Distributed Network Programming

Prativa Nyaupane

Outline

- TCP
- UDP
- IP Address
- Ports
- Socket Programming using TCP and UDP
- Working with URLs and URL Connection Class
- Email Handling using Java Mail API
- Architecture of RMI
- Creating and Executing RMI applications
- Architecture of CORBA
- RMI vs CORBA
- IDL and Simple CORBA Program

Basic Concept:

- Network programming is the general concept of writing programs/codes that allows programs to communicate across a network.
- Java network programming is specific to Java programming language, where we use Java’s built-in libraries(java.net) package for network tasks.
- Java is one of the most premier programming languages for network programming because of the classes defined in java.net package.
- Java is designed for distributed environments of the Internet because it handle TCP/IP protocols.Java also supports Remote Method Invocation. RMI enables a program to invoke methods across a network.

Distributed Network Programming

Distributed network programming is writing programs that split tasks across multiple computers on a network.

A great real-life example of distributed network programming is a large video streaming service like Netflix. Instead of having one giant server delivering videos to everyone, they distribute the content across many servers around the world.This way, users can connect to the closest server and get their movie faster and smoother.

Distributed Network Programming Contd...

- The two main mechanisms through which java supports distributed network programming are:
  - Socket Programming(java.net)
    - A socket identifies an endpoint in a network.
    - A socket allows a single computer to serve many different clients at once through the use of ports.
    - A server process “listens” to a port until a client connects to it.
    - Socket communication takes place via Internet Protocol.
  - Remote Method Invocation RMI(java.rmi)
    - It allows objects in one Java Virtual Machine(JVM) to invoke methods on objects in another Java virtual machine, as if they were local objects.

Java Networking Terminology

1. IP(Internet Protocol) Address: Every computer on the Internet has an IP Address. The address is a number that uniquely identifies each computer on the net. It can be IPv4 or IPv6.

   For Google Search:
   Domain name is www.google.com
   IP address is 142.250.194.228

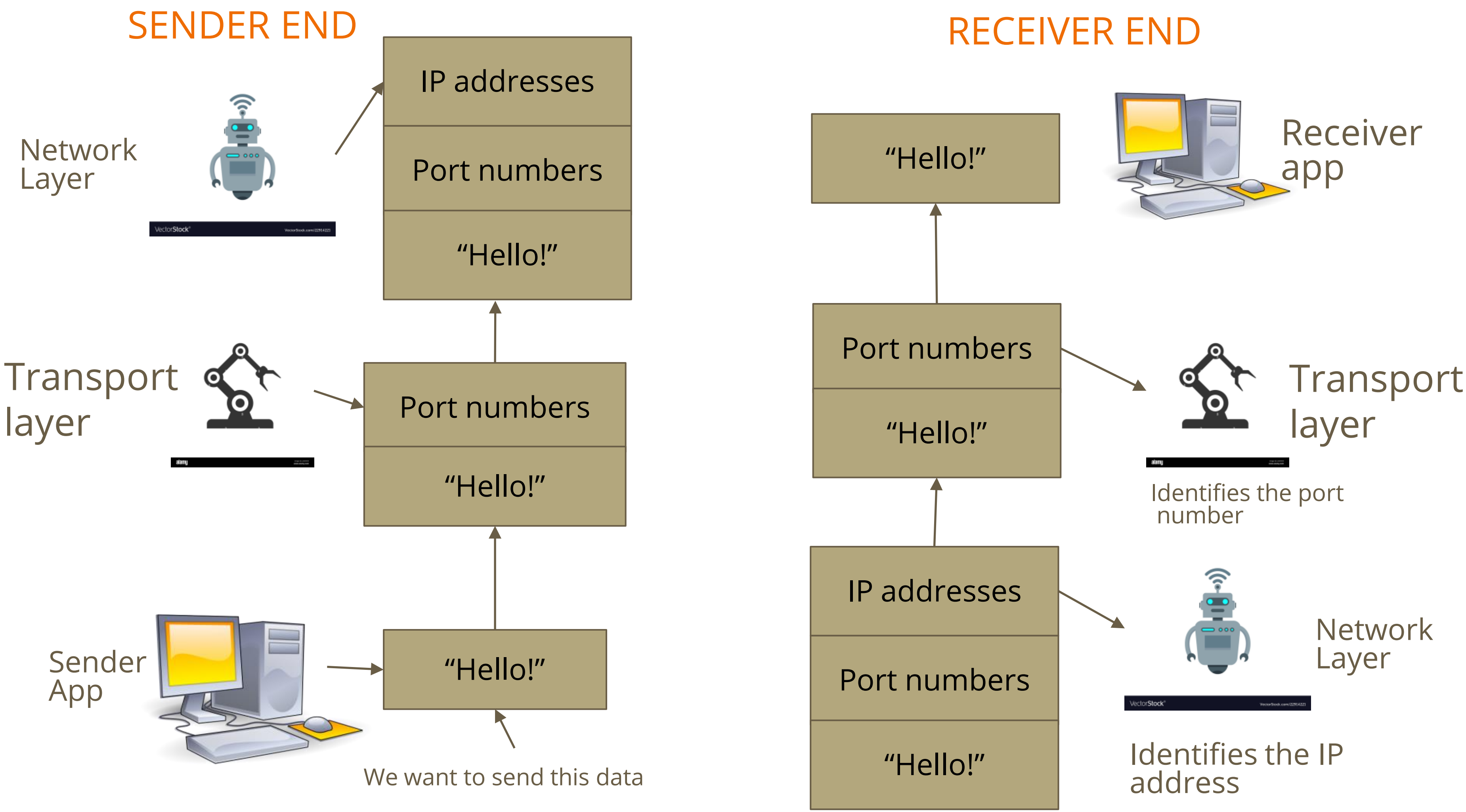
1. Protocol: A protocol is a set of rules that is followed for communication. It is used to define how data is formatted and transmitted. Two commonly used protocols are:
   a. TCP - It is reliable, connection-oriented protocol that guarantees the delivery of data in the order it was sent.
   b. UDP - It is lightweight, connectionless protocol that does not guarantee reliable data delivery or ordering.

Contd...

3. MAC Address: MAC(Media Access Control) address is a unique identifier of NIC(Network Interface Controller). MAC addresses work within a local network for faster device identification, while IP addresses allow routing across the entire internet.
4. Port Number: Port numbers act like labeled doors on a device, directing data to specific applications. A port is a numeric idedntifier that specifier a particular service or application. Common port numbers are:
    a. 80, 443 - Web pages(HTTP, HTTPS)
    b. 21- FTP(File Transfer Protocol)

Contd...

5. Connection-oriented and connection-less protocol: Connection oriented protocols always provide an ACK(acknowledgement) and are reliable because they guarantee the delivery of data in which it was sent but considerably slower than connection-less protocol. Mostly used in email, WWW and File Transfer Protocol. TCP(Transmission Control Protocol) is a connection-oriented protocol. Connectionless protocol does not provide ACK, are lightweight and are less reliable but faster. They are often used for real-time applications like online gaming, video streaming. UDP is connection-less protocol.
6. Socket: A socket is an endpoint between two way communications. Sockets provide a standardized interface for applications to bind to specific network addresses and ports, enabling them to send and receive data streams or datagrams.


SENDER END
Network Layer
IP addresses
Port numbers
"Hello!"
Transport layer
Port numbers
"Hello!"
Sender App
"Hello!"
We want to send this data
RECEIVER END
"Hello!"
Receiver app
Port numbers
"Hello!"
Transport layer
Identifies the port number
IP addresses
Port numbers
"Hello!"
Network Layer
Identifies the IP address

TCP (Transmission Control Protocol)

- TCP is a connection-oriented transport layer protocol that provides reliable, ordered, and error-checked delivery of data between devices on a network.
- Characteristics
  - Connection-oriented: TCP establishes a connection between client and server before data can be transferred, ensuring a reliable and ordered exchange.
  - Reliability: Guarantees data delivery in the order it was sent without loss.
  - Error Checking: Detects and retransmits lost or corrupted data to ensure integrity.
  - eg: File transfer, web browsing

TCP (Transmission Control Protocol) Contd...

- The server must be listening(passive open) for connection requests from clients before a connection is established.
- The client may establish a connection by initiating an active open using the three way handshake:
  - SYN
  - SYN-ACK
  - ACK

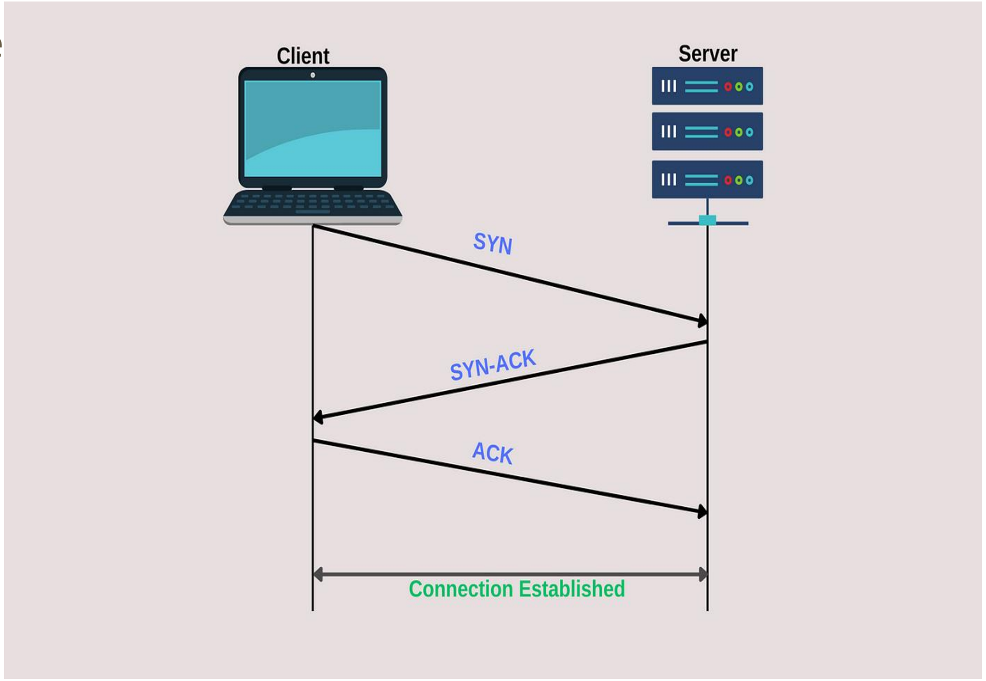


Fig: Transmission Control Protocol

UDP (User Datagram Protocol)

- UDP is a connectionless transport layer protocol that offers a lightweight mechanism for data transfer without the overhead of a connection.
- Characteristics
  - Connectionless: Does not establish a connection before sending data, making it faster but less reliable.
  - Unreliable: Does not guarantee delivery, ordering, or error checking.
  - Low Overhead: Suitable for real-time applications with low latency requirements
  - eg: Video streaming, online gaming

UDP (User Datagram Protocol)

- The target computer is identified and the data packets, called “datagrams”, are sent to it.
- No connection is established between the target computer and source computer.

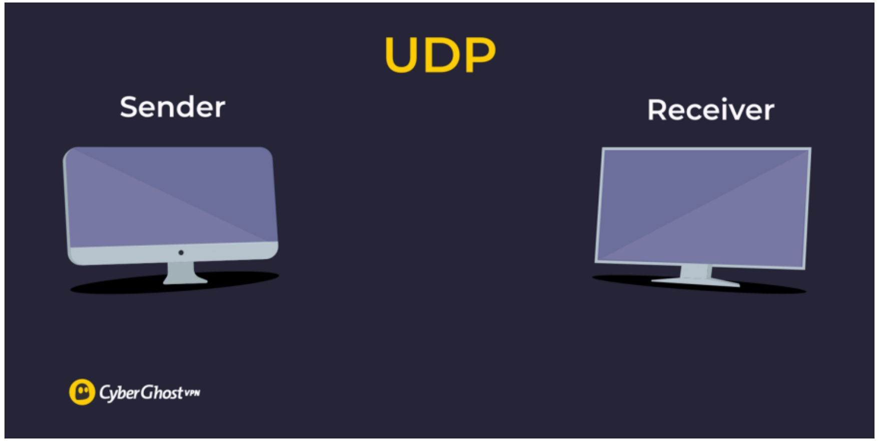
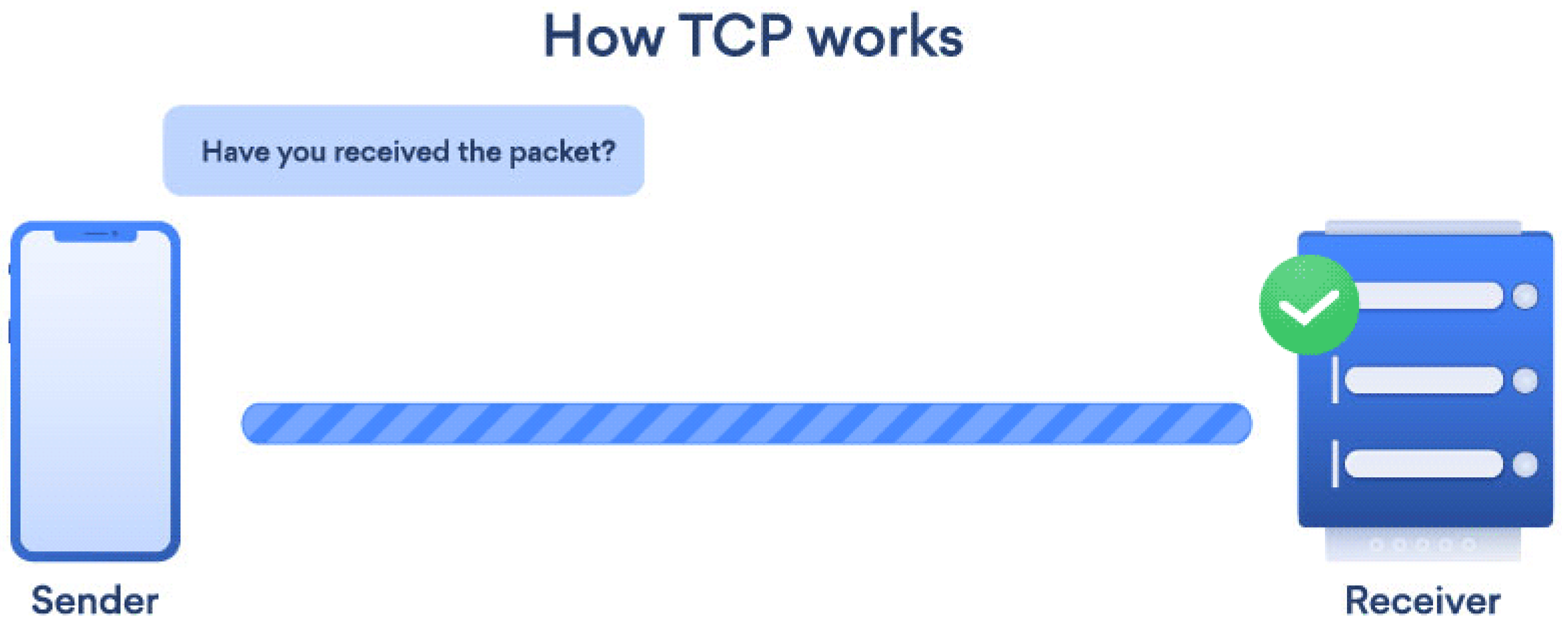
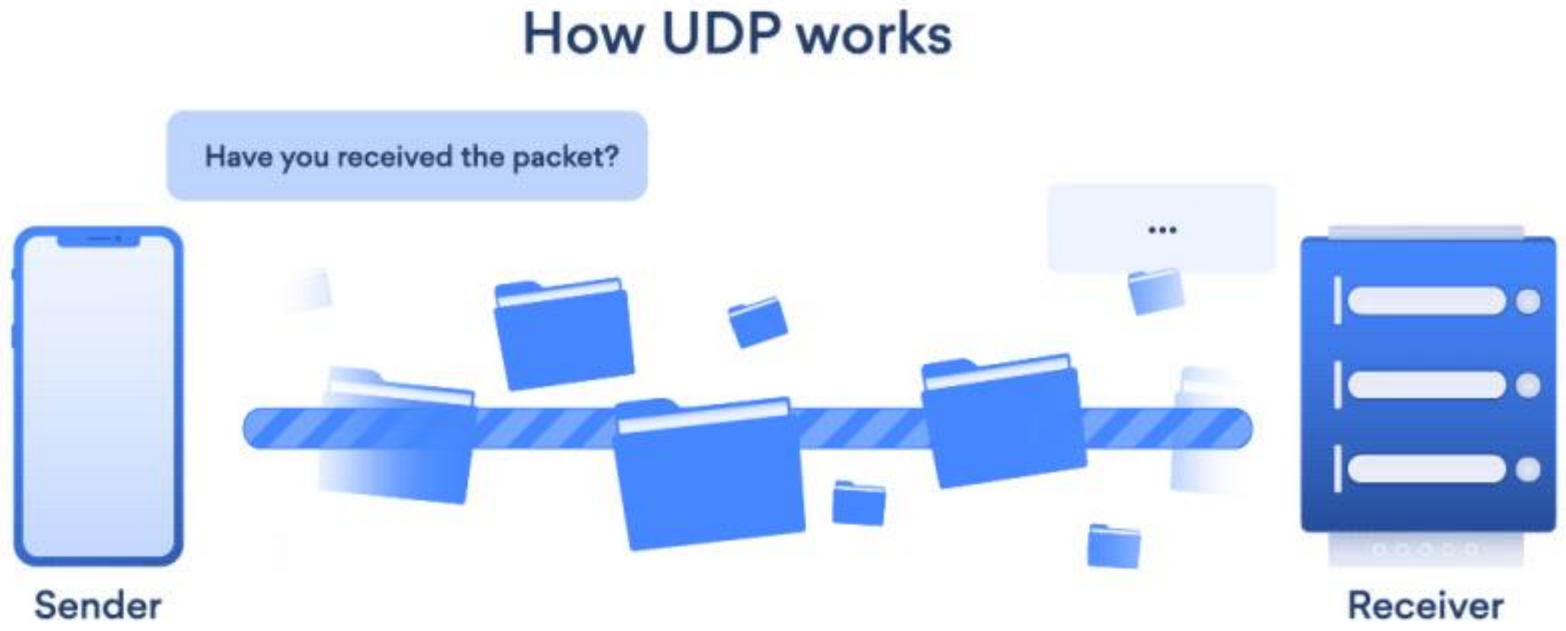


Fig: User Datagram Protocol

TCP vs UDP



TCP vs UDP contd..


How UDP works
Have you received the packet?
...
Sender
Receiver