed, and each copy of the packet is routed independently. Finally, each receiver is delivered an individual copy of the packet. Multicasting is an efficient way of communication between group members as it reduces network traffic.

IP contains some addresses in its address space as multicast addresses. If a packet is addressed to a multicast address, the packet will be delivered to multiple hosts. Both IPv4

and IPv6 support multicast addressing. In IPv4, Class D network addresses are used for multicasting. That is, the four highest order bits are 1110 in a multicast address in IPv4. In IPv6, a multicast address has the first 8 bits set to 1. That is, a multicast address in IPv6 always starts with FF. For example, FF0X:0:0:0:0:0:2:0000 is a multicast address in IPv6.

Anycast IP Address

Anycast is a one-to-one_from_a_group communication where one computer sends a packet to a group of computers, but the packet is delivered to exactly one computer in the group. IPv4 does not support anycasting. IPv6 supports anycasting. In anycasting, the same address is assigned to multiple computers. When a router receives a packet, which is addressed to an anycast address, it delivers the packet to the nearest computer. Anycasting is useful when a service has been replicated at many hosts and you want to provide the service at the nearest host to the client. Sometimes, anycast addressing is also called *cluster addressing*. An anycast address is used from the unicast address space. You cannot distinguish a unicast address from an anycast address by looking at their bit arrangements. When the same unicast address is assigned to multiple hosts, it is treated as an anycast address. Note that the router must know about the hosts that are assigned an anycast address, so that it can deliver the packets addressed to that anycast address to one of the nearest hosts.

Broadcast IP Address

Broadcast is a one-to-all communication where one computer sends a packet and that packet is delivered to all computers on the network. IPv4 assigns some addresses as broadcast addresses. When all 32 bits are set to 1, it forms a broadcast address and the packet is delivered to all hosts on the local subnet. When all bits in the host address are set to 1 and a network address is specified, it forms a broadcast address for the specified network number. For example, 255.255.255.255 is a broadcast address for a local subnet and 192.168.1.255 is a broadcast address for a network 192.168.1.0. IPv6 does not have a broadcast address. You need to use a multicast address as the broadcast address in IPv6.

Unspecified IP Address

0.0.0.0 in IPv4 and :: in IPv6 (note that :: denotes 128-bit IPv6 address with all bits set to zero) are known as unspecified addresses. A host uses this address as a source address to indicate that it does not have an IP address yet, such as during the boot up process when it is not assigned an IP address yet.

8. What is a port number and why is it used?

Answer:

A port number is a 16-bit unsigned integer ranging from 0 to 65535. It is a logical number used by the transport layer to identify a destination process for a packet using a specific protocol on a computer.

9. What is a socket? What is the difference between a connection-oriented socket and a connectionless socket? Give an example of a protocol that supports these types of sockets.

Answer:

Operating systems provide an application program interface (API) called a *socket*, which lets two remote applications communicate, taking advantage of lower level protocols in the protocol stack. A socket is an interface between the transport layer and the application layer. It provides a standard way of communication between the two layers, which in turn provides a standard way of communication between two remote applications. There are two kinds of sockets:

- A *Connection-Oriented Socket*
- A *Connectionless Socket*

A connection-oriented socket is also called a *stream socket*. Transmission Control Protocol (TCP), which is used in a transport layer, is one of the most widely used protocols to provide connection-oriented sockets. Transmission Control Protocol (TCP), which is used in a transport layer, is one of the most widely used protocols to provide connection-oriented sockets. The application hands over data to a TCP socket and the TCP takes care of streaming the data to the destination host. The TCP takes care of all issues like ordering, fragmentation, assembly, lost data detection, duplicates data transmission, etc., on both sides of the communication, which gives the impression to the applications that data is flowing like a continuous stream of bytes from the source application to the destination application. No physical connection at the hardware level exists between two hosts that use TCP sockets. It is all implemented in software. Sometimes it is also called a *virtual connection*.

A connectionless socket is also called a *datagram socket*. User Datagram Protocol (UDP), which is used in a transport layer, is the most widely used protocol that provides a connectionless socket. It is unreliable, but much faster. It lets you send limited sized data—one packet at a time, which is different from TCP, which lets you send data as a stream of any size, handling the details of segmenting them in appropriate size of packets. Data delivery is not guaranteed when you send data using UDP. However, it is still used in many applications and it works very well. The sender sends a UDP packet to a destination and forgets about it. If receiver gets it, it gets it. Otherwise, there is no way to know—for the receiver—that there was a UDP packet sent to it. There is another important difference between UDP and TCP. UDP does not guarantee the ordering of data. That is, if you send five packets to a destination using UDP, those five packets may arrive in any order. However, TCP guarantees that packets will be delivered in the order they were sent. Java supports UDP sockets.

10. What does an instance of the `InetAddress` class represent? Write a program that prints the computer name and the IP address of the computer on which the program is executed.

Answer:

An object of the `InetAddress` class represents an IP address. It has two subclasses, `Inet4Address` and `Inet6Address`, which represent IPv4 and IPv6 addresses, respectively.

The following program prints the name and IP address of the computer on which the program is run:

```
// NameIPPrinter.java
package com.jdojo.network.exercises;

import java.net.InetAddress;
import java.net.UnknownHostException;

public class NameIPPrinter{
    public static void main(String[] args) {
        try {
            InetAddress addr = InetAddress.getLocalHost();
            System.out.println("My computer name: " + addr.getHostName());
            System.out.println("My computer IP address: " + addr.getHostAddress());
        } catch (UnknownHostException e) {
            e.printStackTrace();
        }
    }
}
```

11. What does an instance of the `InetSocketAddress` class represent?

Answer:

An object of the `InetSocketAddress` class represents a socket address.

12. What do the instances of the `ServerSocket` and `Socket` classes represent?

Answer:

An object of the `ServerSocket` class represents a TCP server socket. A `ServerSocket` object is used to accept a connection request from a remote client.

An object of the `Socket` class represents a TCP client socket.

13. What do the instances of the `DatagramSocket` and `DatagramPacket` classes represent?

Answer:

An object of the `DatagramSocket` class represents a UDP socket that is used to send or receive a datagram packet.

An object of the `DatagramPacket` class represents a UDP datagram that is the unit of data transmission over a UDP socket.

14. UDP sockets do not support an end-to-end connection like the TCP sockets. The `DatagramSocket` class, which represents UDP sockets, contains a `connect()` method. What is the purpose of this `connect()` method?

Answer:

The `connect()` method of the `DatagramSocket` class allows an application to restrict sending and receiving of UDP packets to a specific IP address at a specific port number.

15. What do the instances of the `MulticastSocket` class represent? Does a socket have to be a member of a multicast group to send a datagram packet to a multicast address?

Answer:

An object of the `MulticastSocket` class represents a multicast socket.

A socket itself does not have to be a member of a multicast group to send a datagram packet to a multicast address.

16. What are URI, URL, and URN? How do you represent them in a Java program?

Answer:

A Uniform Resource Identifier (URI) is a sequence of characters that identifies a resource. A resource identifier can identify a resource by a location, a name, or both.

A URI that uses a location to identify a resource is called Uniform Resource Locator (URL). For example, `http://www.yahoo.com/index.html` represents a URL that identifies a document named `index.html` at the host `www.yahoo.com`. Another example of a URL is

`mailto:ksharan@jdojo.com` in which the `mailto` protocol instructs the application that interprets it to open up an email application to send an email to the email address specified in the URL. In this case, the URL is not locating any resources. Rather, it is identifying the details of an email. You can also set the subject and the body parts of an email using the `mailto` protocol. Therefore, a URL does not always imply a location of a resource. Sometimes the resource may be abstract, as in the case of the `mailto` protocol. Once you locate a resource using a URL, you can perform some operations, such as retrieve, update, or delete, on the resource. The details of how the operations are performed depend on the scheme being used in the URL. A URL just identifies the parts of a resource location and scheme to locate it, not the details of any operations that can be performed on the resource.

A URI that uses a name to identify a resource is called a Uniform Resource Name (URN). For example, `URN:ISBN:978-1-4302-6661-7` represents a URN, which identifies a book using International Standard Book Numbers (ISBN) namespace.

URL and URN are subsets of URI.
