CCNA

ICND-1 Notes

Interconnecting Cisco
Networking Devices,
Part 1

Somenath Sinha

May 2018

Contents

Part I

Fundamentals of Networking

Chapter 1

Network Reference Models and Protocols

1.1 Introduction

When talking about the components of a network such as Network devices and protocols, it's useful to have a common frame of reference. This is provided by the OSI and TCP/IP models. These reference models help us understand how a networking device works simply by comparing its functionality to an equivalent working device in the reference model.

1.2 The OSI Model

The **OSI(*Open Systems Interconnect*)** model is used to categorize or classify network components. It consists of 7 layers - each dealing with a specific type of functionality that a networking device must perform to meet its goal. Thus, certain devices operate primarily on one particular layer of the OSI model. For example, switches *live* (i.e., primarily operate on) Layer 2 of the OSI model while routers live on Layer 3.

The layers of the OSI model are stacked with the lower layers at the bottom, with Layer 1 at the base. Each layer interfaces with both the layers above and below it and passes them *data* in the format that they're expecting.

1.2.1 Terminology related to the OSI Model

To understand the different layers of the OSI stack, we first need to understand some basic terminology and devices:

Terms	Description
Ethernet Switch	A device which allows data to flow between multiple computers in a network and only sends the data to the correct device by the use of a physical address.
Physical Address	A **Layer-2** address *burned into* (programmed within) the Network Interface Card (**NIC**) via which the computer is connected to the Network.

Terms	Description
NIC	A **NIC(*Network Interface Card*)** is a physical device which is used to connect to the network and has physical interfaces (ports) for wired mediums.
MAC Address	In a PC, this physical address burned into the NIC is called the **MAC (*Media Access Control*)** address. It's a 48-bit address.
Logical Address	A logical address is an address that's assigned to a device by an external agent (such as a DHCP server) for the primary purpose of routing data within the network and among devices.
IP Address	The IP Address (i.e., the **IPv4 (*Internet Protocol version 4*)** address is a 32-bit logical address used by Layer-3 devices such as routers.
TCP	The **TCP *(Transmission Control Protocol)*** is a framework or set of rules or method to send data from one device to another that's reliable because the receipt of data is acknowledged by the recipient.
UPD	The **UDP *(User Datagram Protocol)*** is a similar protocol to TCP, but doesn't acknowledge the receipt of data and thus data loss in transit is possible, making it less reliable, but faster.

1.2.2 The 7 layers of the OSI Model

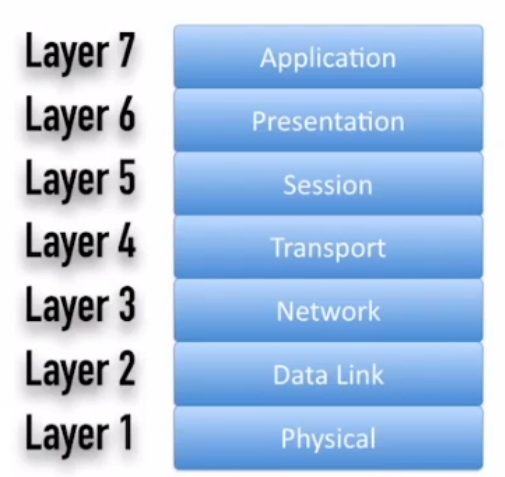

Figure 1.1: OSI Stack

Terms	Description
Physical Layer	This layer is concerned with actually sending the bits (0s & 1s) through the physical media (i.e, wires like Ethernet cable or via wireless mediums). It deals with a way to *electrically* represent them (in case of copper wire based mediums) or *optically* represent them (in case of fiber optics), etc.
Data-Link Layer	Layer-2 devices make forwarding decisions based on a physical address. An Ethernet switch in its basic form is a Layer-2 device. It makes forwarding decisions (i.e, where to send the incoming data) on the basis of a **physical** address, such as the 48-bit MAC Addresses in case of PCs.

Terms	Description
Network Layer	Layer-3 devices such as a router make forwarding decisions based on a **logical** Address such as the IP addresses of the machines in the network. A router analyzes the incoming data on it's ports and determines where to send the data next by finding out which of its other ports lead to the destination IP address.
Transport Layer	The TCP and UDP protocols operate on this layer, and determine how the data is transmitted over the network.
Session Layer	Sets up, maintains and tears down sessions. E.g., **SIP (*Session Initiation Protocol*)** used by IP phones for VoIP calls.
Presentation Layer	Deals with how data is represented on the network. For example, data may be encoded in ASCII (American Standard Code for Information Interchange) or UTF-8(Unicode Transformation Format 8-bit), etc. **Encryption** is also performed within this layer.
Application Layer	The name of this layer is a bit misleading since it is not any application which lives on this layer, but a Network service that allows other desktop applications to take advantage of that service. E.g., The Microsoft Active Directory (AD) service provides the end-user applications with the functionality of logging in to the AD via the network.

Note that it's perfectly possible for a devices to operate in more than one layers, and it's not required to neatly arrange a device or protocol in a single layer. The OSI model is like a book shelf. Similar devices and protocols are arranged in a layer but just like it's possible to have empty shelves in a book-shelf, there's no necessity for a networking device/protocol to have a component present in every single layer.

Since this stack forms the basis of all discussions in the domain of Networking, it's important to memorize the list of the components of the OSI stack and understand what each of the layers in the stack does. If we want to remember the names of the layers from Layer 7 downwards, the acrostic ***A**ll **P**eople **S**eem **T**o **N**eed **D**ata **P**rocessing* can be helpful. To remember from bottom-up, the acrostic is: ***P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way*.

An important point to remember is that the OSI model was **designed** to be *generic* and comprehensive, to act as a reference model for more protocols than just IP. While IP can have certain features that live on certain layers, that's just not universally true for other protocols. Most of our networks run on the **TCP/IP(*Transmission Control Protocol/Internet Protocol*)** stack or the **DoD (*Department of Defence*)** stack.

1.3 TCP/IP Stack

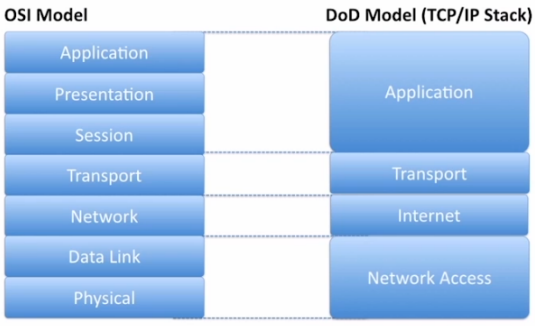

Figure 1.2: TCP/IP Stack

The **TCP/IP(*Transmission Control Protocol/Internet Protocol*)** stack or the **DoD (*Department of Defence*)** model is another reference model that deals directly with the TCP/IP protocols and have a direct mapping to the layers in the OSI stack. This stack was created by the United States Dept. of Defence (DoD) and is thus named after it.

Terms	Description
Network Access Layer	Corresponds to the Physical and Data-link layers of the OSI stack. It's concerned with addressing via physical addresses as well as the representation of the data on physical mediums such as cables. Another name for this layer is the **Network Interface** or the **Link** layer.
Internet Layer	Corresponds to the Network Layer of the OSI stack and deals with logical addressing via the IPv4 and IPv6 protocols.
Transport Layer	This is the equivalent of the Transport Layer of the OSI stack and also deals with the same function of determining the mode of transport, i.e., the protocol to be used for data transmission.
Application Layer	This is the equivalent layer to the combination of the Application, Presentation and Session layers of the OSI stack, and performs all their functions.

Note that some literature that discusses the DoD stack show it as a 5-layer model with the bottom Network Access Layer sub-divided into Data-link and Physical layers:

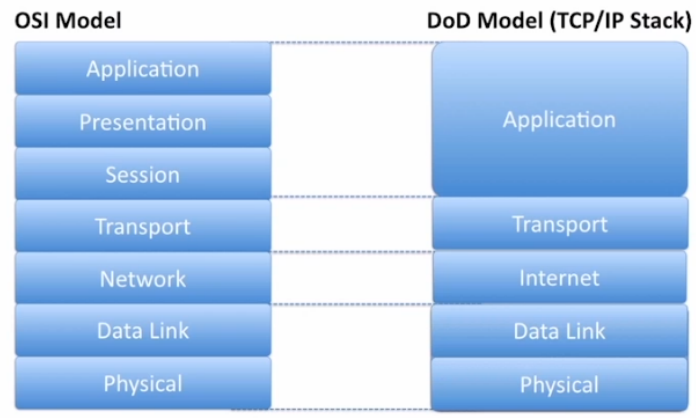

Figure 1.3: TCP/IP Stack Alternate Model

Some may even choose to call this Data-link layer in the TCP/IP stack as the Network Interface Layer. Irrespective of the naming, the functioning of these layers remain consistent.

1.4 TCP/IP Protocol Suite

The TCP/IP protocol suite consists of **IP, ICMP, TCP** and **UDP** protocols. In Layer 3, the Network layer of the OSI stack or the Internet layer of the TCP/IP stack, some of the most useful protocols are IP and ICMP.

IP (*Internet Protocol*) is used to forward *packets* of data to the right (intended) recipient. **ICMP (*Internet Control Message Protocol*)** is used to test the reachability of remote network devices by *pinging* them and can also report error conditions in a network.

1.4.1 Layer-3 (Network/Internet Layer) Protocols

Internet Protocol (IP)

IP is a protocol which contains the data of other higher layer protocols as it's payload. Thus, it's payload consists of segments from UDP or TCP that need to be encapsulated within IP packets to be correctly formatted and sent over the network. Thus, routers, which are layer-3 devices can make forwarding decisions based on the logical address (destination IP address) packed by following the Internet Protocol.

ICMP

A really common use of the **ICMP (*Internet Control Message Protocol*)** is to perform *pings*. The **ping** utility is used to test if one network device is reachable from another network device.

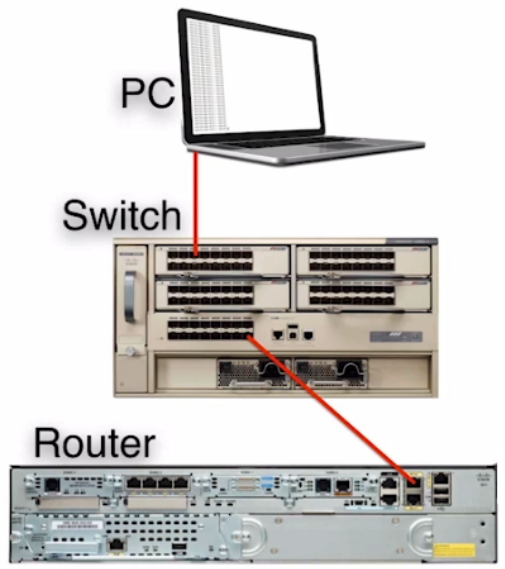

Figure 1.4: Pings using ICMP

Let us consider the scenario above. A PC is connected to a switch which in turn is connected to a router. The switch, a layer-2 device, has two connections - one to the Ethernet port on the PC, which has a MAC address and another to the router's port, which also has it's own MAC address. These MAC addresses (and which port on the switch they're connected to) are known to the switch, which is how it makes forwarding decisions and facilitates the transmission of data.

Now, if the PC can't connect to the internet, during the troubleshooting, one of the first things that we do is to check if we can *reach* the next hop router (the router that connects us to everybody else on the internet/network) from the PC. For this we can use a utility called **ping** that's built into most **OS (*Operating Systems*)**. When ping is used, the PC sends out an **ICMP Echo Request** to that router's IP address. If the router receives that echo request (and if the settings permit it), the router then sends an **ICMP Echo Reply** back to the PC.

The PC can then display that the router is reachable and tell us the *round-trip time*, i.e., the amount of time it took for the packet to reach the router from the PC and then for the ICMP echo reply to acknowledge connectivity to reach the PC.

If our **next-hop router** or our **default gateway**, i.e., the devices that connects us to the rest of the network, has an IP address of `10.10.30.1`, we could ping it from the command line using (The `-c` option specifies the number of *ICMP echo requests* to send.):

```
# ping -c 4 10.10.90.1
PING 10.10.90.1 (10.10.90.1) 56(84) bytes of data.
64 bytes from 10.10.90.1: icmp_seq=1 ttl=64 time=2.06 ms
64 bytes from 10.10.90.1: icmp_seq=2 ttl=64 time=1.31 ms
64 bytes from 10.10.90.1: icmp_seq=3 ttl=64 time=1.42 ms
64 bytes from 10.10.90.1: icmp_seq=4 ttl=64 time=3.37 ms

--- 10.10.90.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.313/2.044/3.377/0.821 ms
```

1.4.2 Layer-4 (Transport Layer) Protocols

User Datagram Protocol (UDP)

UDP is a connection-less and unreliable protocol, because it doesn't receive acknowledgments for the segments that it transmits. While sending a UDP segment, while we hope that it reaches its destination, there's no guarantee that the data will reach the recipient. Further, there's no retransmission for data that failed to reach the destination because we don't know which data was received and which wasn't.

Thus, UDP is used for real-time network applications such as in VoIP phones, where it's not important that every segment of data reach the destination, but the speed at which the segments flow is important. This is applicable for VoIP since we don't care if there's a slight stutter, but delay in communications (such as lag in voice and video, etc.) is unacceptable. Further, the use of TCP in such an application would be an extra overhead that would cause further delays, since a TCP header is much larger than the header on an UDP segment. Another reason TCP isn't used for VoIP is because even if a dropped segment is retransmitted, it may (and probably will) arrive out of order and is thus useless.

Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol that's reliable since it can detect if segments are *dropped*, i.e., lost in transit. In TCP, a **connection** is set up between the two parties involved in the communication and then the data transfer occurs, with acknowledgement for each of the segments received from the recipient to the sender.

A **Three-way handshake** is performed by the two-parties involved to establish a TCP connection. The steps involved are:

- The sender sends a **SYN** (Synchronization message) to indicate that the sender wants to set up a *session*.
- The receiver now needs to send back a **SYN+ACK** message, i.e., acknowledge the `SYN` message from the sender and then send a `SYN` message of its own to the sender.
- The final step in the 3-way handshake is for the sender to reply to the `SYN` message from the recipient.

Now the TCP session is set up. One feature of TCP is that we can send varying amounts of data before expecting an acknowledgement, called TCP windowing.

Sliding Window Protocol & TCP Windowing

TCP's sliding window protocol deals with its ability to vary the amount of data that a sender can send before expecting an acknowledgement. If we're on a highly reliable network, it'd be prudent to send more data at one time before expecting an acknowledgement, because there's less time wasted waiting for acknowledgement. This is called having a *larger window size*. TCP can try to exponentially grow that window size. Let us consider the following case:

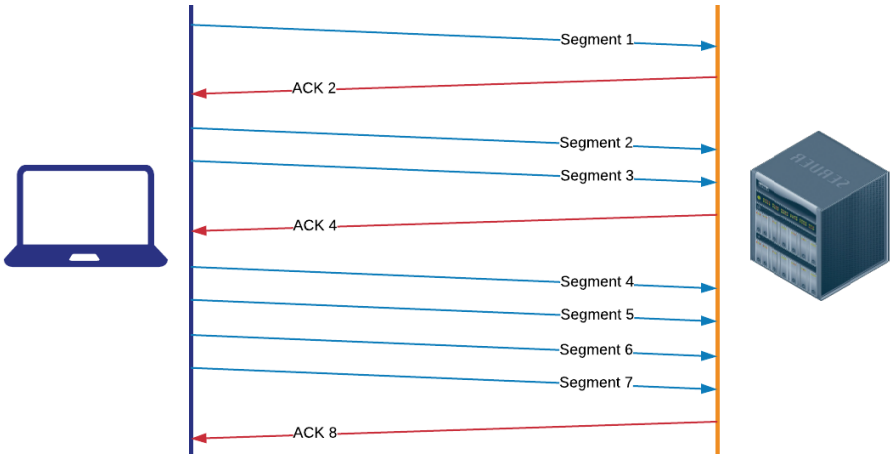

Figure 1.5: TCP's Sliding Window Protocol

Let us consider the laptop on the left is trying to communicate with the server on the right. After the three-way handshake is complete, the laptop sends the first segment of data. The server responds to this with and acknowledgement, and asks for the next segment, with **ACK 2**. The laptop now doubles the number of segments that it sends to 2. The server again responds with an acknowledgement, and demands segment 4 with `ACK 4`. Once more the laptop doubles the number of segments it sends, this time to 4. The server, again responds with `ACK 8`.

This doubling continues till no acknowledgement is received from the server. When that happens, the laptop realizes that it's sending data too aggressively. This might mean that either a packet was dropped, or maybe the sender needs to wait longer for the acknowledgement. Now the sender (i.e., laptop) is going to drop its window size back down and grow at a slower rate, much more cautiously.

1.5 Domain Name System (DNS)

Websites are content hosted on a server accessible by connecting to the IP address of a server (on the port on which the web-server is running). Thus, each public website on the internet has its own public IP. We could connect to the website using its IP, but it's impossible to remember the IPs for such a large number of websites. This is why we have domain names - an organized way to store information about websites. Domain names are hierarchical in nature for ease of indexing and categorization. To connect to a website, we can use its **FQDN (*Fully Qualified Domain Name*)**, a complete domain name specifying each level in that server's DNS hierarchy, instead of its IP. The job of a **DNS (*Domain Name***

System) server is to map each FQDN to an IP address to which our host machine can connect.

Chapter 2

Infrastructure Components

Chapter 3

Network Architecture

Chapter 4

Network Cabling

Chapter 5

Basic Troubleshooting

Chapter 6

IPv4 Addressing

Chapter 7

IPv6 Addressing

Part II

LAN Switching

Chapter 8

Fundamentals of Ethernet

Chapter 9

Basic Cisco Catalyst Switch Configuration

Chapter 10

Virtual LANs (VLANs)

Chapter 11

Trunking

Chapter 12

Troubleshooting Switch Operation

Chapter 13

Basic Switch Security

Chapter 14

Voice VLANs

Part III

IP Routing

Chapter 15

Basic Router Operation

Chapter 16

Basic Router Configuration and Verification

Chapter 17

Routing Fundamentals

Chapter 18

Routing Information Protocol (RIP)

Part IV

Network Services

Chapter 19

Dynamic Host Configuration Protocol (DHCP)

Chapter 20

Network Address Translation (NAT)

Chapter 21

Network Time Protocol (NTP)

Part V

Network Management

Chapter 22

Network Management Protocols

Chapter 23

Device Management

Chapter 24

Troubleshooting with Cisco IOS Tools