LPIC-1 102-500 – Lesson 13

109.1 Fundamentals of internet protocols

The *TCP/IP* Protocol Suite

- The TCP/IP (Transport Control Protocol/Internet Protocol) protocol suite, is a collection of protocols used in the Internet and Networking in general.
- All modern operating systems have their own “TCP/IP stack”.
- The TCP/IP stack consist of 4 (in some literature 5) layers: **Application**, **Transport**, **Internet** and **Link**.
- Most protocols belong to one layer but there are some protocols that cross layers.
- The basic protocols we will examine are: TCP, UDP, IP (IPv4, IPv6), ICMP and others.

TCP/IP model

Layer	Description	Protocols
Application	Provides communication between network applications, session management and data presentation in a human readable form.	HTTP, SMTP, DNS, DHCP
Transport	Provides transport of messages and service separation using **ports**. Provides reliability, error correction flow control and data segmentation.	TCP, UDP
Internet	Responsible for routing data packets in an internetwork. IP addresses are defined here.	IP (IPv4, IPv6), ICMP
Link	It is responsible for converting the data for transport into the physical elements of a network. Not really a part of TCP/IP.	Ethernet, Wi-Fi Token Ring, PPP, SLIP

The format of an IPv4 address

10101100000011110001100000000110 - 172.16.24.6

- An IPv4 address is 32 bit long and represented in 4 octets (bytes) in decimal separated by dots (dotted-decimal notation).
- Every address is separated in two portions:
  - **Network portion**: defines the network where the address belongs.
  - **Host portion**: defines the unique host id that represents the host in an IPv4 network.
- Every address is unique in a network.
- Addresses are assigned statically or dynamically through the DHCP protocol.

The format of an IPv6 address

2001:0db8:0000:dead.0000:0000:0000:beef

2002:db8.0.dead::beef

- An IPv6 address is 128 bit long and represent in 8 hextets (16 bit words) in hexadecimal separated by colon.
- Every address is separated in two portions:
  - **Network portion**: defines the network where the address belongs.
  - **Host portion**: defines the unique host id that represents the host in an IPv6 network.
- Every address is unique in a network.
- Addresses are assigned statically or dynamically through SLAAC (stateless), DHCP (statefull) or a combination of both.

Finding the boundaries of an IPv4 network using subnet masks

- We have the following IPv4 address: **215.25.17.45** with mask **255.255.255.192 (/26).**

```
$ ipcalc 215.25.17.45 255.255.255.192
Address:   215.25.17.45          11010111.00011001.00010001.00 101101
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
=>
Network:   215.25.17.0/26        11010111.00011001.00010001.00 000000
HostMin:   215.25.17.1           11010111.00011001.00010001.00 000001
HostMax:   215.25.17.62          11010111.00011001.00010001.00 111110
Broadcast: 215.25.17.63          11010111.00011001.00010001.00 111111
Hosts/Net: 62                     Class C
```

- The 26 most significant bits of 215.25.17.0 represent the network and the other 6 represent the unique network id.

Subnetting a bigger network

- If we want to segment the **215.25.17.0/255.255.255.0** networks in smaller subnets using the mask **255.255.255.224** we will get 8 subnets.

```
$ ipcalc  215.25.17.0 255.255.255.0 255.255.255.224 | grep -A1 '[1-8]\.$'
 1.
Network:   215.25.17.0/27       11010111.00011001.00010001.000 00000
 2.
Network:   215.25.17.32/27      11010111.00011001.00010001.001 00000
 3.
Network:   215.25.17.64/27      11010111.00011001.00010001.010 00000
 4.
Network:   215.25.17.96/27      11010111.00011001.00010001.011 00000
 5.
Network:   215.25.17.128/27     11010111.00011001.00010001.100 00000
 6.
Network:   215.25.17.160/27     11010111.00011001.00010001.101 00000
 7.
Network:   215.25.17.192/27     11010111.00011001.00010001.110 00000
 8.
Network:   215.25.17.224/27     11010111.00011001.00010001.111 00000
```

Variable length subnet masks

- It is not always efficient to use the same subnet mask for all the networks because we may have different needs in each one. Let's segment the network **215.25.17.0/255.255.255.0** in other subnets with different masks.

```
$ ipcalc  215.25.17.0 255.255.255.192 | egrep -i "(network|broadcast)"
Network:   215.25.17.0/26       11010111.00011001.00010001.00 000000
Broadcast: 215.25.17.63         11010111.00011001.00010001.00 111111

$ ipcalc  215.25.17.64 255.255.255.224 | egrep -i "(network|broadcast)"
Network:   215.25.17.64/27      11010111.00011001.00010001.010 00000
Broadcast: 215.25.17.95         11010111.00011001.00010001.010 11111

$ ipcalc  215.25.17.96 255.255.255.224 | egrep -i "(network|broadcast)"
Network:   215.25.17.96/27      11010111.00011001.00010001.011 00000
Broadcast: 215.25.17.127        11010111.00011001.00010001.011 11111

$ ipcalc  215.25.17.128 255.255.255.248 | egrep -i "(network|broadcast)"
Network:   215.25.17.128/29     11010111.00011001.00010001.10000 000
Broadcast: 215.25.17.135        11010111.00011001.00010001.10000 111
```

Private IP Addresses

- Because of the exhaustion of IPv4 addresses internationally, the private IPv4 address were created. These IP addresses are not routable over the Internet and they are supposed to be used only on internal networks .
- If a computer from a private network wished to access the Internet, from a private network, its private address must be “translates” to a Public IPv4 address using the “Network Address Translation” (NAT) mechanism.
- Every private network that accesses the Internet should have one or more public ipv4 addresses. The private addresses, which are usually more than the public addresses, are overloaded to the public address. That means that public address may represent more than one private address.

Private IP Addresses

Address block	IPv4 Address range	Number of addresses
10.0.0.0/8 (255.0.0.0)	10.0.0.0 – 10.255.255.255	16.777.216
172.16.0.0/12 (255.240.0.0)	172.16.0.0 – 172.31.255.255	1.048.576
192.168.0.0/16 (255.255.0.0)	192.168.0.0 – 192.168.255.255	65.536

The solution: IPv6

- Version 6 of IP (IPv6) was created to counter the problem of IPv4 exhaustion.
- There are many improvements but the most important in the use of 128 bit address which come up to: $2^{128} = 3,4 \times 10^{38}$ addresses!
- The addresses are represented in hexadecimal and separated with "**:**" in 16 bit words, e.g.:
- **2001:0db8:85a3:0000:0000:8a2e:0370:7334** -> **2001:db8:85a3:0:0:8a2e:370:7334** -> **2001:db8:85a3::8a2e:370:7334**
- **0:0:0:0:0:0:0:1 -> ::1 (loopback address) , 0:0:0:0:0:0:0:0 -> :: (any address)**
- **2001:db8:a::/64** (/64 is the network prefix)

Special addresses

- **loopback**: the network **127.0.0.0/8** is used to test the health of the TCP/IP stack. The **127.0.0.1** is set on the **lo** interface and the **localhost** hostname resolves to 127.0.0.1. The 127.0.0.0/8 can not be used for routing neither on the Internet or internal networks.
- **link-local (APIPA)**: Addresses from **169.254.1.0** to **169.254.254.255** are used for the automatic assignment of an IP on a network interface card (e.g. Ethernet) when there is no DHCP in the network. The nodes with link-local IPs can only communicate within the same network segment with other link-local IPs. They cannot reach nodes outside their network because link-local is not supposed to be routable.

TCP/IP protocols

- **IP (Internet Protocol)**: it is the backbone of TCP/IP and used by almost every other protocol.
  - It’s basic task is routing data from one network to the other using IP addresses.
  - It is unreliable i.e. it does not provide error correction, re-transmitions works on the best-effort principle.
  - It does not provide flow control) .
  - It is connectionless.
  - It is implemented at the Internet layer.

TCP/IP protocols

- **TCP (Transport Control Protocol)**: it is the basic protocol for creating connections between applications. It transports data using ports, which are essentially service IDs.
  - It is reliable.
  - It provides flow control.
  - It is connection-oriented.
  - It has a bigger overhead comparing to UDP.
  - It only supports **unicast** i.e. communication between two nodes.
  - Implemented at the transport layer of the TCP/IP model and uses IP for routing.

TCP/IP protocols

- **UDP (User Datagram Protocol)**: it is implemented at the transport layer and just like TCP it uses ports to send data in the form of datagrams.
  - It is unreliable.
  - It provides no flow control .
  - It is connectionless.
  - It is faster than TCP because of the lower overhead.
  - Supports unicast, broadcast and multicast.
  - It is implemented at the transport layer and uses IP for routing.

TCP/IP protocols

- **ICMP (Internet Control Message Protocol)**: it is used for troubleshooting and notifying other protocols about the behavior of the network.
  - Flow control: notifies TCP about network congestion.
  - Notifies other protocol about unreachable destinations (Destination Unreachable).
  - Re-routing of network paths (Route Redirection).
  - Checking remote destination e.g. using the **ping** command.
  - It is connectionless.
  - It is implemented at the Internet layer and uses IP for routing.

Ports and services

- Ports are implemented at the transport layer by **TCP** and **UDP**, for identifying services on a network node.
- To transfer a data packet, the application need to know the IP address of the node and the id of the service (aka port) for which the data is destined to.
- A node (server) can offer many services and these are distinguished by ports.
- Services (aka applications) are implemented at the application layer while ports are implemented at the transport layer.
- Some examples of application protocols of TCP/IP: ftp, ssh, http, dns etc.
- The **/etc/services** file contains a list of well-known services and the ports they use.

Well-known ports and services

Service	Port(s)	Description
FTP (File Transfer Protocol)	20 (data), 21 (control)/TCP	FTP is used for file transfers over the Internet
SSH (Secure Shell)	22/TCP	Secure Remote control protocol
TELNET	23/TCP	Insecure remote control protocol
SMTP, SMTPS (Simple Mail Transfer Protocol)	25, 465 (TLS), 587 (STARTTLS)/TCP	Mail Sending Protocol
DNS (Domain Name Service)	53/TCP-UDP	Resolving hostnames to IPs
DHCP	67 (Server), 68 (Client)/UDP	Automatic IP Address assignment
HTTP/HTTPS (HyperText Transfer Protocol)	80, 443 (TLS)/TCP	World Wide Web
POP3, POP3S (Post Office Protocol)	110, 995 (TLS)/TCP	Receiving mail locally
Netbios	139/TCP-UDP	File/printer sharing for Windows networks
IMAP, IMAPS (Internet Message Access Protocol)	143, 993 (SSL)	Corporate Mail receiving (on a cental server)
SNMP (Simple Netowork Management Protocol)	161, 162/TCP-UDP	Monitoring and Netork Management
LDAP (Lightweight Directory Access Protocol)	389, 636 (TLS)/TCP	Directory Information/Authentication Protocol
Syslog (System Log Protocol)	514/UCP, 6514 (TLS)/TCP	Sending logs over the network

License