AISE 4430

Intro to Networking, Security, and IoT Systems

Arnav Goyal

2025-2026

Contents

1 Sample Problems & Solutions

1.1 Sample Problem Set 5

This sample problem set deals with the IP architecture, subnetting, etc.

1.1.1 Problem I

***Question*:**

Do Routers have IP addresses? If so, how many?

***Answer*:**

A Router is a device that joins different **subnets** together, To communicate on each of these networks it needs an **interface** on each one. So yes, a Router has one IP address for every interface it connects to.

1.1.2 Problem II

***Question*:**

What is the 32 bit binary equivalent of the IP Address 223.1.3.27?

***Answer*:**

IP Addresses follow the **four-dotted-decimal** convention, in decimal this consists of four numbers from 0-255 placed together with dots in between them. To convert these into binary, we must write out the corresponding 8 bits of binary for each number. An easy way to do this is to first convert the number into **hexadecimal** format, and then write the binary out for it.

To convert a decimal into hex, we perform the following recursive operation. Suppose our 8-bit hex number is labeled H, where H_0 and H_1 correspond to the LSh and MSh respectively. Lets also suppose our decimal number is labeled D, Note here that the % sign corresponds to the mod operator

$$H_n = \left\lfloor \frac{D}{16^n} \right\rfloor \% \, 16$$

This means our corresponding LSh and MSh become F and D respectively

$$H_0 = \left\lfloor \frac{223}{1} \right\rfloor \% \, 16 = 16 = F$$

$$H_1 = \left\lfloor \frac{223}{16} \right\rfloor \% \, 16 = 13 = D$$

Thus 223 decimal is equivalent to DF in hex. Now we just write it out in binary. Repeating this process for each number in the IP address provides the following result for the binary IP address.

1101_1111 0000_0001 0000_0011 0001_1011

1.1.3 Problem III

***Question*:**

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 223.1.17/24. Also suppose that Subnet 1 is required to support up to 125 interfaces, and Subnets 2 and 3 are each required to support up to 60 interfaces. Provide three network addresses (of the form a.b.c.d./x) that satisfy these constraints.

***Answer*:**

To do this problem we need to figure out what this notation means first. Here we have an IP address followed by a slash. The main part of this address is 223.1.17.0. Note that this number has an implicit zero, because there is no number specificed before the slash. We also have the /24 part, which means that we have 24 bits in the *entire* IP address reserved for the host (main part of the IP). This means we have 32 - 24 = 8 bits leftover to partition subnets into, Note that since IP addresses are four numbers within 0-255 it will always fit into 32 bits.

So with the 8 bits leftover we need to partition it properly into three subnets of sizes 125, 60, and 60. When we partition a subnet, we need to understand that two addresses for *each* subnet are unusable by other interfaces, these are the **network address** and the **broadcast address**. These reserved addresses take up the lowest and highest slots of the subnet partition respectively.

Thus the number of usable bits per partition becomes

$$\text{slots} = 2^b - 2$$

Our challenge is finding out the number of bits we need in each partition, we always start with the largest one (125).

so we will need a slot that fits 125 interfaces + 2 reserved addresses = 127 slots in total. This fits super will into a 7 bit value. We have all 8 bits left over since the end part of the main IP is 0, thus we have one bit left in the IP address, so our notation becomes

$$\text{Subnet 1 } = 223.1.17.0/25$$

This means we have IP addresses for this subnet ranging from 223.1.17.0 to 223.1.17.127, where the boundaries are reserved, and anything in between is usable by interfaces.

For the other two subnets, we do something similar. We will need 60 + 2 slots, which fits into a 6 bit value, this means we have the following 223.1.17.x/26

this 'x' value will need to be found though, its the start of the subnet partition, so here since our larger net ends at 127, we put 128

so our first subnet of 60 becomes

$$\text{Subnet 2 } = 223.1.17.128/26$$

This second subnet ranges from 223.1.17.128 to 223.1.17.191, where the boundaries are reserved, and anything in between is usable by interfaces. This process repeats for subnet three, where it becomes:

$$\text{Subnet 3 } = 223.1.17.192/26$$

1.1.4 Problem IV

***Question*:**

Consider a subnet with prefix 101.101.101.64/26. Give an example of one IP address (of form xxx. xxx. xxx. xxx) that can be assigned to this network

***Answer*:**

We know the subnet's boundaries are reserved for network/broadcast address, so anything within these boundaries works! Thus we can just write anything withing the range 101.101.101.65 to 101.101.101.126, .64 and .127 are reserved.

1.1.5 Problem V

Question:

Suppose an ISP owns the block of addresses of the form 101.101.128/17. suppose it wants to create four subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form a.b.c.d/x) for the four subnets?

Answer:

To answer this lets look at how many bits we have left to partition this ID into.

$$32 - 17 = 15$$

Thus we have 15 bits to work with! Note that this is over 8 bits, which means we will be using the last two numbers in the IP address, instead of only the last one. Lets obscure the first two numbers as x = 101.101. for shorthand

Our entire range of values will be

$$2^{15} = 32,768$$

to divide this by four we can essentially shift two bits right, resulting in each subnet having 13 bits to work with

$$2^{13} = 8192$$

So each subnet will have 8192 slots, where only 8190 are usable, the other two are reserved.

We can divide 8192 by 256 to see how many incrments in the third number we get per slot, 8192/256 = 32 increment per 8192 slots

We start at x.128, so lets add 31 each time, four times. This gives the following subnets

Subnet 1 => x.128.y to x.159.z

Subnet 2 => x.160.y to x.191.z

Subnet 3 => x.192.y to x.223.z

Subnet 4 => x.224.y to x.255.z

To find out the y and z values from above we can see whats left over from our increment, here our increment cleanly divides into 32 slots, thus the start and end will be 0 and 255 respectively as we fully fill up the chunk, however if we have any left over addresses, we would have to calculate things manually by hand tediously. Thus our 4 subnets become

Subnet 1 => 101.101.128.0 to 101.101.159.255 (101.101.128/19)

Subnet 2 => 101.101.160.0 to 101.101.191.255 (101.101.160/19)

Subnet 3 => 101.101.192.0 to 101.101.223.255 (101.101.192/19)

Subnet 4 => 101.101.224.0 to 101.101.255.255 (101.101.224/19)

1.1.6 Problem VI

***Question*:**

Suppose an ISP owns the block of addresses of the form 192.168.15.0/24. Suppose it wants to create four subnets from this block. What are the prefixes (of form a.b.c.d/x) for the four subnets? Find the number of interfaces that each subnet can support?

***Answer*:**

This one is very similar to the previous one, here we have 8 bits left over to work with, four subnets as well. This means we have

$$2^8 = 256 \text{ total slots}$$

$$2^6 = 64 \text{ slots per subnet}$$

So each subnet will have an increment of 63 in the smallest number, This means our subnets become

Subnet 1 => 192.168.15.0 to 192.168.15.63 (192.168.15.000/26) Subnet 2 => 192.168.15.64 to 192.168.15.127 (192.168.15.064/26) Subnet 3 => 192.168.15.128 to 192.168.15.191 (192.168.15.128/26) Subnet 4 => 192.168.15.192 to 192.168.15.255 (192.168.15.192/26)

Each subnet has 64 slots, and 60 of them can be used for interfaces, thus we have 240 total slots for interfaces available here throughout all 4 subnets.

1.1.7 Problem VII

Question:

Suppose an ISP owns the block of addresses of the form 132.59.0.0/16. Suppose it wants to create two hundred subnets from this block. What are the prefixes (of form a.b.c.d/x) for the four subnets? Find the number of interfaces that each subnet can support?

Answer:

Here we need 200 subnets, so we need to see how many are used to MUX between the subnets. Here clog2 refers to the ceiling of log base 2 operation, since we are working with binary values it is abbreviated to the common shorthand.

$$b = \text{clog2}\,(200) = 8$$

Thus we have 16 bits reserved for the ID, another 8 bits to select each subnet, and the remaining bits determine the slots per subnet, so we have 24 fixed bits, and thus 8 bits left per subnet. This means each subnet will support 253 interfaces and 2 more reserved slots.

The first four subnets have the following, since each subnet has 256 slots, it increments the second lowest decimal number by 1. Thus our subnets become

Subnet 1 => 132.59.0.0/24 Subnet 2 => 132.59.1.0/24 Subnet 3 => 132.59.2.0/24 Subnet 4 => 132.59.3.0/24

Subnet n => 132.59.$(n-1)$.0/24, for any n in [0, 200], the other slots [201, 255] will be effectively unused.

1.2 Sample Problems Set 6

This problem set deals with Link Layer protocols such as CSMA/CD Ethernet

1.2.1 Problem I

Question:

Recall that with the CSMA/CD protocol, the adapter waits k · 512 bit times after a collision, where K is drawn randomly. For K = 100, how long does the adapter wait until returning to Step 2 for a 10 Mbps Ethernet? For a 100 Mbps Ethernet?

Answer:

First we need to define what a **bit-time** is. This is the amount of time it takes to transmit a single bit onto the link. More specifically:

$$t_{\text{bit}} = \frac{1}{R}$$

In this case our 10Mbps rate has a $t_{\text{bit}} = 100ns$ and our 100 Mbps rate has a $t_{\text{bit}} = 10ns$ With CSMA/CD we wait for K times 512 bit times before retransmitting, where K is randomly chosen. Here the problem fixed K = 100, thus our wait times are:

$$t_{\text{wait, 10Mbps}} = 51200 \cdot 100ns = 5.12ms$$

$$t_{\text{wait, 100Mbps}} = 51200 \cdot 10ns = 512\mu s$$

1.2.2 Problem II

***Question*:**

Suppose nodes A and B are on the same 10 Mbps Ethernet segment, and the propagation delay between the two nodes is 225 bit times. Suppose node A begins transmitting a frame and, before it finishes, node B begins transmitting a frame. Can A finish transmitting before it detects that B has transmitted? Why or why not? If the answer is yes, then A incorrectly believes that its frame was successfully transmitted without a collision

Hint: suppose at time t = 0 bit times, A begins transmitting a frame. In the worst case, A transmits a minimum-sized frame of 512+64 bit times. So A would finish transmitting the frame at t = 512 + 64 bit times. Thus the answer is no, if B's signal reaches A before bit time t = 512 + 64 bits. In the worst case, when does B's signal reach A?

***Answer*:**

Here the hint mentions that the minumum sized frame is 512 + 64 = 576 bits, which takes 576 bit times to transmit, and an extra 225 bit times to propagate to node B. So lets look at this from A's perspective.

Suppose A transmits at $t = 0$, it will finish transmitting at $t = 576$ and due to pipelining the first transmitted bit would arrive at node B at $t = 225$. There are two distinct cases here:

If node B transmits at $t = 0$ as well, the same holds true, so if both A and B transmit at $t = 0$ they would both sense a collision at $t = 225$ after the first bit propagates to the other node. This is the *best* case.

The latest possible time node B could transmit is at $t = 224$, one bit time before the message from A reaches B. If this occurs, the total time it takes for node B to transmit would be $t = 224 + 225 = 249$ for the first bit to propagate and reach node A. This is is smaller than the total time it would take for A to finish transmitting. This is the *worst* case.

So in this case node A will never incorrectly classify its message as in the worst case scenario the first bit from node B arrives earlier than the entire transmission time for node A. In other words, if A doesnt detect another node transmitting during its own transmission, it means no other host has begun transmitting.

1.2.3 Problem III

***Question*:**

Consider a 100 Mbps 100BaseT Ethernet. In order to have an efficiency of 0.5, what should be the maximum distance between a node and the hub? Assume a frame length of 64 bytes and that there are no repeaters.

Does this maximum distance also ensure that a transmitting node A will be able to detect whether any other node transmitted while A was transmitting? Why and why not? How does your maximum distance with the actual 100 Mbps standard?

***Answer*:**

Recall from the slides the efficiency equation, Here we desire $\eta = 0.5 = 50\%$

$$\eta = \frac{1}{1 + 5\left(\frac{t_{\text{prop}}}{t_{\text{trans}}}\right)}$$

We can easily find t_{trans} for transmitting the 64 byte frame + 1 byte of header as shown below.

$$t_{\text{trans}} = \frac{L}{R} = \frac{65 \times 8}{100\text{M}} = 5.76\mu\text{s}$$

With this quantity we can plug this value into the efficiency equation to solve for t_{prop}.

$$t_{\text{prop}} = \left(\frac{1}{\eta} - 1\right) \cdot \frac{t_{\text{trans}}}{5} = 1.152\mu\text{s}$$

We also assume that the propagation speed $s = 1.8 \cdot 10^8$ m/s, then we can solve for the distance d

$$t_{\text{prop}} = \frac{d}{s}$$

$$d = s \cdot t_{\text{prop}} = 1.8 \cdot 10^8 \cdot 1.152\mu\text{s} = 207.36\text{m}$$

Thus we should have a physical distance of 207.36 meters for an efficiency of 50% in this link. It is close to the 100 Mbps Ethernet Standard value of 200m.

The second part of this question asks if node A is transmitting, it can detect any other nodes are transmitting at the same time. For this we require that the below holds true.

$$t_{\text{trans}} > 2 \cdot t_{\text{prop}}$$

In our case it certainly does hold true, so we can be certain that no matter what, A will be able to detect if another node is transmitting during its own transmission, and will correctly classify a collision.

1.2.4 Problem IV

***Question*:**

A token-ring LAN interconnects M stations using a star topology in the following way. All the input and output lines off the token-ring station interfaces are connected to a cabinet where the actual ring is placed. Suppose that the distance from the each station to the cabinet is 100 meters and the ring latency per station is eight bits. Assume that frames are 1250 bytes and that the ring speed is 25 Mbps.

a) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit an unlimited number of frames/token?

b) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit 1 frame/token using single-frame operation?

***Answer*:**

In a token ring, each node basically sends data if it has any, and then passes on a token afterwards

Lets start with part (a), here if stations are allowed to transmit at max speed for unlimited time, the max rate is 25 Mbps because one station might eat up the entire channels bandwidth, leaving the other channels eternally waiting and not transmitting the token, so the arrival rate is the channel datarate, 25 Mbps.

For part (b) each station will get the token, send 1 frame of data, and then pass the token onwards. To find the arrival rate we need to find the time it takes for an entire traversal of the network topology.

In other words each station will experience the following delays:

1. cabinet to station prop delay (receiving the data)
2. processing delay (processing the data, question says 8 bit times)
3. transmission delay (sending the data onto the link)
4. station to cabinet prop delay (sending the data)

We can calculate these delays for each station

$$t_{\text{trans}} = \frac{L}{R} = \frac{1250 * 8}{25\text{M}} = 400\mu\text{s}$$

$$t_{\text{prop}} = \frac{d}{s} = \frac{100}{3 \cdot 10^8} = 333.333\text{ns}$$

$$t_{\text{proc}} = \frac{8}{25\text{M}} = 120\text{ns}$$

Now we can calculate the time required per station:

$$t_{\text{station}} = t_{\text{prop}} + t_{\text{proc}} + t_{\text{trans}} + t_{\text{proc}} = 400.7\mu\text{s}$$

Then a whole cycle around the ring of 5 stations is simply $5 \times t_{\text{station}}$ Which takes approx 2 ms, and in this time we transmit one frame of 1250 bytes.

This means the arrival rate in general for a token ring with M stations is

$$R_{\text{ring}} = \frac{L}{M \cdot t_{\text{station}}}$$

Using this formula in our case gives the following arrival rate:

$$R_{\text{ring}} = \frac{1250 \cdot 8}{2\text{ms}} \approx 5\text{Mbps}$$

This value is very close to the equation $R_{\text{ring}} = \frac{R}{M}$, Its only exactly the same here because i approximated the total value for $4 \times t_{\text{station}}$

1.2.5 Problem V

Question:

Consider six stations that are all attached to three different bus cables. The stations exchange fixed-size frames of length 1 second. Time is divided into slots of 1 second. When a station has a frame to transmit, the station chooses any bus with equal probability and transmits at the beginning of the next slot with probability p.

Find the value of p that maximizes the rate at which frames are successfully transmitted.

Answer:

In order to do this we require basic probability and calculus :(First lets analyze this first with probability.

We have 6 stations here, each station has the following basic stats.

- There is a $\frac{p}{3}$ chance that the station transmits onto bus A
- There is a $1 - \frac{p}{3}$ chance that the station doesnt transmit onto bus A (chooses B or C instead)

We have 6 six stations in total, so the probability that station 1 transmits onto bus A, is composed of the following

1. station 1 transmits onto bus A
2. AND stations 2-6 do not transmit onto bus A

We can write this mathematically as shown below

$$P_{\text{1 on A}} = (p) \cdot \left(1 - \frac{p}{3}\right)^5$$

this probabilty can be repeated for the remaining stations 2 - 6, to get the same expression as the one above. Thus the total probability that any station successfully transmits on bus A is 6 times the previous value

$$P_{\text{success, bus A}} = 6 \cdot P_{\text{1 on A}}$$

here since the entire system is symmetrical we have the same probabilty for a success on bus B and bus C as well. Regardless we have the probabilty as a function of p so we can now do the basic calculus optimization thing

$$\frac{dP_{\text{success, bus A}}}{dp} = 6 \cdot \frac{d}{dp}\left[(p) \cdot \left(1 - \frac{p}{3}\right)^5\right]$$

Painful product rule here

$$\frac{dP_{\text{success, bus A}}}{dp} = 6 \cdot \left[\left(1 - \frac{p}{3}\right)^5 - \frac{5}{3} \cdot (p)\left(1 - \frac{p}{3}\right)^4\right]$$

Now we do algebra sadly

$$\frac{dP_{\text{success, bus A}}}{dp} = 6\left(1 - \frac{p}{3}\right)^4 \left(\left(1 - \frac{p}{3}\right) - \frac{5p}{3}\right)$$

We set this equal to 0 and solve for possible values of p here, which occurs when any of the factors are equal to 0. This means we can enumerate solutions to this equation above as p_n

p_0 can be found by setting the first factor equal to 0

$$0 = \left(1 - \frac{p_0}{3}\right)$$
$$p_0 = 3$$

p_1 can be found by setting the second factor equal to 0

$$0 = \left(1 - \frac{p_1}{3}\right) - \frac{5p_1}{3}$$
$$0 = \left(1 - \frac{6p_1}{3}\right)$$
$$p_1 = \frac{1}{2}$$

we cannot have a probabilty value of $p = 3$, thus we exclude it as a valid solution, leaving $p = 0.5$ as the optimal solution here to maximize the probability of successful transmissions on cable A, B, or C from any of the six stations.