Set 10. Routing with Redundancy

Skill 10.01: Describe the purpose of IP packets

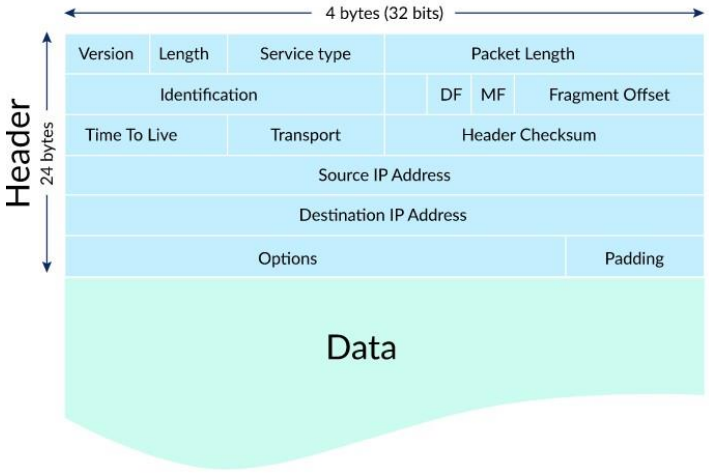
Skill 10.01 Concepts

On the massive network known as the Internet, computing devices send all kinds of messages to other computing devices. A message might be a tiny ping to check if another device is online or a message could be an entire webpage.

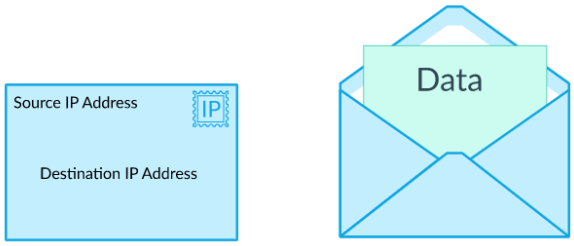
But there's a limit to how large a message can be, since there's a limit to how much data can be reasonably transmitted at once by the physical network connections between devices.

That's why many networking protocols split each message into multiple small packets referred to as **IP packets**. The Internet Protocol (IP) describes the structure of the packets that whizz around the Internet.

Each IP packet contains both a header (20 or 24 bytes long) and data (variable length). The header includes the IP addresses of the source and destination, plus other fields that help to route the packet. The data is the actual content, such as a string of letters or part of a webpage.



You can think of IP packets like postal letters: the header is the envelope with all the routing information that's needed by the post office, and the payload is the letter that's read only by the recipient.



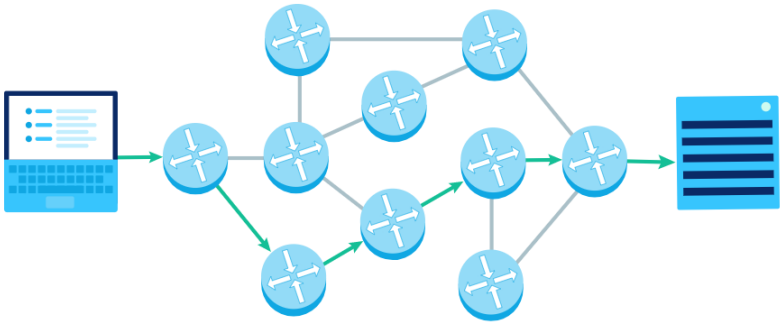
Just like the postal system routes postal letters around the world, the Internet Protocol routes IP packets around the Internet. Watch the video below to learn how IP packets can be used to send content on the Internet.

https://www.youtube.com/watch?v=AYdF7b3nMto&feature=youtu.be

Skill 10.02: Describe the Internet Protocol for routing IP Packets

Skill 10.02 Concepts

In the Internet Protocol (IP), computers split messages into packets and those packets hop from router to router on the way to their destination:



Let's step through the process of routing a packet from a source to a destination.

Step 1: Send packet to router

Computers send the first packet to the nearest router. A router is a type of computing device used in computer networks that helps move the packets along.

You likely have a router in your home or classroom right now, and that's the first stop for your current computer's packets.

Step 2: Router receives packet

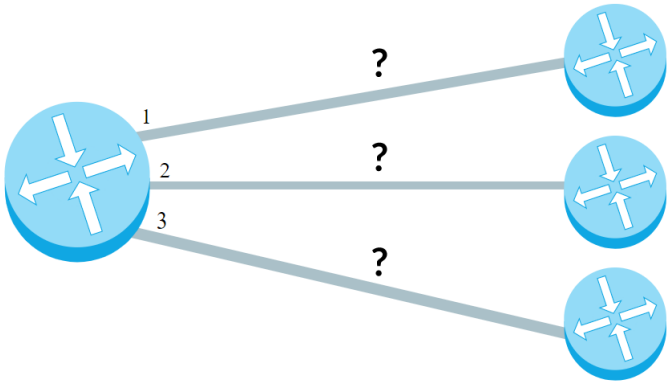
When the router receives a packet, it looks at its IP header. The most important field is the destination IP address, which tells the router where the packet wants to end up.

IP header

Field	Content
Source IP Address	`216.3.192.1`
Destination IP Address	**`91.198.174.192`**
Version	`4`
Time to Live	`64`
... plus 10 more fields!	

Step 3: Router forwards packet

The router has multiple paths it could send a packet along, and its goal is to send the packet to a router that's closer to its final destination.

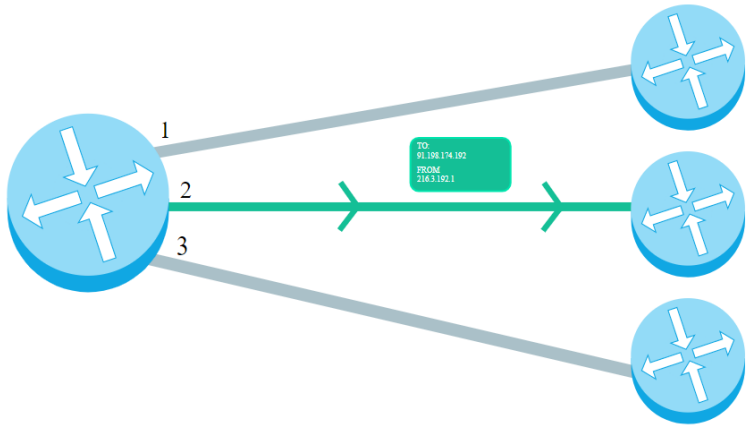


How does it decide? The router has a **forwarding table** that helps it pick the next path based on the destination IP address. That table does *not* have a row for every possible IP address; there are 2^{32} possible IP addresses, and that's far too much to store. Instead, the table has rows for IP address *prefixes*.

IP address prefix	path
91.112	#1
91.198	#2
192.92	#3

IP addresses are hierarchical. When two IP addresses start with the same prefix, that often means they're on the same large network, like the Comcast SF network. Router forwarding tables take advantage of that fact so that they can store far less information.

Once the router locates the most specific row in the table for the destination IP address, it sends the packet along that path.

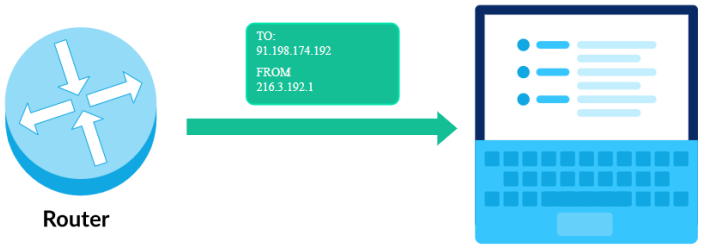


Step 4: Final router forwards message

If all goes well, the packet should eventually arrive at a router that knows exactly where to send it.

IP address prefix	path
91.112	#1
91.198.174.192	Direct
192.92	#2

The router can now send the message to the destination IP address, which may be a personal computer or a server.

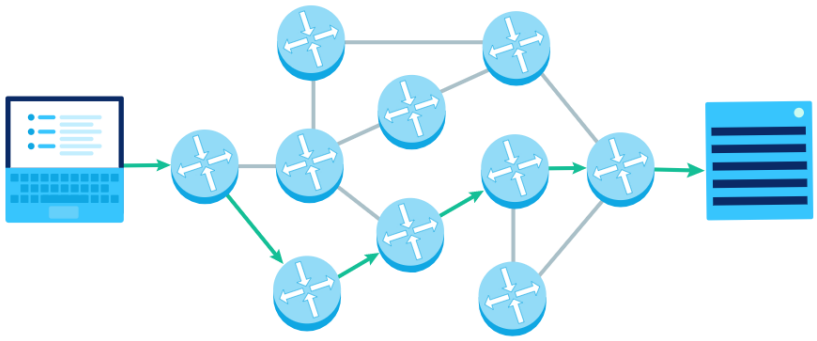


[Skill 10.02 Exercise 1]

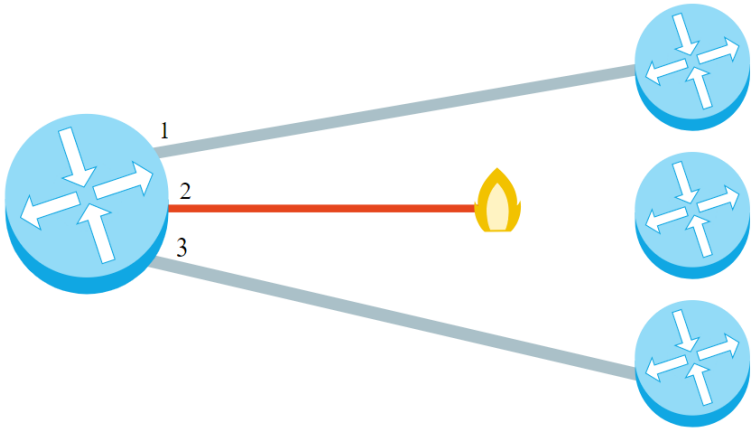
Skill 10.03: Explain redundancy as it applies to the Internet

Skill 10.03 Concepts

In the Internet Protocol (IP), computers split messages into packets and those packets hop from router to router on the way to their destination:

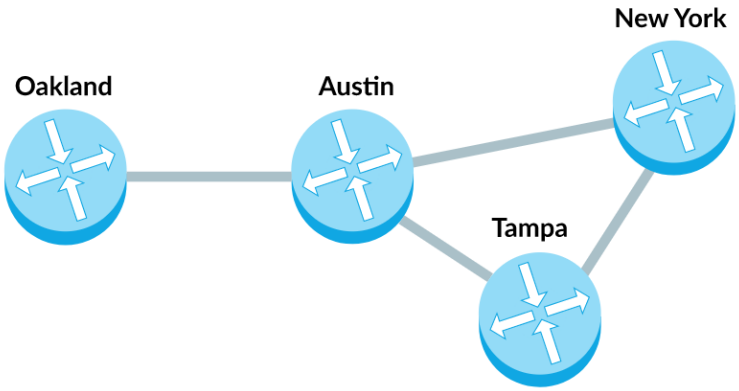


What happens if a network path is no longer available, like due to a natural disaster physically destroying it or a cybercriminal hijacking it? Is the packet doomed to never reach its destination?



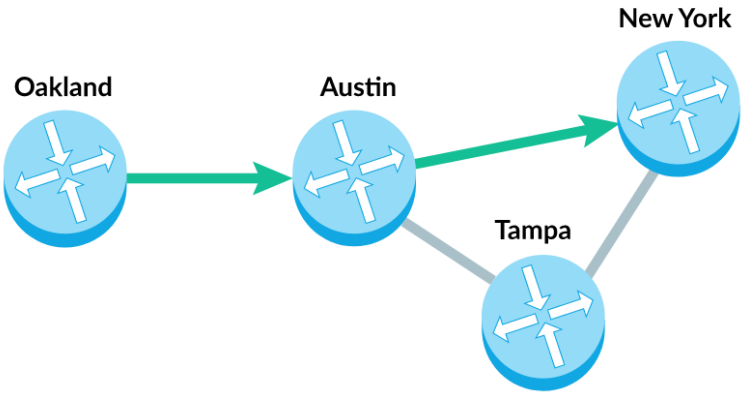
Fortunately, there are often many possible paths a packet can go down to reach the same destination. The availability of multiple paths increases the **redundancy** of a network.

Consider this simplified network connecting routers in four major cities:

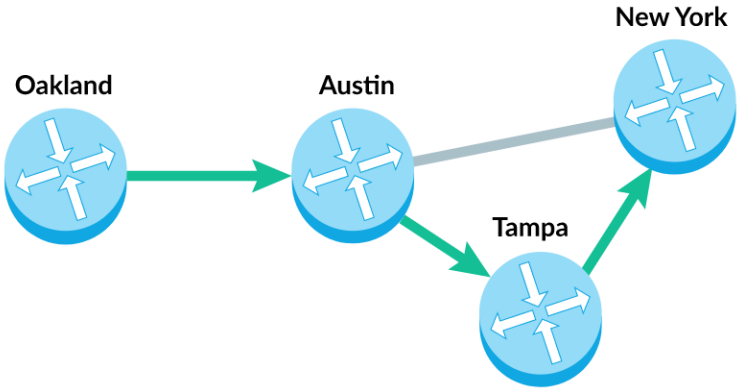


There are multiple paths from the Oakland router to the New York router.

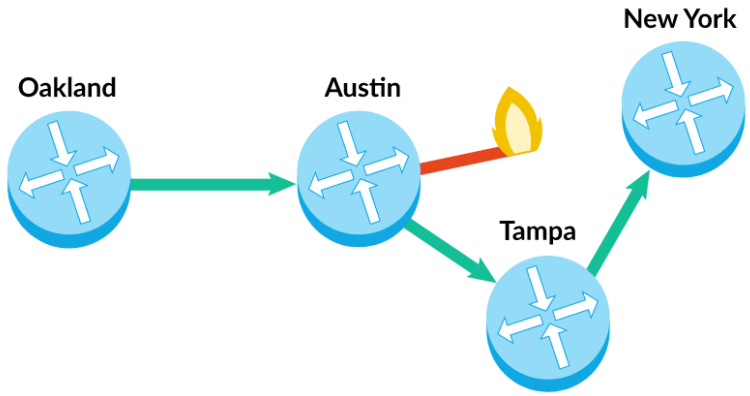
The first and shortest path goes from Oakland to Austin to New York:



A slightly longer path goes from Oakland to Austin to Tampa to New York:



Why is this redundancy so important? If the connection between the Austin and New York router is no longer available, then there's still another way for the packet to reach its destination.



The redundancy of the paths in the network increases the number of possible ways that a packet can reach its destination.

[Skill 10.03 Exercise 1]

Skill 10.04: Explain what is meant by fault tolerant

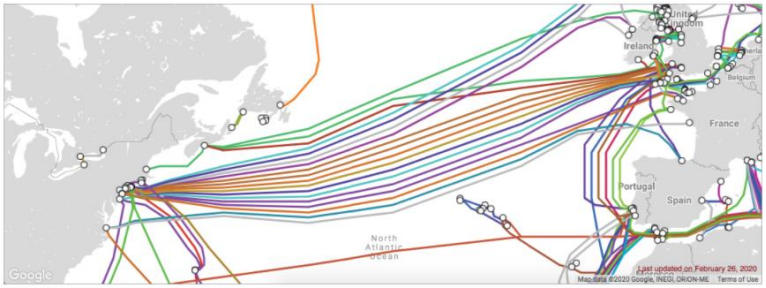
Skill 10.04 Concepts

A **fault-tolerant** system is one that can experience failure (or multiple failures) in its components, but still continue operating properly.

The Internet is a massive and complex system with millions of components that can break at any time—and many of those components *do* break. But as of 2020, nobody has managed to break the entire Internet.

A big contributor to the fault tolerance of the Internet is the redundancy in network routing paths.

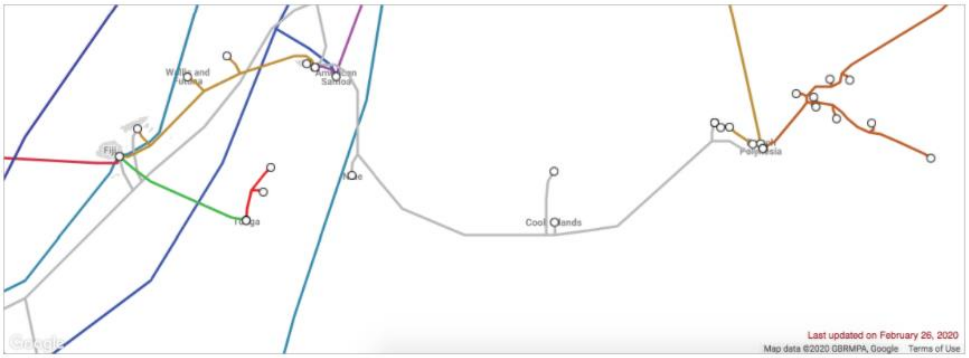
Consider the number of underseas cables connecting the eastern side of the United States to the western side of Europe:



If one of those cables is damaged, there are multiple other cables that can carry Internet traffic over the Atlantic ocean.

Or, to put it another way, there is no **single point of failure** between the coasts. A single point of failure is a component in the system that will bring down the entire system if it fails. When we're trying to make sure a system is fault tolerant, we look for single points of failure and find ways to add redundancy at those points.

Now consider the meager number of undersea cables between these Polynesian islands in the South Pacific:



If a cable is cut between Cook Islands and French Polynesia, how will that affect the Internet on those islands?

In some cases, a cable cut can bring down an entire country. In 2019, a ship anchor dragging along the sea floor cut the cable to Tonga and cut their Internet access off for 11 days!

It doesn't take much to cut a cable. In 2011, a grandmother in the country of Georgia accidentally damaged a cable with her shovel, resulting in all of Armenia losing Internet access for 5 hours.

Cable cuts happen relatively frequently—"around every 3 days", according to networks analyst Stephan Beckert. Most of the time, the average Internet user doesn't even notice when cuts happen and the cable gets fixed up by one of many cable repair ships. When we do notice the cable cuts, that usually means there's a single point of failure and it's time to add redundancy to the system.

Why don't we start off with redundancy everywhere? As you might guess, it's expensive. The underseas cable that connects Tonga to Fiji cost about $30 million, and that's a relatively short cable. When Google installed a high speed fiber optic cable between the US and Tokyo, it cost $300 million dollars.

When it's too expensive to duplicate a resource, it may be possible to find ways for the system to gracefully degrade in the face of failure. During the Tonga outage, satellite providers rushed to provide Internet access. They may not have been able to provide the same speeds as the fiber cable connection, but any Internet connection is better than no Internet connection at all.

Skill 10.04 Exercise 1