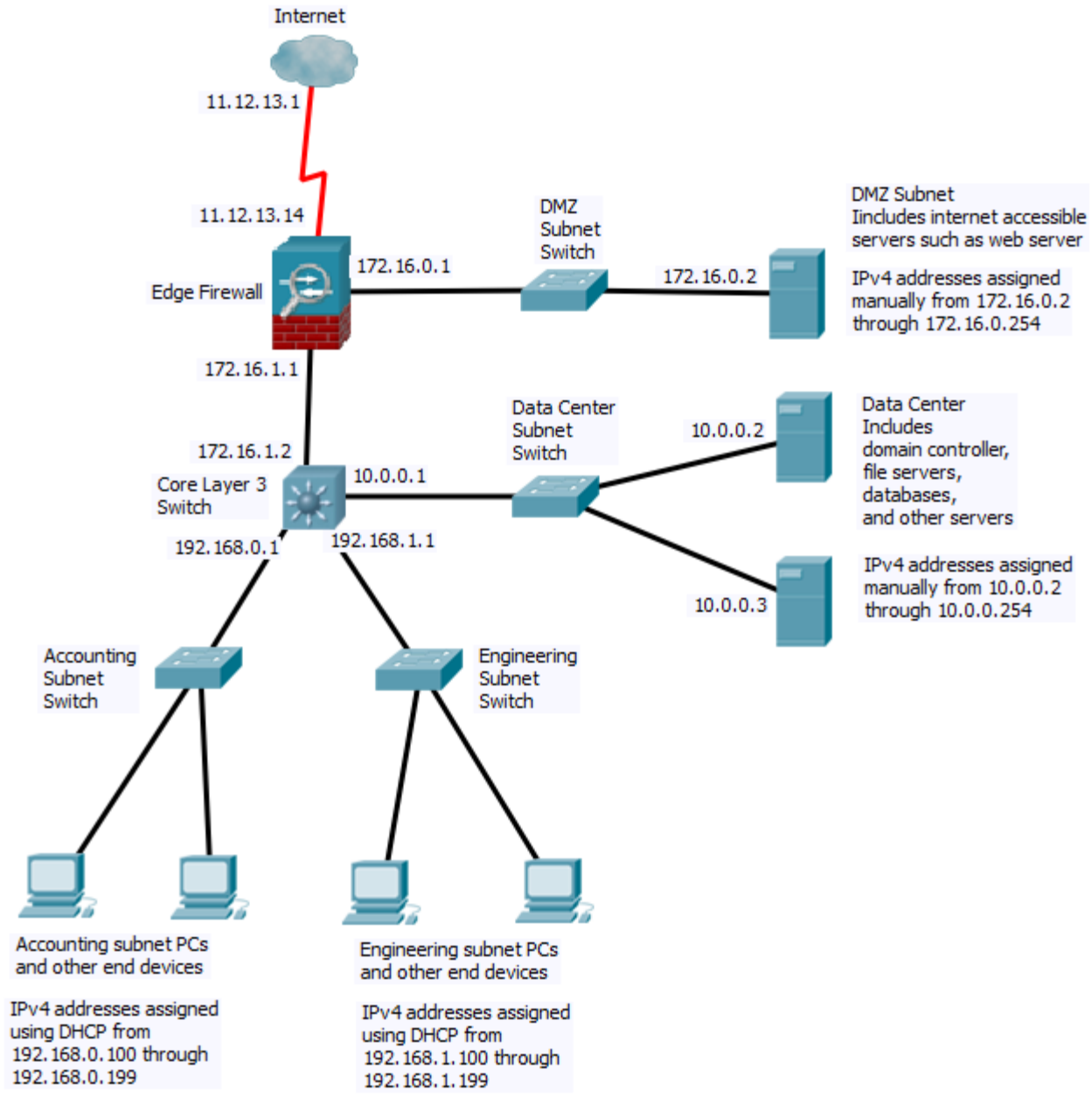
Chapter 10 Instructor Comments

This chapter could easily be two, or maybe even three separate chapters. An incredible amount of information here. And the length of these comments reflects that!

Segmentation and Subnetting

It feels like the textbook dives into this without any real background, so I'll try to fill that in.

I think the easiest place to start is the Example LAN. Here it is again, as updated for Chapter 8.



When this Example LAN was first presented in Chapter 1, I explained the use of the different subnets very briefly:

We'll talk more about why you create separate subnets in Chapter 10, but the basic reason is to simplify management and to separate devices according to security requirements.

We could connect all of the devices in the accounting, engineering and data center subnets to one single switch and had them form one single subnet. In some ways that would have been simpler. This is known as a "flat network" - it has no subnet structure. Most small networks, like your home network, are flat.

But the apparent simplicity of a flat network becomes a liability as the network grows, and the textbook provides three reasons why larger networks are divided, or *segmented*, into smaller pieces called *subnets.*

The first reason for segmenting a network is to enhance security. We've actually already seen two reasons why this is true.

Back in Week 4, you wrote a report about ARP Spoofing attacks. Those attacks only work within a subnet. In our network, if an engineering computer is compromised, it can at most perform ARP spoofing attacks against other computers in the same subnet. But it can't perform the attack against an accounting computer, or worse, against a server. There are other attacks that are similarly limited by segmenting.

And in the Example LAN post for Chapter 8, you saw how the core router (or in our case, the core multilayer switch), can perform basic packet filtering to prevent devices in the accounting and engineering subnets from communicating directly with each other. This can help prevent attacks and also can slow attacks from spreading.

How a Computer Uses a Subnet Mask

You may have noticed that in your home network, all of the devices have IPv4 addresses that start with the same three numbers (octets). For instance, your computer's IPv4 address might be 192.168.0.116, your printer might be 192.168.0.24, and the default gateway (your router) might be 192.168.0.1.

This is not just a coincidence, one of the requirements on a subnet is that all of the devices in the subnet have IP addresses that come from a single range of IP addresses.

I won't go into all of the reasons why this feature is important, but I will at least mention one.

Remember routing tables from back in Chapter 3? Routers maintain routing tables that tell them where to forward any packet they receive. If IP addresses were simply assigned randomly from all possible IP addresses, a router would have to know where every individual IP address was located, and the routing tables would be enormous! But because devices with similar IP addresses are located in similar places, routing tables only have to keep track of ranges of IP addresses, not individual addresses, keeping the routing tables smaller.

As the textbook says, an IP address is really divided into two portions:

- The first part of the address is the network portion. This is usually referred to as a network or subnet address, but I'll use “network ID” since that's what this textbook uses. The network ID is the same for all devices in a subnet.
- The second part of the address is the host portion. It is different for each device in the subnet.

And this is where the subnet mask comes in. It's entire purpose is to identify which part of an IP address is the network ID and which part is the host portion.

To see how a subnet mask works, we'll need to talk a little bit about binary. Luckily, you won't have to work with it again after this short description.

A subnet mask is a 32-bit number just like an IPv4 address, and it is written the same way, as four decimal numbers divided by periods. But a subnet mask has a very specific pattern – in binary, it always consists of a series of 1's followed by all 0's.

For instance, the subnet mask might be

11111111.11111111.11111111.11000000

Now suppose your IPv4 address, when written in binary, looks like:

11010010.11101010.01010010.11101101

To divide that IPv4 address into network and host portions, you simply divide it in the same place where the subnet mask is divided between 1's and 0's. The first part (where the subnet mask is all 1's), is the network ID, and the second part (where the subnet mask is all 0's), is the host portion.

For instance, repeating the subnet mask and IP address above, the highlighted portion of the IP address is the Network ID, and the un-highlighted portion is the host ID.

11111111.11111111.11111111.11000000
11010010.11101010.01010010.11101101

To be more accurate, we think of the network ID as being the same as the IP address with the host portion set to zero:

11010010.11101010.01010010.11000000

That's really pretty simple, IF you are a computer, or you are writing out an IPv4 address in binary. But most of us don't want to write out IPv4 addresses in binary!

So how do you do this without converting everything to binary? We'll get to that a bit later.

CIDR (Classless Interdomain Routing)

So above you've seen a subnet mask written out in binary. Luckily, nobody ever writes subnet masks that way.

A common way to write a subnet mask is the same way that an IPv4 address is written, by converting that binary to decimal. For instance, the subnet mask above, when converted to decimal notation, is:

`255.255.255.192`

I don't expect you to do the binary conversion yourself, just take my word for it!

This is probably the most common way that subnet masks are written. For instance, this is the way Windows displays your computer's subnet mask (for instance, using the command **`ipconfig`**).

But there's an even easier way to write a subnet mask. Remember that a subnet mask is just a series of 1's followed by a series of 0's? CIDR notation (pronounced “cider”), also commonly known as “slash notation” uses this fact and simply tells you how many 1's are in the subnet mask. For instance, the subnet mask above had 26 1's in it, and we can just write that as:

`/26`

The combination of an IPv4 address and its subnet mask is written together like this:

`192.168.0.116/26`

I think we can all agree that of the three different ways to write a subnet mask, CIDR (slash) notation is by far the easiest. Unfortunately, there's a lot of software (such as Windows) that still insists that you write it out in decimal form. There's a chart on page 500 you can use to convert between the two.

Why Subnets?

I really like my example of subnetting by department much better than their example of subnetting by location. It is true that this was once a more common way to subnet, so you may well run into networks that are still subnetted that way, but it doesn't provide the security benefits of subnetting.

Now you come to a long green section in the textbook titled “Subnetting in IPv4”.

All introductory networking courses teach subnetting. It is the process of taking a large set of IPv4 addresses and breaking into smaller subnets. You may have heard other students talk about “subnetting”, learning this is almost a form of hazing. And like hazing, it's completely unnecessary. There is actually very little subnetting of the type taught in these classes done in the real world, and none of it is done by anyone who has only taken an introductory networking course. That's because the hard part of subnetting is not even the math, it's determining how and why you want

to subnet in a particular way. And that topic is way beyond an introductory course.

So you can safely skip everything in the green box that runs from page 495 to 499.

You aren't completely off the hook though, you still need to be able to determine the network ID of an IPv4 address from its IPv4 address and subnet mask. This is a common task when troubleshooting networks, and you'll get practice doing it in this week's written assignment.

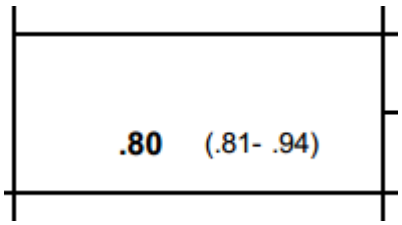
You could learn how by reading through everything in the green box. But you'll be converting IPv4 addresses to binary and other distasteful tasks. Nobody really does it that way!

The easiest subnets to work with are /24 subnets, where the subnet mask is **`255.255.255.0`**. This is what you will almost certainly have on your home network. In a /24 subnet, the network ID is the first three "octets" of the IPv4 address, and the host portion is the last octet. For instance, in the IPv4 address **`10.11.12.13/24`**, the network ID is **`10.11.12.0`**, and the host portion is **`13`**. Doesn't get much easier than that.

For subnet masks /25 and longer an easy way to identify subnets is the subnet chart in Canvas. You'll want to open that chart and follow along here.

The purpose of this chart is to identify the range of IP addresses associated with every possible subnet having a subnet mask between /24 and /30. Every rectangle in the subnet chart represents one possible subnet.

Across the top of the chart are the different subnet sizes, identified by CIDR (slash) notation. In each column below the size are all possible subnets of that size. For instance, in the column for **`/28`**, you'll find the following rectangle:



The number in bold is the last octet of the network ID. The numbers in parentheses show the range of all usable IP addresses in that subnet. Though not shown in the chart, the broadcast address for the subnet is always one address larger than the end of that range. So for this subnet, the broadcast address is **`.95`**.

As an example of how to use the chart, suppose your computer's IP address is **`192.168.25.179`**, and your subnet mask is **`255.255.255.192`**.

First, look at the chart on page 500. You'll see that the CIDR notation for this subnet mask is **`/26`**. Next, look down the column for **`/26`** until you find a range that includes the last octet of your IP address, **`179`**. That rectangle looks like this:

.128

(.129- .190)

From this, you can read off the network ID for your subnet to be **192.168.25.128**.

The broadcast address is one address past the last usable address, so it is **192.168.25.191**.

Finally, for many organizations the standard way of assigning IP addresses to routers is to assign the first IP address in a subnet to the default gateway for devices in that subnet. This makes it easy to know what the default gateway should be for any computer. If that is done in your organization, the default gateway for your computer is the first usable address in the subnet, **192.168.25.129**.

You'll get to practice with this in the written assignment this week.

Supernetting

Supernetting is relevant to advanced router configurations beyond anything in an introductory course. It's rather silly to teach it independent of it's purpose, so you should feel comfortable skipping this section.

Subnetting in IPv6

This section is just about the right length. The theory of subnetting is the same in IPv6 as in IPv4, but the usual practice is that all subnets are /64. In fact, some equipment even assumes that subnets are never smaller than /64. There may be larger supernets, but again, that's really not relevant at this point.

Virtualization

No you are not crazy, there is absolutely no connection between this topic and the previous topic. Nor as you will find, is there any connection between this topic and the rest of the chapter. It's a fine topic to cover, but a strange place to cover it.

This is also a good place to provide a warning – one of the difficulties for newcomers to IT is that some words are used in different contexts to mean completely different things. There is probably no better example of that than the word “virtual”. In the next few pages of this chapter, you will

see this word used in three completely different contexts. Don't be confused into thinking that there is any connection between the three, there isn't.

I have to quibble with their list of disadvantages on pages 508 to 509.

The first disadvantage is “compromised performance”. Sure, if you put 10 virtual machines on a server that used to do one thing, it will run slower. But given equivalent hardware resources, virtualization improves performance because all of the virtual machines share the same set of resources, so when one isn't using them, another can.

The third disadvantage is “increased licensing costs”. It is true that licensing costs usually increase, but that cost is usually more than made up for by decreased hardware costs. Virtualization almost always leads to decreased overall costs, which is the primary advantage.

The fourth disadvantage is “single point of failure”. Again, if you put ten virtual machines on one physical server, then yes, that's a single point of failure. But when configured correctly, you can use fewer hardware resources and get higher reliability with virtualization.

Truly, the only significant disadvantage is complexity. Transitioning from a network where all servers are separate physical machines to a network where they all share the same hardware resources as virtual machines is not a simple task, and you'll want outside expertise at least to guide the process.

Virtual Network Components

This is a continuation of the previous topic. Not only can computers be virtualized, but so can switches and routers. There's far more detail here than necessary in an introductory course, but it's a fine description.

At least it's fine until we get to page 517.

VRRP and HSRP

The authors include this topic in their discussion of virtual network components. Other than the fact that the “V” in VRRP is short for “virtual”, there's no connection at all between this topic and the use of virtual machines. None.

One topic I wish this book talked about more directly is the need for redundancy in networks. Network downtime is very expensive, and it's often much cheaper to have redundant systems than it is to pay for the downtime when something eventually fails.

Some examples:

- There may be multiple connections between devices, so that if one NIC fails, there's another one that can do the job.
- Networking equipment often has two power supplies, each one connected to a different

electric circuit. If a power supply fails, or a circuit breaker breaks, there's a backup that takes over instantly.
- You may have two devices that provide the exact same function, so that if one fails, the other can continue to provide that function.

This third example is where VRRP and HSRP are used. Consider the Core Router in the Example LAN. If it fails, essentially all internal traffic will stop. So you might have two Core Routers that are wired and operational. But the PCs can only have one default gateway, so how can this work?

The answer is that the two routers share the IP address of the default gateway. This leads to some complications, and VRRP and HSRP are protocols that the two routers use to manage those complications, such as coordinating which one gets to use the default gateway IP address at any time.

Again, those protocols have nothing at all to do with virtualization or virtualized network components.

SDN (Software Defined Networking)

Unfortunately, the authors have also confused this topic with virtualization. It can certainly be used with virtualization, but it can also be used with physical devices. Unfortunately, even a short explanation of this topic is beyond an introductory networking course. I would guess they felt obligated to mention the topic, because it is a current hot topic, or at least a current buzzword, but I'd just recommend you skip this section.

VLANs and Trunking

The “V” in VLAN is short for virtual. But this is yet another different use of the term virtual, it has nothing to do with virtualization and it has nothing to do with the V in VRRP.

This is the second most important topic in this chapter, after all of that discussion of subnet masks and network addresses and so on.

Referring back to the Example LAN, all of the Accounting devices are connected to one switch, and all of the Engineering devices are connected to another switch. Imagine that the accounting department is on the first floor, so their switch is on the first floor, while the engineering department and their switch is on the second floor.

Now suppose there's a reorganization, and some of the people in the Engineering department are moved to the first floor. That's a problem, because the computers in their new offices will be connected to the Accounting switch. You are going to need to run cables that can connect their offices to that switch on the second floor. That may not be easy, and the cables may even end up being too long. Even if it all works, clearly, you do not want to have to run new cables every time employees change offices!

This is where VLANs come to the rescue. VLANs allow a single switch to behave as if it is multiple switches, each one on a different subnet. If I had my way, the feature wouldn't even be called Virtual LANs, it would be called Virtual Switches, because that describes the feature much better.

It's actually very simple. As part of the configuration of a switch, you just tell the switch which of its ports are in which subnet. When traffic comes into one port, the switch only looks for the destination port among the ports assigned to the same subnet.

Note that I keep saying “subnet”, where the book will say “VLAN”. Note the equation on the top of page 520. A VLAN is essentially the same thing as a subnet. Most of the time, the only difference between the terms is that we say “subnet” when talking about IP addresses, and VLANs when talking about switch ports.

So that's it, that's essentially all that VLANs are.

There is however a bit of a catch which you may not have noticed. In the Example LAN, there is only one connection between the Accounting switch and the core router. I said that every port on the switch has to be assigned to one subnet. What about the port that is connected to the router? If that is assigned to the Accounting subnet, then devices on the Engineering subnet cannot connect to the Core Router!

The solution to this problem is “trunking”. In reality, each port on that switch can be in three different modes:

- It can be assigned to the Accounting subnet
- It can be assigned to the Engineering subnet
- It can be assigned to both simultaneously, which is referred to as a “trunk port”

Sounds good enough, but that creates yet another problem. Suppose the switch receives a packet on a trunk port. How does it know which subnet that packet is a part of?

The answer is that the router adds a “tag” to every packet it sends to a trunk port. That tag, also referred to as an “802.1Q tag “or simply a “dot1q tag” is yet another header that is added to a packet!

Luckily, that's the last complexity, and that's the basics of VLANs and trunking. It really is quite easy. Don't let the book convince you it is hard, they get into a bunch of details that aren't really necessary in an introductory course, and that can be very confusing if you don't first understood the big picture. But even those details aren't difficult once you get used to it. Truthfully, VLANs are one of the simplest “advanced” topics in networking, as long as you keep in mind the fundamental idea that all you are doing is using a single switch as if it is multiple switches.

You can read more about the use of VLANs in the Example LAN page in Canvas.

STP (Spanning Tree Protocol) and SPB (Shortest Path Bridging)

Again, I have to wonder why the authors put this topic where they did. STP and SPB have nothing to do with VLANs, they are solving a problem that exists with or without VLANs.

If you can understand the problem that STP and SPB are trying to solve, you have 95% of what you need from this section. Study what it says in the second paragraph of this section, comparing what they say with Figure 10-28. Take your time with it.

There are many different solutions to this problem, and each one has its own acronym. I don't know why you would need to know the details of the solutions in an introductory course. You only need to know that when you are deciding which one to use (or when troubleshooting), and those aren't jobs for someone with only an introductory course. In fact, switches will by default choose a solution that works, the only time a network administrator needs to intervene at all is to ensure that the chosen solution is the most efficient.

Switch Configurations

The only thing I want to add to the textbook is that larger organizations or any organization that is particularly concerned about security should NEVER be using unmanaged switches. You lose too many features, including security features such as those that can stop ARP spoofing attacks.

But don't be surprised if you find unmanaged switches in a network, and don't be surprised if no one in IT even knows they are there. When someone decides they need a new port off in some corner of the organization, an unmanaged switch may be put there “temporarily” and then forgot. Or maybe IT never knew about it in the first place. Your job as a network administrator is to search out and destroy them!

Wireless VLANs

This is an interesting topic, but given about as much attention as it should be.

Troubleshooting VMs and VLANs

You've had plenty to learn in this chapter. Feel free to skip this final section.