Networking

Neil Kingdom

October 3, 2024

Contents

1 Introduction

When you think of networking, IP addresses, servers, and running ethernet cables probably comes to mind. As I continue to become more experienced, however, I notice that networks are actually a very natural phenomena - a pattern which appears in mathematics under graph theory, under sociology and human interactions, and likely other fields with which I'm less knowledgeable on. Although we'll be specifically looking into the nitty gritty details of how the internet was built, it is helpful to remember that networking is all about maintaining a series of connections and being able to manage interactions between devices. Being able to visualize problems at various levels of abstraction will increase your problem solving abilities tremendously.

2 Terminology

Before we really begin, I'd like to start off right away by defining certain terms that will arise often as we speak about networks, the internet, etc. Most of the definitions should be quite straight forward.

- **Network:** A network is a system of connections which are linked by some form of media. In computer science, networking refers to a system of connections between devices (often computers or servers) via wires, light, or wave transmissions.

- **Internet:** The internet is a world-wide web (where the term www comes from in web addresses) of connections between routers. The internet is how you are able to connect to anyone or anything outside of your local household. Your Internet Service Provider (ISP) is in charge of linking your household to the internet.

- **Intranet:** An intranet is similar to the internet, but on a much smaller scale. This is often found within large companies where employees might need access to resources about the company, but may still have restricted access to the internet. For example, when I worked at CRA, we had an intranet where we could search for information about particular branches or sects.

- **Extranet:** A less common form of network are extranets. An extranet is simply an intranet which has been extended to public use. For instance, a company may want to keep certain things within their intranet private to employees, but other data may be acceptable for the public to view. The data that is acceptable for the public to view would belong to the companies extranet.

Often times, we talk about networks in terms of their size. This helps us get an idea about the scale of the network. The following terms apply to network areas:

- **Peer to Peer (P2P):** Peer to peer is the smallest type of network possible – a connection between two devices. P2P is a specification of the broader Personal Area Network (PAN). For example, a phone connected to your car radio via bluetooth would be considered a PAN.

- **LAN:** A very common term in networking, LANs are Local Area Networks. Your household is considered a LAN.

- **MAN:** Metropolitan Area Networks often refer to a neighbourhood or large building like a hotel or retirement home.

- **WAN:** Wide Area Networks are very large networks. Typically, WAN refers to the internet.

- **SAN:** Storage Area Networks are often used to refer to server centers. A SAN is a connection of storage devices.

- **CAN:** Campus Area Networks refer to a campus' network like the one you may or may not be attend-

ing!

Note: Prepending a 'W' to any of the abbreviations mentioned above means "wireless". For example, a WLAN is simply a wireless local area network.

- **Host:** The formal definition of a host is any computer or device connected to a network. I do not really like this definition though. In computer science, the host often refers to the primary provider of data. Servers are almost always hosts because they provide data to the clients that connect to them.

- **Client:** The client always refers to the reciever of media. The client can request data, and often times send data, however, they are typically not the primary provider of the data.

- **Server:** A physical device (often located in a remote location such as a special building) which stores data. Servers are most often responsible for storing information about web pages or customer data.

- **Node:** The term node (sometimes also called a member) is any device which belongs to a group or cluster. The term node is intentionally vague, as it may refer to a laptop, a PC, a server, a gateway, or any other thing which belongs to the cluster.

- **Cluster:** The term cluster (sometimes called a group) is a group of nodes which are connected in some fashion. The type of connection doesn't matter, so long as the nodes have a method of communicating with one another.

3 Network Topology

A topology diagram is used to help network engineers understand the details of how a particular network is configured. Typically, a topology diagram has both a physical and logical counterpart. The physical diagram shows the network engineer where devices reside physically within a building. This might include room numbers, number of devices, etc. The logical diagram is usually the one we care about most. It shows an abstract overview of the connections between each device. This helps us see how the network is split, which devices are connected to which, and how the network connects to the internet if applicable.

Since we wont really be concerning ourselves with physical topology diagrams in this paper, I will show you the various symbols that are used to represent various devices and media.

3.1 Topology Types

When searching for network topology, you'll frequently come across the various common topology types that we find in both networking and graph theory. A lot of these structures appear in programming as well, funny enough. Here is the list:

- **Point-to-Point (P2P):** We've already briefly discussed P2P networks. They are the smallest type of network consisting of two nodes which have a direct connection to one another. To reuse the same example, Bluetooth device connections can be a great example of a P2P network topology.

- **Bus:** Busses are probably most commonly found when looking at electrical circuits. Another example of a bus off the top of my head is the Blackberry QNX microkernel. In a bus, we usually have some central physical or logical line which other processes or devices (or in this case nodes) can connect to. Nodes do not have direct connections to other nodes, but communications can still take place over the central bus which unites them.

- **Ring:** A ring is pretty self-explanatory. It is a network which involves 3 or more nodes, each connected such that they form a "circle" (at least an approximation of one) or ring. Side note, a ring is a type of data structure found in programming, such as a ring buffer or circular linked list.

- **Tree:** Another topology type which has roots in programming data structures. A tree is a top down heirarchy which resembles a tree. Each node with children is considered a parent, and nodes which do not have children are considered to be leaf nodes. Sibling nodes are nodes which appear on the same heirarchical level relative to another node, but which do not have a direct connection with that node.

- **Star:** A star topology is formed when you have one central device such as a server and many other nodes which are connected to it, but not to each other. Personally, I think star networks look more like snowflakes, but to each their own.

- **Mesh:** A mesh network does not necessarily have as concrete a defintion as the others, but I would consider mesh networks to be ones in which you have nodes which relay messages on behalf of other nodes. A mesh network is usually very interconnected by having more than one connection per node.

- **Hybrid:** Hybrid networks are what we tend to actually deal with in real life. The internet is the perfect visualization of a hybrid network, because it is composed of every other topological structure that we've looked at.

3.2 Broadcast, Multicast, Unicast, and Anycast

Within a network topology, there are various ways that a single node is able to communicate with other nodes in the cluster. The four primary methods of communication are the ones listed above (broadcast, multicast, unicast, and anycast). Here is a list of each and the differences between them:

- **Broadcast:** Broadcasting is when a node sends out a message to every other connected node within the cluster. This is great for Data Distribution Services (DDS) where we have a publish-subscribe model. A publisher sends out one or more messages to everyone on the network, and the members which are subscribed to said message type consume it, and those who aren't simply ignore the message.

- **Multicast:** Multicasting, unlike broadcasting, sends data to specific devices on the network. Although the node may be connected to several other nodes, it may only want to send data to a select few. Multicasting is more specific than broadcasting in that we choose who will receive the messages.

- **Unicast:** Even more specific than multicasting, unicasting sends data to a single recipient. No other members in the group will receive that message.

- **Anycast:** Anycast is a bit more niche than the rest, although it still has its usages for sure. In anycast, we send to a single recipient, similar to unicast, but unlike unicast we don't select who the message gets sent to - we simply send it to the "nearest" node. The word "nearest" usually refers to the node which is the fewest hops away.

4 The OSI and TCP/IP Models

A network connection sends data as a series of packets of information encoded as bits. Although computers can process raw bits with ease, humans have a much harder time reading and understanding simple bits. This is why we create protocols, which are formal defintions of the data layout. For instance, we might say that field x is represented by bits 0-7, and field y is represented by bits 8-23. Protocols make up the internet as we know it. There are many protocols, each of which serve a particular purpose, as we'll come to see. In order to better categorize these protocols, we divide them into logical layers which are effectively based upon their level of abstraction. There are two primary models which are used to categorize protocols. These are the OSI model and the TCP/IP model. Luckily the TCP/IP model is mostly just a simplification of the OSI model, so if you can memorize the OSI model, the TCP/IP model should be quite easy to remember as well.

Here is the layout:

// TODO: Make a table

OSI TCP/IP ——————————————— Application Layer Presentation Layer Application Layer Session Layer ——————————————— Transport Layer Transport Layer ——————————————— Netork Layer Network Layer ——————————————— Data Link Layer Network Access Physical Layer Layer ———————————————

13 Simplex, Duplex, Half Duplex, Full Duplex, and Multiplexing

Boy oh boy, when does terminology ever become easy? My aim is to help clarify all of the above terms in a very simple to understand way. The reason I've grouped these terms is because a) they are related, but also b) because you'll hear them thrown around a lot and I wouldn't want them to get muddled together in your head. First, let's separate 'multiplexing' from the rest of the list and focus first on what a simplex, duplex, half duplex and full duplex are. In fact, a half duplex and full duplex are just sub-categories of a duplex, so we've already simplified to focusing on simplex vs duplex.

The terms simplex and duplex first began with two way communications between radios. A simplex is a radio which uses one band i.e., one frequency, for receiving and transmitting the audio signal. If you've ever held two cheap radios, you'll know that as the button is pressed on radio a to talk, radio b can only

listen - not speak, and vice versa. This is a simplex, and its called that because it's the simplest mode of receiving and transmitting data. In a duplex, we increase the channels to two. This is easy to remember because the 'du' in duplex stands for duo i.e., two. There are two kinds of duplex communications though, as I mentioned. These are half duplex and full duplex. It's quite simple - even though half duplex systems have two channels, they are still limited to only using one channel at a time. This means they can receive data or send data, but not do both at the same time. Of course, a full duplex is when you *can* do both at the same time.

I've segregated multiplexing from the list because unlike the terms I've defined thus far, multiplexing is not really a concept that applies to radios, and it mostly only applies to network switches. Multiplexing, unlike a duplex, can have more than just two channels or bands or whatever term you want to use. With multiplexing, we take multiple input signals and combine them into one signal (this is sometimes called muxing). A process called demuxing decouples the combined signal back into the appropriate output signals. An example is when we have multiple phone calls transferred through the same wire, but that each reach their intended destinations.

14 IP Addresses

Amazing that we've come so far and not yet discussed IP addresses! Hopefully everyone reading this has some general clue as to what an IP address is. The word address should make it clear that it is a unique identifier which identifies a destination. IPv4 is currently the most commonly used protocol for sending or receiving data across the internet, though IPv6 seems to be becoming more heavily utilized. IPv4 addresses are composed of four fields, each separated by a period, and each of which are 8 bits long. Since 2 to the power of 8 is 256, this means that each field can support a range of numbers between 0-255. With four fields we have four to the power of 256 possible combinations, which comes out to nearly 4.3 billion unique addresses. The smallest possible address is 0.0.0.0, and the largest is 255.255.255.255. Unfortunately, the designers of IPv4 made a small little oopsie and didn't realize there would eventually come a time where more than 4.3 billion devices would be connected to the internet. We'll discuss the workaround they came up with in a bit, but suffice to say that IPv6 was created to alleviate the problem of IPv4 address exhaustion.

IPv6 went a wee bit overboard when trying to compensate for IPv4s shortcomings and decided to make the new standard have 8 fields this time, each encoded as 16-bit hexadecimal numbers (i.e. double the bit count for each field in IPv4). This gives us a range of IP addresses so astronomically large that it far exceeds the amount of estimated stars in the universe.

14.1 Subnets and Subnet Masks

When we divide a network into smaller networks, we call the smaller networks subnetworks (pretty simple right)? But the question is, how do we divide a network? The answer is by using subnet masks. If you know some bit twiddling tricks, you are probably familiar with the concept of a mask, but if not I'll attempt to explain. A bitmask is a series of bits which shadow another series of bits by performing a bitwise AND operation. In a bitwise AND operation, any bits which are set to 0 within the mask will force any bits which are 1 in the original series of bits to become 0 (bits which were already 0 remain 0). Any bits within the mask which are 1 will cause the original set of bits to remain the same. Assume that we have the IP address 192.168.4.89. Now assume that we apply the mask 255.255.0.0 to this IP address. Since 255 in binary is eight 1s the IP address would become 192.168.0.0. A subnet mask is used to determine which bits will be reserved for the actual subnet groups and which bits will be reserved for individual IP addresses for devices on those subnets. To put it another way, the subnet mask 255.255.0.0 reserves the left-most two bit fields for subnets and the right-most two bit fields for individual IP addresses. You might have subnet 192.168 followed by device X which has the IP 2.89. Another device Y might be on the same subnet (192.168) followed by its IP 3.76. Finally, we might have device Z on a different subnet 122.122 and with IP 187.23. In reality we don't separate the subnet portion from the IP address portion, so X would be 192.168.2.89, Y would be 192.168.3.76 and Z 122.122.187.23. It should also be noted that this will probably not matter

much to you unless you are studying to become an IT person who manages large networks. A subnet mask of 255.255.0.0 will give you 65025 possible subnetworks, each of which can have 65025 unique addresses (that's a lot). If you need to extend the number of subnets, we can make the mask 255.255.255.0 which gives us 255 to the power of 3 (16,581,375) possible subnets each with 255 unique IP addresses. Note how the number of available IP addresses is inversely proportial to the number of available subnets.

14.2 Reserved IPv4 Addresses

Multiple address ranges are reserved by the Internet Assigned Numbers Authority (IANA) for special purposes. Perhaps the most notable of these reserved addresses is the local loopback address. The loopback address is an address which routes traffic back to the host. The DNS name for the loopback address is called localhost. For me personally, I mostly associate the loopback address with serving local content that only I can access. For instance, hosting a website on localhost will allow you (and only you) to access and view the site, usually for testing purposes. You could really host anything here though. The IPv4 address 127.0.0.1 is reserved as the loopback address. As stated, you can also enter localhost into the address bar, although just note that there will be a very small delay in the response due to the fact that this will route to the DNS server (which could potentially be far away) and then return 127.0.0.1. A loopback adapter is a file in Unix systems which acts as a physical device. It emulates a NIC, but is sandboxed so that it cannot connect to anything which is exterior to the host. The reserved IP 0.0.0.0 will achieve the same purposes as the loopback address because it listens for traffic on all available NICs, including the loopback adapter.

15 Ports

Imagine this. You have two applications running on two separate machines. The application running on device A wants to send data to the application running on device B. How does this work in practice? Well, we know that we'll need to send the data to device B's IP address, that much is obvious. However, a problem arises, which is that device B does not know what to do with the data once its received. A port is 16-bit number which uniquely identifies an application or service running a device. If were told which port to send the received data to, then device B can pass along the data it receives to the specified port. Applications create sockets, which establish a connection with a specific port number. The application then listens on that socket for any incoming traffic and processes it accordingly. One of the things that a firewall does is that it may block certain ports. For example, we might blacklist port 22 if we want to prevent anyone from SSHing into our server.

15.1 Reserved Ports

16 Gateways

16.1 The Default Gateway

16.2 Routing Tables

16.3 Hops and Time to Live (TTL)

16.4 Network Address Translation (NAT)