# CONTENT DELIVERY

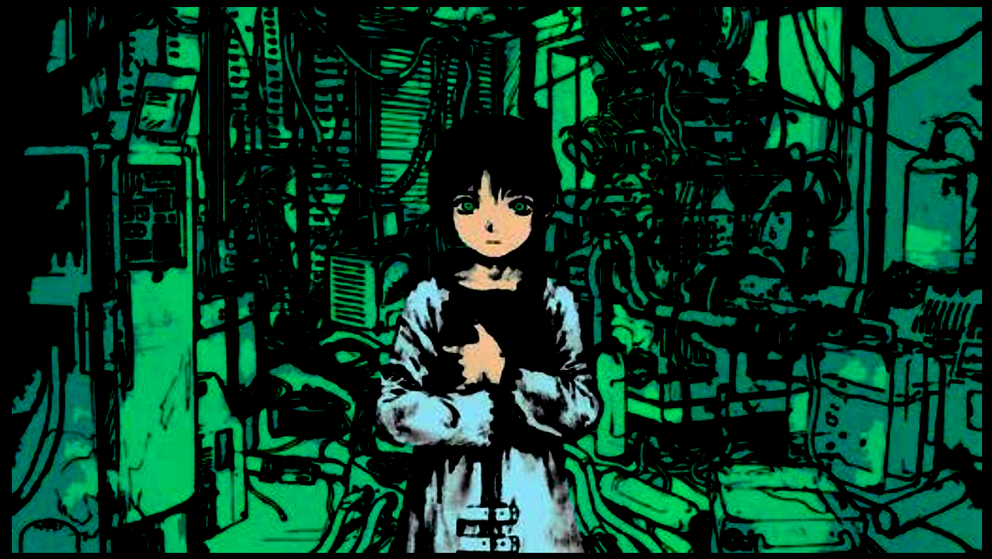
## A Technical Primer on the Internet.

Hardware Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing
- IP (Internet Protocol) Address
- LAN (Local Access Network)

Software Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)
- Transport Layer Security (TLS)
- Certificate Authority (CA)

# What's actually going on?

As much as we like to imagine the internet as an ephemeral, all-connecting ether we tap into through digital devices, it's incredibly physical. The internet is a huge connection of high-transit wires spreading across the globe (and a couple satellites in the upper atmosphere providing cellular connection). When you connect to a server in Japan, you are quite literally facilitating beams of light from your router to Japan and back.



# In the Home

Before the “Internet,” much of this cable-based infrastructure was already in place: the global phone network. This is why early internet was called “dial-up”—to connect, you would “dial” into a special phone service which would communicate bytes of information through tones across a standard telephone line.

The advent of modern broadband cut out this temporary connection and “dial up” process—once you connect to broadband Internet, you stay on a connected, provisioned line through your Internet Service Provider, who maintains the physical cables and infrastructure in your area connecting all of the devices to the global internet network. This long-lived connection is facilitated by two devices:

- A modem, which translates signal from your ISP into an Ethernet connection to either a router (think WiFi connection) or a device.
- A router, which often disperses internet signal over a WiFi network to the local area and can be connected to by multiple devices at once.

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router

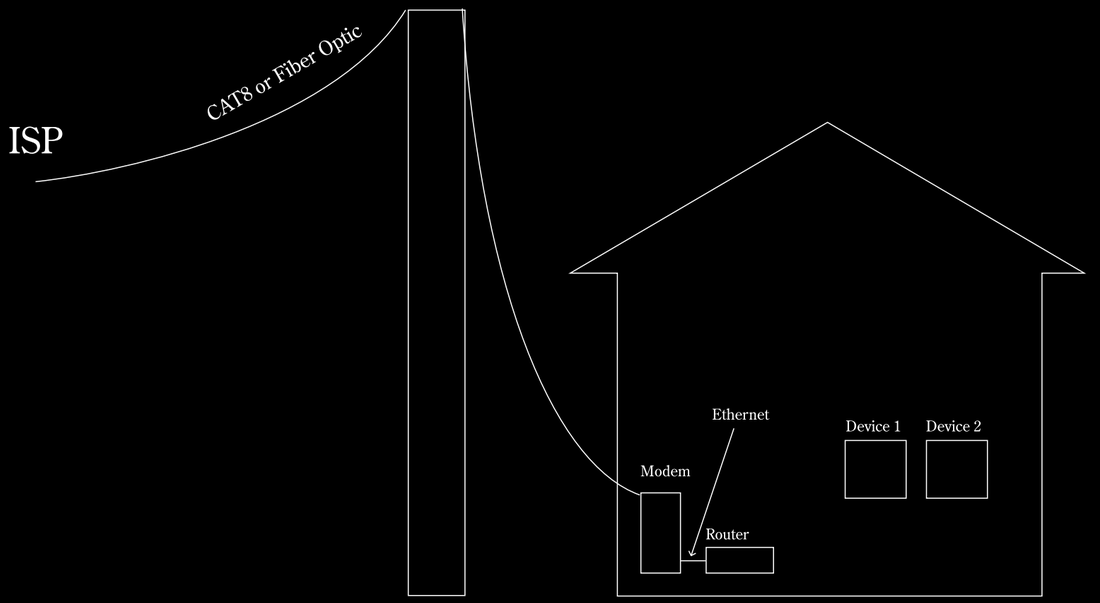
# 1. Hardware

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router

A modem.

A router.

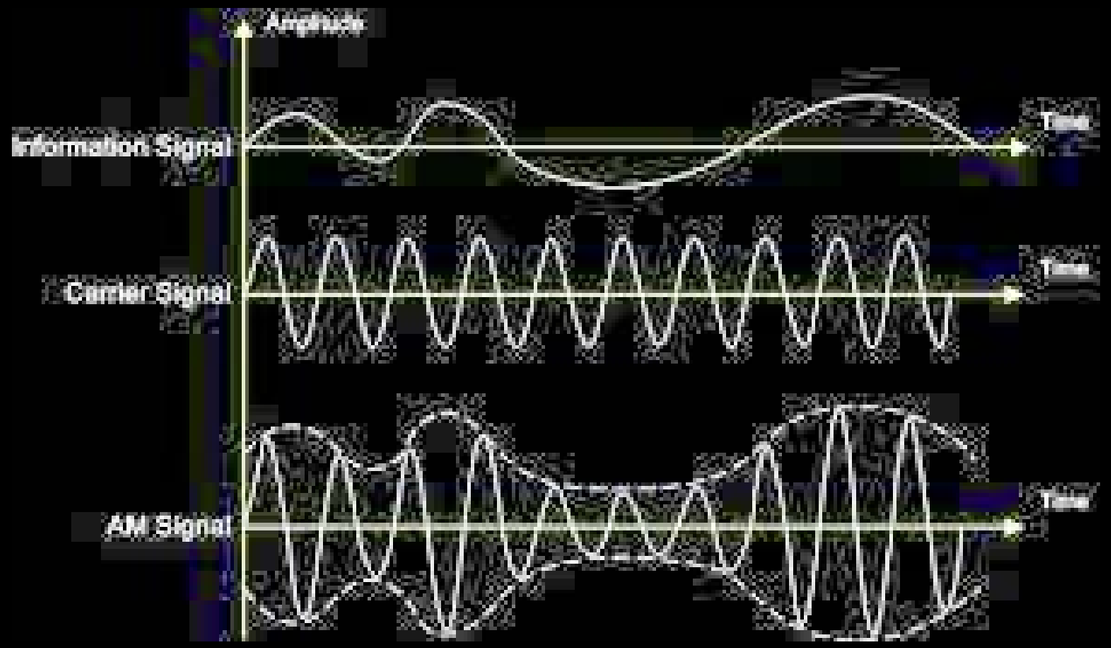
Ethernet cables.

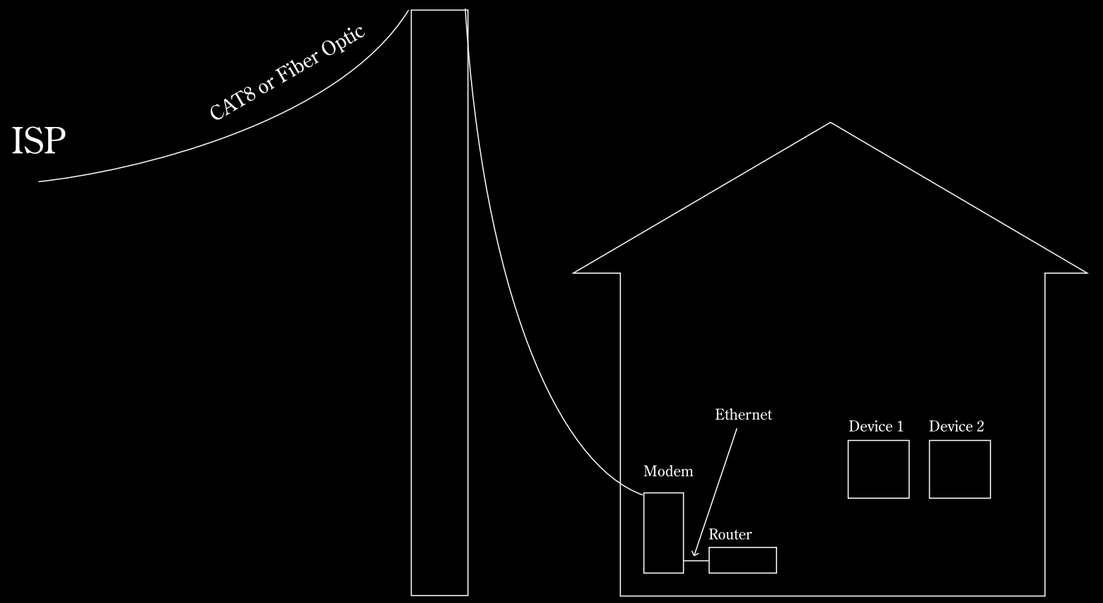


# 1. Hardware

Your router is the device that transmits the local WiFi network to which your device connects to. WiFi (Wireless Fidelity) operates on two different frequencies, which you may recognize from WiFi network names: 2.4Ghz and 5Ghz (Ghz = Gigahertz). That is, 2.4 (or 5.5) billion cycles per second in the radio signal wave. Your router is a high-frequency radio over which your internet connection is transmitted through amplitude modulation, which is then converted to bits on your device:

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation

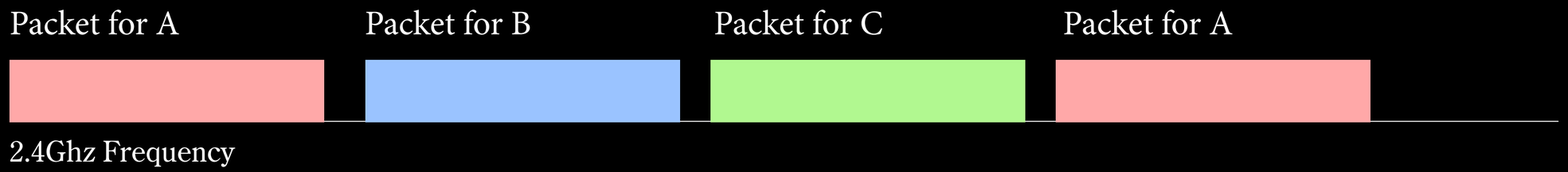


Despite the perception that multiple devices are connected and streaming at the same time on the WiFI network, only a single frequency is actually used, and so this parallelism is actually just rapid switching between broadcasting “frames” to all devices (but each frame only being consumed by a single device). This is facilitated by a complex same-signal scheduling process called time division multiplexing (TDM). To be honest, I don’t know very well how it works. The essence of it is that the signal producer in your device waits until the frequency channel is clear, emits the frame, and then waits a specified time offset before trying to send the next frame.

The idea of “frames”—splitting up a transmission into small pieces—is quintessential to networking and is similar to the packets of the Transmission Control Protocol (TCP), which we will see later.

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing



Within your computer, there exists a device similar to a router called a WiFi card. This little device is a radio much like the router—where the router's card operates in AP (Access Point) mode as a signal emitter, the WiFi card in your computer operates in WiFi mode as a signal receiver. This converts WiFi signal into bits readable by your computer.

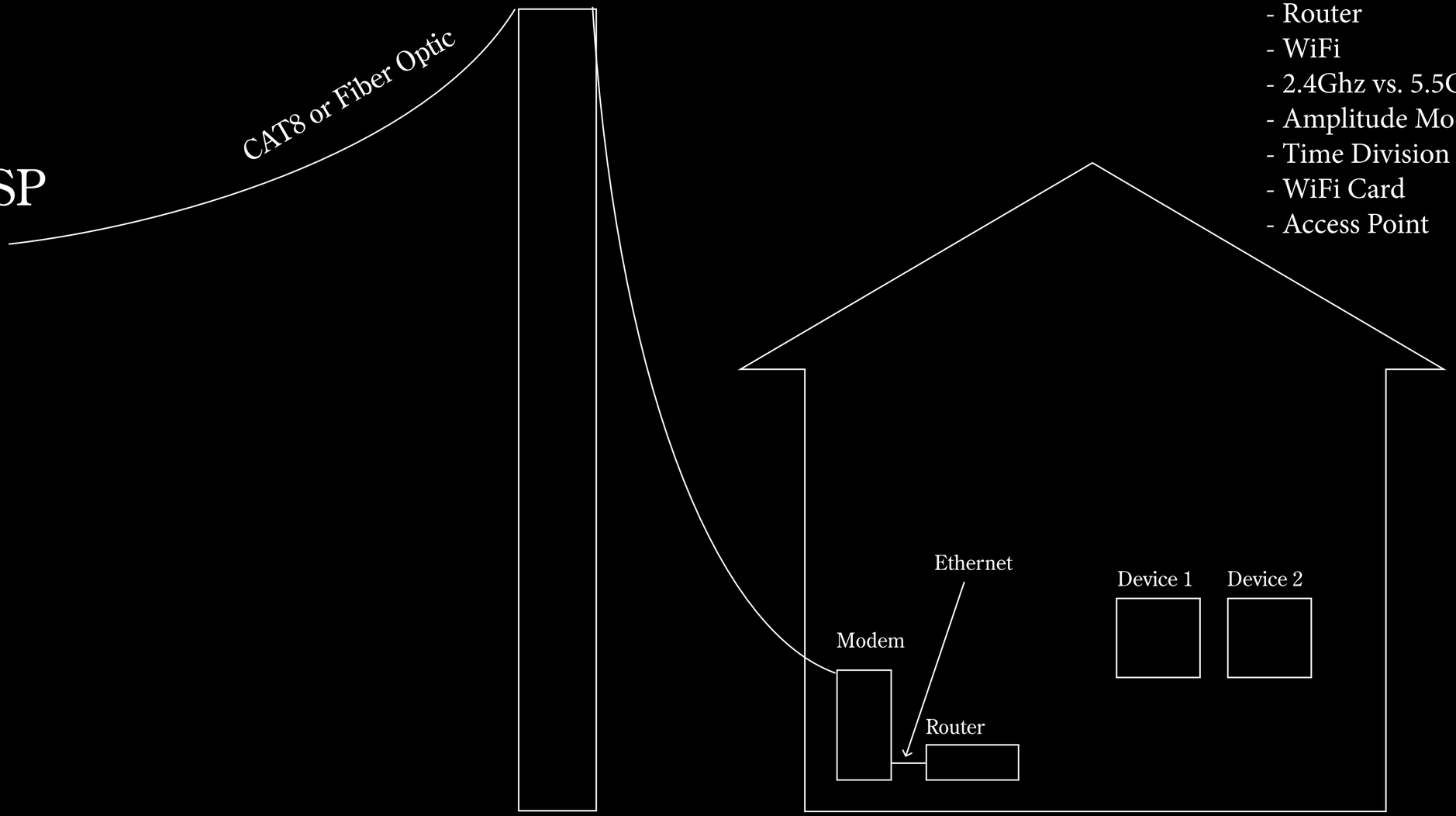
Interestingly, any WiFi receiver can also generally function as an Access Point. For example, this is how your iPhone hotspot works—when you turn this setting on, the internet signal received from your phone's cell receiver is re-transmitted through its WiFi card in AP mode. This is why you can't host a hotspot and turn on WiFi at the same time.

Actually, a lot of electronic emitter / receiver devices are like this. For example, any speaker can function as a microphone and any microphone can function as a speaker—it turns out the electrical circuit involved in emitting signal is often just the electrical / mechanical inverse of that to receive a signal.

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing
- WiFi Card
- Access Point

Let’s zoom out a bit. As a recap, we have the following components:

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing
- WiFi Card
- Access Point



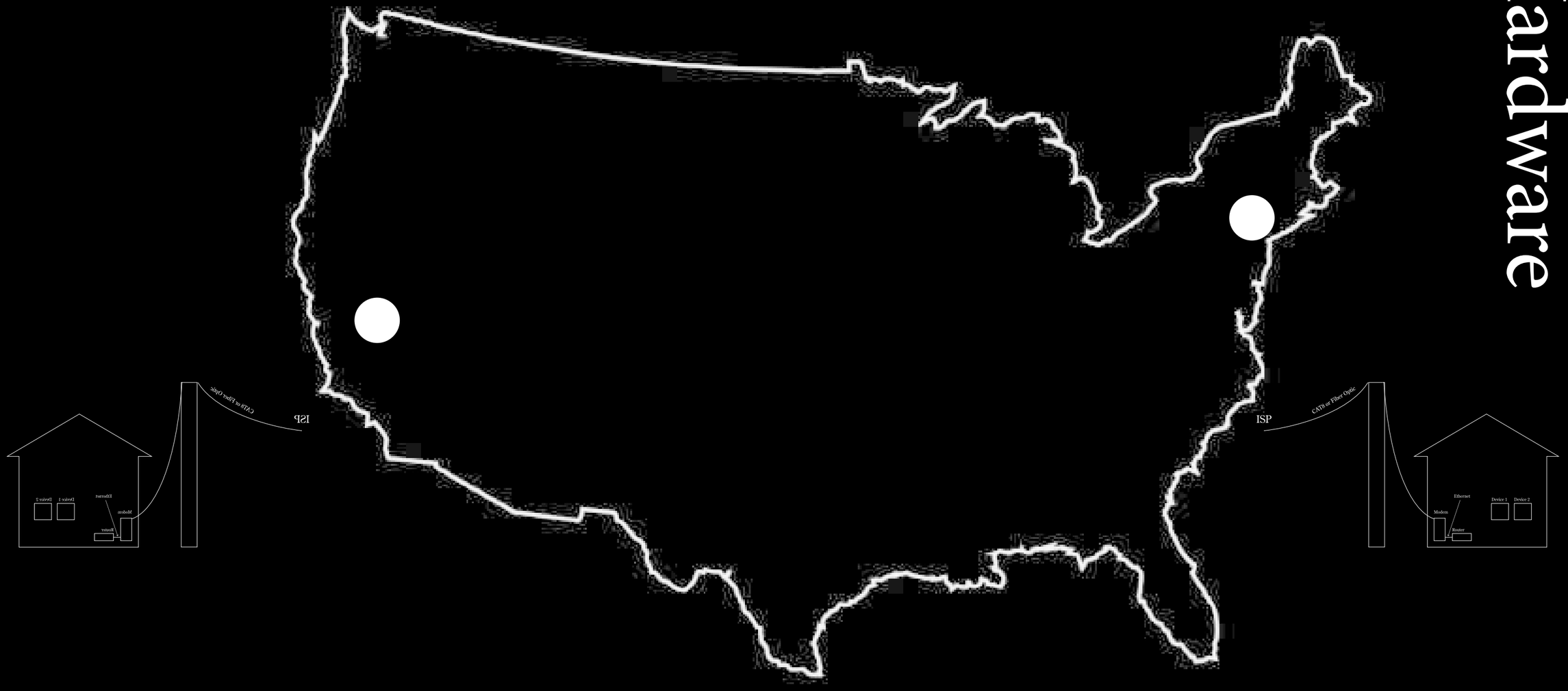
# Across the World

How do all of these localized networks provided by ISP-supplied routers connect together at a larger scale? We understand that ISPs maintain a massive web of cables providing a medium of communication to each of their customers. However, how do these home networks and computers all link together to connect the entire world with Internet?

This is the essence of the Internet Protocol. Quite literally, if you think of the word “Internet”, it stands for Inter-Network—the facilitation of communication across networks.

Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing

The example we'll use is between a computer / server in Connecticut and a computer / server in California. This is the connection we are trying to make:



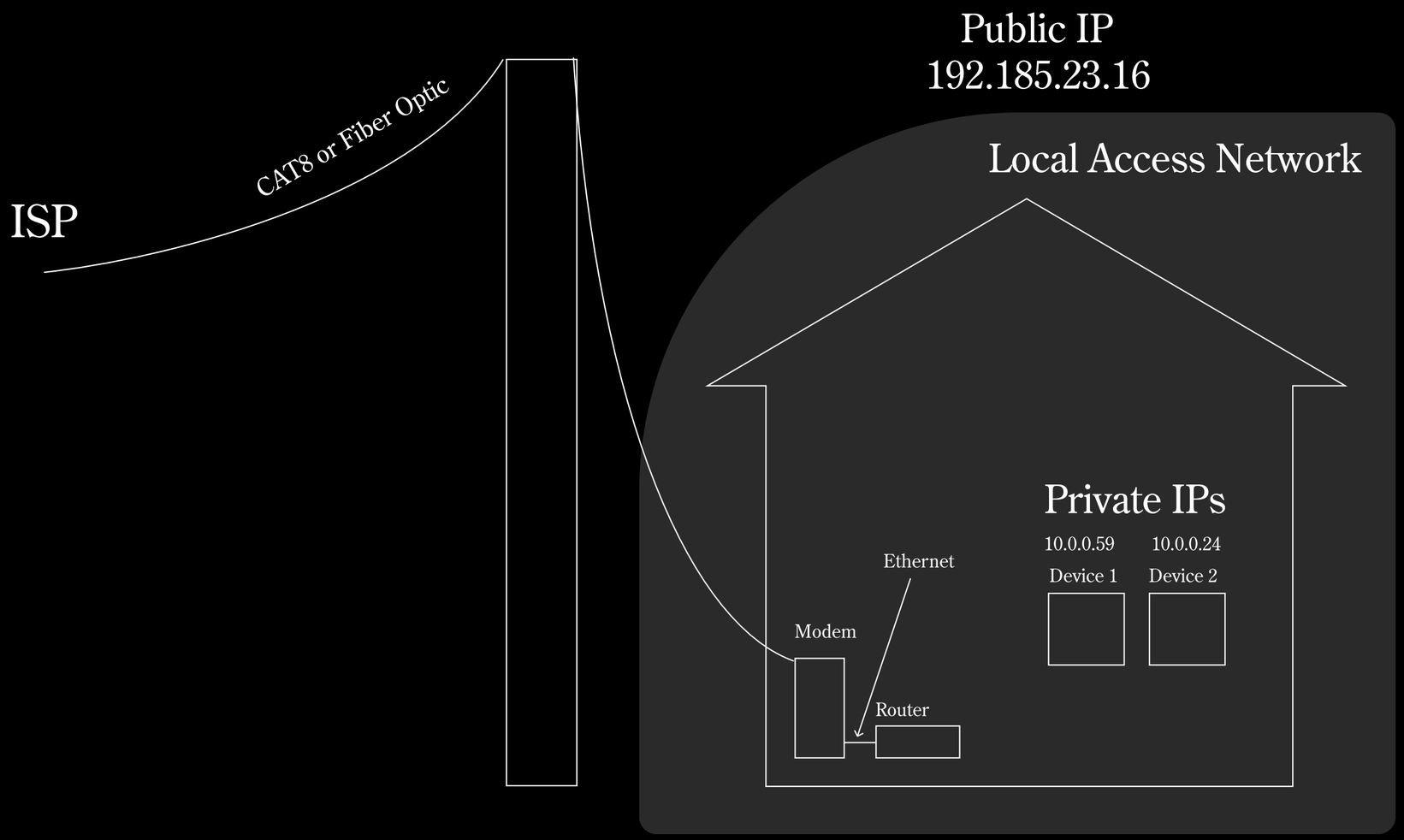
Wait, we can't do that yet. How do we address a computer in the first place? We need to talk about IP (Internet Protocol) addresses.

These are unique network identifiers across the entire internet, and are assigned by your ISP to your router such that no public IP address is the same.  If you connect with ethernet directly to your modem, your computer instead receives the IP address (not the router).

Within the network provided by a router, IP addresses are also assigned—however, these are not unique across the entire Internet, only within the Local Access Network (LAN) of the WiFi Network. The router uses these private IP addresses to identify devices connected to it. When you create a LAN Minecraft server, for example, you can communicate only with devices in the LAN by the internal network IP addresses assigned by your router—the connections never reach the global internet network where your router's public IP address is used.

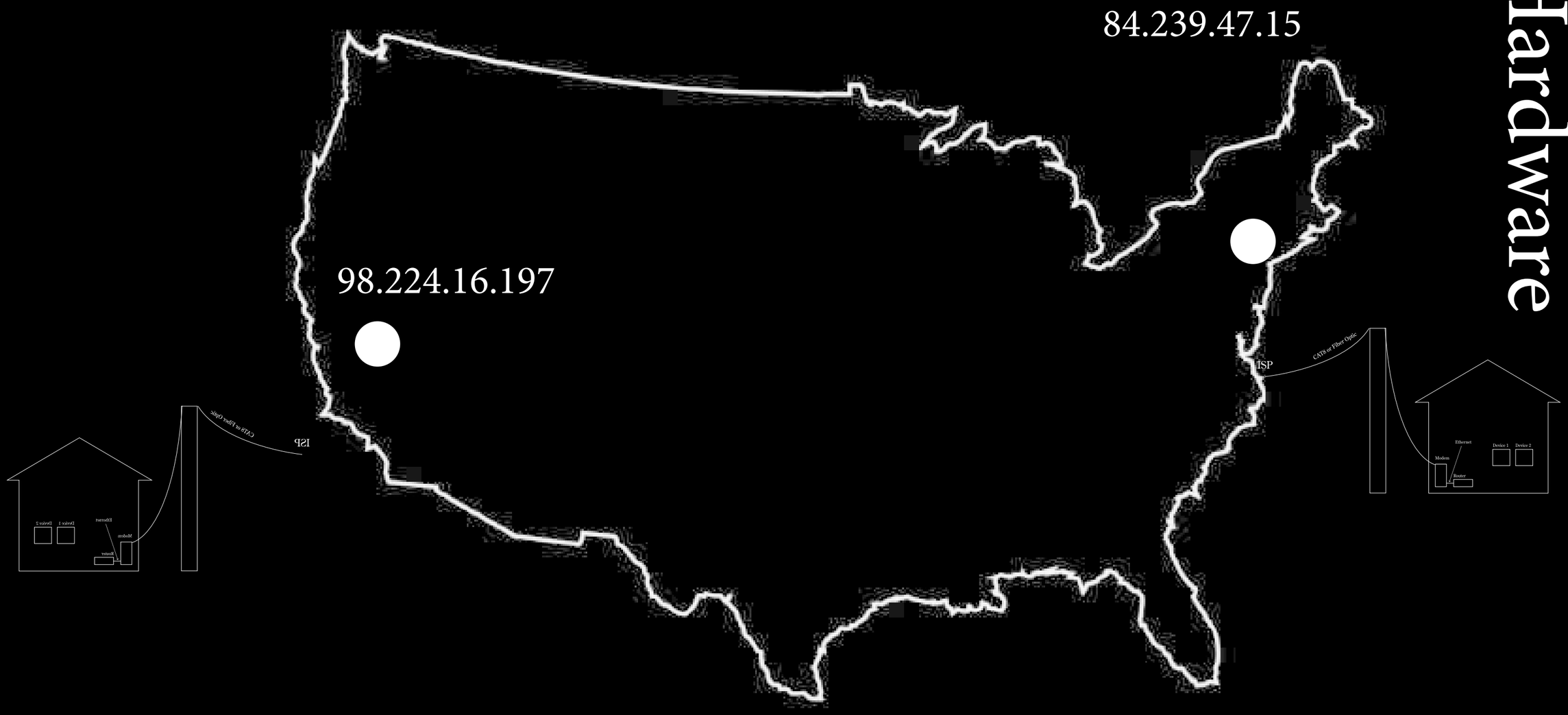
Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing
- IP (Internet Protocol) Address
- LAN (Local Access Network)

Here's a visualization:




Terms
- Dial-Up Internet
- Broadband
- ISP (Internet Service Provider)
- Modem
- Ethernet
- Router
- WiFi
- 2.4Ghz vs. 5.5Ghz
- Amplitude Modulation
- Time Division Multiplexing
- IP (Internet Protocol) Address
- LAN (Local Access Network)

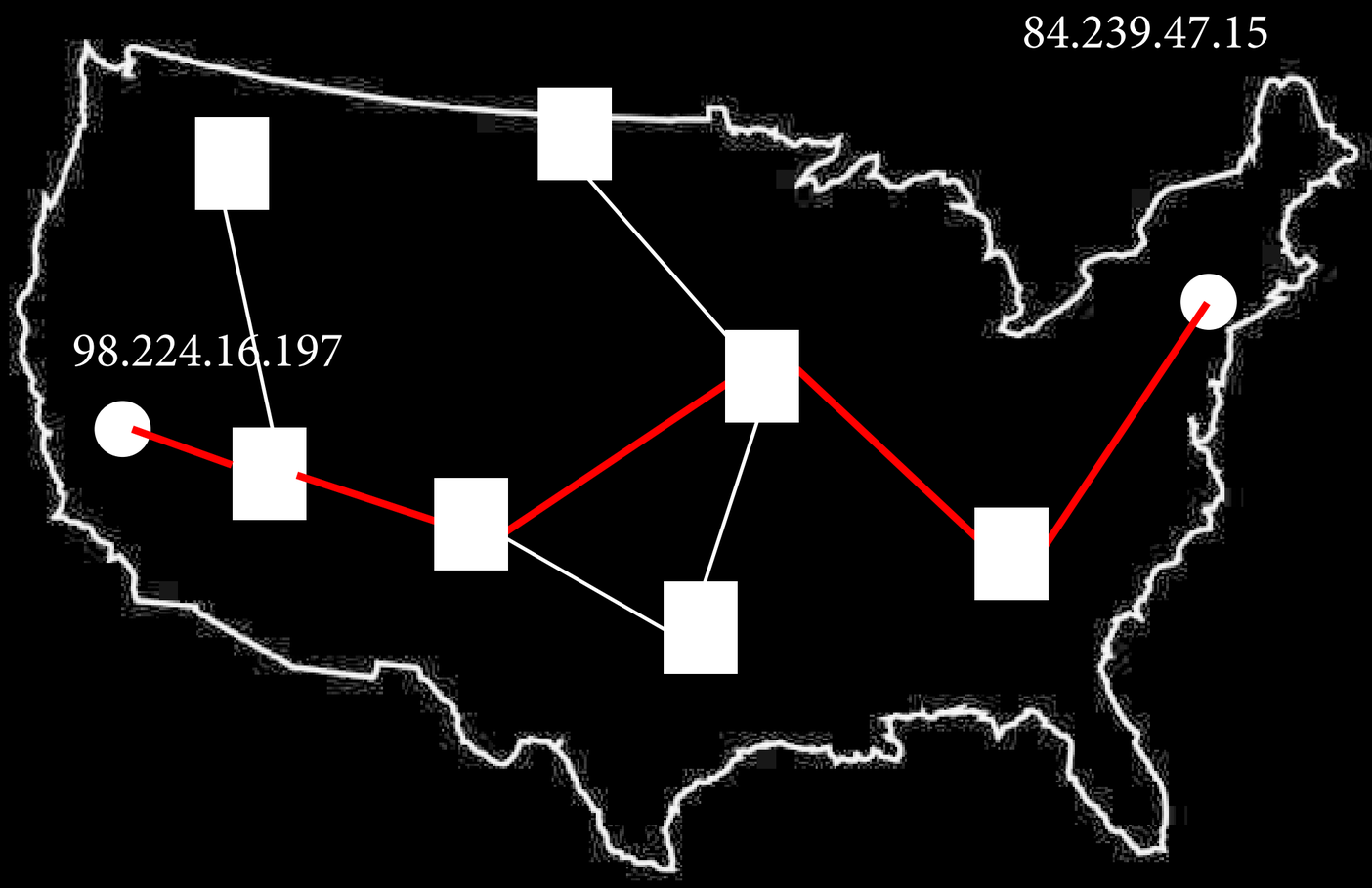
Let’s say we have the following IP addresses for our California and Connecticut networks, then:



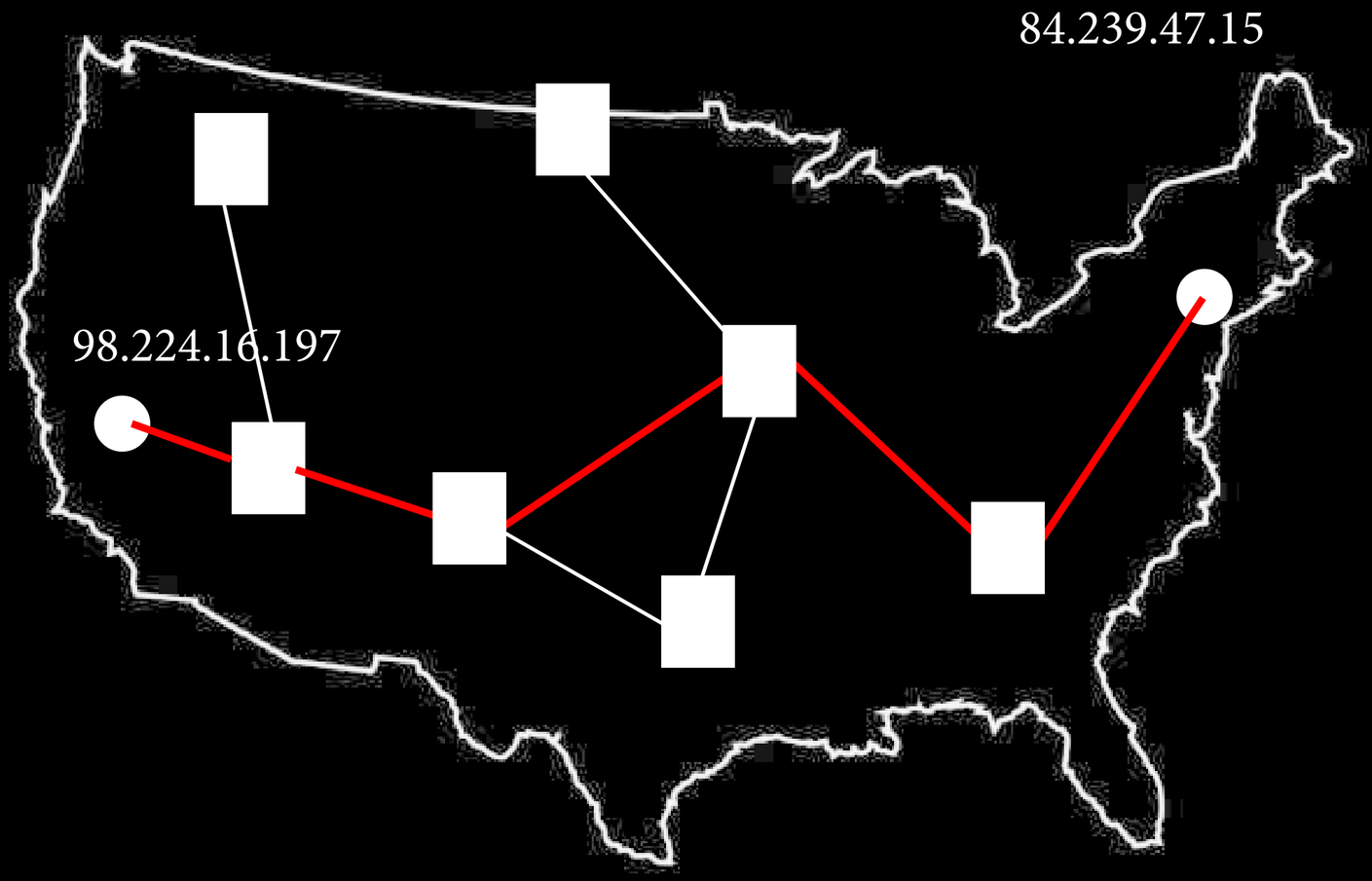
There isn't a cable that runs straight-shot from our California network to our Connecticut network. Instead, there are series of router nodes provided by ISPs (often different companies) through which a connection "hops".

At each router node in the network, the IP's prefix is used to determine the best next-node to traverse to.

Once a route is established, this traversal is cached such that the routers are ready for frames to resolve the best path for data to travel between the two networks. Now, communication can happen quickly and the data can be sent between ISPs.



This web of routers and networks spans the globe thousands of times over. I've glossed over some intricacies (like how frames can end up taking different routes to the same location), but in general this "hopping" between networks through copper and fibre optic cables is how data traverses the globe in the Internet from one IP to another.



That's it for now for the hardware aspect of the Internet. Some fun bits:

The Terminal command `traceroute` can be used to show the exact route (in IP addresses) a packet takes from your computer to a specific domain / IP address. Try the following command, for example:

```
traceroute -I 2024.software-for-people.net
```

From my house with Comcast / Xfinity as an ISP, the output:

```
traceroute to 2024.software-for-people.net (69.163.229.108), 64 hops max, 48 byte packets
 1  104.28.0.0 (104.28.0.0)  30.375 ms  17.745 ms  19.764 ms
 2  172.68.53.1 (172.68.53.1)  22.980 ms  94.543 ms  28.955 ms
 3  * * *
 4  nyk-bb1-link.ip.twelve99.net (62.115.122.202)  33.139 ms  102.860 ms  26.203 ms
 5  palo-b24-link.ip.twelve99.net (62.115.118.121)  98.663 ms  92.764 ms  89.976 ms
 6  port-b3-link.ip.twelve99.net (62.115.115.25)  108.207 ms  113.418 ms  119.008 ms
 7  rcntelecom-ic-337495.ip.twelve99-cust.net (62.115.33.125)  112.317 ms  104.443 ms  121.650 ms
 8  216.243.25.90 (216.243.25.90)  126.698 ms  104.925 ms  128.946 ms
 9  pdx1-cr-1.sd.dreamhost.com (66.33.200.2)  114.113 ms  111.304 ms  180.621 ms
10  pdx1-a7u27-acc.sd.dreamhost.com (66.33.200.56)  113.792 ms  110.611 ms  112.563 ms
11  apache2-xenon.foss.dreamhost.com (69.163.229.108)  111.194 ms  109.494 ms  110.238 ms
```

Seems like Rosa is using Dreamhost :) Also, interestingly, twelve99.net is an ISP in Sweden—so the path a packet takes isn't always direct!

That's it for now for the hardware aspect of the Internet. Some fun bits:

The Terminal command `traceroute` can be used to show the exact route (in IP addresses) a packet takes from your computer to a specific domain / IP address. Try the following command, for example:

```
traceroute -I 2024.software-for-people.net
```

From my house with Comcast / Xfinity as an ISP, the output:

```
traceroute to 2024.software-for-people.net (69.163.229.108), 64 hops max, 48 byte packets
 1  104.28.0.0 (104.28.0.0)  30.375 ms  17.745 ms  19.764 ms
 2  172.68.53.1 (172.68.53.1)  22.980 ms  94.543 ms  28.955 ms
 3  * * *
 4  nyk-bb1-link.ip.twelve99.net (62.115.122.202)  33.139 ms  102.860 ms  26.203 ms
 5  palo-b24-link.ip.twelve99.net (62.115.118.121)  98.663 ms  92.764 ms  89.976 ms
 6  port-b3-link.ip.twelve99.net (62.115.115.25)  108.207 ms  113.418 ms  119.008 ms
 7  rcntelecom-ic-337495.ip.twelve99-cust.net (62.115.33.125)  112.317 ms  104.443 ms  121.650 ms
 8  216.243.25.90 (216.243.25.90)  126.698 ms  104.925 ms  128.946 ms
 9  pdx1-cr-1.sd.dreamhost.com (66.33.200.2)  114.113 ms  111.304 ms  180.621 ms
10  pdx1-a7u27-acc.sd.dreamhost.com (66.33.200.56)  113.792 ms  110.611 ms  112.563 ms
11  apache2-xenon.foss.dreamhost.com (69.163.229.108)  111.194 ms  109.494 ms  110.238 ms
```

Seems like Rosa is using Dreamhost :)

You can use a website like whatismyipaddress.com to look up the locations of IP addresses, too. Some interesting locations my packets for 2024.software-for-people.net visited before ending up in California at DreamHost's servers:

```
 2  172.68.53.1 (172.68.53.1)  22.980 ms  94.543 ms  28.955 ms
```

This is CloudFlare's datacenter in Boston, Massachusetts.

```
 4  nyk-bb1-link.ip.twelve99.net (62.115.122.202)  33.139 ms  102.860 ms  26.203 ms
```

This is a datacenter in Belgium provided by a Swedish ISP, Arelion.

```
 8  216.243.25.90 (216.243.25.90)  126.698 ms  104.925 ms  128.946 ms
```

This is a datacenter in Portland, Oregon built by Wave Broadband.

```
11  apache2-xenon.foss.dreamhost.com (69.163.229.108)  111.194 ms  109.494 ms  110.238 ms
```

And this is the final location where Rosa's site is ultimately hosted—Brea, California under ISP New Dream Network, LLC.

It might seem counter-intuitive to send data all over the place like this, but these networks are being traversed by billions of bytes every second at lightspeed. Sometimes a detour is far faster than taking a route along congested electric super highways.

# 2. Software & Protocols

I've already used a lot of slides going over the hardware, so I'll keep this second section as brief as I can. I mostly want to go over the protocols that open the connection and read a file from a server (what happens when you visit a webpage).

This takes place in four steps:
1. DNS Lookup
2. TCP Handshake
3. TLS Handshake
4. HTTP Request

There are 4-5 agents involved:
1. Your Web Browser
2. A DNS Server
3. A Certificate Authority
4. An Origin Server
5. (Sometimes) A CDN

# 1. DNS Lookup

Terms
- Domain Name Server (DNS)

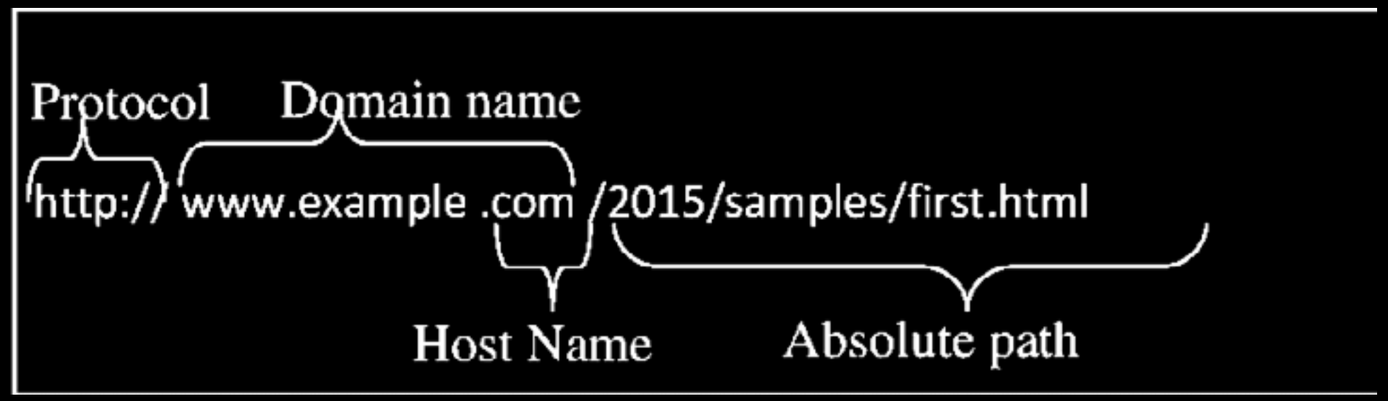
First, you open your web browser—Google Chrome, Safari, Firefox, etc. and type in https://2024.software-for-people.net. You hit enter, and—wait... don't we need an IP address to connect to a server?

Yes, we do, and 2024.software-for-people.net is not an IP address, but rather a Universal Resource Locator (URL). This is where the Domain Name Server (DNS) comes into play.

In the early days of ARPAnet, users had already begun writing programs that would map readable text "domains" to numeric IPv4 addresses so they could easily remember and type in the locations of servers they wanted to connect to. In 1983, Paul Mockapetris invented the concept of a Domain Name Server—essentially a standardized database of text-to-IP address mappings that users could "register" in and provide a single source of truth for their readable server addresses.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)

Consider the parts of a URL (assuming a GET request):




A Universal Resource Locator (URL) is simply an address. Here:

- With the HTTP protocol,
- On the www.example.com server
- GET the resource at /2015/samples/first.html.

In order to figure out the IP address of www.example.com, your browser first has to query a DNS Server which will send it back the corresponding IP address. With that in hand, the browser internally builds an IP-based URL that it can actually connect to:
`http://93.184.216.34/2015/samples/first.html`

# 2. TCP Handshake

Now, your browser has to open a connection to the server specified by the URL. This happens on a protocol-level through the computer's operating system in what is known as a Transport Control Protocol (TCP) handshake.

TCP is the communication protocol—defined by exchange of packets of data—on which most of the internet is built. A key feature of TCP is the request/response workflow, where delivery of packets is guaranteed at the price of higher overhead and latency. Every time a packet of information is sent, it must be acknowledged with a ACK signal to indicate it has been received—else it must be re-sent.

An alternative to TCP is the Universal Datagram Protocol (UDP), where packets and data are not acknowledged and may be lost, at the benefit of far less overhead. This is commonly used in livestreaming services where packet loss is acceptable.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)

To “open a connection” with TCP, three things must happen.

1. Your computer sends a SYN segment to the target server’s IP, indicating a request to open a connection.
2. The server, if it allows and can open a connection, responds with an ACK segment, acknowledging the connection request. It also sends a SYN segment back to your computer, requesting to open the other direction of a connection.
3. Your computer responds with an ACK segment, acknowledging the completion of the handshake and officially “opening” the connection.

These three steps are known as the Three-Way Handshake and serve to (a) let both sides know that they are ready to transfer data and (b) set a couple initial flags that preserve the integrity of the connection and make spoofing / hijacking the connection harder for attackers.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)

# 3. TLS Handshake

At this point, the TCP connection has been opened—data transmitted on it will be sent to the other server, but is entirely unencrypted. There is no agreed-upon encoding scheme that protects transmitted data from attackers / spoofers.

To establish a common secure baseline, most websites nowadays opt into a security layer of TCP known as Transport Layer Security (TLS), or its predecessor Secure Sockets Layer (SSL). When you connect to a site using HTTP(S)—S for secure—, you are using a TLS connection.

This also occurs with a handshake like opening a TCP connection, but, unlike TCP, this handshake involves the exchange of special files known as certificates. The client and server agree on an a cipher suite and exchange cryptographic certificates which are used to verify each others' identities and generate a session key, encrypting all data passing between them.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)
- Transport Layer Security (TLS)
- Secure Sockets Layer (SSL)

How is identity verified? This involves what is known as a Certificate Authority. When you host a site on a specific domain, in order to enable secure HTTPS connections, you must be issued an SSL certificate by a trusted entity known as a certificate authority (CA).

This certificate is essentially a file verifying your domain's identity with key cryptographic information your server can then use in the TLS handshake to safely encrypt data between it and a client.

These third-party Certificate Authorities issuing identities are crucial to the integrity of the internet and act as an unopininiated identity verification source. When you visit a website and your browser complains that it is "unsafe" or "unsecure," this is generally from an expired SSL certificate or a certificate in which the domain name does not match the domain name the certificate was issued for.

Fun fact: Mark Shuttleworth, the other South African billionaire space explorer, made his money by selling his certificate authority company Thawte Consulting in 1999.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)
- Transport Layer Security (TLS)
- Certificate Authority (CA)

Now, we're all set to begin transmitting data! As a recap:

1. We first determined the IP address of the server behind the domain we wanted to visit through a DNS lookup.
2. We opened a connection to said server using the TCP Three-Way Handshake, enabling transmission of data between client and server.
3. We established a secure session using a TLS Handshake, verifying the server's identity through a certificate signed by a Certificate Authority and using this certificate to encrypt all data in our connection.

From this point on, it's smooth sailing. Your browser initiates a GET HTTP request, which is split up into packets, encrypted with TLS, and transferred securely over the TCP Protocol to the origin server. The origin server receives the packets, assembles them in order, decrypts, does some processing, and sends back a response in the same manner.

Terms
- Domain Name Server (DNS)
- Universal Resource Locator (URL)
- Transport Control Protocol (TCP)
- Universal Datagram Protocol (UDP)
- Transport Layer Security (TLS)
- Certificate Authority (CA)