* **The OSI Open systems Interconnect Model**

1. Th OSI reference model is a standard of the international Organization for Standardization (ISO).
2. It is a general-purpose framework that characterises and standardises how computers communicate with one another over a network.
3. Its seven-layered approach to data transmission divides the operations into specific related groups of actions at each layer.
4. A layer serves the layer above it and is served by the layer below it.

e.g.

Sender--------------Switch--------------Receiver (E-mail server)

Scenario – sending an email to the email server.

* + - The sender is going to compose the traffic that’s going to be sent to the receiver and as it builds the packet which will be sent, it is going to start from the top level, and then it’s going to work its way down to the bottom level.
    - First off, **it will create the Layer 7 information,** which is the Application layer. So, if sending an email, that will contain information such as from and the to field, etc.
    - Then, **it will encapsulate the layer 7 information in the layer 6 information**. Layer 6 is the Presentation layer.
    - Then, it will encapsulate that in the layer 5, Session layer Information. **Layer 5, 6, 7 are known as the Upper Layers and they’re more important for application developers rather than Network Engineers.**
    - So, we’ve got our Upper layer traffic composed, that then gets encapsulated with our Layer4 header, which is the Transport Layer. **Layer 4, this is going to be either TCP or UDP, and it will include the port number**, such as port 80 if its web traffic, port 25 for our email example.
    - The next thing that happens is that **it will then be encapsulated with our Layer 3 Header**. Layer 3 is the Network layer, and important information that is included the source and destination IP address. A network device that operates on Layer 3 is Routers.
    - The next thing that will happen, as we compose our packet, is the **sender will then encapsulate that information in the Layer 2 Header.** Layer 2 is the Data Link layer, and important information here is the MAC address, if we’re sending this over an ethernet network. The network device at Layer 2 is our switches.
    - Finally, this packet is now ready that we can actually transmit it over the wire. When we do that, we’re down to Layer 1, which is the Physical Layer. A network device, which you don’t really see anymore which operate at Layer 1 was Hubs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Name** | **Includes** | **Devices** |
| 7 | Application |  |  |
| 6 | Presentation |  |  |
| 5 | Session |  |  |
| 4 | Transport | TCP/UDP, Port |  |
| 3 | Network | IP Address | Routers |
| 2 | Datalink | Ethernet MAC Address | Switches |
| 1 | Physical |  |  |

So, we’ve now composed the packet at the sender side. We’ve put it into the wire. It’s going to travel over the network, and it’s going to reach our receiver. The order of operations is going to be reverse now because, obviously, it comes in at the physical Layer. The receiver will process the packet from the bottom layer back up to the top layer now.

**Process on the receiver side:**

* + - The packet comes at the Physical Layer.
    - The receiver will then look at the outside header, which is the Layer 2 header, and it will check that this packet is for it. It’ll check that the destination MAC address in the layer 2 header is its MAC address. If it’s not for it, it will just discard the packet.
    - Next up, it will continue with de-encapsulation now. It will then look at the layer 3 header. It will check the destination IP address, and again, it will check that the packet is for it. If not, it will discard the packet.
    - Then we carry on with our de-encapsulation on the receiver. It will look at the layer 4 header. What this is useful for is, imagine the receiver is not just an email server listening on port 25 for email traffic. Let’s say it’s also a web server as well, listening on port 80 for web traffic. Well how does it know if this traffic is for its email server application or its web server application? It can tell by looking at the port number.
    - Next, we’ll carry on with the de-encapsulation. We’ll strip off the Layer 4 header. We’ll look at the layer 5 at the session layer.
    - **We’ll carry on up. We’ll then look at the layer 6 header and then the layer 7 header.**

**So, that’s the whole thing, that’s how we got the traffic from the sender to the receiver as far as the OSI model was concerned. At the sender side, again, we start off at the top layer. We work our way down to the Physical Layer. It goes across the Physical Layer, it will hit the receiver there, and it will then de-encapsulate going into the opposite direction, going from the bottom layer up to the top layer.**

**ACRONYM FOR REMEMBERING THE OSI LAYERS: PLEASE DO NOT THROW SAUSAGE PIZZA AWAY**

**PLEASE – Physical Layer**

**DO – Data Link Layer**

**NOT – Network Layer**

**THROW – Transport layer**

**SAUSAGE – Session layer**

**PIZZA – presentation Layer**

**AWAY – Application Layer**

**PLEASE DO NOT TEACH STUDENTS POINTLESS ACRONYMS**

**TCP/IP Stack (Transmission Control Protocol/ Internet Protocol)**

TCP/IP was developed during the 1960s by the US department of Defence’s (DoD) Advanced Research Projects Agency (ARPA).

It is a protocol stack which consists of multiple protocols including TCP and IP.

It is the main protocol used in computer operations today.

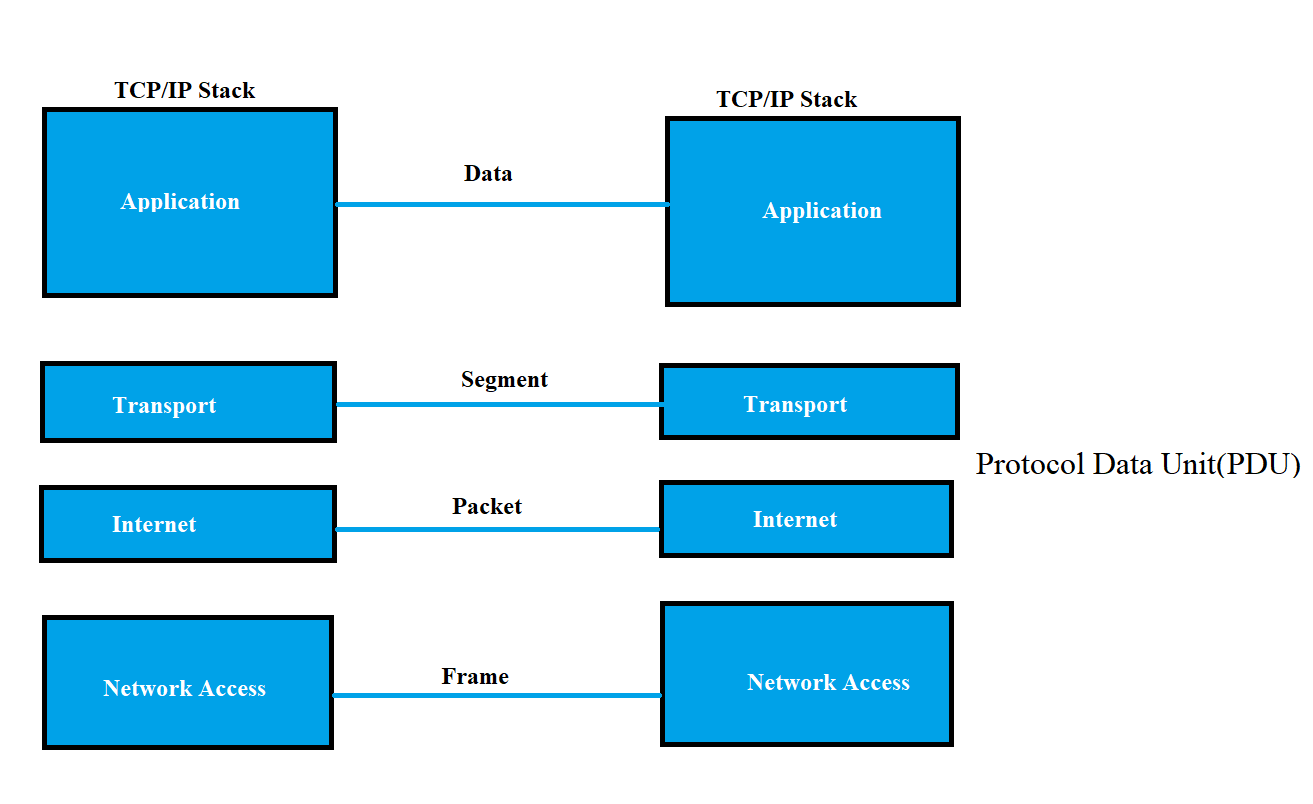
Whereas the OSI reference model is conceptual, the TCP/IP stack is used to transfer data in production networks.

TCP/IP is also layered, but does not use all the OSI layers, though the layers are equivalent in operation and function.

Comparison of OSI model with TCP/IP stack:

|  |  |  |
| --- | --- | --- |
| OSI model | TCP/IP stack | CISCO TCP/IP stack Layer definition |
| Application | Application | Represents data users, encodes, and controls the dialog |
| Presentation |
| Session |
| Transport | Transport | Supports communication between end devices across a diverse network |
| Network | Internet | Provides logical addressing and determines the best path through the network |
| Data Link | Network Access | Controls the hardware devices and media that make up the network. |
| Physical |

**Host communication terminology**

When two hosts talk to each other, they’re going to be exchanging PDUs, Protocol Data Units. The PDU is the entire communication all the way from Layer 7 down to Layer 1 the OSI Stack. We can also divide that into different terms depending on which layer of the TCP/IP suite we’re talking about.

**The Upper OSI Layers**

1. **Layer 7 – The Application layer**
   * The application layer provides network services to the applications of the user.
   * It differs from the other layers in that it does not provide services to any other OSI layer.
   * The Application layer establishes the availability of intended communications partners. Intended communication partners would be the hosts that this host is communicating with.
   * It then synchronizes and establishes agreement on procedures for error recovery and control of data integrity. Data integrity means checking that data has not been altered or corrupted in transit.
2. **Layer 6 – The Presentation Layer**
   * The Presentation layer ensures that the information that is sent at the application layer of one system is readable by the application layer of another system.
   * The presentation layer can translate among multiple data formats using a common format (e.g., computers with different encoding schemes)
3. **Layer 5 – The Session Layer**
   * The session layer establishes, manages, and terminates sessions between two communicating hosts.
   * The session layer also synchronizes dialog between the presentation layers of the two hosts and manages their data exchange.
   * For example, web servers have many users, so there are many communication processes open at any given time to track.
   * It also offers efficient data transfer, CoS(Class of Service), and exception reporting of upper layer problems.

**The Lower OSI Layers**

1. **Layer 4 – The Transport Layer**
   * The main characteristics of the Transport Layer are whether TCP or UDP transport is used, and the port number.
   * Definition:
     1. The Transport layer defines services to segment, transfer, and reassemble the data for individual communications between the end devices.
     2. It breaks down large files into smaller segments that are less likely to incur transmission problems.
   * If we want the communication for two hosts to be reliable, then we will use TCP.
   * If speed is more important than reliability, like for voice or video of traffic, then we’ll use UDP instead.
2. **Layer 3 - The Network Layer**
   * The most important information at the Network layer is the source and destination IP address.
   * Routers operate at Layer 3.
   * Definition:
     1. The Network layer provides connectivity and path selection between two host systems that may be located on geographically separated networks.
     2. The network layer is the layer that manages the connectivity of hosts by providing logical addressing( IP addressing is our logical addressing).
3. **Layer 2 – The Data Link Layer**
   * The most important information at the Data-Link Layer is the source and destination layer 2 address (other information also included in layer 2 header)
   * For example, the source and destination MAC address if Ethernet is the Layer 2 technology. Different Layer 2 technologies use different formats for their addressing. For example, old legacy Frame relay uses DLCI or DLCI numbers for the addressing. With ethernet, which is what is always used in our local Area networks, it’s the MAC address that is used here. Switches operate at Layer 2. Our Switches are Layer 2 aware devices.
   * Definition:
     1. The data link layer defines how data is formatted for transmission and how access to physical media is controlled.
     2. It also typically includes error detection and correction to ensure a reliable delivery of the data.
4. **Layer 1 – The Physical Layer**
   * The Physical layer concerns literally the physical components of the network, for example the cables being used.
   * Definition:
     1. The Physical link enables bit transmission between end devices.
     2. It defines specifications needed for activating, maintaining, and deactivating the physical link between end devices.
     3. For example, voltage levels, physical data rates, maximum transmission distances, physical connectors, etc.

**The IOS Operating System: Introduction**

* Two ways to configure the devices:
  1. Using GUI
  2. Using CLI
* **A short history of CISCO Operating Systems:**
  1. Most people think of cisco as Primarily a routing and switching company, but they actually started out with just routers in 1984
  2. IOS is the Operating system that has been used on cisco routers since the inception.
  3. CISCO catalyst switches evolved from the acquisition of Crescendo in 1993.
  4. The original CISCO switch operating system was CatOS, which has not been deprecated.
  5. CISCO firewalls evolved from the acquisition of Network Translation’s PIX firewall with Finesse Operating system in 1995
  6. CISCO switches and firewalls were ported over to the IOS operating system over the following years.
  7. IOS remains as the operating system used on the majority of CISCO enterprise grade network devices.
  8. Other operating systems have been developed for some more recent router and switch platforms as mentioned below:
     + The CISCO Nexus and MDS data centre switch product lines run on NX-OS.
     + The IOS-XR operating system runs on the service provider NCS, CRS, ASR9000 and XR12000 series routers.
     + IOS-XE runs on the ASR1000 series service provider routers.
     + The command line Interfaces for the other operating systems are nearly identical to IOS.
  9. **Why is there the different Operating systems?**
     + IOS has got a monolithic kernel, meaning that if one process running on the router crashes, it can crash the entire router.
     + The other newer Operating systems have got microkernels, and the processes run in separate, protected memory address space. So, if one of the process crashes, it shouldn’t affect the running on the rest of the system.
     + These other Operating systems are mainly on the higher-end routers and switches, but on the enterprise-grade routers and switches, they’re still running IOS.

**Connecting to a CISCO Device over the Network**

**Objective:**  To connect to CLI on CISCO routers and switches over the corporate network.

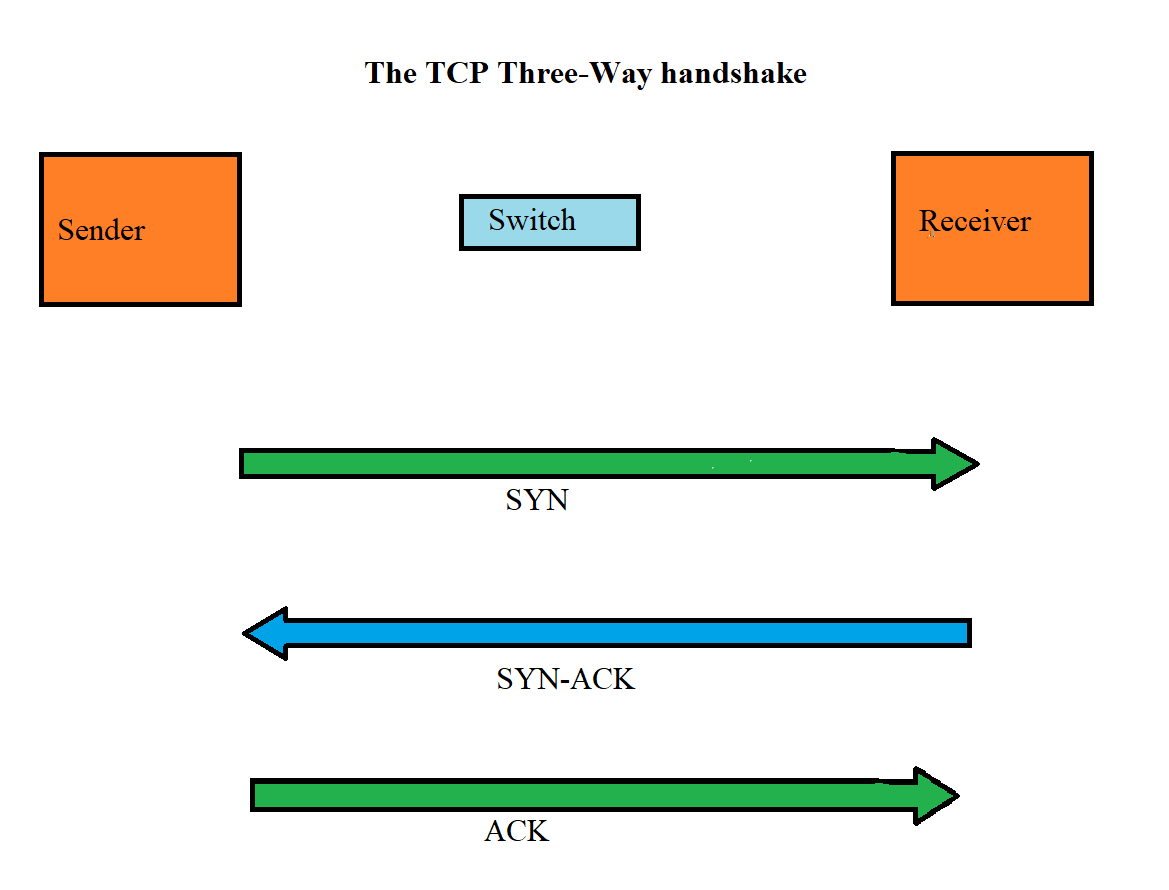
* To get the command line interface for day-to-day management of a Cisco device, you will use Secure Shell (SSH) to connect to its management IP address over the network.
* Telnet is also supported but not recommended because it is insecure.
* In enterprise networks, secure login will typically be enforced through integration with a centralized AAA (Authentication, Authorization and Accounting) server.
* **AAA server - It’s got access to the central database of all your usernames and passwords, so you’ll be having them on each separate device.**

**Navigating the CISCO IOS Operating System**

* The IOS operating system is stored in Flash.
* Then startup configuration is stored in NVRAM.
* The running configuration is stored in RAM. (Loaded into RAM from the Startup config when the device boots up

**Layer 4 – Transport Layer**

* The transport layer provides transparent transfer of data between hosts and is responsible for end-to-end error recovery and flow control.
* Flow control is the process of adjusting the flow of data from the sender to ensure that the receiving host can handle all of it. **If the sender is sending too quickly, maybe because we’ve got faster network connections on that side, and it’s sending more than the receiving host can accept, then if flow control is enabled, the receiving host will have a mechanism to signal back to the sender, telling it to slow down.**
* Another thing that is supported at layer 4 is session multiplexing. Session multiplexing is the process by which a host is able to support multiple sessions simultaneously and manage the individual traffic streams over a single link.
* It’s the Layer 4, The Transport Layer, that is responsible for tracking and keeping control of the different sessions on a host.
* **Layer 4 Port Numbers:**
  + The Layer 4 destination port number is used to identify the upper layer protocol.
  + for example, HTTP used port 80, SMTP email uses port 25.
  + The sender also adds a source port number to the layer 4 header.
  + The combination of source and destination port number can be used to track sessions.
* **TCP Protocol**
  + TCP (Transport Control Protocol) and UDP (the User Datagram Protocol) are the most common layer 4 protocols.
  + TCP is connection oriented – once a connection is established, data can be sent bi-directionally over that connection.
  + TCP carries out sequencing to ensure segments are processed in the correct order and none are missing.
  + TCP is reliable – the receiving host sends acknowledgements back to the sender. Lost segments are resent.
  + TCP performs flow control. (So, if the sender is sending a rate too high and the receiver cannot handle it, the receiver can signal back to the sender, telling it to slow down. So, TCP, it’s connection-oriented, reliable protocol. The way that, that connection is set up between the two hosts is it used the **TCP three-way handshake**.

****

Here we have got sender in the left. It’s going to initiate the connection. It sends a SYN, a synchronized message, over to the receiver on the right. When the receiver receives that, it will send a SYN-ACK back, so a synchronized acknowledgement, and then finally, to complete the connection, the sender will send an acknowledgement, ACK. We now have the connection set up between the two hosts., and they can send traffic over it.

* **Make Up of a TCP Header**

Quick reminder about how a packet is composed:

|  |  |  |  |
| --- | --- | --- | --- |
| **Layer** | **Name** | **Includes** | **Devices** |
| 7 | Application |  |  |
| 6 | Presentation |  |  |
| 5 | Session |  |  |
| 4 | Transport | TCP/UDP, Port |  |
| 3 | Network | IP Address | Routers |
| 2 | Datalink | Ethernet MAC Address | Switches |
| 1 | Physical |  |  |

Now, we will understand how Layer-4 header is composed, what comprises it:

So, we’ve got the source port and destination port numbers, as we spoke about just there earlier.

1. We then have a sequence number and the acknowledgement number.
2. We have a header length, a reserved field, which is for any reserved information later.
3. Code bits, window, which can be used for flow control.
4. A checksum, which can be used to check and see if the traffic got corrupted in transit.
5. We’ve got an optional urgent part of the header there as well.
6. We can add other options.
7. And then we’ve got the Data.

So, there’s a lot that goes in the TCP Header.

|  |  |  |  |
| --- | --- | --- | --- |
| Bit 0 |  | Bit 15 | Bit 16 Bit 31 |
| Source Port (16) | | | Destination Port (16) |
| Sequence Number (32) | | | |
| Acknowledgement Number (32) | | | |
| Header length (4) | Reserved (6) | Code bits (6) | Window (16) |
| Checksum (16) | | | Urgent (16) |
| Options (0 or 32 If any) | | | |
| Data (Varies) | | | |

* **UDP – User Datagram Protocol**
  + The User Datagram Protocol sends Traffic best effort. Meaning, we don’t have the connection, we don’t have reliability. The sender just makes up the packet, sends it over to the receiver, and hopes that it’s going to get there.
  + UDP is not connection oriented. There is no handshake connection setup between the hosts.
  + UD does not carry out sequencing to ensure segments are processed in the correct order and none are missing.
  + UDP is not reliable – the receiving host does not send acknowledgements back to the sender.
  + UDP does not perform flow control.
  + If error detection and recovery is required, it is up to the upper layers to provide it. It’s not going to be provided by UDP.
  + **UDP Header:**

|  |  |
| --- | --- |
| 0 16 31 | |
| Source port | Destination port |
| Length | UDP checksum |
| Data | |

There’s much less Overhead with UDP, which leads us to where TCP or UDP would be used. Now, this is up to the designer of the application. Whenever a designer designs an application, they can choose whether it’s going to use TCP or UDP for its transport. They will typically use TCP fir traffic which requires reliability. But real time applications such as voice and video can’t afford that extra overhead of TCP, so they would use UDP. Voice and Video, it’s very sensitive to delay.

You’ve probably watched TV before. You’ve seen a news report where the newscaster is doing it over a satellite phone, and you can see it’s very laggy because satellites are famously high latency connections. So, voice and video, it’s very sensitive to latency. We don’t want to use TCP for real-time traffic like that because the extra overhead is going to slow it down, and it’s going to affect the quality. So, for real-time traffic that’s sensitive to delay, we’ll usually use UDP. For other applications, we’ll use TCP. And because there’s a lot more other application than voice and video, TCP is the most used Layer 4 Transport.

There are some applications that can use both TCP and UDP as well.

**So here, we’re going to look at some of the common applications and their destination ports.**

* **TCP**

**FTP (21)**

**SSH (22)**

**Telnet (23)**

**HTTP (80)**

**HTTPS (443)**

* **UDP**

**TFTP (69) – Trivial File Transfer Protocol**

**SNMP (161) – Simple Network Management Protocol**

* **TCP and UDP**

**DNS (53)**

**Layer 3 – Network Layer**

We are going to cover the format of IP Header and the format of IP addresses.

We will learn about the different types of traffic we can have on our network that’s Unicast, Broadcast, and we’ll also learn about the subnet mask and about how it defines the boundary between our different subnets.

* The network layer is responsible for routing packets to their destination and for Quality of Service.
* IP (Internet Protocol) is the best know Layer 3 protocol. IPv4 is the focus of this section.
* It is a connectionless protocol with no acknowledgement at layer 3.
* Other Layer 3 protocols include ICMP (Internet Control Message Protocol) and IPsec (for secure encrypted communications).

**IP Addressing:**

* IP addressing is a logical addressing scheme which is implemented at Layer 3.
* The network designer uses IP addressing to partition the overall network into smaller subnetworks, or commonly called subnets.
* This improves performance and security and makes troubleshooting easier.
* Layer 2 MAC addresses use one big flat addressing scheme. There is no logical separation between networks at Layer 2, it’s done at layer 3 with our IP addressing.
* **IP Header:** The explanation is given below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <-32 bits-> | | | | |
| 4-bit version | 4-bit hdr length | Type of service | 16-bit total length( in bytes) | |
| 16 bit identification( ID) | | | 3-bit flags | 13-bit fragment offset |
| 8-bit time to live (TTL) | | 8-bit protocol | 16-bit header checksum | |
| 32-bit source IP address | | | | |
| 32-bit destination IP address | | | | |
| Header options, if any(0-40 bytes) | | | | |
| Data (variable length) | | | | |

* **1st row:**
  + First part is 4-bit version, its either going to be IPv4 or IPv6 that’s referenced in that field.
  + We then have the 4-bit header length, the length of the IP header. It can be a different length because the header options that you see further down can be variable length.
  + We then have the type of service byte. This is used for Quality-of-service information. So, we can put a marking on the packet to specify what kind of Traffic this is, and on our routers later, we can take an action based on that marking to give it better service if we need to, for example, for our Voice over IP (VOIP) traffic.
  + We then have the 16-bit total length of the packet.
* **2nd row:** This row is used for fragment information. With our different media types, for example, Ethernet, there is a maximum transmission unit size, the maximum size of packet. Maximum size by default in Ethernet is a 1500 bytes MTU. If we try a packet onto a wire that is longer than that size, it has to get split up into smaller parts which are called as fragments.
  + The second row of information on the IP header is used to help keep track of those fragments.
* **3rd row:**
  + **8- bit time to live field –** Every time a packet goes through a router, the router will decrement the TTL field by one. If it gets down to zero, then that router will drop the packet. What this is for is to prevent the routing loops. We might have an error in our network somewhere that is causing packets to endlessly loop around the network without ever getting to their destination. We don’t want packets to loop around forever, so TTL will prevent that and drop them. Now, it doesn’t fix the underlying problem. We still need to cure the loop. So that traffic, once they get there, that stops us from having a huge amount of traffic getting build up on our network, slowing it down, which is just looping.
  + The next field, the 8-bit protocol, that will specify the Layer 4 information type, typically, TCP or UDP.
  + We then have a checksum which is used to check that packet has not been corrupted in transit.
* **4th row and 5th row:**
  + Next, we have the source IP address specifying where the packet came from and then the destination IP address specifying where the packet is going to.The next field is the header options, where we can put
* **6th row:**
  + The next field is the header option, where we can put in additional information. It’s not commonly used.
* **7th row:**
  + Finally, we have the data, the rest of the packet.

**IP Traffic types:**

* There are three main IP traffic types: Unicast, broadcast and multicast.
* Unicast Traffic is to a single destination host.
* Broadcast traffic is to all hosts on the subnet.
* Multicast traffic is to multiple interested hosts.
* **Unicast traffic:**

**A diagram of a computer network

Description automatically generated**

* + We have got the sender on the left, and it sends some particular traffic to the PC, up at the top of the picture there, and it also sends some other traffic to the PC up at the top right.
* **Broadcast Traffic:**

**A diagram of a computer network

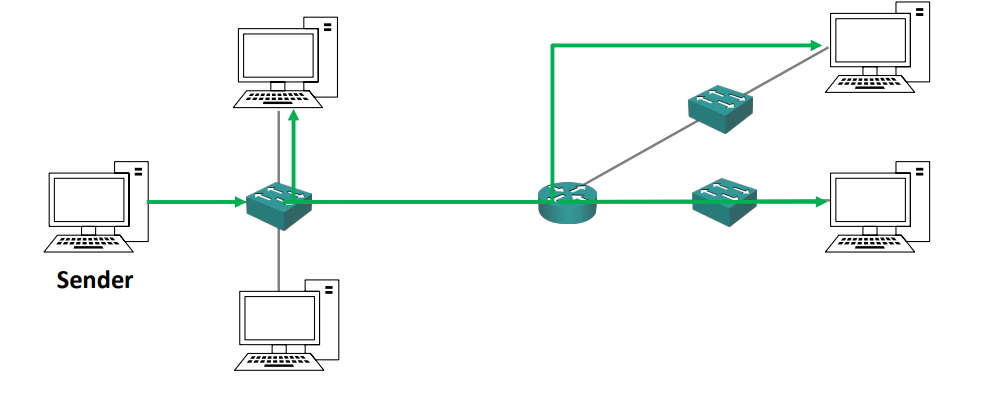
Description automatically generated**

* In Broadcast traffic, the sender sends one copy of the traffic that will come into the switch, and then it gets flooded out to all other hosts on that particular part of the network. So, see here, it goes to all of the different PCs that are attached to the switch.
* There is only one copy of the traffic that gets sent everywhere. You’ll see it also hits the router in the middle there, which has got an interface which is on that network.
* What happens then is the router will drop the traffic, so routers do not forward broadcast traffic on. We can see the other two interfaces on the router, we don’t forward the broadcast traffic out the other side of the router.
* This is a good thing because if you think about it, like with the internet, imagine if you sent a broadcast traffic and it got sent to everybody on the internet. Well, there’s a lot of broadcast traffic, and that would really badly affect performance, and also, we could see the security concerns there as well.
* So, in Broadcast traffic, one copy gets flooded everywhere throughout that particular part of the network.
* **Before going to Multicast traffic, let’s understand how unicast traffic works to multiple hosts first.**
  + This will help us in understanding why we have multicast traffic.

**A diagram of a computer network

Description automatically generated**

* + Let’s say, we have sender over on the left. It’s going to stream out some video traffic, and let’s say the bandwidth of the video stream is 1 Megabyte. So, it sends it to the PC at the top, it also sends it to the PC up at the top right, and it’s sending separate copies of the same traffic to each of those individual hosts. It also sends it to the PC over in the middle on the right as well.
  + So, we’re sending a video stream, it’s 1 Megabyte in bandwidth. We’re sending three separate copies of it, so that would take 3 Megabytes worth of bandwidth from the host over on the left, also throughout that part of the network as well.
  + **With Multicast, the sender sends one copy of that same video stream, and that one copy gets sent out to all the interested receivers.**
  + So now, rather than sending three separate copies taking up 3 Megabytes in our example, we send one copy. It only takes up 1 Megabyte, so we’ve got good bandwidth savings there, but it goes to everybody that’s interested.

****

* + **Example of Multicast –** Tuning into a radio station. So, all of the different receivers, they’ve tuned in that particular station, they’re going to get that particular traffic.

**Converting from Decimal to Binary:**

IP addresses are written down in a decimal format. But to understand how they control the logical separation between networks, it really helps if you think about then the same way that a computer does, and that is in Binary.

Conversion from decimal to binary is going to help understand IP addressing as we work through the rest of this section.

**IPv4 Addresses:**

* An IPv4 address is 32 bits long.
* It is written as 4 ‘octets’ in dotted decimal format.
* For example: 192.168.10.15
* Each octet is 8 bits long (4\*8 = 32).

**Static vs Automatic Addressing:**

* The IP address is usually set manually n servers, printers, and network devices such as routers and switches. It is usually assigned automatically through the Dynamic Host configuration protocol (DHCP) on desktop computers.
* To understand how the logical separation between subnets works, you need to understand the IP address in binary.

**Calculating and IPv4 address in Binary:**

* Each octet in the IP address has a value ranging from 0 to 255.

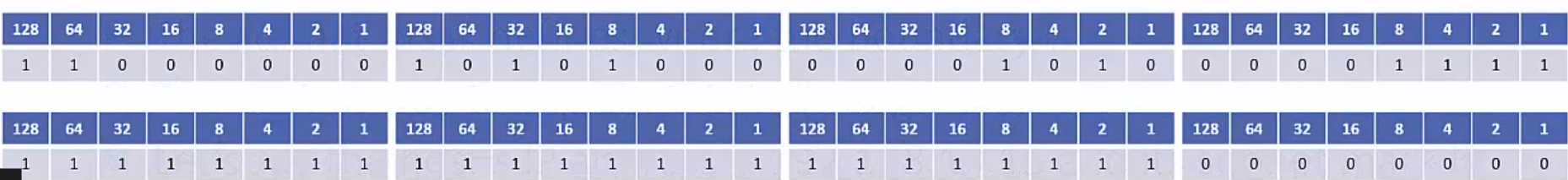
**A screenshot of a number grid

Description automatically generated**

* On each of the octets, the bit pattern in here on each of those different bits can be any combination that you want of 1s and 0s.
* **192.168.10.15 in binary is: 11000000.10101000.00001010.00001111**

**Subnet masks:**

* To set the boundary between logical networks (subnets), the IP address is combined with a subnet mask.
* A host can send traffic directly to another host on the same subnet via switches.
* For a host to send traffic to another host in a different subnet, it must be forwarded by a router. **A router is a device that links our different subnets together and route the traffic between them.**
* The host therefore needs to understand if the destination is on the same or a different subnet in order to know how to send it.
* The subnet mask is used for this.
* The subnet mask is also 32 bits long and can be written in dotted decimal or slash notation.
* **A host’s IP address s divided into a network portion and a host portion, and it’s the subnet mask that defines where the boundary is between the network part and the host part of the address.**
* **Explanation by an example:**
  + Let’s say the host’s IP address is 192.168.10.15, and the subnet mask is 255.255.255.0.
  + When we convert the IP address and subnet mask to binary we get as following:



* The IP address is compared (‘masked’) with the subnet mask.
* A ‘1’ in the subnet mask indicated that bit in the IP address is part of the network.
* A ‘0’ indicated the bit is part of the host address.
* In our example, the network address portion is 192.68.10
* The host address portion is .15
* If the host wants to communicate with another host with an IP address which also begins with 192.168.10. (for example 192.168.10.20), it knows it’s on the same subnet and it can send the traffic directly
* If it wanted to communicate with another host with any other network address (for example 192.168.11.20), it knows it has to send the traffic via a router.
* For a destination address to be in the same subnet, the network portion must be exactly 192.168.10.
* Otherwise, it’s in a different subnet and traffic must be sent via a router.
* **The subnet mask always begins with contiguous ‘1’s.**
* **For example, 11111111.11110000.00000000.00000000 is a legal subnet mask.**
* **11101101.11110000.11100000.00001111 is not a legal subnet mask.**
* The host portion of the address is available to be allocated to the different hosts on the subnet (example PCs, Servers< Printers, Router Interfaces and Switch Management Addresses)
* The host portion of the address specifies the individual host and must be unique on that subnet.
* Hosts do not have to be numbered sequentially.
* If the network portion of the address is 10.10.10, you can have a host with IP address 10.10.10.20
* You can’t have two different hosts both with IP address 10.10.10.10. That would be a duplicate IP address. Whenever another host sent traffic to 10.10.10.10, the network would not know which one to send it to.
* We could have host 10.10.10.10 on one subnet and host 10.10.20.10 on another subnet.

**EXCEPTIONS:**

* **All 0’s in the host portion designates the network address and is not allowed to be allocated to a host.**
* **In our example, the network address is 192.168.10.0. It signifies the network address, which is the bottom address in that particular subnet.**
* **All 1’s designates the directed broadcast address for the subnet.**
* **Traffic with this destination address will be sent to all hosts in the subnet.**
* **In our example, the broadcast address is 192.168.10.255**
* **That leaves 192.168.10.1 to 192.168.10.254 available to be allocated to hosts.**

**Slash notation:**

* As we know that the subnet mask is always contiguous 1’s and then followed by a block of contiguous 0’s.
* Because it’s contiguous, we can count how many 1’s that we have in a row.

A screenshot of a computer

Description automatically generated

* In our example above, we have got 24 1’s in a row. Therefore, we can write it as /24.
* This allows us to write the subnet mask in slash notation which is more convenient than dotted decimal for network diagrams or in conversation.

**IP ADDRESS CLASSES - INTRODUCTION**

The IP Address classes are important to understand before we get into the more advanced topics of subnetting.

The IP address classes are Class A, B, C, D, and E.

**Important points to note:**

1. The bigger the host portion of the network, the more hosts we can have.
2. If the subnet mask is /8, we have 24 bits available to allocate to hosts.
3. If the subnet mask is /24, we only have 8 bits available to allocate to hosts.

**How Internet Addressing was meant to Work:**

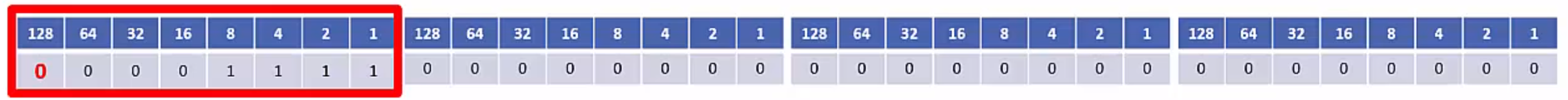
When IPv4 was first conceived, the designers didn’t know about what was going to happen with the Internet. They didn’t realize that there would be a huge explosion of usage, and everybody would be using it, and that everybody would require am IP address. So, when they first designed it, they designed it for what was right at that time. And as we go through this section, you need to think about the internet how it was then to understand why IPv4 was designed the way that it is.

The original way that IPv4 addressing was meant to work, when a company wanted to communicate on the internet, they would apply for a range of IP addresses.

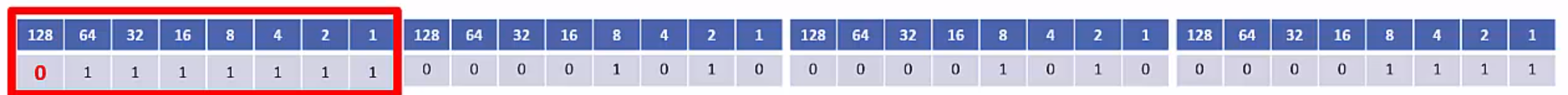
* The global assignment of IP addresses is handled by IANA (Internet Assigned Numbers Authority).
* This is the way it was originally supposed to work.
* When a company wants to communicate on the internet, they apply for a range of IP addresses.
* If they have 6000 hosts, they ask for a range of IP addresses big enough to cover that, plus room for growth.
* They then allocate their addresses to their hosts in their various offices.
* Unfortunately, when IPv4 was created, the designers didn’t realise how big the internet was going to get, and they didn’t create a big enough address space – there’s not enough address for everyone.
* The long-term solution to this problem is IPv6 which has a much bigger address space. IPv6 is a 128-bit address.
* Private IP addresses with NAT (Network Address Translation) are currently deployed in the majority of enterprise networks as a workaround.

**Class A IP Addresses:**

* The internet authorities split the IPv4 address spaces into separate classes.
* Class A addresses are assigned to networks with a very large number of hosts.
* The high-order (first) bit in class A address is always set to zero.
* The default subnet mask is /8.
* Valid network addresses range from 1.0.0.0 to 126.0.0.0/8
* This allows for 126 networks and 16,777,214 hosts per network.



* Therefore, the actual host address ranges from 1.0.0.0 to 126.255.255.254
* We may notice here, if we counted this up, the available values if we set the first bit as 0s, we could have all 0s, and we could go up to 127. But all 0s and 127 are not in the valid address that we can assign to out hosts because they are reserved addresses.
* 0.0.0.0/8 is reserved and signifies ‘this network’, and it’s used by some protocols. So, 0.0.0.1 and 0.255.255.255 are not Valid addresses that you can assign to hosts. The entire range is not available.
* Also, 127.0.0.0/8, which is also in the class A space is also reserved. That is used as the loopback address, and it’s used for testing the IP stack on the local computer.
* 127.0.0.0 to 127.255.255.255 are not valid host addresses. You see, I’ve written down at the bottom here, whoops! They just wiped out over 33 million addresses (33,554,428) that could have been used for addressing actual hosts on the public internet. If you think about it, ‘this network’, they could have just used a single address for that. One address rather than 16 million addresses. And the same for the loopback, we didn’t really need 16 million addresses to be used for loopback testing, and if you’re thinking, “Well, there’s this huge shortage of addresses on the internet, why would they do something crazy like that?”. The reason is, when IPv4 was designed, they didn’t realize that they were going to run out to addresses, so they thought, “Hey, it’s no problem, we can assign 16 million addresses for testing, it doesn’t matter, because we don’t have a shortage of addresses. “They didn’t realize what was going to happen later on.



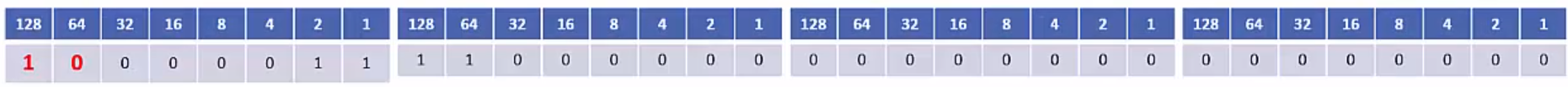
* So, all 0s would begin with o, obviously, and 127 is 0 and then all 1s. Class A always begins with a 0 as the first bit.

**Subnetting**

* Obviously, a company wouldn’t put all 16,777,214 hosts into a single logical network, this would be terrible for performance and security.
* They would split their /8 address allocation into smaller subnets and allocate these to different offices and types of hosts.
* For example, if they received 15.0.0.0/8, they could allocate ethe subnet 15.0.1.0/24 to sales computer in New York, 15.0.2.0/24 to accounting PCs and 15.0.9.0/24 to sales computer in Boston.
* This is called subnetting.

**Class B IP addresses:**

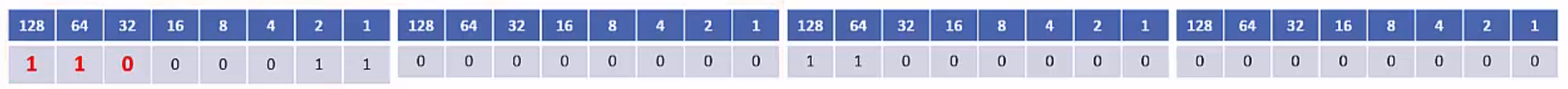
* Class B IP addresses are assigned to medium-sized to large-sized networks.
* The two high-order bits in a class B address are always set to binary 1 0.
* The default subnet mask is /16.
* Valid network addresses range from 128.0.0.0 to 191.255.0.0/16
* This allows for 16,384 networks and 65,534 hosts per network.
* This would also be sub netted in a real-world environment.



* 131.192.0.0/16

**Class C IP Addresses:**

* Class C addresses are used for small networks.
* The three high-order bits in a class C address are always set to binary 110.
* The default subnet mask is /24
* Valid network addresses range from 192.0.0.0 to 223.255.255.0/24
* This allows for 2,097,152 networks and 254 hosts per network.
* This could be allocated as is for a real-world network, or subnetted into smaller subnets.



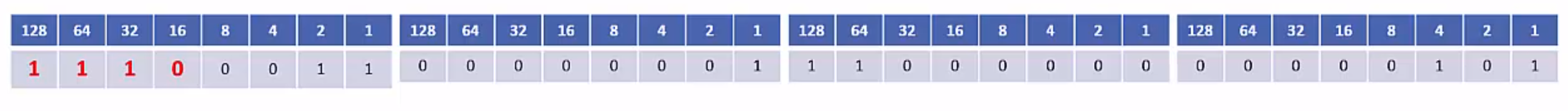
* 195.0.192.0/24

**A Quick Note on Private Addresses**

* There is also a range of reserved Private Addresses in each class.
* These are valid to be assigned to hosts but they are not routable on the public internet.
* They were originally designed for hosts in a closed private network with no internet connectivity.
* Class A: 10.0.0.0 to 10.255.255.255
* Class B: 172.16.0.0 to 172.31.255.255
* Class C: 192.168.0.0 to 192.168.255.255
* Private addresses will be studied later in detail.

**Class D IP Addresses:**

* Class D addresses are reserved for IP multicast addresses.
* The four high-order bits in a class D address are always set to binary 1110.
* These addresses are not allocated to hosts and there is no default subnet mask.
* Valid addresses range from 224.0.0.0 to 239.255.255.255



* 227.1.192.5

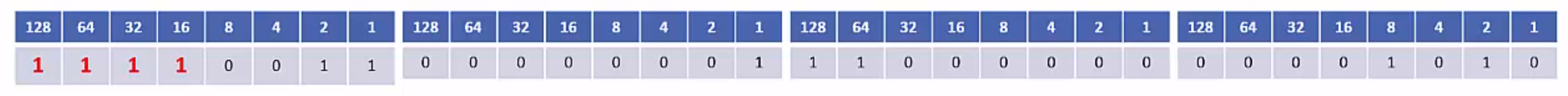
To see how multicast works, let’s do a quick review of unicast first.

A diagram of a computer network

Description automatically generated

**Class E IP Addresses**

* Class E IP Addresses are ‘experimental and reserved for future use’.
* The high-order bits in a class E address are set to 1111
* These addresses are not allocated to hosts and there is no default subnet mask.
* Addresses range from 240.0.0.0 to 255.255.255.255
* 255.255.255.255 is the broadcast address for ‘this network’.



* 243.1.192.10

A table with numbers and letters

Description automatically generated

**IP address Table**

**SUBNETTING**

In the earlier section, we started off on OSI layer 3, and we learned about IP addresses and how we can combine those with the subnet mask to define the boundaries between our logical networks. In this section, we’re still going to be on layer 3, but we’re going to go much deeper into how we can control the boundaries between our networks with the use of subnetting.

**CIDR – Classless Inter-domain Routing**

* A problem with the original implementation of the classful addresses was that when the internet authorities gave out addresses, they always give a complete Class A with a /8, or a complete class B with a /16, or a complete class C with a /24 subnet mask.
* This gave a problem that if a company had more than 254 hosts, they were too big for a class C, they would have to given to Class B. And, if they had, say, 500 hosts, they would actually get allocated addresses for 65,543 hosts which is obviously too much.
* This led to huge amounts of global address space being wasted.
* Therefore, CIDR – Classless Inter domain Routing, was introduced in 1993 as a solution or a partial solution for this problem.
* **CIDR removed the fixed /8, /16 and /24 requirements for the address classes and allowed them to be split or ‘subnetted’ into smaller networks.**
* **for example – 175.10.10.0/20**
* **Companies can now be allocated an address range which more closely matches their needs and does not waste addresses.**
* **We get another benefit from CIDR as well, which is route summarization. (Aggregate blocks of networks can be advertised on the internet)**

**A diagram of a network

Description automatically generated**

we can see in the example here that we have got ISP - A, and they have allocated the address blocks that we see on the left. So, one company got 175.10.0.0/24, another one got 175.10.1.0/24, 175.10.2.0/24, etc., all the way up to 175.10.255.0/24. So, they’ve given out 256 address blocks there.

We’ve also got ISP – B, and they’ve given out 175.11.0.0/24, 175.11.1.0/24, and so on.

ISP-A and ISP-B get connected.

Now, if we didn’t have CIDR and we weren’t able to do route summarization, ISP-A would advertise all 256 address blocks to ISP-B, and ISP-B would, vice-versa, advertise all their 256 addresses blocks to ISP-A. But when we got CIDR and route summarization, what we can do is the two ISPs can advertise just an aggregate block. So, ISP-A rathe than advertising all 256/24s, it advertises 175.10.0.0/16, which is superset of all those 256 smaller networks. So now, ISP-B, it only learns one route to all the networks behind ISP-A rather than learning 256, and ISP-B will vice-versa just advertise one route 175.11.0.0/16 to ISP-A.

**Route summarization benefits:**

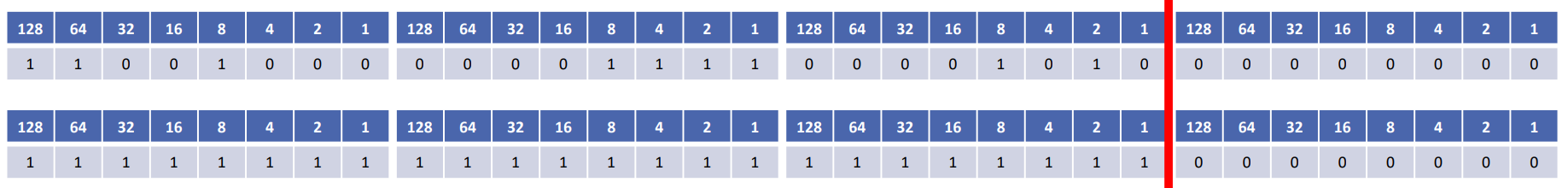
1. ISP-A does not know about all 256/24 networks reachable in ISP-B.
2. It only has the single 175.11.0.0/16 summary route.
3. This reduces the size of ISP-A’s routing table and takes up less memory.
4. If an individual link goes down in ISP-B, it has no impact on ISP-A. The single summary route does not change.
5. (Routers in ISP-B would have to recalculate their routing table if a link went down).
6. This restricts issues to the local part of the network and reduces CPU load.

**So, this was about CIDR, which is closely related to VLSM, our Variable Length Subnet Mask.**

**Subnetting Overview:**

At the end of this section, we will be able to understand how we can implement subnetting.

* To understand this, let’s think about it from the point of view of the originally intended IPv4 design again, where all host which can communicate on the internet have a public IP address.
* Let’s say we’re the network designer for a small business with four departments spread over two offices, and we want to manage our own public address space.
* Rathe than purchasing separate address ranges for the different departments, we can purchase a single range and subnet it into smaller portions.
* Let’s say we’ve got only a handful of hosts in each of those departments. Rather than buying 256 addresses for each, we can buy a single Class C range, and then we can divide that network up into smaller networks and assign it to the different parts of our networks.
* So, in an example, let’s say we’ve been allocated Class C 200.15.10.0/24, the default class C subnet mask.



* So, to do our subnetting, to divide that network into smaller subnets, we need to borrow some host bits and add them to the network portion of the address. So, we’re going to move the line that separates the network portion of the address and the host portion further over to the right.
* So, we’re going to take some of our host addresses away and give them to the network portion of the address.
* When we do subnetting, the line always moves to the right, and the further to the right we go, the more subnets that we’re going to have, but the less the number of hosts that we’ll have on each subnet.

**CALCULATING THE NUMBER OF NETWORKS:**

* To calculate the number of available subnets, the formula is 2subnet-bits
* If a Class C network used a /28 subnet mask, then we’ve borrowed 4 bits from the default /24.
* 24 = 16 available subnets
* If a class B network used a /28 subnet mask, then we’ve borrowed 12 bits from the default of /16.
* 212 = 4096 available subnets.
* Hosts on different subnets need to go via a router if they want to communicate with each other. That’s the whole point of having our IP addressing and doing subnetting. It’s to divide our network up into different logical parts of the network. And it’s the routers that are the devices that know how to get everywhere, and it can direct the traffic.

**TO CALCULATE THE NUMBER OF HOSTS:**

* To calculate the number of available hosts, the formula is 2host-bits minus 2. ( 2 to the power of how many bits there are minus 2)
* We subtract 2 because the network address and broadcast address cannot be assigned to hosts.
* If a class C network uses a /28 subnet mask, then we have 4 bits left for hosts.
* 24 -2 = 14
* If a class B network uses a /28 subnet mask, then we have 4 bits left for hosts.
* 24 -2 = 14
* Therefore, the number of hosts we’re going to get is going to be dependent on the subnet size, and it’s going to be the subnet mask’s size, it’s going to be the same whether it’s class A, Class B, or Class C. But the number of subnets we’ll get, if I go back and slide, it’s going to be different for class A, B, or C because we’ve got different default subnet mask sizes.
* For example, if we use a /28 with a class B, then we’re going to have 4096 available subnets. If we use a /28 with a class C, we’re going to have 16 available subnets. There’s a difference here, but for the hosts, it’s always going to be the same.

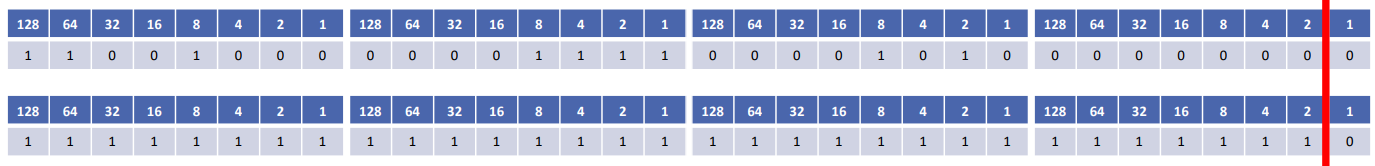
**A quick note on ‘ip subnet-zero’ command.**

* Just like we have to subtract 2 to get the number of valid hosts, we used to have to subtract 2 to get the number of available networks also.
* This is because, in the original Internet standards, it wasn’t allowed to use network bits of all 0’s or all 1’s (just like we can’t use all host bits of all 0’s or all 1’s). So, that took away two of our available subnets.
* There wasn’t really any practical need for this, and it wasted address space. But there is a practical need with the host bits because we’ve got that network address and the broadcast address that are actually used.
* The ‘ip subnet-zero’ command on a router overrides the limitation and is enabled by default.

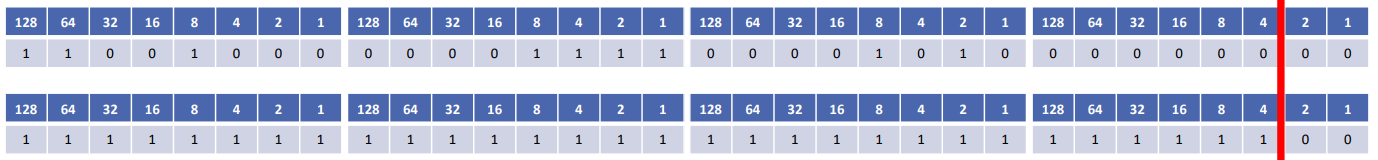
**Note: On cisco routers now, for quite a long time, there’s a default command of ‘ip subnet-zero’, which disables this behaviour. By default, the command is enabled, so those two extra network addresses are available in cisco networks.**

**Subnetting Class C Networks and VLSM**

* Let’s say we’ve been allocated class C 200.15.10.0/24



* **If we move the line all the way up to the right we’re now using /31 (or 255.255.255.254)**
  + **We can’t do a /32 if we want to have more than one available host now. You can use a /32 subnet mask, but there is only one host there that’s mostly used for a loopback addresses.**
  + So, if you do need to have multiple hosts, then the furthest right you can go is a /31.
  + That leaves one bit for the host address, with a possible value of 0 or 1.
  + It borrows 7 bits for the network address.
  + This gives us 128 subnets (27) which can accommodate 2 hosts each.
  + But wait, if we’re using a /31, there’s only 2 possible values to assign to hosts. What about the network and the broadcast address? A /31 subnet breaks the standard rules of IP addressing, where we have to have the network address and the broadcast address at the top and the bottom of the range.
  + /31 subnets are supported on cisco routers for point-to-point links. (Which have no need for a network or broadcast address.)
* **We’re going to create a/30 now.** 
  + The subnet mask in dotted decimal, 255.255.255.252 as shown below:



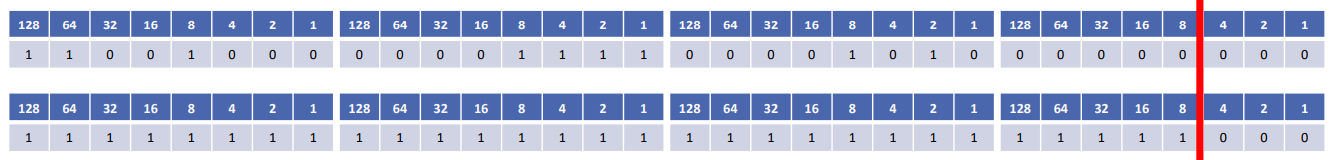
* + This leaves 2 bits for the host address, 22=4, minus 2 for the network and broadcast address = 2 possible hosts.
  + It borrows 6 bits for the network address.
  + This gives us 64 subnets (26), which accommodate 2 hosts each.
  + The valid addresses are on the first subnet.
  + The network address is 200.15.10.0, and the broadcast address is 200.15.10.3
  + The valid addresses are:
    - 200.15.10.1 to 200.15.10.2(network – 0, broadcast – 3)
    - 200.15.10.5 to 200.15.10.6(network – 4, broadcast – 7)
    - Etc., to
    - 200.15.10.253 to 200.15.10.254(network – 252, broadcast – 255)
  + As we look at this, the way that we can calculate what the different valid subnets are going to be, notice where the line is, it’s after the number 4. So the network address is going to go up in values of 4.

**Differences between /31 and/30**

|  |  |
| --- | --- |
| **/31** | **/30** |
| Can accommodate 2 hosts per subnet | Can accommodate 2 hosts per subnet |
| Supports 128 subnets | Supports 64 subnets |
| Useful if you need to maximise use of your address space | More standard and commonly used |

**Note: For the CCNA exam, use /30 when a subnet to support 2 hosts is required, unless told to use /31**

* **Class C /29 subnet**
  + Let’s say we’ve been allocated Class C 200.15.10.0/24

****

* + Let’s mode the line back a place. We’re now using /29(or 255.255.255.248)
  + This leaves 3 bits for the host address, 23 minus 2 = 6 possible hosts.
  + It borrows 5 bits for the network address.
  + This gives use 32 subnets (25) which accommodate 6 hosts each.
  + Valid host addresses are mentioned below. Notice that the line is after the 8, so the network addresses are going to go up in values of 8.
    - 200.15.10.1 to 200.15.10.6(network – 0, broadcast – 7)
    - 200.15.10.9 to 200.15.10.14(network – 8, broadcast – 15)
    - Etc.,
    - 200.15.10.249 to 200.15.10.254(network – 248, broadcast – 255)

**Variable Length Subnet masks (VLSM)**

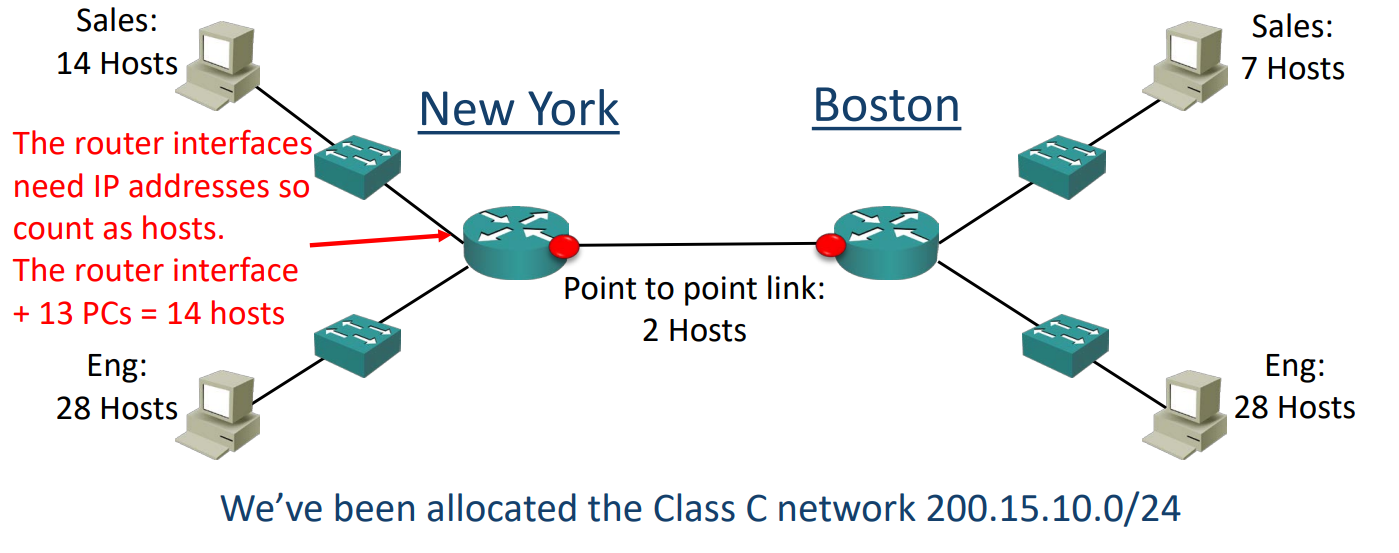
* With early routing protocols, like RIPv1, they only supported fixed length Subnet Masks (FLSM), which meant you could subnet, but all subnets in a particular network had to be the same size.
* So, all of your different subnets had to accommodate 14 hosts, for example, or 30 hosts, for example. And if you had one subnet that needed up to 30 hosts, you had to have all your subnets at the size of 30 hosts, or even if one of them only had 3 hosts in it.
* Then, later routing protocols, they came out with support for Variable Length Subnet Masks, that means that within the same range, you can have different sizes of subnets in there to see.
* So, say I’ve got one part of my network which has got 10 hosts in it. I could give them 14 available hosts and another part of a network that’s got 28 hosts there. I could give them a subnet mask that gives them 30 available hosts. So, Variable Length subnet Masking, it means we can use different length subnet masks within the same network.
* This is a good thing because it lets us be much more precise with the size of our networks, and we’re going to be wasting a lot less addresses.

**Variable Length Subnet Masks (VLSM) Example**

In this section, we’re going to carry on with our subnetting, and by the end of this section, we will be able to carry out a Variable Length Subnet Mask design for a real-world network topology.

The things we need to consider when we are going to do the design are:

* How many locations do we have in the network?
* How many hosts are there in each location?
* What are the IP addressing requirements for each location? (Should different departments or types of hosts be in different subnets/)
* What size is appropriate for each subnet? (Don’t waste addresses but leave room for growth.)



For our example, this will be our topology diagram.

We’ve got an organization, they’ve got an office in New York and a branch office in Boston. New York is their headquarters, and they’ve got 28 hosts in the engineering department and 14 hosts in Sales. In Boston, they’ve also got engineering and sales. They’ve got 18 in engineering, the same as in New York, and they’ve only got 7 hosts in the sales department.

So, those are the different parts of the network that we need to apply IP addresses to, and we’ve been allocated the Class C network of 200.15.10.0/24 from our internet Service Provider.

Another thing we need to do is don’t forget about our point-to-point links between the routers. They need to have connectivity with each other, so they’re going to need IP addresses too. It’s a point-to-point link, so we’ve just got 2 host IP addresses there. The outside interface on the New York Router is connected to the outside interface on the Boston router.

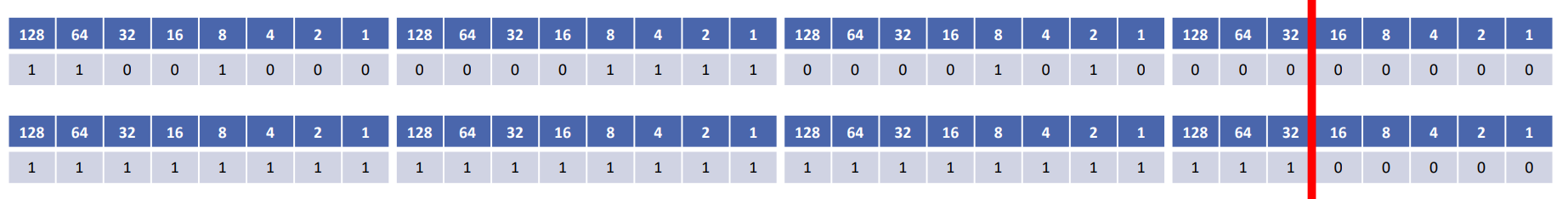
**Subnetting design steps:**

* Find the largest segment and allocate a suitable subnet size for it.
* Allocate this subnet at the start of the available address space.
* Continue going down the list.
* In the real world, you want a scalable design – you will likely allocate spare subnets for future growth, and then leave space in the subnets for additional hosts. (Let’s say that I’ve got a subnet that has got 14 hosts on there. Don’t give a subnet that size because maybe in a few weeks’ time, a couple of extra people are going to join the department and now, how are you going to fix that problem?)
* In the CCNA exam, do exactly what the question asks, don’t worry about whether it’s best practice or not.

The Engineering departments in both sites have 28 hosts. For our example, we’ve been told that the departments will not grow, and we need to use the smallest subnets possible to maximise our address space. **We will calculate the optimal subnet mask for the Engineering departments. Also determine the network and broadcast addresses that will be allocated to both Engineering departments, and the range of host addresses.**

**Solution:**

So, we’ve been allocated 20.15.10.0/24. We’ve been allocated a class C, and we’ve got 2 different departments that we want 2 different subnets for, and they had got 28 hosts each. A /27 supports 30 hosts. A /28 is 14, so it’s not big enough, we can’t use that.



* /27 (or 255.255.255.224) supports 30 hosts
* **New York Engineering subnet:**
  + Network address: 200.15.10.0/27
  + Broadcast address: 200.15.10.31
  + Hosts: 200.15.10.1 to 30
* **Boston Engineering subnet:**
  + Network address: 200.15.10.32/27
  + Broadcast address: 200.15.10.63
  + Hosts: 200.15.10.33 to 62
* **New York Sales subnet:**
  + Network address: 200.15.10.64/28
  + Broadcast address: 200.15.10.79
  + Hosts: 200.15.10.65 to 78
* **Boston sales Subnet:**
  + Network address: 200.15.10.80/28
  + Broadcast address: 200.15.10.95
  + Hosts: 200.15.10.81 to 94

After this, we need to allocate the addresses for the point-to-point link between the routers in Boston and New York. Another thing, we would do in the real world is we would also allocate address space for our loopback addresses. Loopback addresses are used for management. They’re a logical address, so, there’s no anything physical on the other, so we’ll usually allocate a /32 to out loopback addresses.

So, for link between the New York and Boston routers:

* **Point-to-point connection between New York and Boston:**
  + 2 hosts are there.
  + So, we’ll use a /30 here.
  + Network address: 200.15.10.96
  + Broadcast address: 200.15.10.99
  + Hosts: 200.15.10.97 to 98

A diagram of a diagram of a couple of circles

Description automatically generated with medium confidence