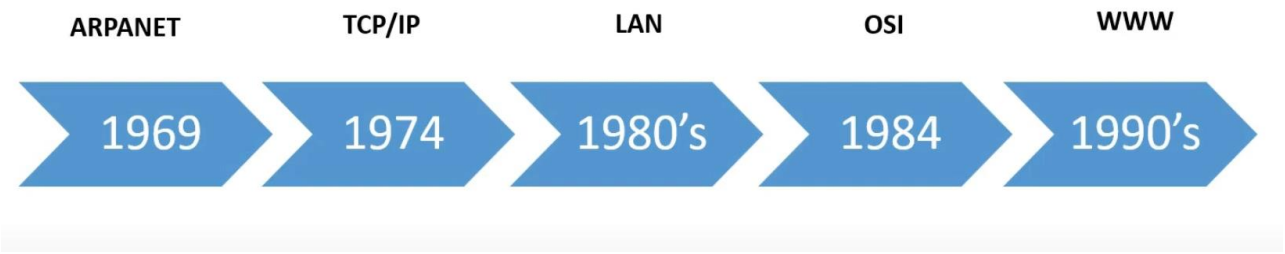
NETWORK DOCUMENT

Melih Yesilyurt

Contents

1. Network Layers and How it Works

In the first years when they started to communicate using the Internet, two computers had to use the same brand/model devices to communicate with each other. For this reason, some standards have been developed for the computers produced to work seamlessly with each other. Two of the most used of these standards are OSI and TCP/IP. OSI was developed by the Open System Connections committee, and TCP/IP was developed by the US Department of Defense. In the OSI model, a 7-layer network system has been proposed for the communication to run smoothly. In TCP/IP, on the other hand, a 4-layer system has been created to solve this communication without any problems.



TCP/IP MODEL
Application Layer
Transport Layer
Internet Layer
Network Access Layer

OSI MODEL
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

1.1. OSI (Open System Interconnect)

1. **Physical Layer**
    - Enables physical devices for transmission and reception of data
    - Establishment of a connection between two nodes
    - Termination of connection between two nodes
    - Defines the medium of transmission of data such as:
        - One-way transmission i.e. Simple
        - Two-way transmission but partially, i.e. half duplex
        - Two-way transmission of data, i.e. full duplex

2. **Data Link Layer**
    - Permits Access to data between devices
    - Controls devices

- Sends data in the form of packets
- Controls error checking
- Allows packet synchronization
- Example of protocols in this layer as ATM, PPP and HDLC

3. **Network Layer**
    - Sends the data to its destination address or node
    - Splits the data into several fragments
    - Delivers each fragment by a separate path
    - Reassembles the fragments
    - Reports delivery errors of each fragment
    - Protocols in this layer are for example IP, ARP, ICMP, RARP and BOOTP

4. **Transport Layer**
    - Manages connections
    - Handles errors in delivery of data
    - Protocols in this layer are for example TCP and UDP

5. **Session Layer**
    - Starts, manages and stop the connection between nodes.
    - Checkpoints procedures
    - Adjourns procedures
    - Terminates procedures
    - Restarts procedures
    - Protocols in this layer are for example SQL, NFS and NetBios

6. **Presentation Layer**
    - Encryption of the data
    - Sends data to application layer
    - Protocols in this layer are for example MIME, MPEG, TLS and SSL

7. **Application Layer**
    - Interacts with software
    - Protocols in this layer are for example HTTP, TFTP, DNS and SMTP

1.2. TCP/IP

1. **Link Layer**
    - The link layer determines the characteristics of the communication medium, the communication speed and the coding scheme.
    - Protocols in this layer are for example ARP and NDP

2. **Internet Layer**
    - This layer provides the sending and routing of data.

- Functions include traffic routing, traffic control, fragmentation, and logical addressing.
- Protocols in this layer are for example IP, ICMP and IGMP

3. **Transport Layer**
    - Functions include message segmentation, acknowledgement, traffic control, session multiplexing, error detection and correction (resends), and message reordering.
    - Protocols in this layer are for example TCP and UDP

4. **Application Layer**
    - There are a number of different functions which are carried out by this layer, including session establishment, maintenance and termination, character code translations, data conversion, compression and encryption, remote access, network management and electronic messaging to name a few.
    - Protocols in this layer are for example MIME, TLS, SSL, FTP, and HTTP

HOW DO LAYERS AND PROTOCOLS WORK?

Below is the path that a web page takes until it arrives on your screen.

• On the server where the web page is located, applications create an HTML format of the page. The data, which is output in HTML format, is sent with the HTTP protocol. After these steps that take place in the application layer. The data is transferred to the transport layer.

• Data related to this layer (such as port information and data size) to the data in the transport layer is added.

• To the data reaching the network layer, the computer (server) to which the data will be sent and your your computer's IP addresses are added.

• At the physical layer, physical addresses and the size of the final data are added.

• The data packet leaves the server and follows the path between the server and your computer. Reaches your computer.

• When the data reaches your computer, this time in reverse order (Physical – Network – Transport – Application) protocols are operated in layers. Finally the data packet sent to your web browser and the process is complete.

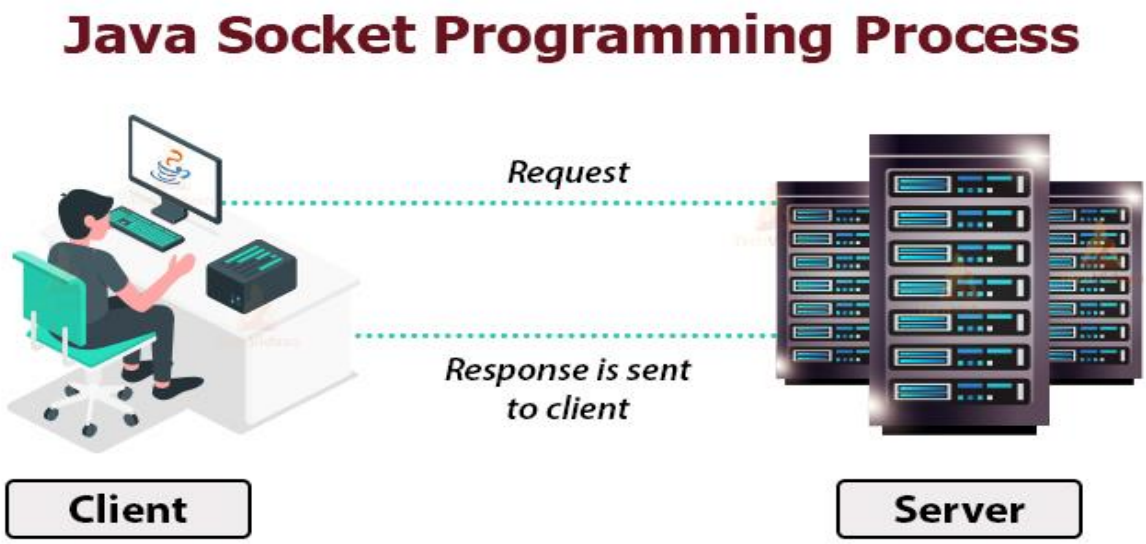
NOTE: Different hardware is used in each layer. For example, at the physical layer Switch, network hardware such as Router is used in the transport layer and NAT is used in the transport layer.

2. Socket Programming

Socket: a door between application process and end-end-transport protocol (UCP or TCP).
Socket API is available for many languages on many platforms:
• C, Java, Perl, Python,…

Client must contact server:
• server process must first be running
• server must have created socket (door) that welcomes client's contact

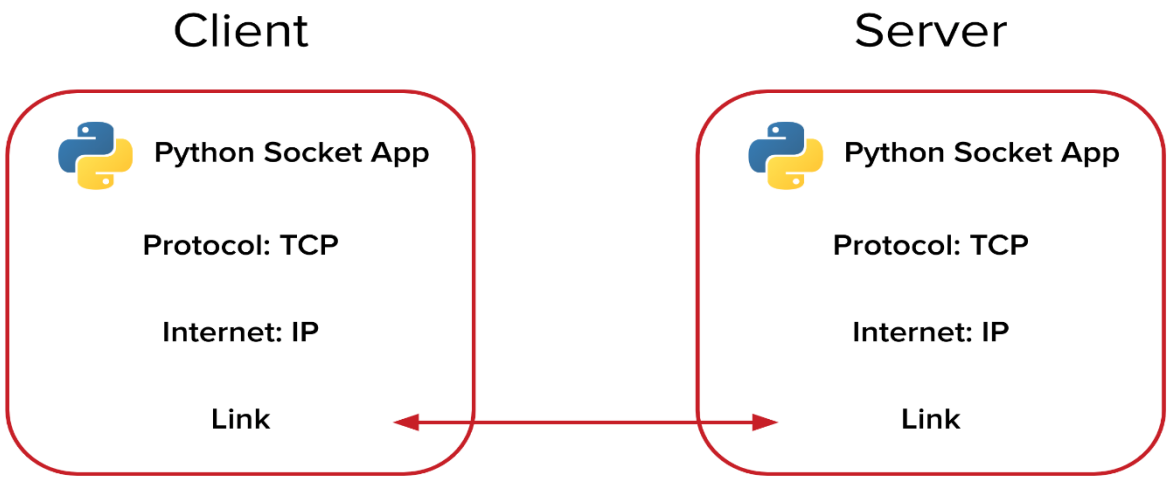
Client contacts server by:
• creating client-local TCP socket
• specifying IP address, port number of server process
• When client creates socket: client TCP establishes connection to server TCP
• When contacted by client, server TCP creates new socket for server process to communicate with client
– Frees up incoming port
– allows server to talk with multiple clients



Sockets are used for interprocess communication. Interprocess communication is based on the client-server model. In this case, client-server are applications that interact with each other. Interaction between client and server requires a connection. Socket programming is responsible for establishing the connection between applications that will interact. In short, it is an adaptation that ensures network communication.

Suppose you have created a platform where users meet online and you will bring instant messaging between users, when a message is sent from one user to another user, there is a change on the server side and the server cannot notify the client. Structures such as polling, long polling or websocket are used to detect this change. Polling periodically makes requests to the server, the server creates and sends a response to each request, and this creates a lot of traffic, which goes beyond the instantaneous concept. Long polling sends the request to the server, but waits for a new request to be made by the server for a response. Websockets, on the other hand, simplify the complex structure of concurrent web applications that are not compatible with the HTTP protocol. Websockets require less bandwidth than polling. With Websocket, you can listen to the communication port between users through the port we created with a permanent connection, and you can instantly handle communication between users in a cheap and fast way.

To give an example from web sockets, I think socket.io would be a good example. Socket.io is a web socket emulator that standardizes all protocols (xhr, long polling, websocket) related to web socket and offers a stable usage. It works on every platform, browser and device. It focuses equally on reliability and speed.



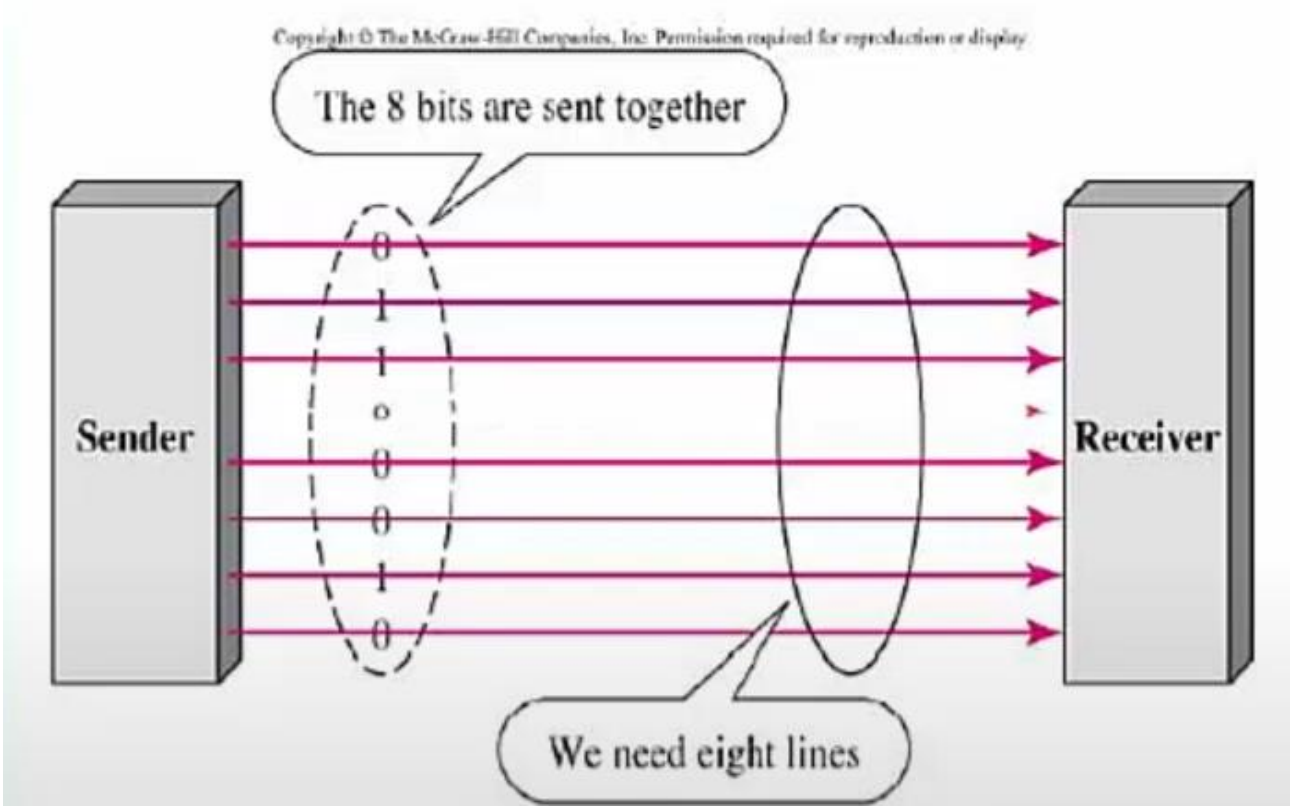
3. How the Data is Transmitted From One Computer/Node To Another?
In the Open System Interconnection(OSI) Layer Model, the Physical Layer is dedicated to data transmission in the network. It mainly decides the direction of data in which the data needs to travel to reach the receiver system or node.

In most communication systems, the transmitting point applies source coding, followed by channel coding, and lastly, line coding. This produces the baseband signal. The presence of filters may perform pulse shaping. Some systems then use modulation to multiplex many baseband signals into a broadband signal. The receiver un-does these transformations in reverse order: demodulation, trellis decoding, error detection and correction, decompression.

Some communication systems omit one or more of these steps, or use techniques that combine several of these steps together. For example, a Morse code transmitter combines source coding, channel coding, and line coding into one step, typically followed by an amplitude modulation step. Barcodes, on the other hand, add a checksum digit during channel coding, then translate each digit into a barcode symbol during line coding, omitting modulation.

Parallel Transmission
If all bits of digitally encoded information are transferred simultaneously, it is called "parallel data transmission". In parallel data transmission, a separate cable connection is provided for each bit of information to be transmitted.

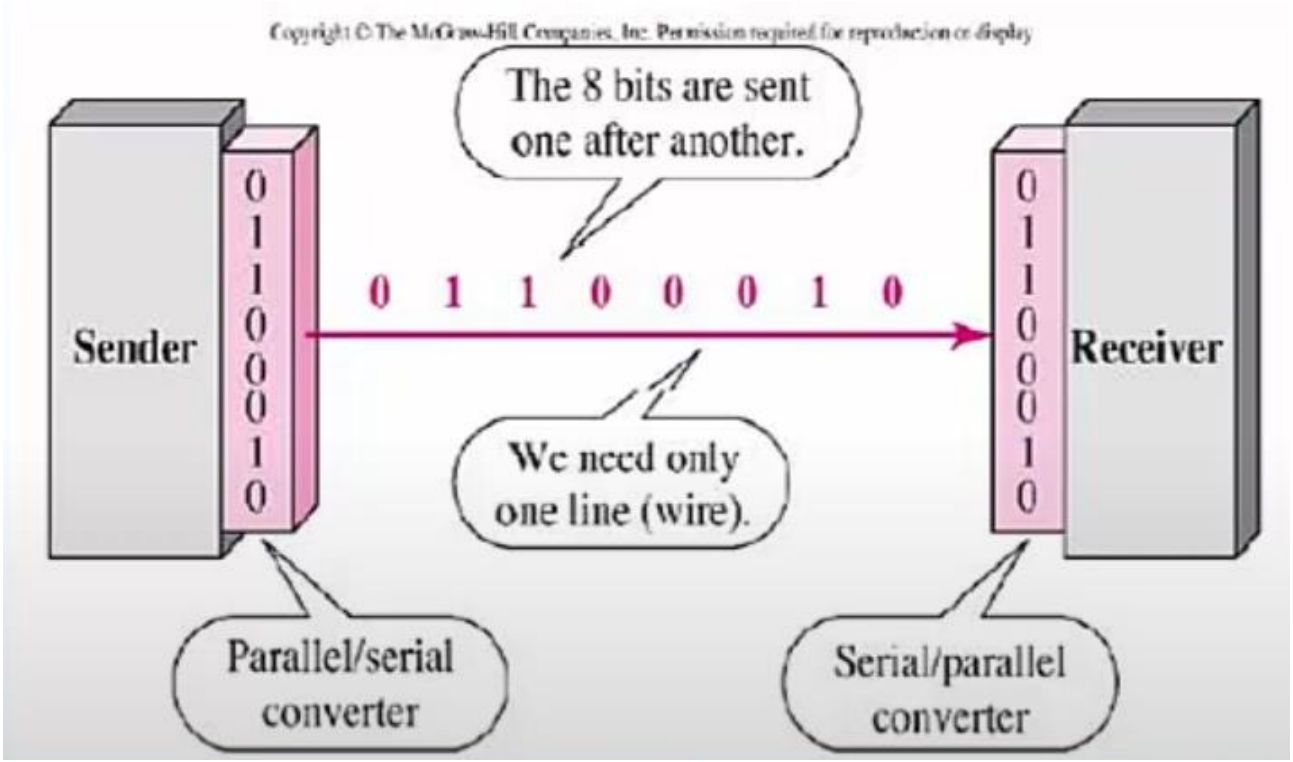
In parallel data transmission, start-end bits are not needed as all bits of a character are transmitted simultaneously. Therefore, its accuracy is higher. Parallel data transmission is very fast due to the simultaneous transmission of all bits of information.



Serial Transmission

It is the transmission of information in n-bit order over a single transmission path. Communication on computer networks is serial communication.

In serial data transmission, only one bit of a character is transmitted at a time. The receiving machine must know the character length, start-end bits and transmission speed for correct communication.



Serial communication is of two types. These are Asynchronous and Synchronous communication.

A. Asynchronous Serial Communication

- It can send data at any time.
- When no data is sent, the line remains idle.
- It is slower than synchronous serial communication.
- Each data group is sent separately.
- The sent data is left on the line one character at a time.
- Bits are added at the beginning and end of the character to detect errors. The start bit (0) is used to start, and the stop bit (1) is used to terminate the data communication.

B. Synchronous Serial Communication

- Synchronous communication means the simultaneous operation of the transmitter and receiver.
- First, the sending party sends a specific character. This is the communication initiation character known to both parties.
- If the receiving party reads this character, communication is established.
- The transmitter sends the information.
- The transfer process continues until the data block is completed or the pairing between the transmitter and receiver is lost.

4. Peer to Peer Network

In its simplest form, a peer-to-peer (P2P) network is created when two or more PCs are connected and share resources without going through a separate server computer. A P2P network can be an temporary connection, a couple of computers connected via a Universal Serial Bus to transfer files. A P2P network also can be a permanent infrastructure that links a half-dozen computers in a small office over copper wires. Or a P2P network can be a network on a much grander scale in which special protocols and applications set up direct relationships among users over the Internet.

Unstructured P2P networks

Unstructured P2P networks do not have a specific node organization. Participants communicate randomly with each other. These systems are considered to be resistant to high I/O movements (eg, often several nodes join and leave the network).

While unstructured P2P networks are easier to build, they require higher CPU and memory usage because search queries are sent to as many peers as possible. This causes the network to be flooded with queries, especially if the queried content is served by a small number of nodes.

Structured P2P networks

Structured P2P networks have an organized architecture, allowing nodes to efficiently search for files even on content that is not widely available. In most cases, hash functions are used to facilitate database searches to achieve this.

Although structured networks are more efficient, they tend to be more centralized and often require higher installation and maintenance costs. In addition, structured networks are more vulnerable to high I/O mobility.

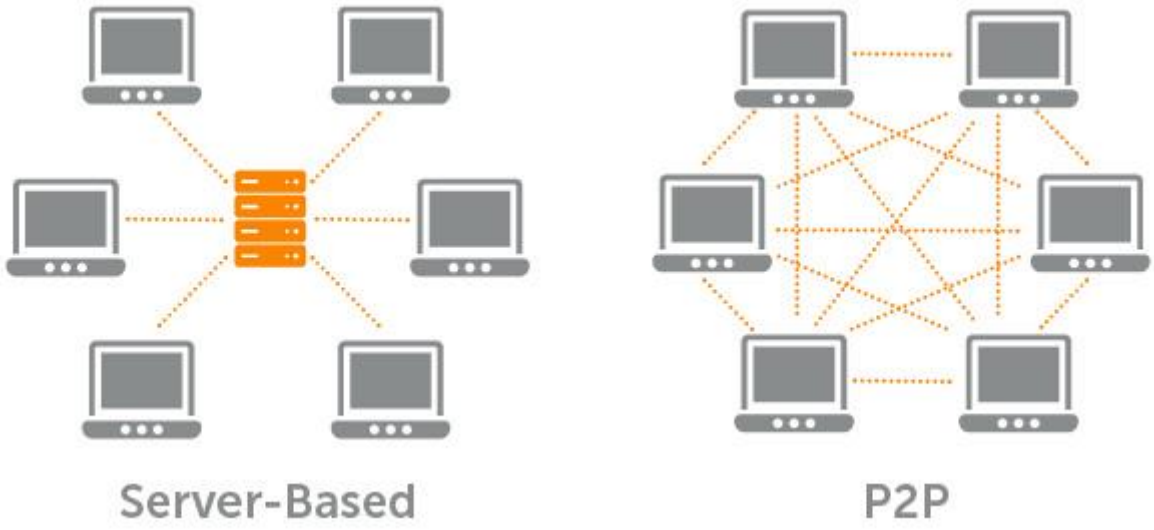
Hybrid P2P networks
Hybrid P2P networks combine the traditional client-server model with some features of peer-to-peer architecture. For example, they may include a central server design that supports peer-to-peer connectivity.

Compared to the other two types, hybrid models have better overall performance. Combining the main advantages of both approaches, they can offer significant efficiency and decentralization at the same time.

Comparison of centralized and decentralized
It is important to note that although the P2P architecture is inherently distributed, there are also levels of decentralization. So not all P2P networks are decentralized.

In fact, most systems need a central authority to direct network movements, which makes them centralized to some degree. For example, some P2P file sharing systems allow users to search and download files from other users, but these users cannot be involved in other processes such as handling search queries.

In addition, it can be said that small networks controlled by a limited number of user bases with common goals have a high degree of centrality even though they do not have a central network infrastructure.



The role of P2P in blockchains
Satoshi Nakamoto defined Bitcoin as “Peer-to-Peer Electronic Cash System” in its early days. Bitcoin was created as a type of digital money. It can be transferred from one user to another via a P2P network managed by a distributed ledger called blockchain.
In this context, it is the P2P architecture inherent in the blockchain that enables Bitcoin and other cryptocurrencies to be transferred worldwide without the need for intermediaries or any central server. Also, anyone who wants to participate in the process of verifying and confirming blocks can operate as a Bitcoin node.
Therefore, there is no bank in the Bitcoin network that records and processes transactions. Instead, the blockchain acts as a digital ledger that publicly records all transactions. Basically, each node keeps a copy of the blockchain and compares it to other nodes to ensure the accuracy of the data. Malicious acts or inaccuracies are quickly rejected by the network.

In the context of cryptocurrency blockchains, nodes can have different roles. For example, full nodes ensure network security by verifying transactions according to the system's consensus rules.
Each full node maintains a complete and updated copy of the blockchain, which can thus contribute to the collaborative effort to verify the true state of the distributed ledger. However, it is important to note that not all fully validator nodes are miners.

5. SDN (Software Defined Networking)
Physically separating the network control plane from the dispatch plane, SDN also allows the control plane to control some devices. SDN, which is an application compatible with high bandwidth and dynamic structure in today's technology, is a cost-effective, dynamic, adaptable and manageable structure. This structure can program the network control directly, as well as separate the control and routing functions. OpenFlow protocol is a requirement to use SDN solutions.

SDN Structure

- Network control can be programmed directly, as routing and control are separated.
- Managers needs can change over time. In line with these needs, SDN offers routing control. As a result, traffic flow is dynamically adjusted across the network.
- Network intelligence is logically centralized in SDN controllers. Applications and policy engines are aggregated into a single logical operation.
- Network administrators automatically configure, manage, optimize and secure network resources using SND programs.
- SND combines network design and operation when implemented through open standards, as instructions are provided by SDN controllers rather than vendor-specific devices and protocols.
- Conventional networks cannot meet the dynamic computing and storage needs of today's data centers. For this, a new networking paradigm, like SDN, is needed.

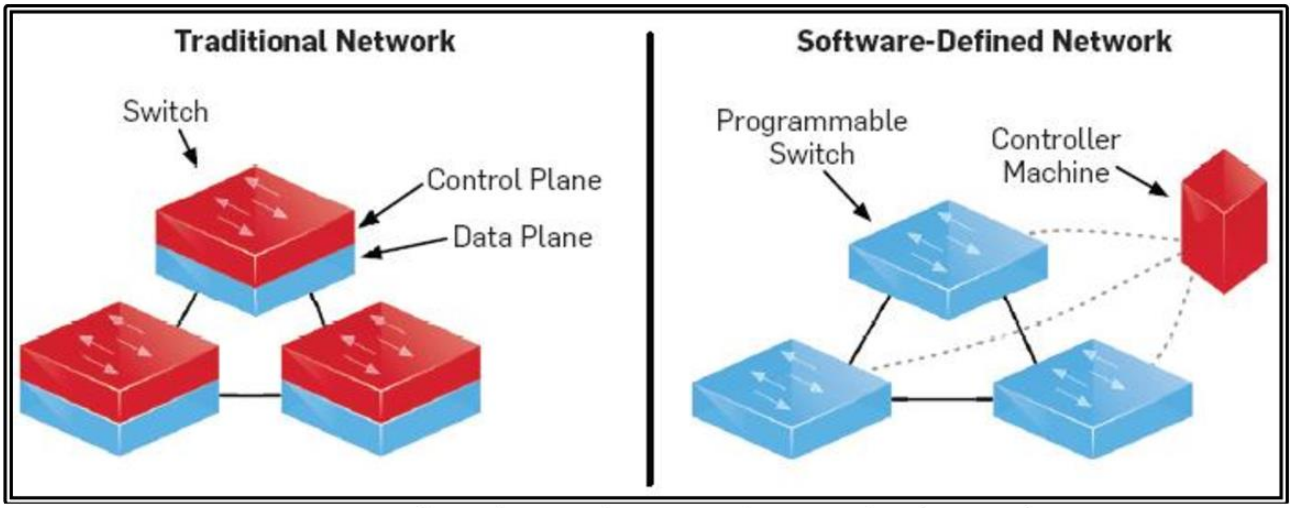


Figure 1. Traditional Network versus Software Defined Network

What are the Benefits of SDN?
SDN, which makes network management very easy, provides the opportunity to change the behavior of devices by controlling them quickly and simply. In order for the devices to communicate with each other, the devices on the network over a common language have a hardware-independent central management. The development of technology day by day has led

to the growth of data centers and, as a result, to the increase of virtualization. In short, cloud architectures have also become a necessity for network management.

Playing a major role in the unification of network and management systems, SDN also shapes the future advancements of networks around itself. Any application that needs processing power, storage or bandwidth to run can be requested by SDN before launching.

Providing the services to be provided over the network via SDN will be as easy as creating a virtual server. With the start of service over virtual networks, the hardware dependency on the manufacturer will be eliminated. This will also reduce investment costs. Since SDN will accelerate network definition and management processes, operational costs will also decrease.

Today, companies operating in the IT sector pay attention to cost, performance, speed and security the most. If the devices on the network are integrated into the SDN structure, things will be much easier. Since there are so many products and services in the market, companies should progress in harmony with each other in network management. It is necessary to handle this progress quickly, without creating security vulnerabilities, by keeping the performance high. With SDN, all of these elements can be provided and possible problems can be prevented.

OpenFlow Protocol

The Openflow protocol, in software-defined networks, allows the network administrator to manage changes in the network's topology and filtering packets in the network. The most important components in this protocol are the controller and OpenFlow network switches. To briefly explain Openflow, it is a method of running a program on another server so that network packets can find the right path.

The OpenFlow protocol is a solution developed to eliminate all the deficiencies in the system. Besides being a software-based network, OpenFlow offers network administrators the ability to manage the tapology and packet filtering areas in the network.

6. References

https://www.researchgate.net/publication/348183262_Computer_Networking_Layers_Based_on_the_OSI_Model

https://sadievrenseker.com/wp/wp-content/uploads/2018/09/fhuy.pdf

https://www.researchgate.net/publication/274639418_TCPIP_PROTOCOL_LAYERING

https://www.pearsonitcertification.com/articles/article.aspx?p=1804869

http://www.cs.cornell.edu/courses/cs519/2004sp/519-fa04-03-sockets-v0.pdf

https://pdf4pro.com/download/462-socket-programming-by-limi-kalita-ijcsit-4d7805.html

https://www.ktu.edu.tr/dosyalar/bilgisayar_d9a2b.pdf

https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html

https://www.networkworld.com/article/3209131/what-sdn-is-and-where-its-going.html

https://www.slideshare.net/KemalYiitzdemir/software-defined-networking-turkish-yazlm-tanml-alar-nokia

https://en.wikipedia.org/wiki/Node-to-node_data_transfer

https://afteracademy.com/blog/what-are-the-data-transmission-modes-in-a-network