Please join the `eduroam` wifi network.

If it asks you for a username/password, use your SUNET ID (first part of email) and password, and accept any questions about certificates.

CS45, Lecture 8
Networking

Spring 2023
Akshay Srivatsan, Ayelet Drazen, Jonathan Kula

A Grain Of Sand

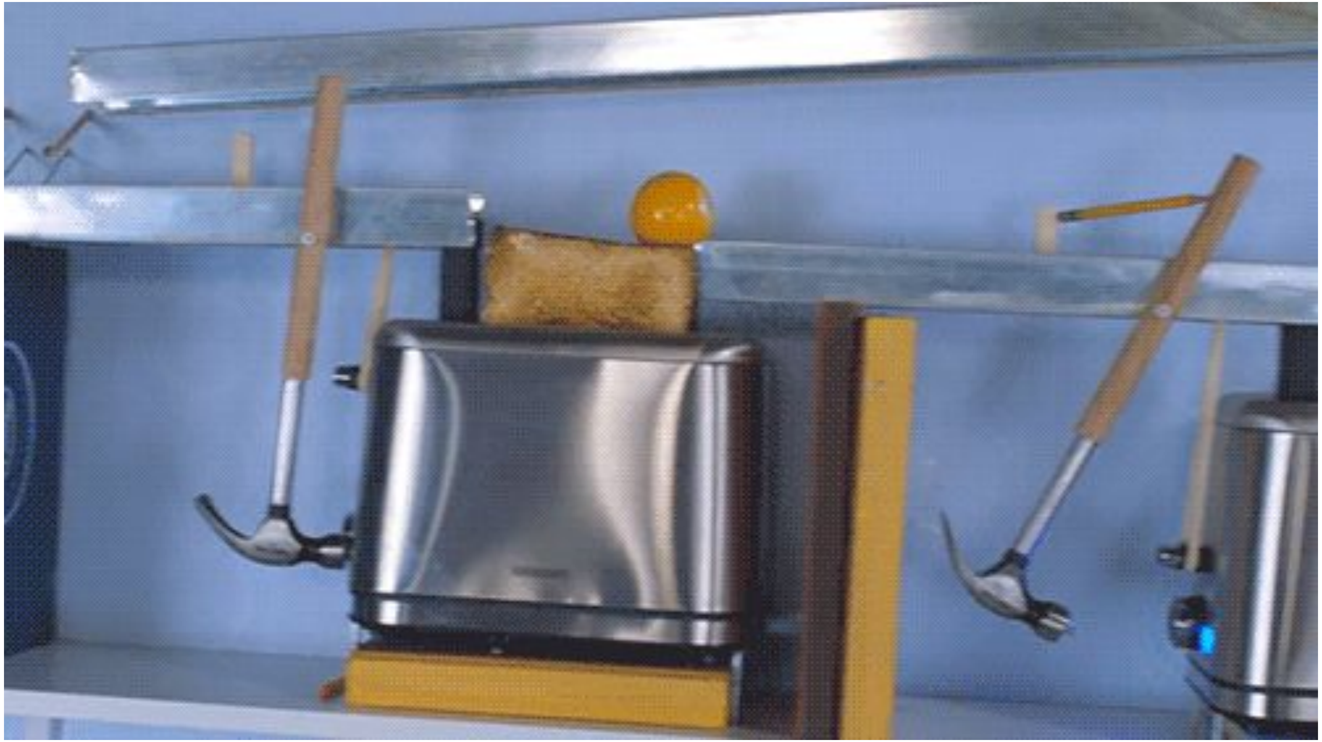
A Touch From Across The World

The distinction between "virtual" and "physical" is fuzzy.

It may be small, but every piece of data, every process, involves the physical presence, absence, or movement of *at least* electrons.

The internet lets us affect *real* objects, anywhere, anywhere.

Whether it's just causing a CPU to do some work, or adjusting the position of a mechanical arm constructing a car– our actions over a network set off a veritable rube goldberg machine that actually makes physical changes somewhere else– as far as across the globe!



Learning Goals

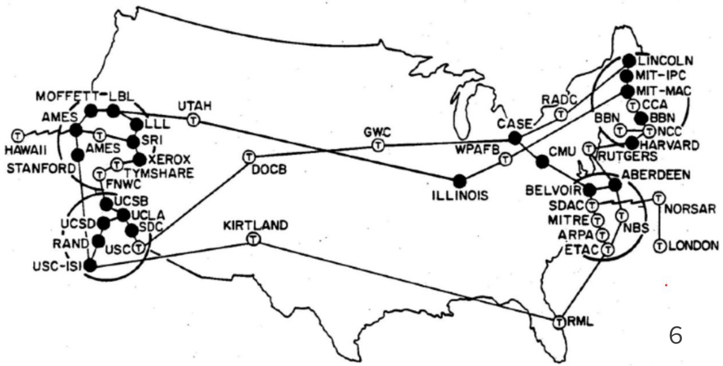
- Understand basics of computer networking
  - Circuit-switched vs packet-switched networks
  - Ethernet, IP, TCP; NAT, DHCP, DNS
- Understand the structure and use of Ethernet, and IPv4 addresses
- Understand how a packet is routed around Stanford's campus
- Understand how a basic home network works
- Obtain a rudimentary understanding of DNS
- Understand what a server and client are
- **Understand how to use ping and traceroute tools**
- **Understand how to use nslookup and dig tools**
- **Run your first server and have a friend/classmate connect to it!**

Research At A Distance

The Inception of Modern Computer Networking

- ARPANET was the progenitor of the modern internet.
- It was created with the goal of facilitating resource sharing between faraway computers, especially for research purposes.
- Key innovation:

  packet-switched wide area network



Here to There and Back Again

[demo circuit switched network]

Here to There and Back Again

[demo packet switched network]

Concepts from the Demo

Packets *(the envelope)*

Chunked-up data, including any information about how to transport the data and where to.

Routers *(that's you!)*

Computers whose special job is to ingest incoming packets and very quickly decide where to forward those packets to. (They often have special hardware to make this happen faster.)

Addresses

We wrote things like "back left of the room," but computers that want to talk to each other must have an address, and often have multiple of different kinds for different purposes. More on this in a minute.

Hops

Each time the "packet" was "forwarded" (each time the envelope exchanged hands), that was a "hop."

Concepts from the Demo

Routers *(that's you!)*

By the way, consumer "routers" you might buy at the store or be given by your ISP are actually much more than just a router: they're usually a **router, firewall, access point,** and **network switch** all at once.

Addresses I

IP Addresses

These are used to **identify a computer on a network.** That means that **an IP address is unique to the network it's in.**

IP Addresses are structured in a particular way that makes routing efficient. More on this in a moment.

IP Addresses are used for getting information across a network, even if it needs to hop through many different points to get there.

Ethernet Address

These are used to **identify a computer globally.** They are **unique across all devices***, and **cannot be changed***.

Ethernet addresses are used to send information **locally**, i.e. to other computers you are directly connected to.

Addresses II: IP Addresses

IP Addresses

These are used to **identify a computer on a network.** That means that **an IP address is unique to the network it's in.**

Addresses II: IP Addresses

IP Addresses

These are used to **identify a computer on a network.** That means that **an IP address is unique to the network it's in.**

If that network is **your home network**, your IP addresses will be unique only in that network. (This is commonly referred to as a Local Area Network, or a LAN).

Addresses II: IP Addresses

IP Addresses

These are used to **identify a computer on a network.** That means that **an IP address is unique to the network it's in.**

If that network is **your home network**, your IP addresses will be unique only in that network. (This is commonly referred to as a Local Area Network, or a LAN).

If that network is **the Internet**, your IP address(es) will be unique on the entire internet. (This is commonly referred to as a Wide Area Network, or a WAN).

Addresses II: IP Addresses

IP Addresses

These are used to **identify a computer on a network.** That means that **an IP address is unique to the network it's in.**

If that network is **your home network**, your IP addresses will be unique only in that network. (This is commonly referred to as a Local Area Network, or a LAN).

If that network is **the Internet**, your IP address(es) will be unique on the entire internet. (This is commonly referred to as a Wide Area Network, or a WAN).

Sometimes, you need to go **between different networks** (e.g. between a LAN and the Internet). In order to do this, you use something called Network Address Translation (or NAT).

Addresses II: IP Addresses

Here is what your typical IPv4 address looks like:

192.168.1.1

Addresses II: IP Addresses

Here is what your typical IPv4 address looks like:

192.168.1.1

`11000000101010000000000100000001`

It might look like a set of four numbers, but it's really just a special format of a single **32-bit number,** chunked up in 8-bit sections.

Addresses II: IP Addresses

Here is what your typical IPv4 address looks like:

192.168.1.1

```
General <-----------------------------------------------> Specific
```

IP Addresses are **hierarchical.** Networks are defined in terms of this hierarchy.

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

192.168.x.x

`11000000` `10101000` `xxxxxxxxxxxxxxxx`

IP Addresses are **hierarchical.** For example, any IP address starting with 192.168 (that is, starting with `1100000010101000`) is defined as being for LANs.

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

192.168.0.0/16

`11000000101010000xxxxxxxxxxxxxxxx`

One common way to **define a subnet** is using CIDR notation. The above says that "the first 16 bits are locked in, and the rest of them are for the network"

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

192.168.1.0/24

11000000101010000000000 1xxxxxxxx

This is a common LAN subnet used by home routers.

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

192.168.1.0/24

`11000000101010000000000 1xxxxxxxx`

Every network has two special addresses: the **first** address (e.g. 192.168.1.0, for the above network) is **unused** (it defines the network). The **last** address (e.g. 192.168.1.255 for the above network) is called the **broadcast** address. Messages sent to broadcast are received by *all computers* on the network.

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

10.0.0.0/24

00001010000000000000000000000000xxxxxxxx

Here's another common LAN network.

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

10.0.0.0/8

00001010xxxxxxxxxxxxxxxxxxxxxxxx

In fact, any IP address in the subnet 10.0.0.0/8 is reserved for LANs!

Addresses II: IP Addresses: Subnets

Here is what your typical IPv4 address looks like:

10.0.10.0/23

`00001010.00000000.0000101x.xxxxxxxx`

Of course, subnets don't have to only be /8s, /16s, or /24s– they can be any number of bits you want*. For example, my home network uses this subnet.

Addresses II: IP Addresses: Subnets

This is one of the blocks of **public IPv4 space** that Stanford owns:

128.12.0.0/16

`10000000.00001100.xxxxxxxx.xxxxxxxx`

I.e., this is a set of IP addresses **for the public Internet** (not local area networks!) that Stanford owns exclusively. Nobody else can use these.

Addresses II: IP Addresses: Subnets

Meanwhile, most home internet connections give you only a single, **temporary** public IP address. Some of them don't even give you that.

172.217.14.78/32

10101100110110010000111001001110

A /32 is a single IP address.

Addresses II: IP Addresses: Subnets

Meanwhile, most home internet connections give you only a single, **temporary** public IP address. Some of them don't even give you that.

172.217.14.78/32

`10101100110110010000111001001110`

A /32 is a single IP address. Unless you're running an ISP or extremely large business, you'll likely be given an IP address **from a DHCP** (Dynamic Host Configuration Protocol) **server**.

Addresses II: IP Addresses: Subnets

Meanwhile, most home internet connections give you only a single, **temporary** public IP address. Some of them don't even give you that.

172.217.14.78/32

10101100110110010000111001001110

A /32 is a single IP address. Unless you're running an ISP or extremely large business, you'll likely be given an IP address **from a DHCP** (Dynamic Host Configuration Protocol) **server**. DHCP servers oversee some set of IP addresses and give them out to computers that ask, ensuring no duplicates.

Addresses III: Ethernet Addresses

However: **Public IP addresses are owned by people/companies, and private IP addresses have duplicates and are given dynamically.** We need some way to identify *specific computers* in order to send them information.

Enter: MAC Addresses (a.k.a. Ethernet addresses).

Addresses III: Ethernet Addresses

Here is what your typical MAC address looks like:

2A:78:A3:5F:5D:02

Addresses III: Ethernet Addresses

Here is what your typical MAC address looks like:

2A:78:A3:5F:5D:02

`001010100111100010100011010111110101110100000010`

MAC Addresses are also a chunked-up single number, in this case, a 48-bit number.

Addresses III: Ethernet Addresses

Here is what your typical MAC address looks like:

2A:78:A3:5F:5D:02

`001010100111100010100011010111110101110100000010`

They are split into two parts: the **OUI** (an ID given to a particular manufacturer) and the **NIC ID** (the ID for your computer*).

Addresses III: Ethernet Addresses

Here is what your typical MAC address looks like:

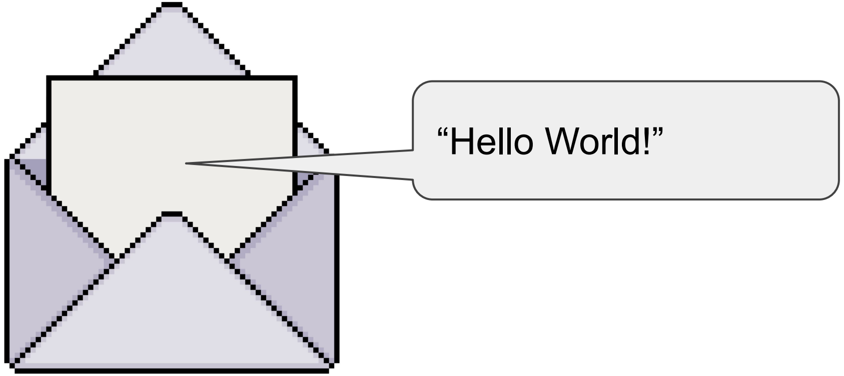
2A:78:A3:5F:5D:02

`001010100111100010100011010111110101110100000010`

Ethernet addresses (MAC addresses) are used to send packets **link-locally** (i.e. to other computers on the same logical “wire” as you. Usually the same as your LAN.)

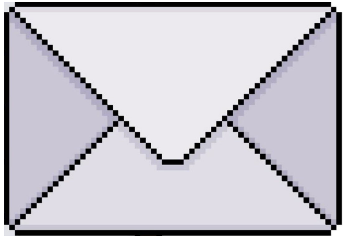
Visiting My Dorm Room: I

Let's send a message to my dorm room!



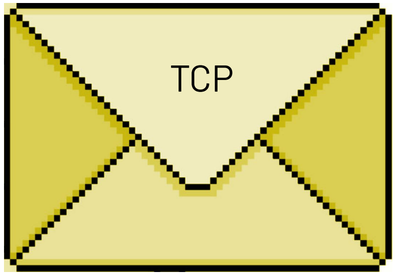
Visiting My Dorm Room: I

Let's send a message to my dorm room!



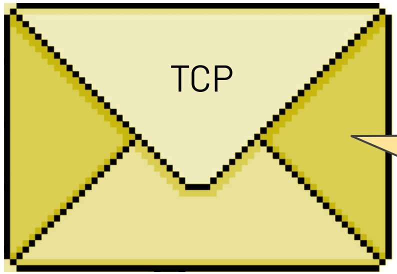
Visiting My Dorm Room: I

Let's send a message to my dorm room!



Visiting My Dorm Room: I

Let’s send a message to my dorm room!

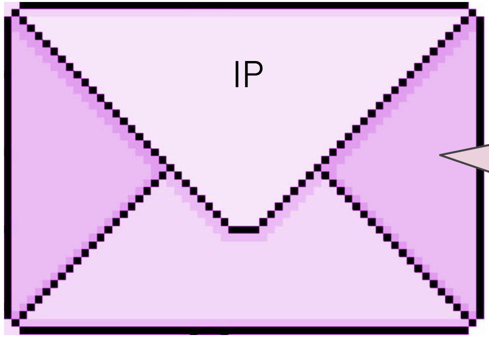


“Please make sure my data gets to where I’m sending it, **in order**, **resending any information that gets dropped along the way.**”

Visiting My Dorm Room: I

Let's send a message to my dorm room!

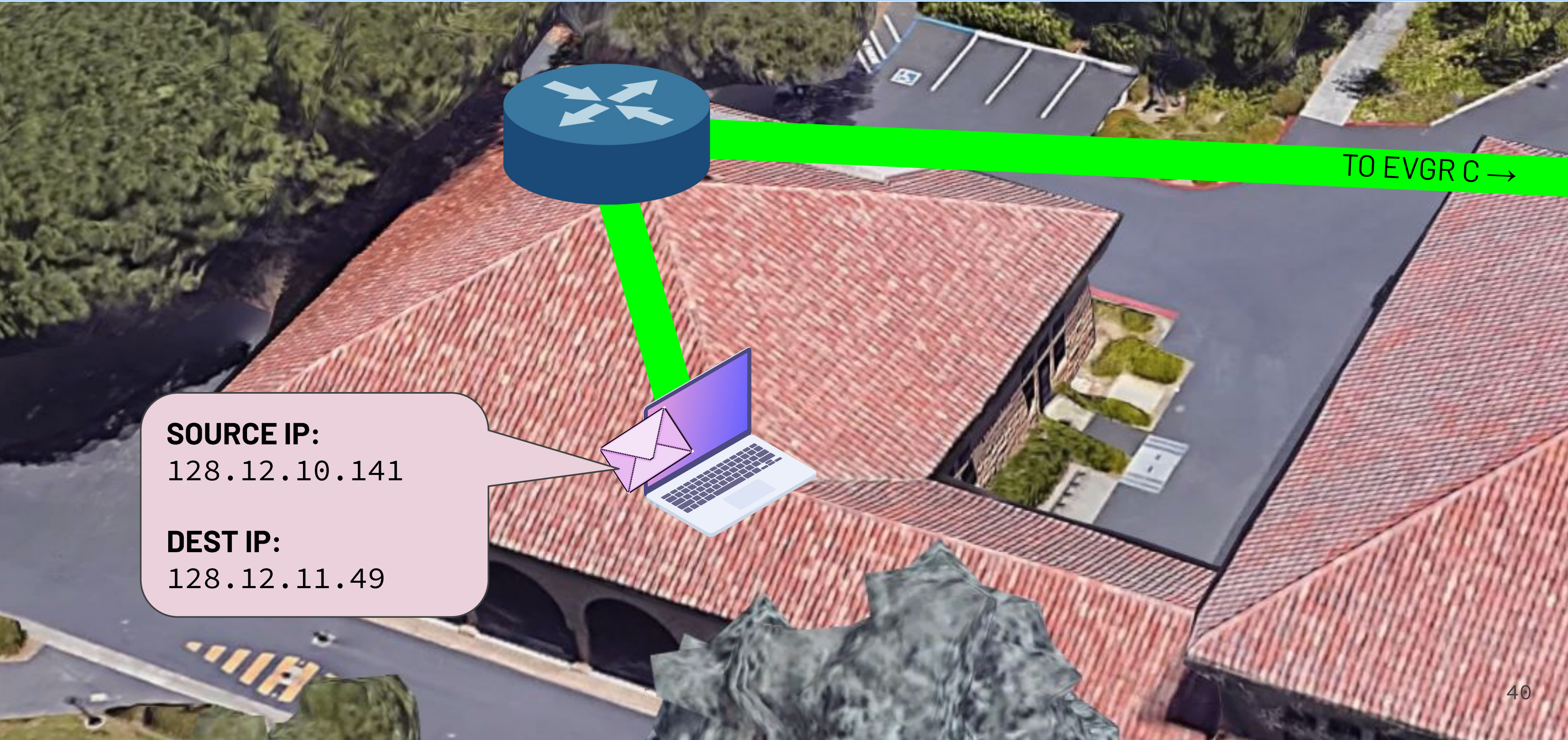
We use **encapsulation** to layer information that we need to transfer our message across the network.



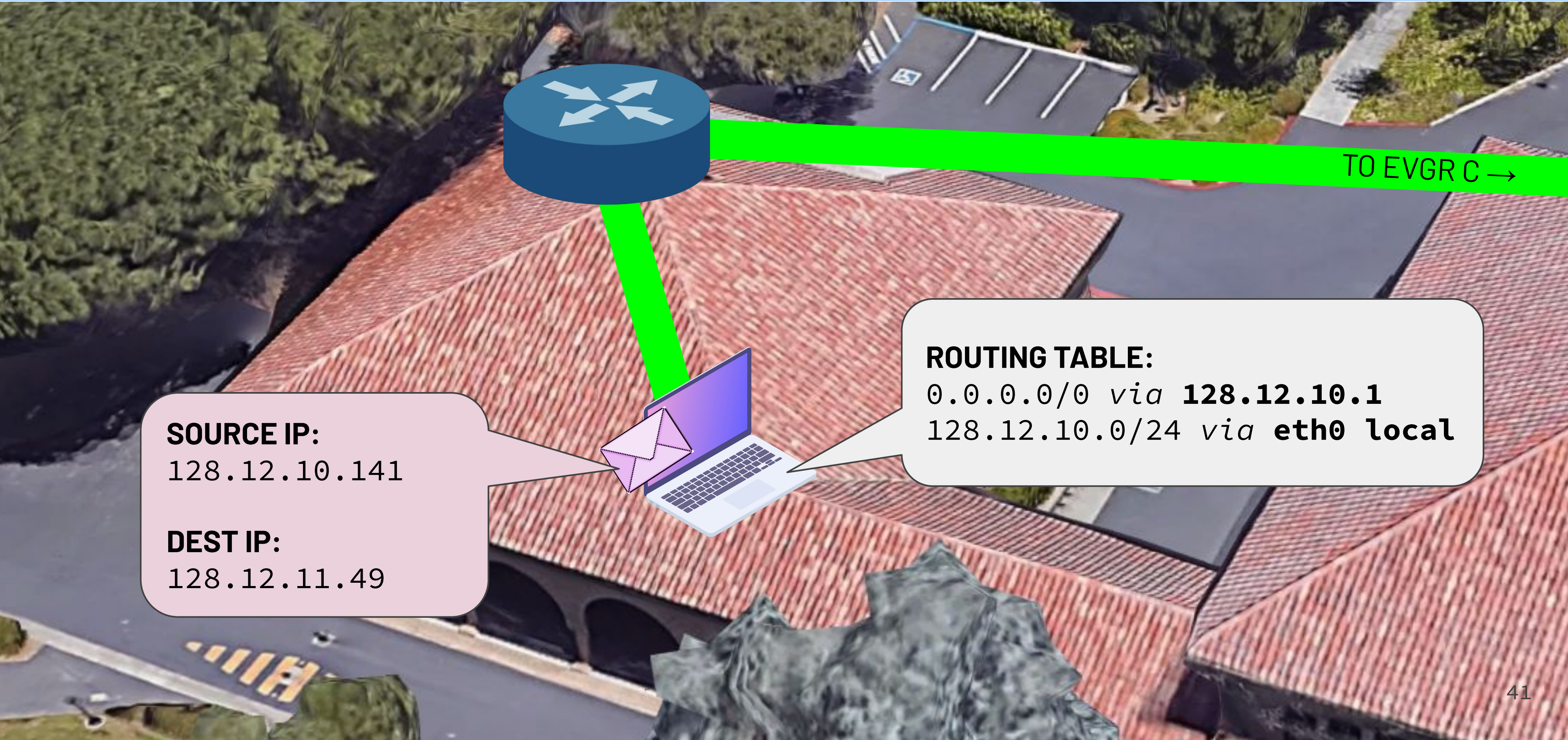
SOURCE IP:
128.12.10.141

DEST IP:
128.12.11.49

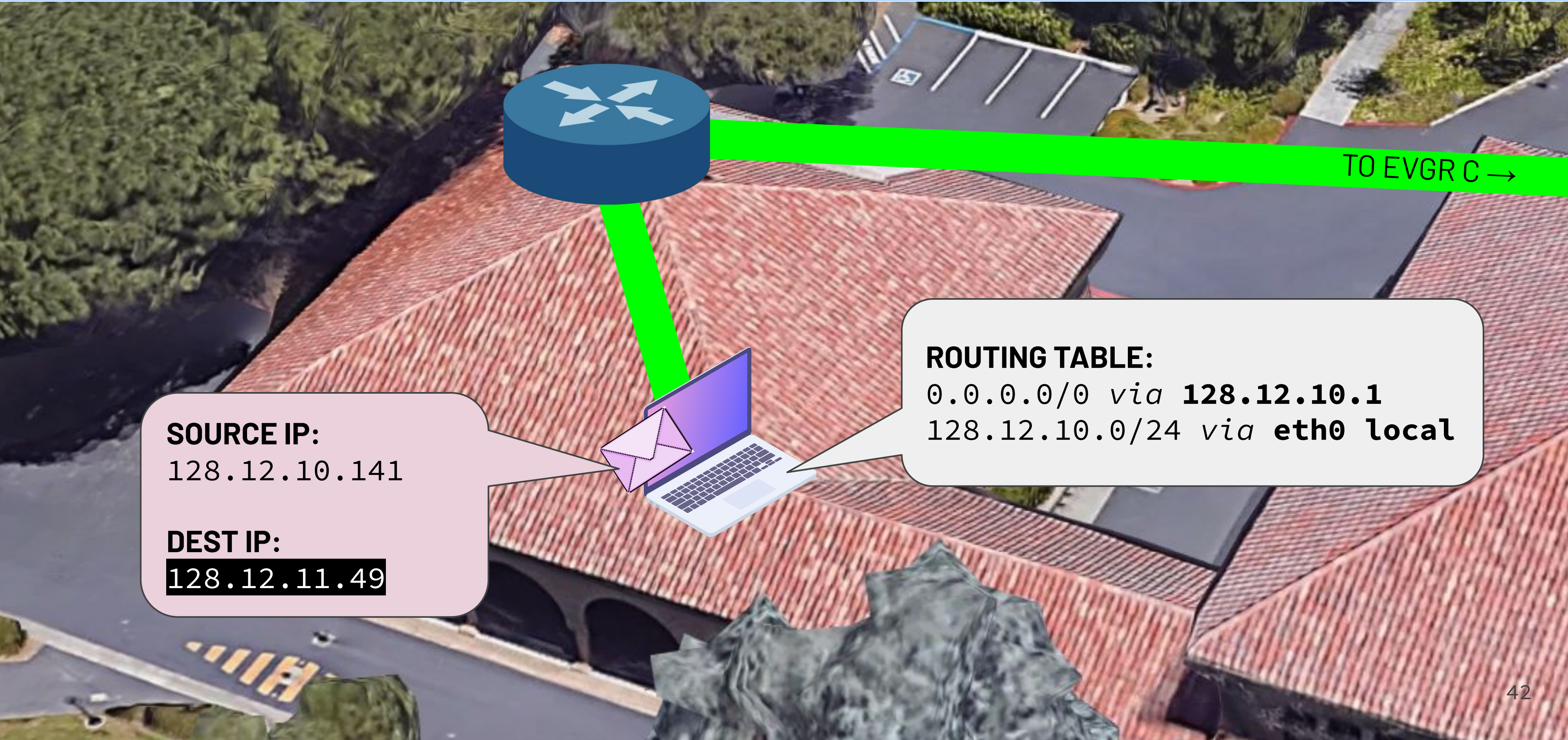
Visiting My Dorm Room: I



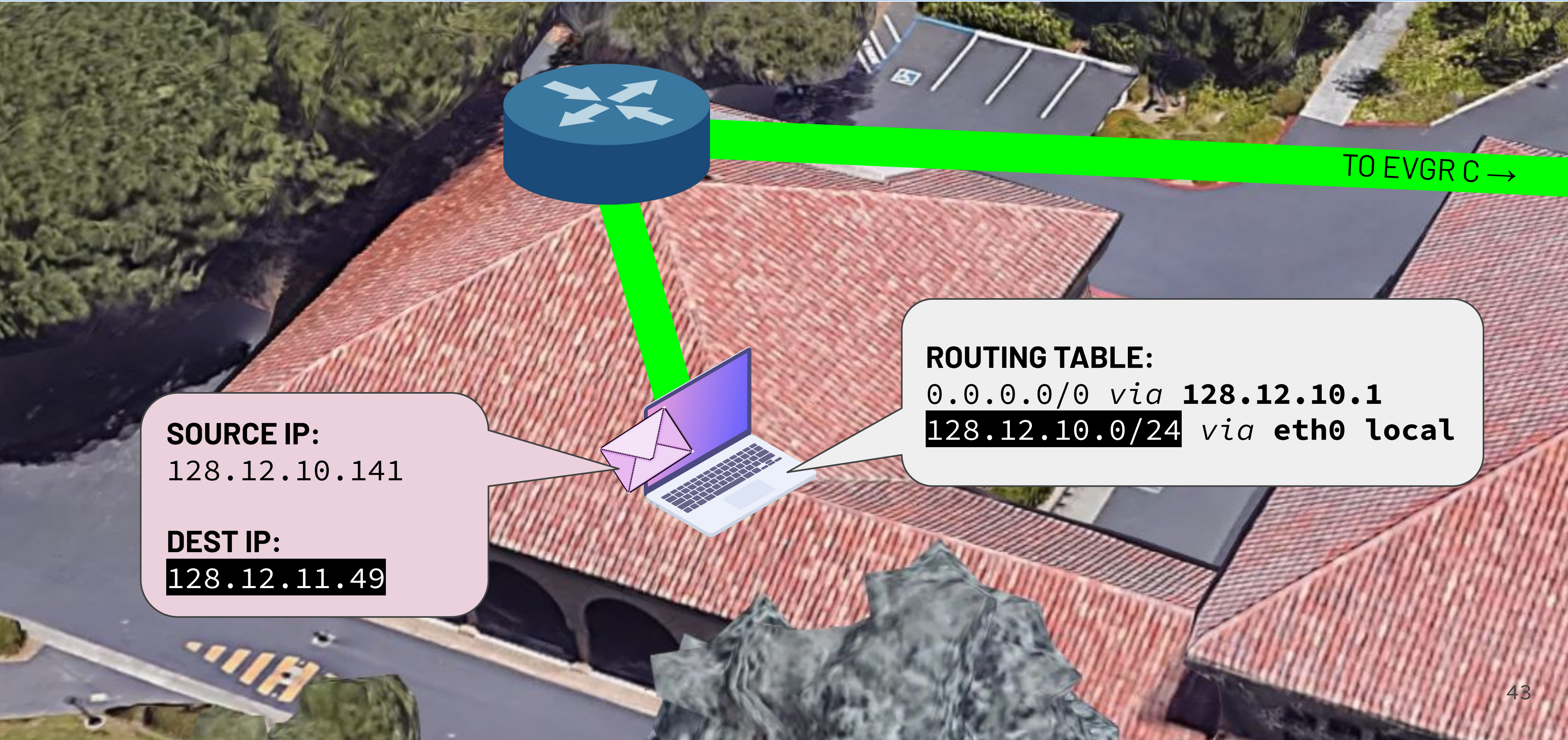
Visiting My Dorm Room: I



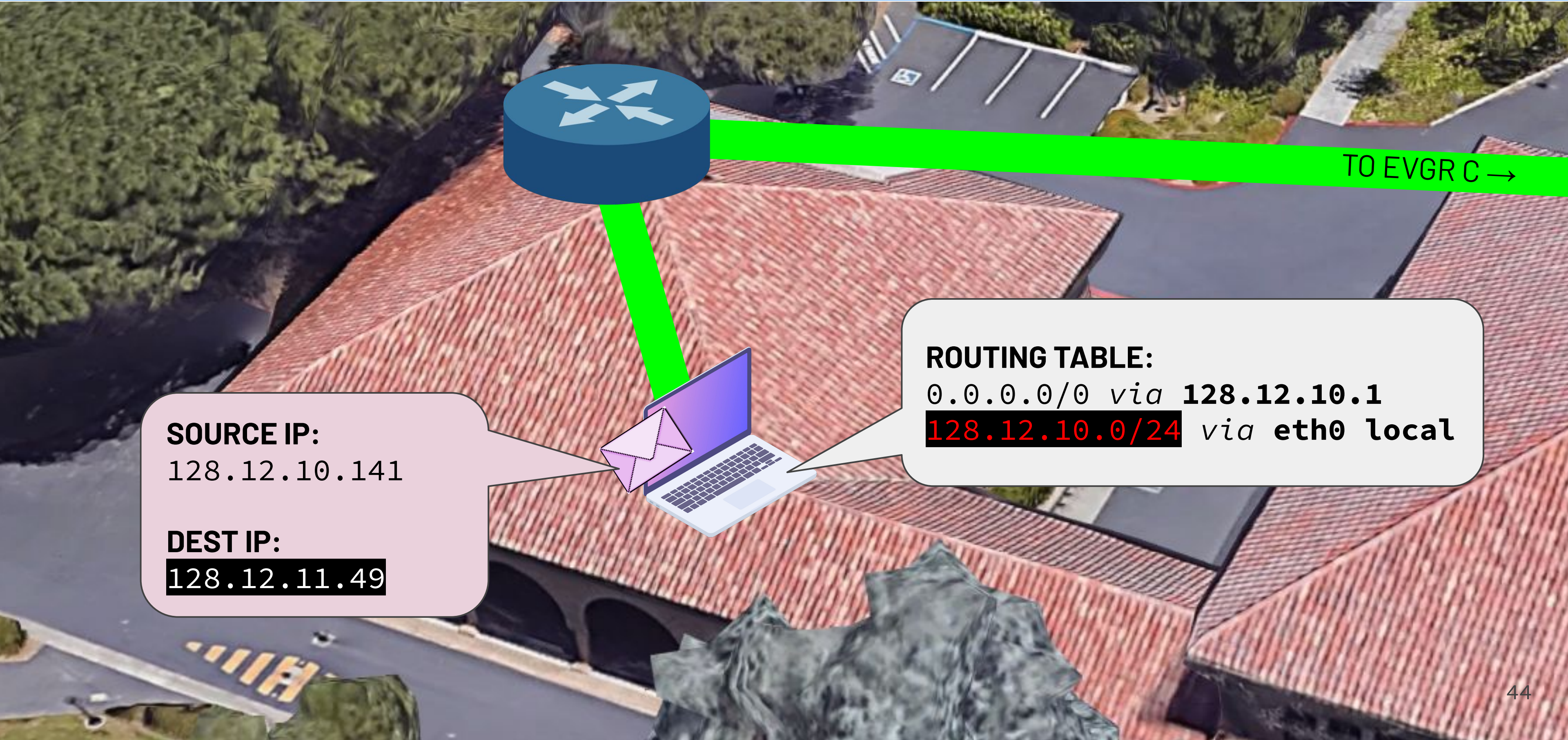
Visiting My Dorm Room: I



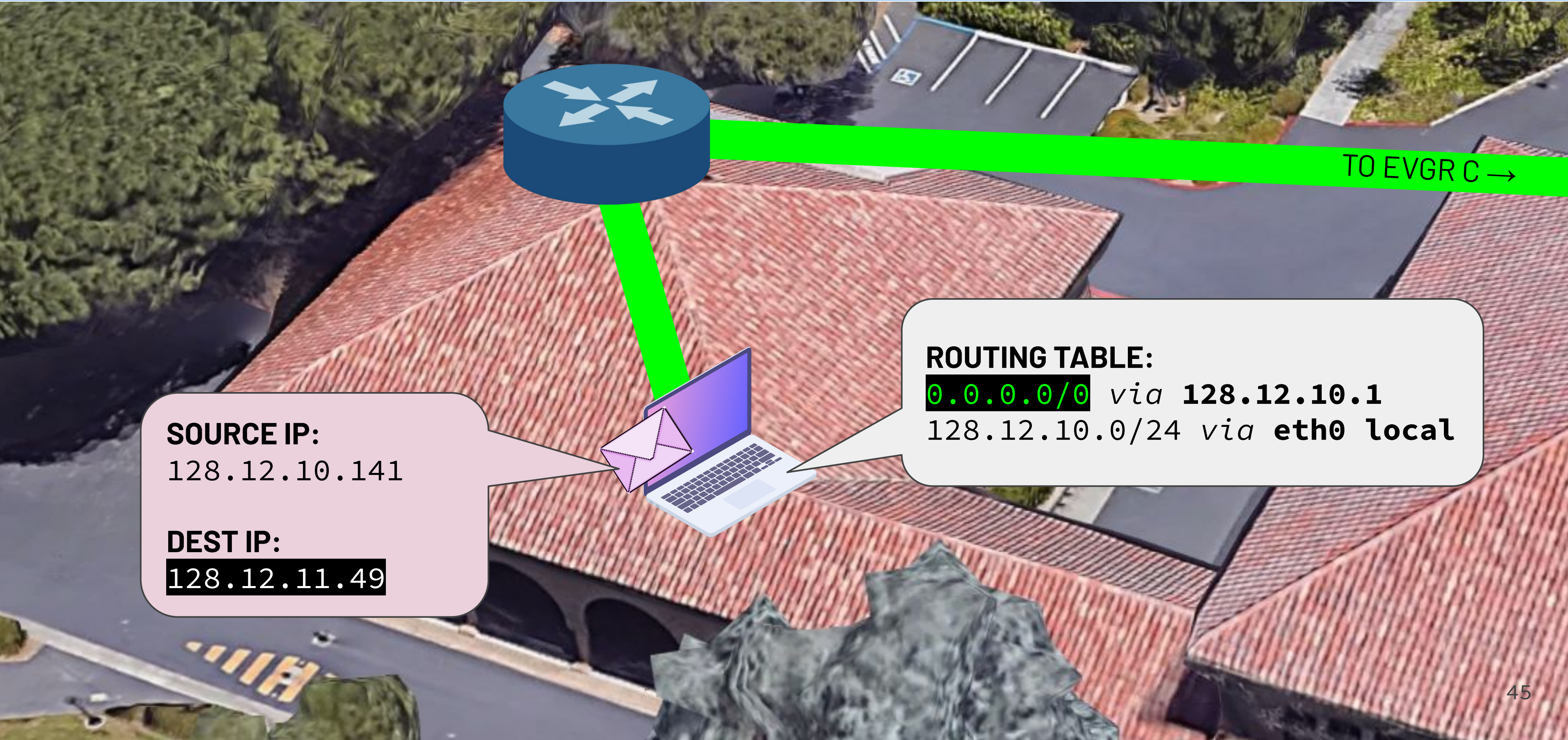
Visiting My Dorm Room: I



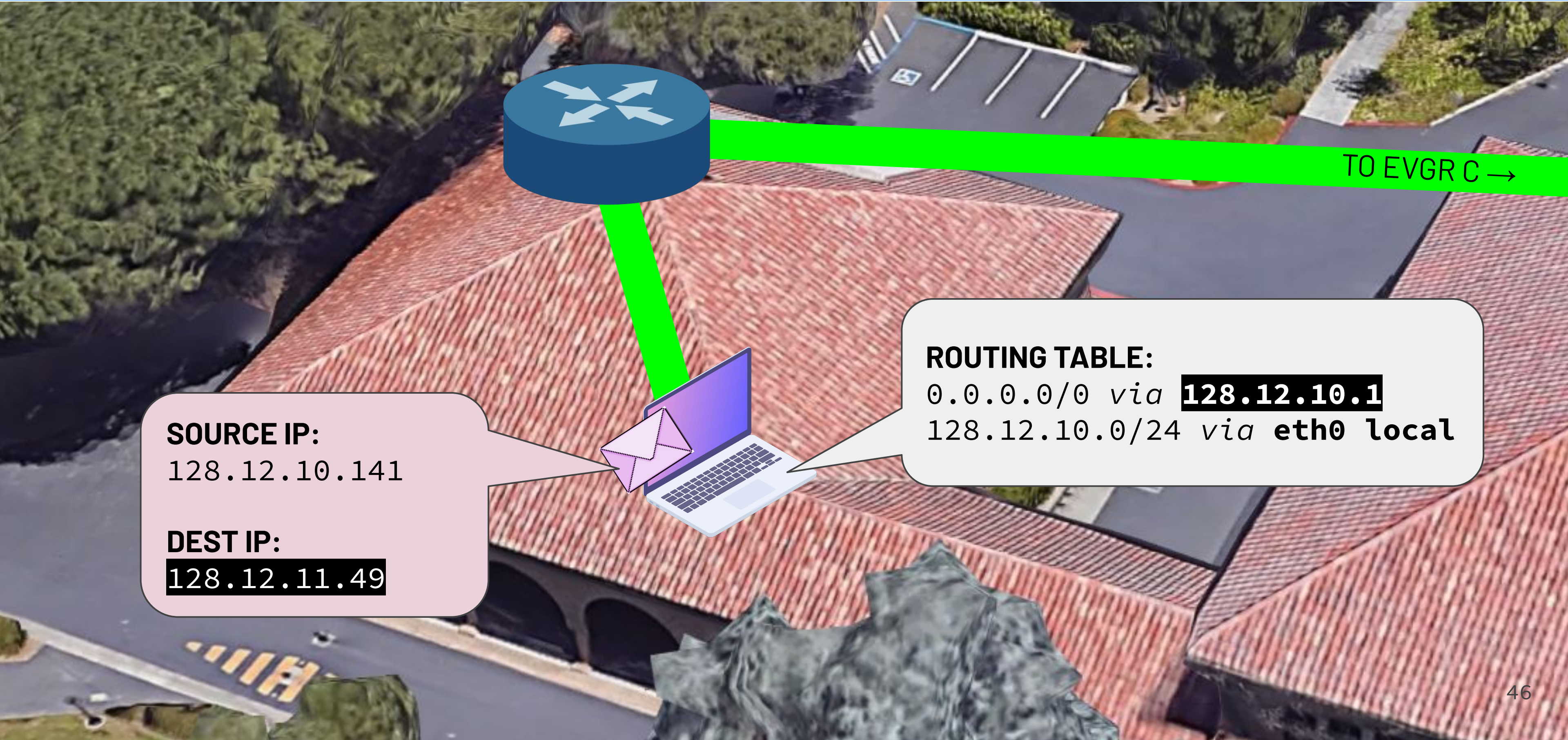
Visiting My Dorm Room: I



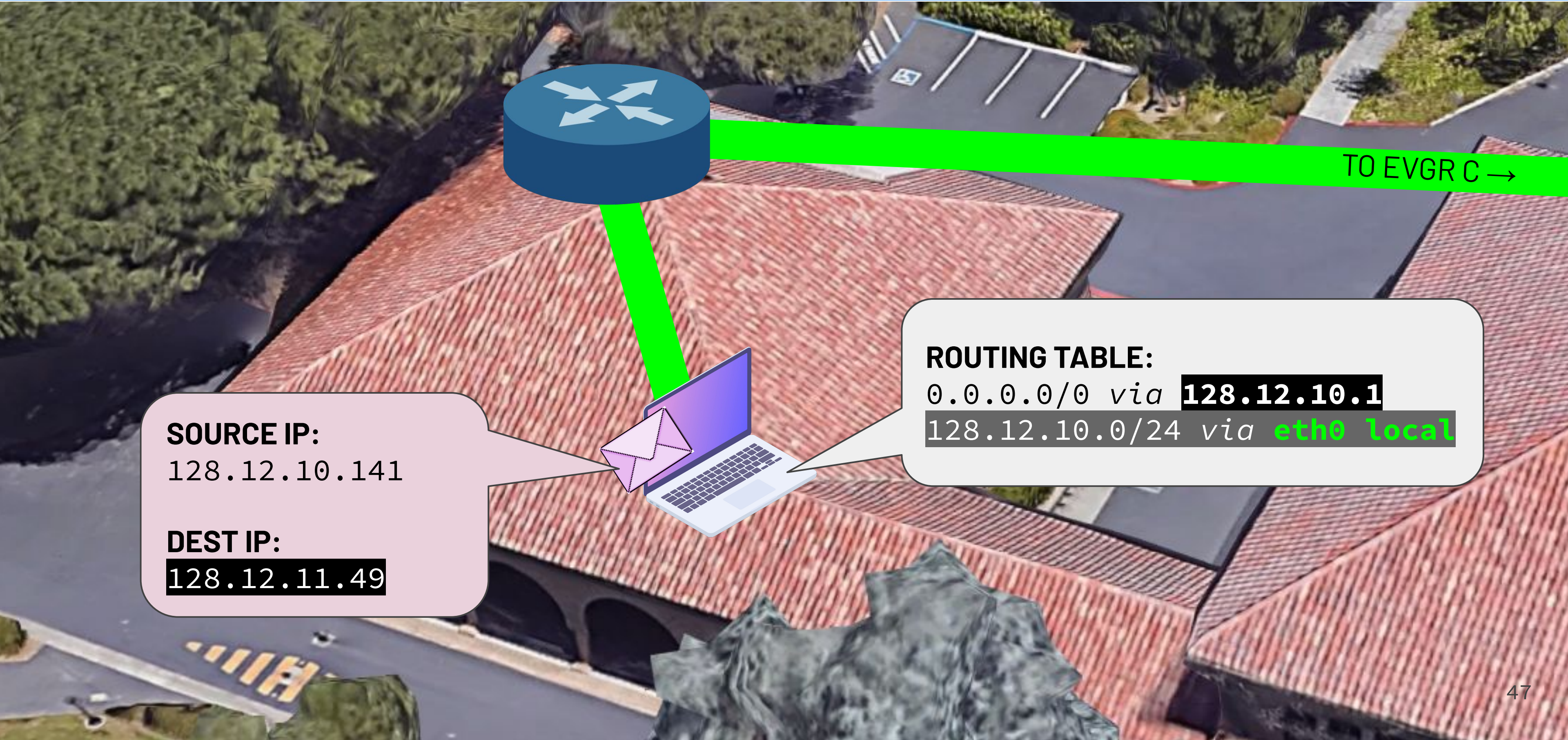
Visiting My Dorm Room: I



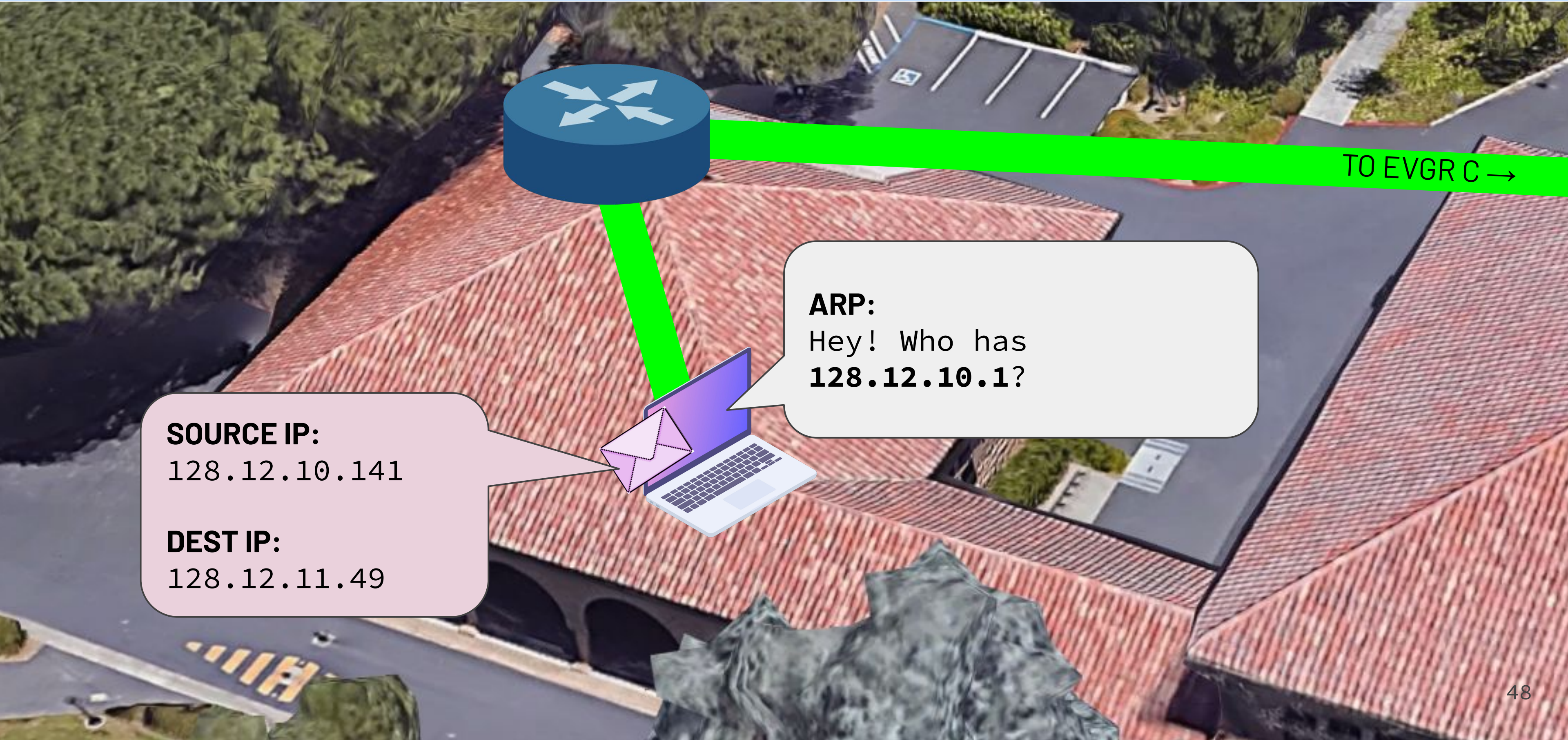
Visiting My Dorm Room: I



Visiting My Dorm Room: I



Visiting My Dorm Room: I



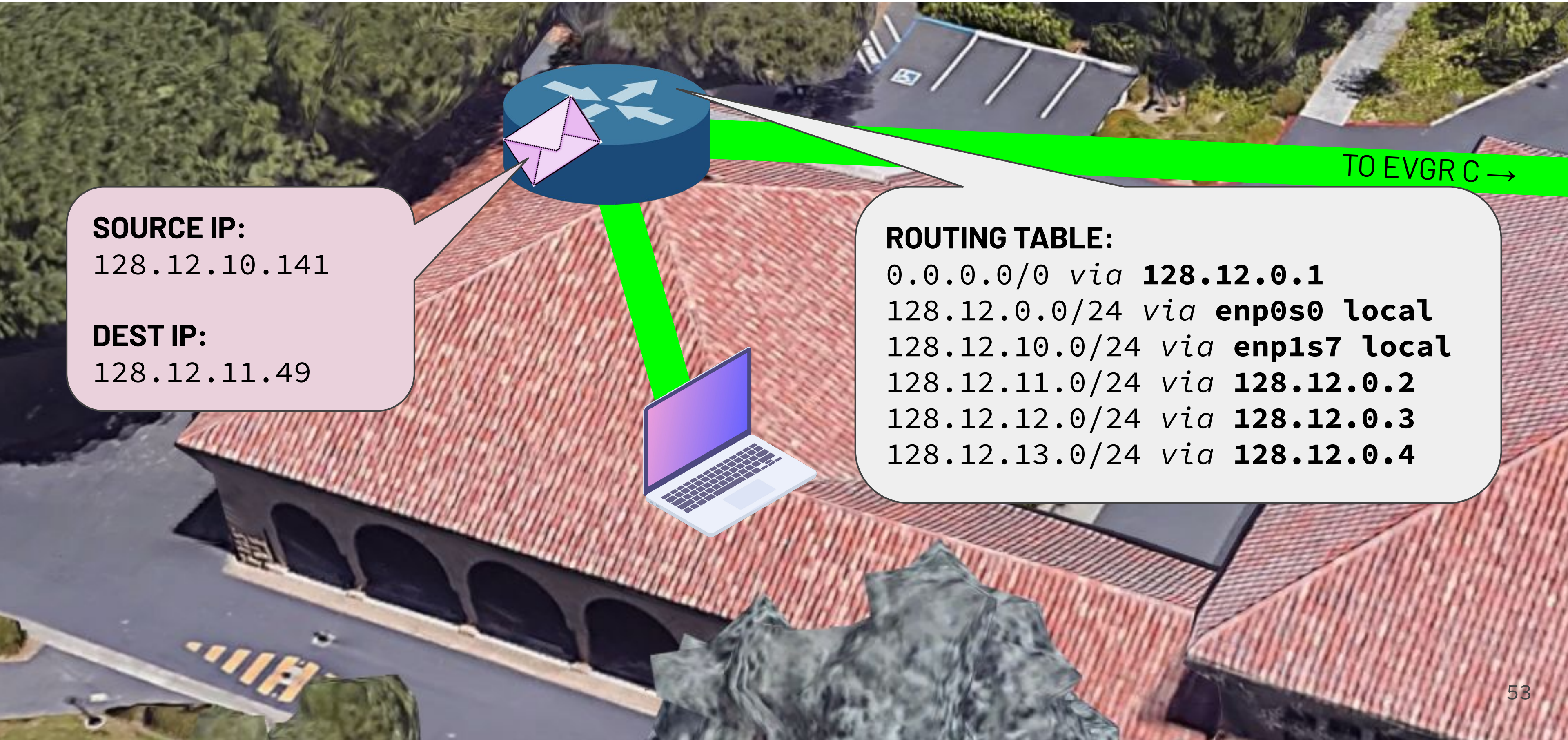
Visiting My Dorm Room: I

Visiting My Dorm Room: I

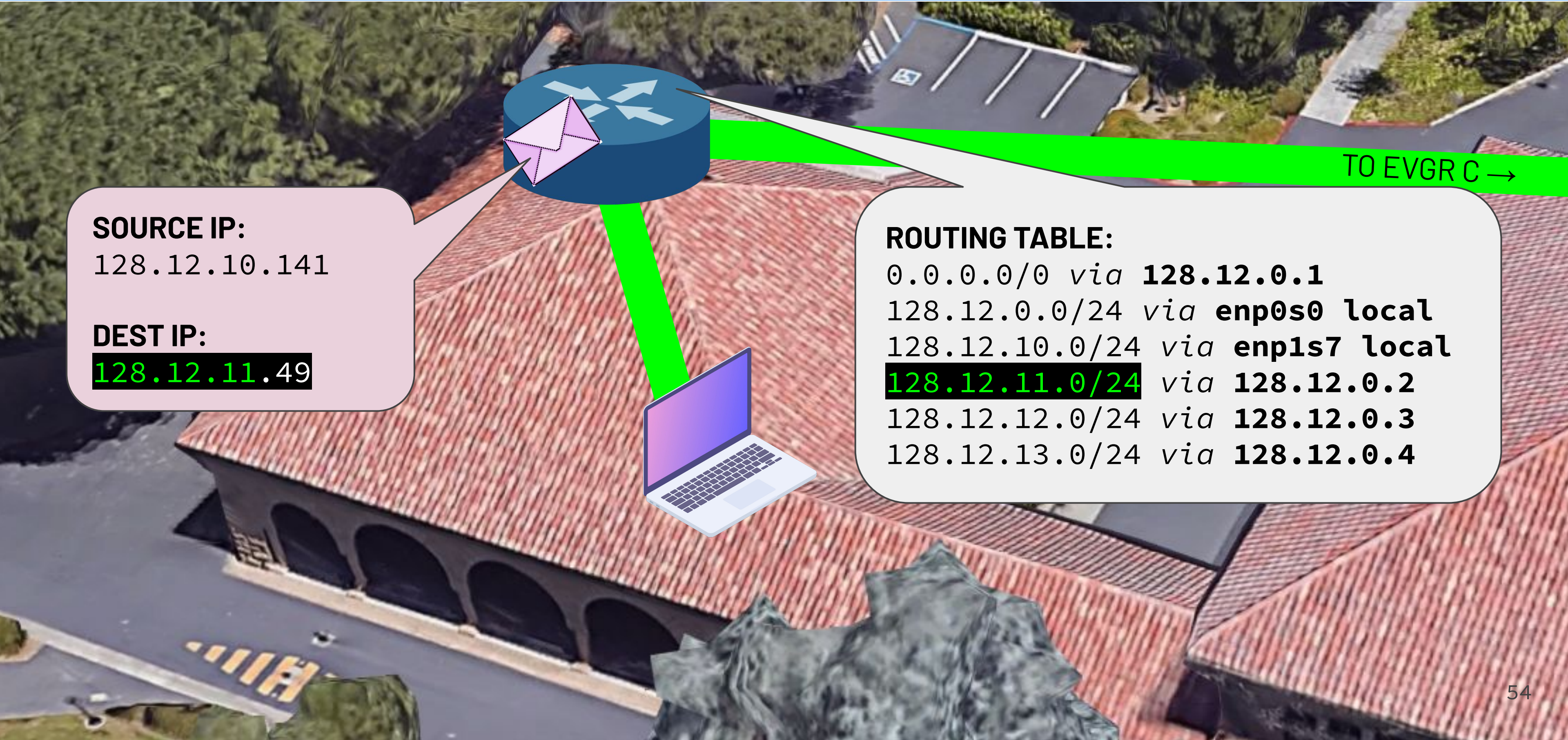
Visiting My Dorm Room: I

Visiting My Dorm Room: I

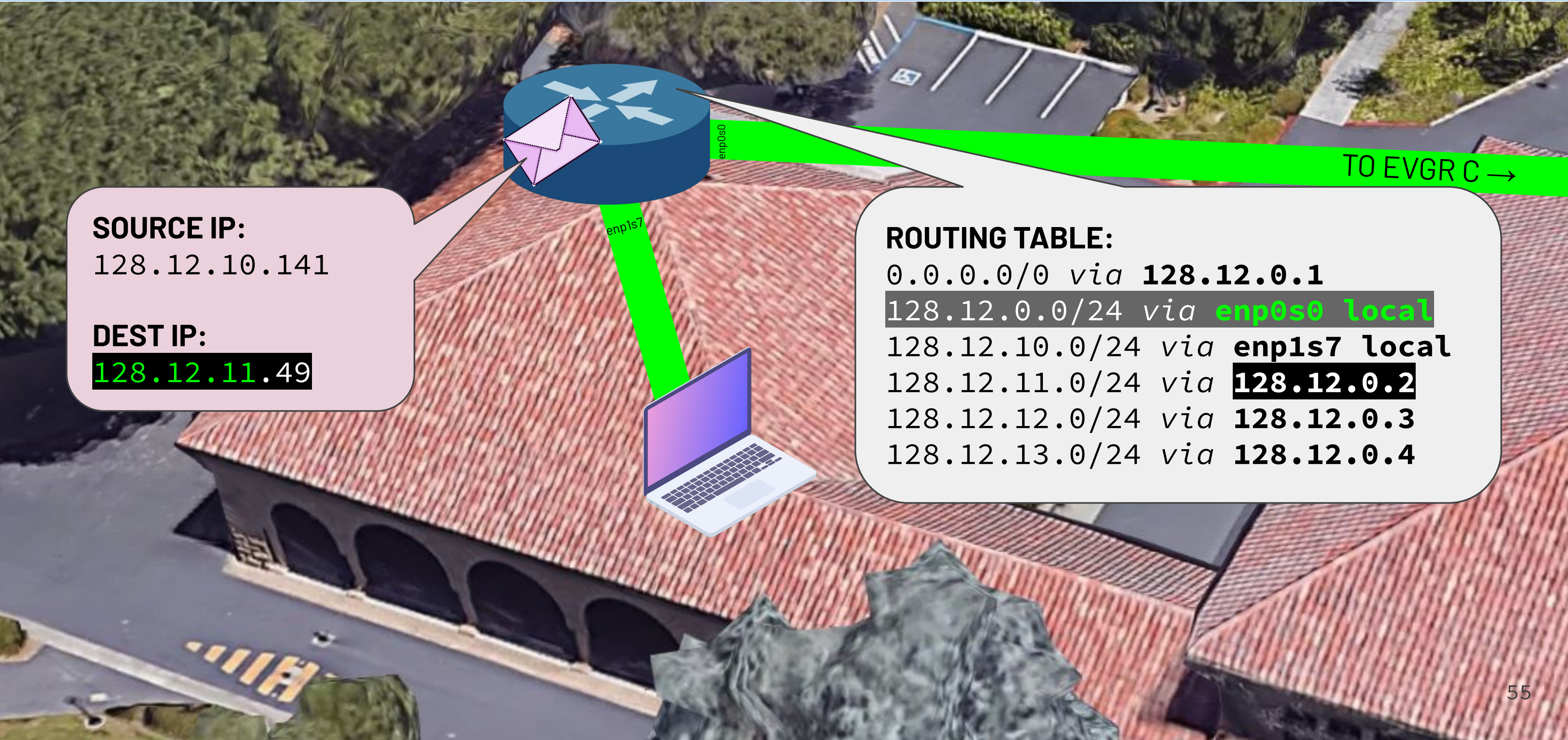
Visiting My Dorm Room: I



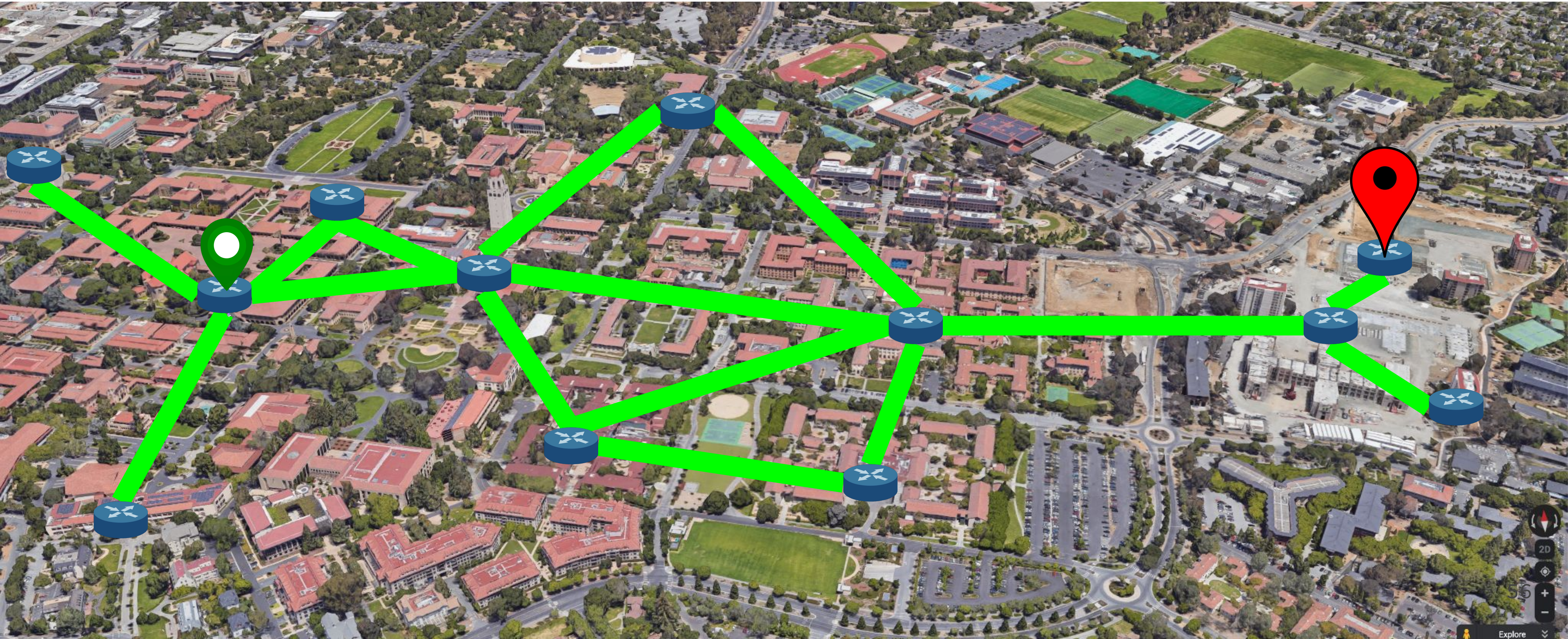
Visiting My Dorm Room: I



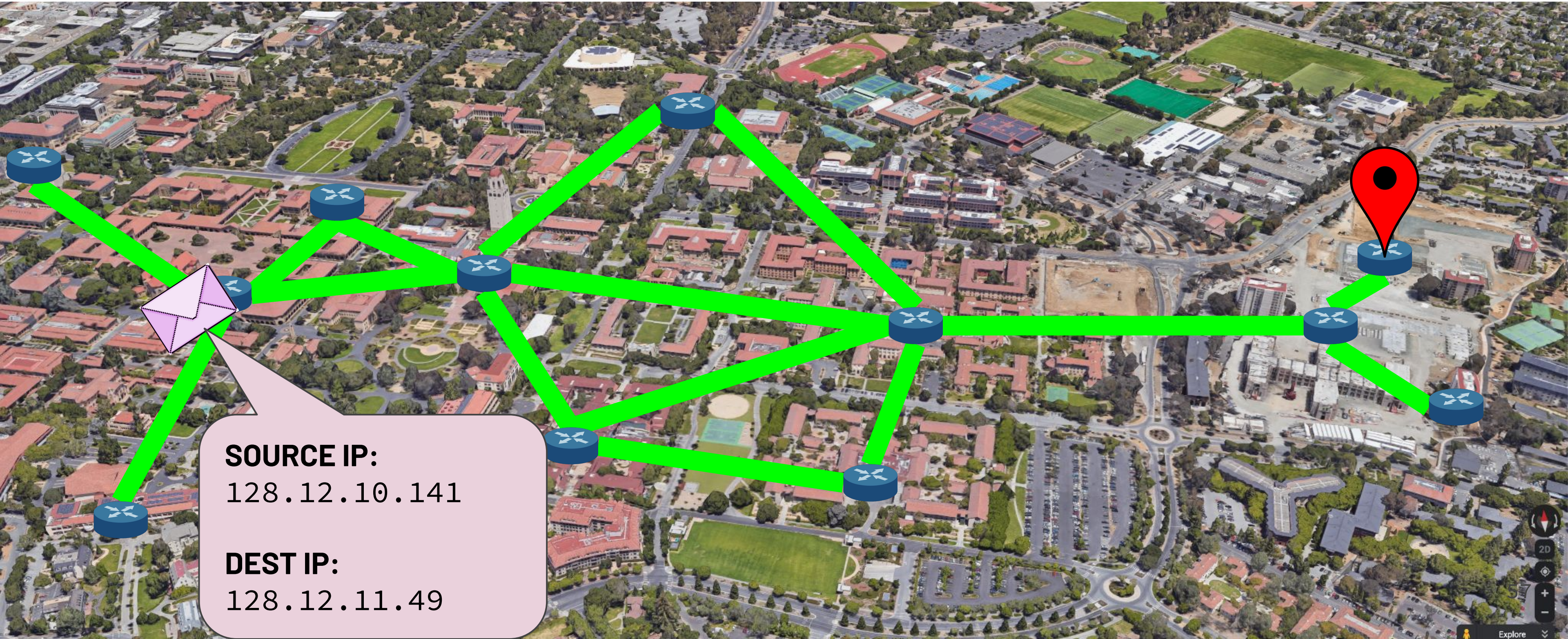
Visiting My Dorm Room: I



Visiting My Dorm Room: II



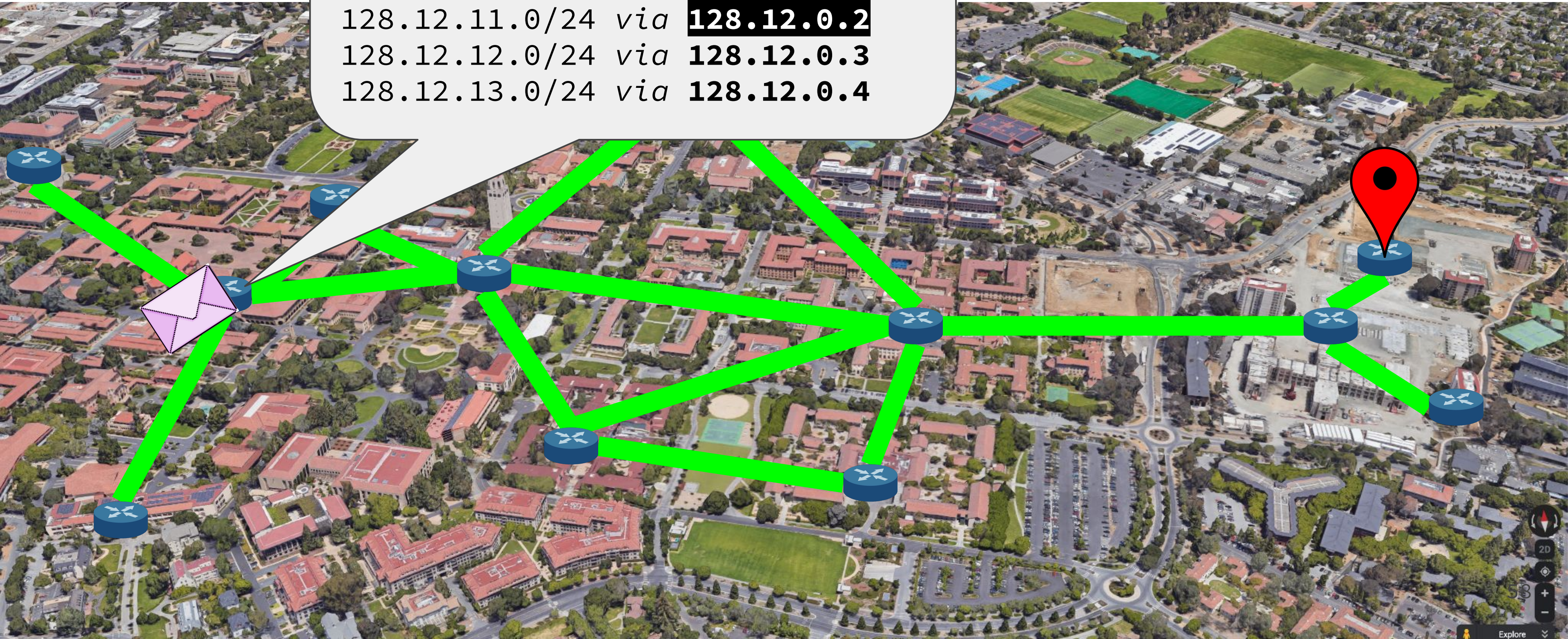
Visiting My Dorm Room: II



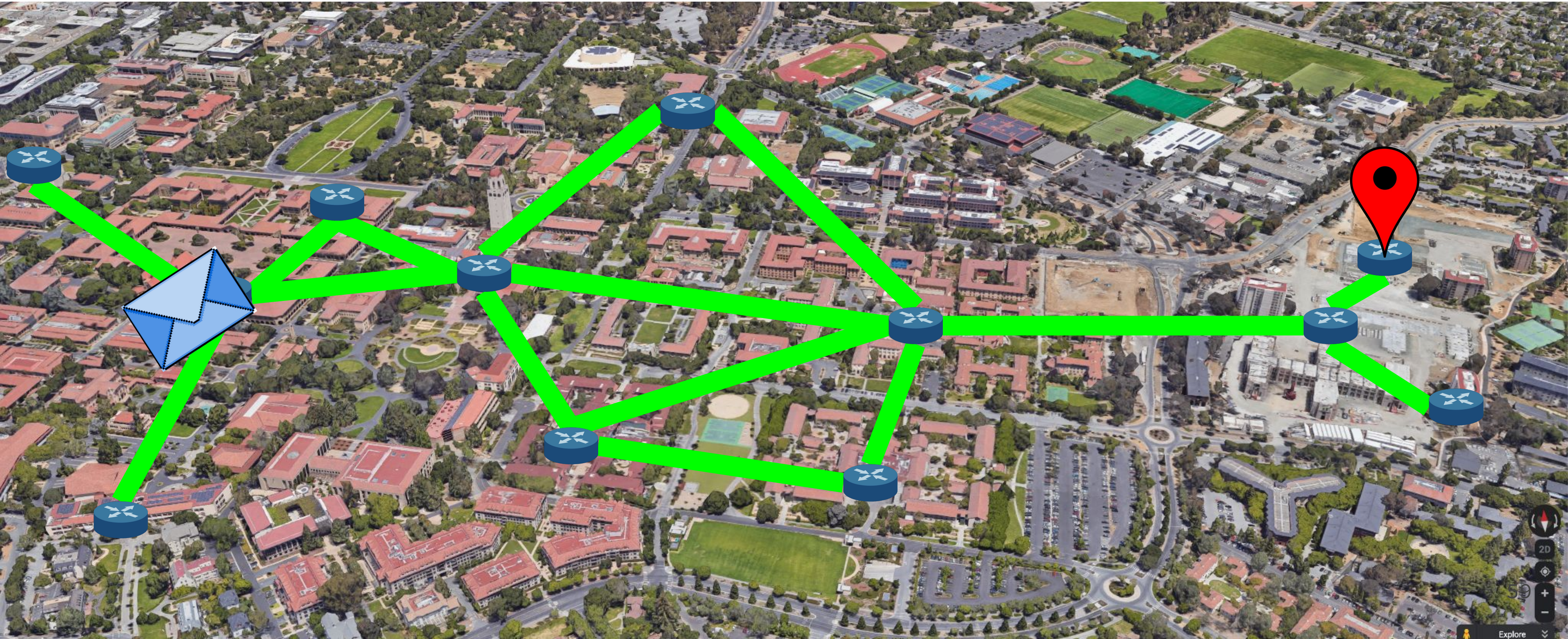
Visiting

ROUTING TABLE:

```
0.0.0.0/0 via 128.12.0.1
128.12.0.0/24 via enp0s0 local
128.12.10.0/24 via enp1s7 local
128.12.11.0/24 via 128.12.0.2
128.12.12.0/24 via 128.12.0.3
128.12.13.0/24 via 128.12.0.4
```

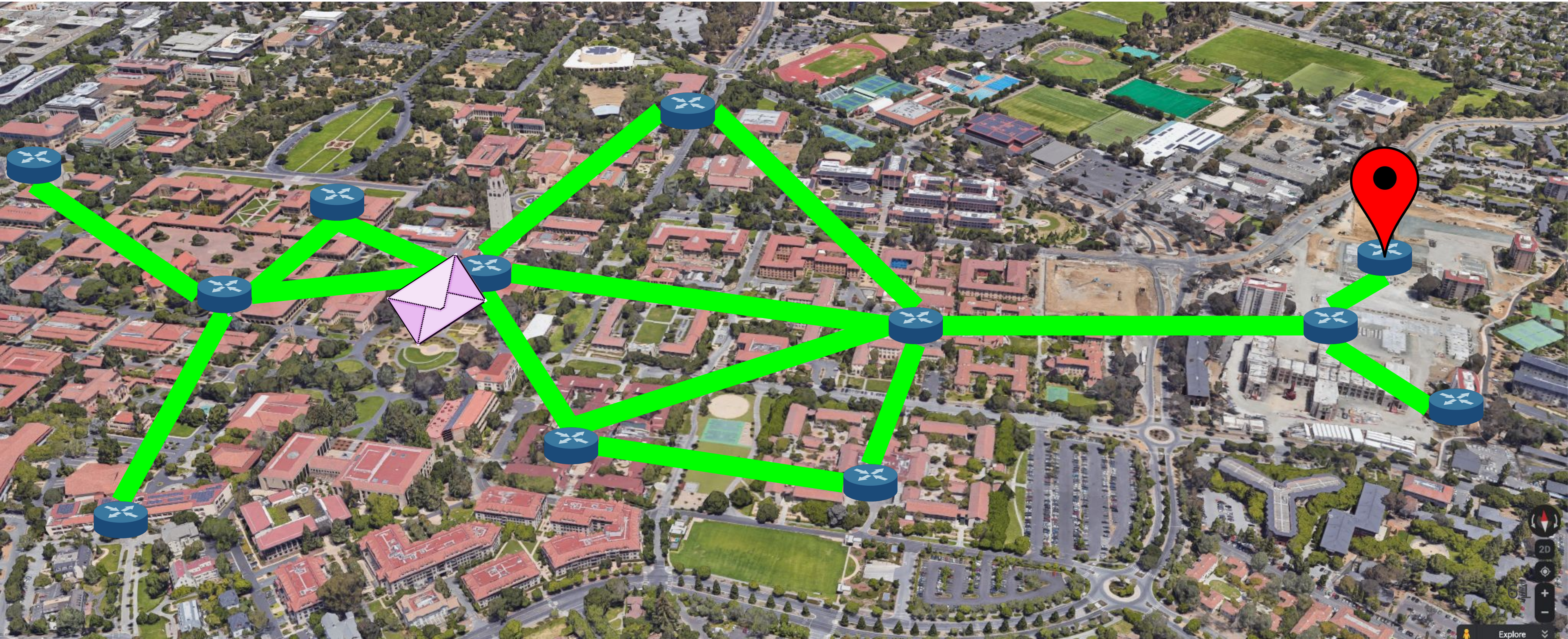


Visiting My Dorm Room: II



Visiting My Dorm Room: II

Visiting My Dorm Room: II

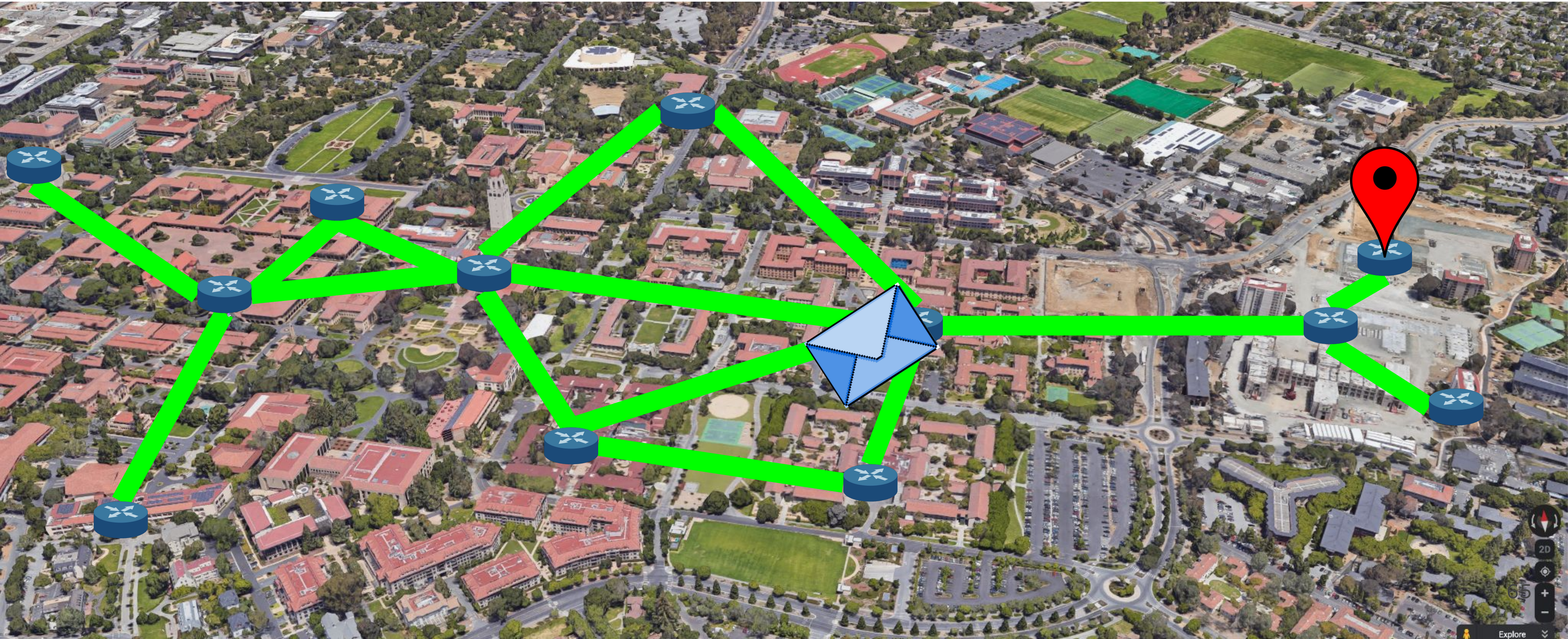


Visiting My Dorm Room: II

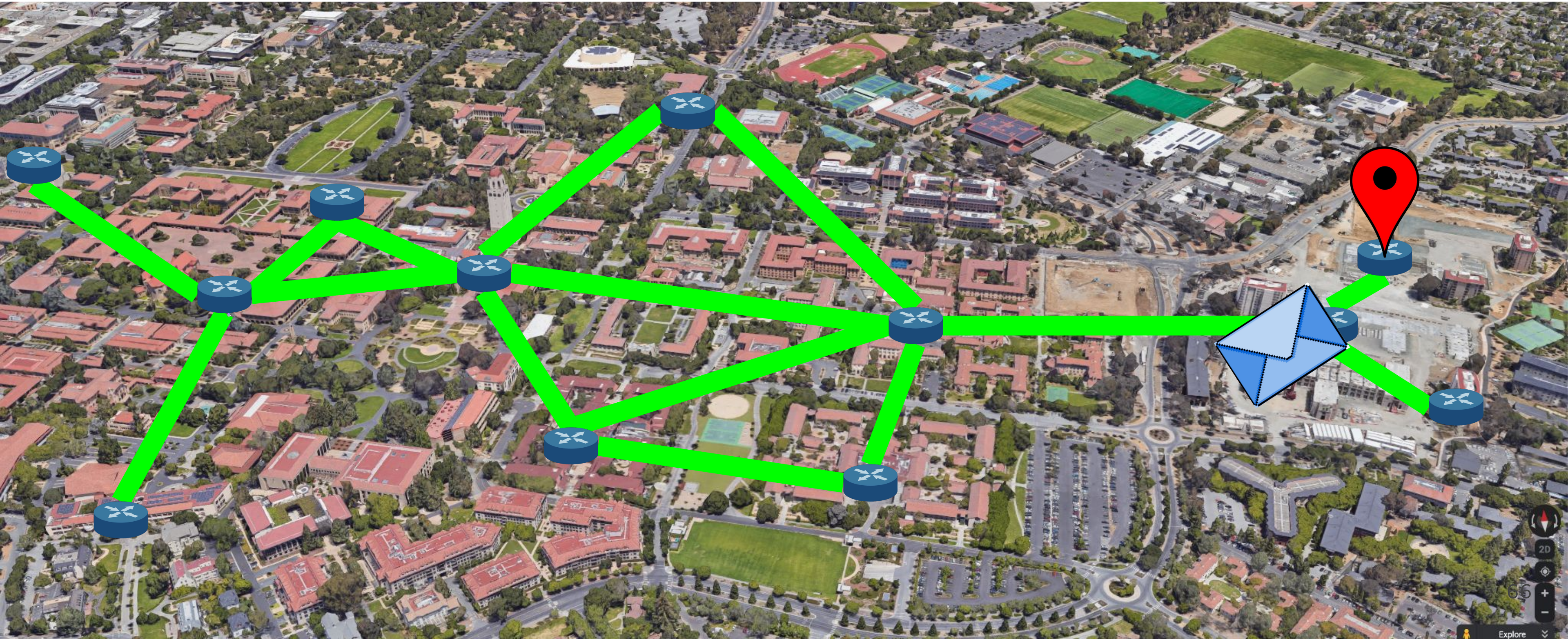
Visiting My Dorm Room: II

Visiting My Dorm Room: II

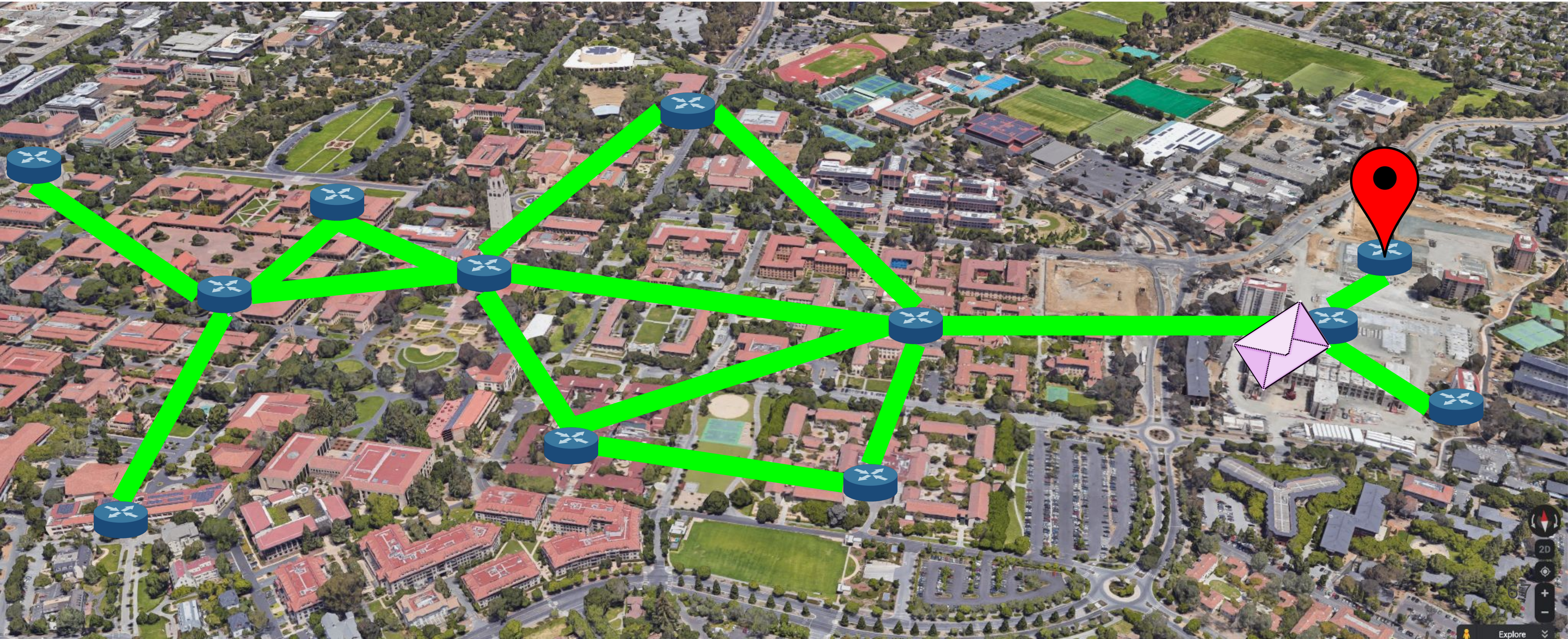
Visiting My Dorm Room: II



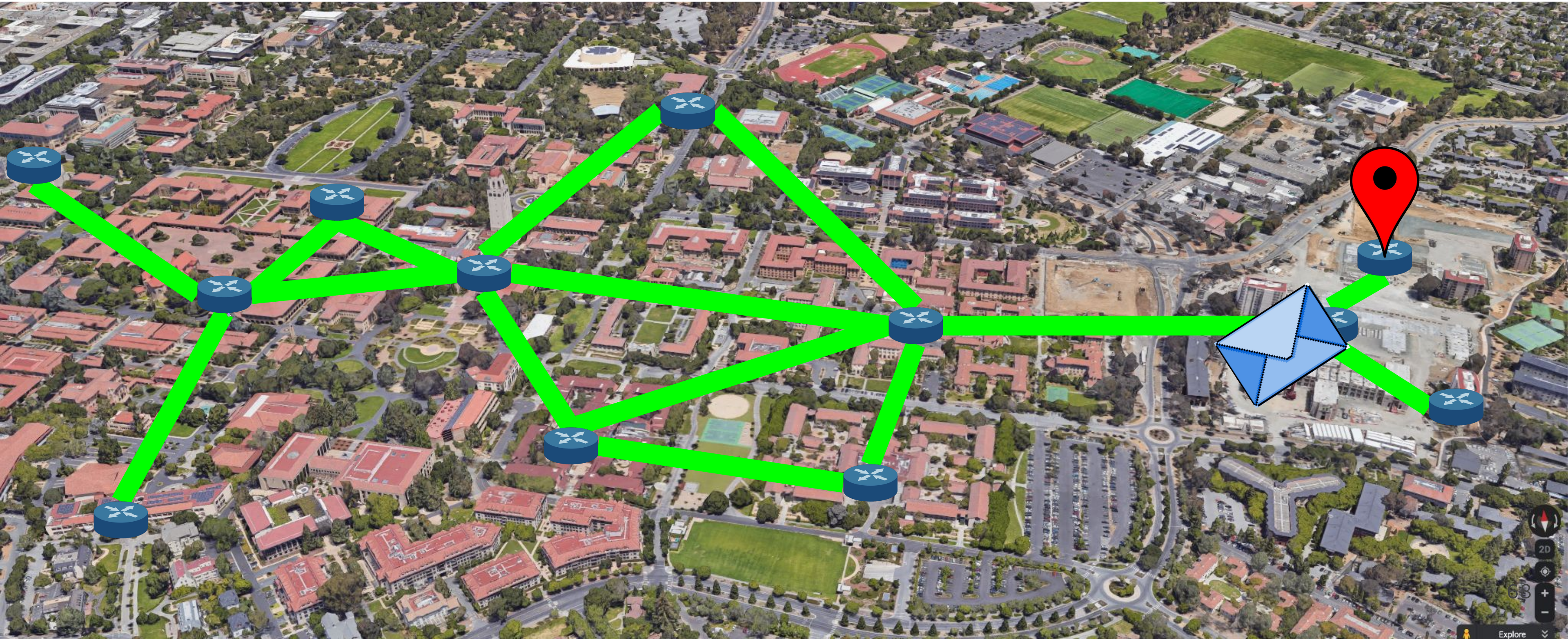
Visiting My Dorm Room: II



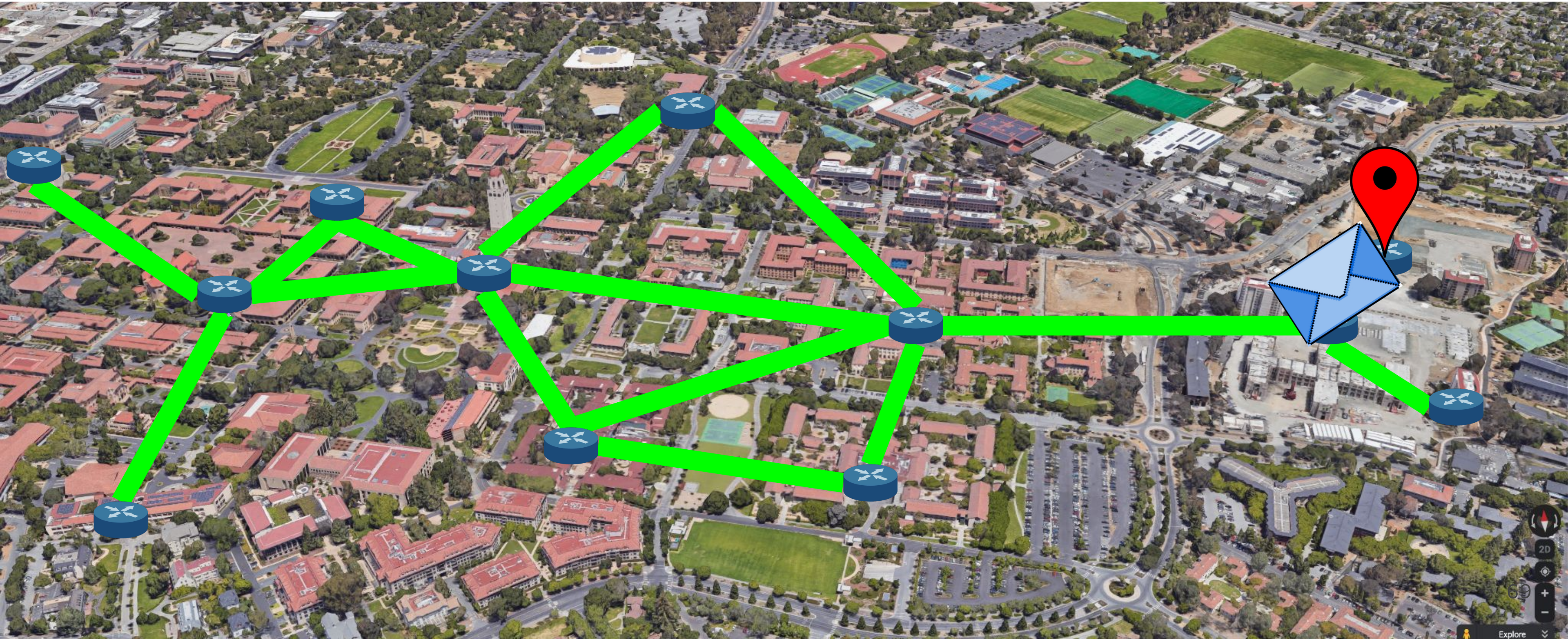
Visiting My Dorm Room: II



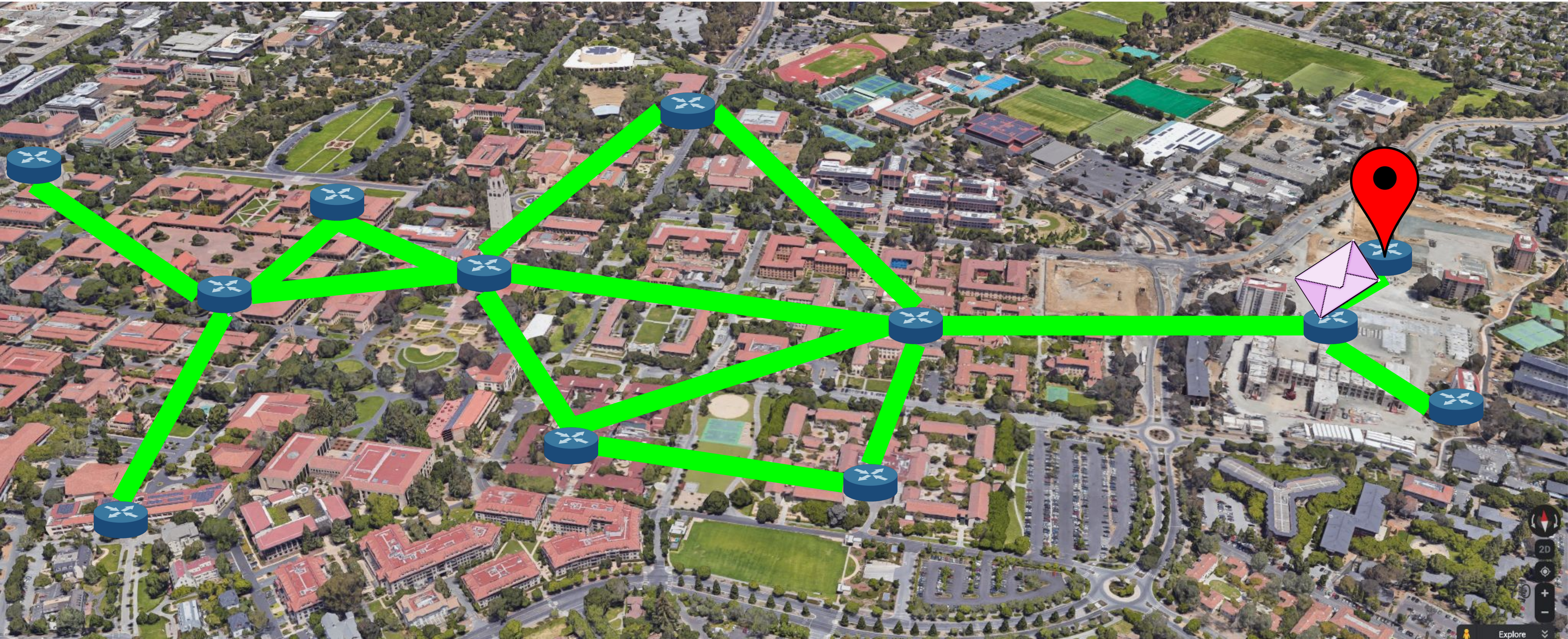
Visiting My Dorm Room: II



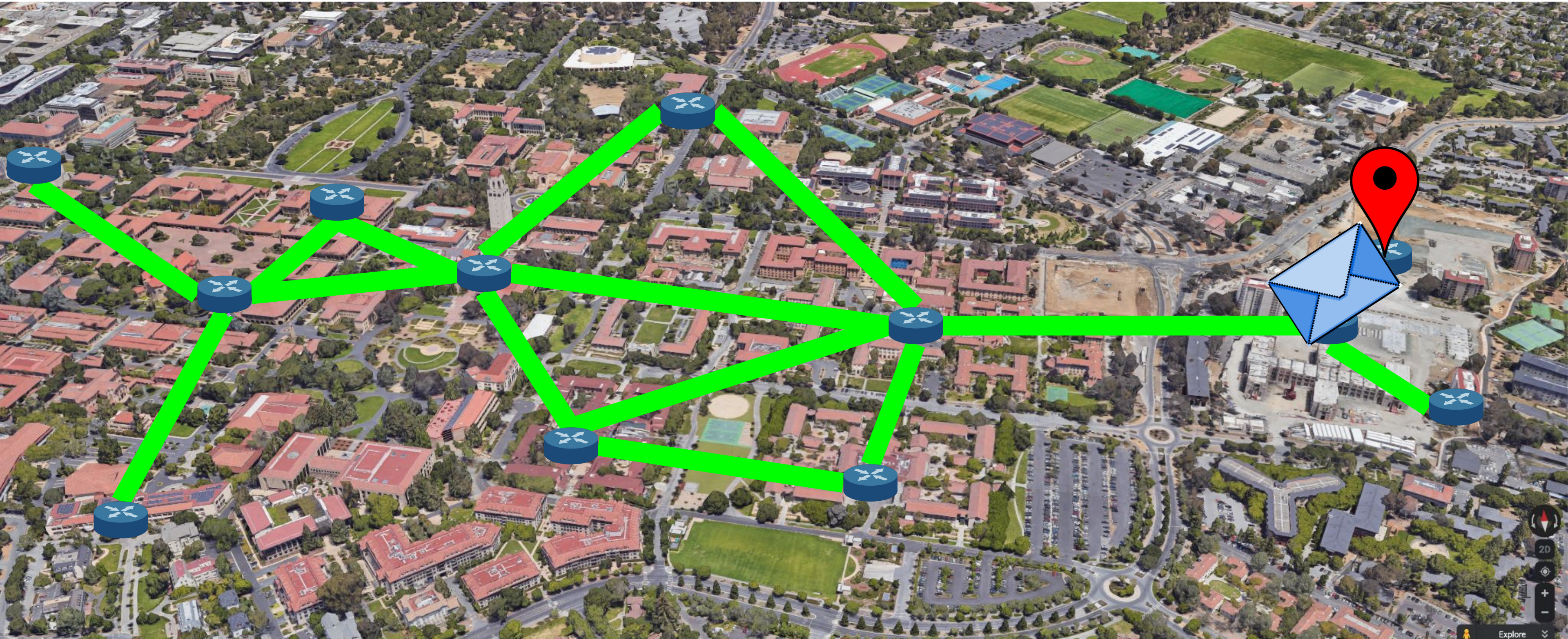
Visiting My Dorm Room: II



Visiting My Dorm Room: II

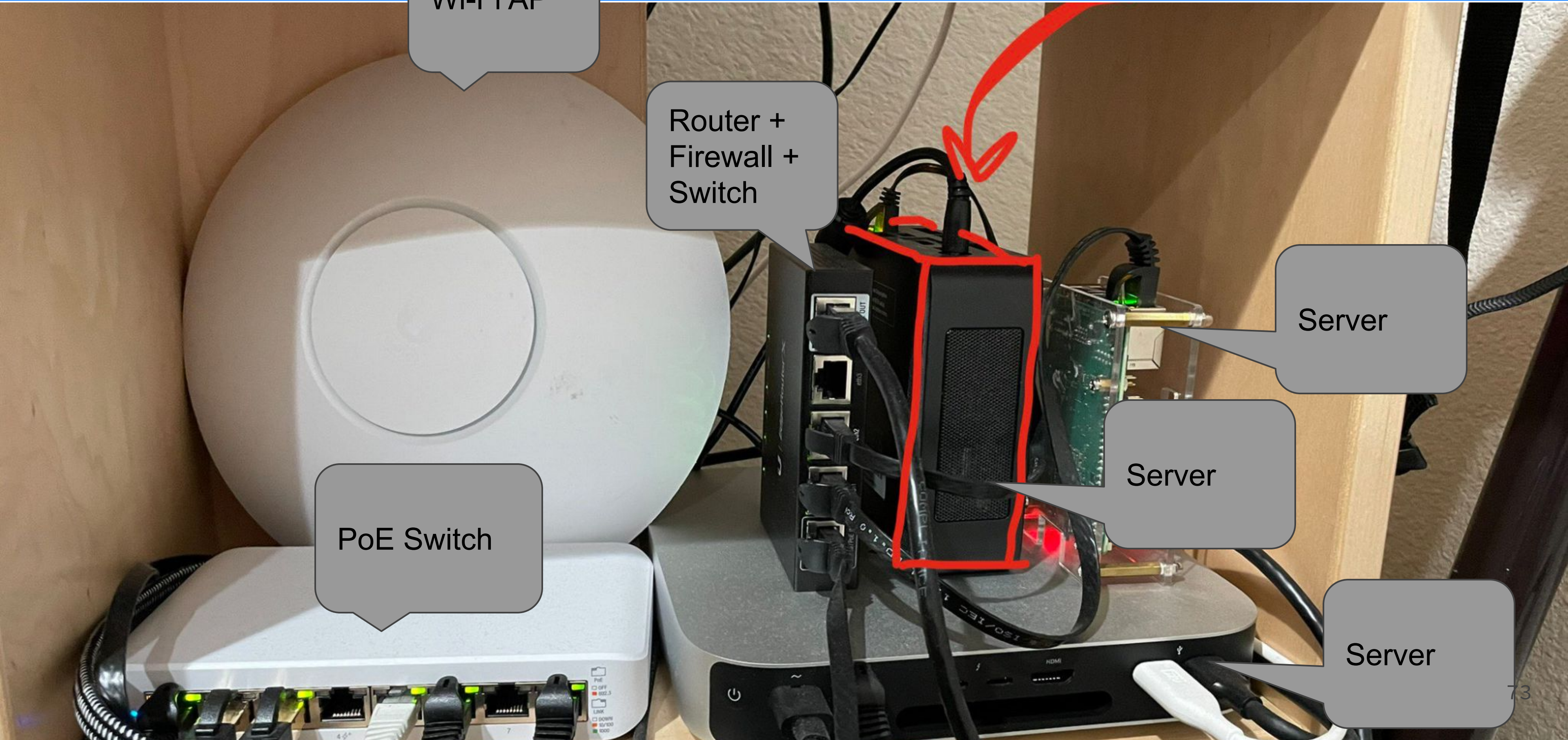


Visiting My Dorm Room: II

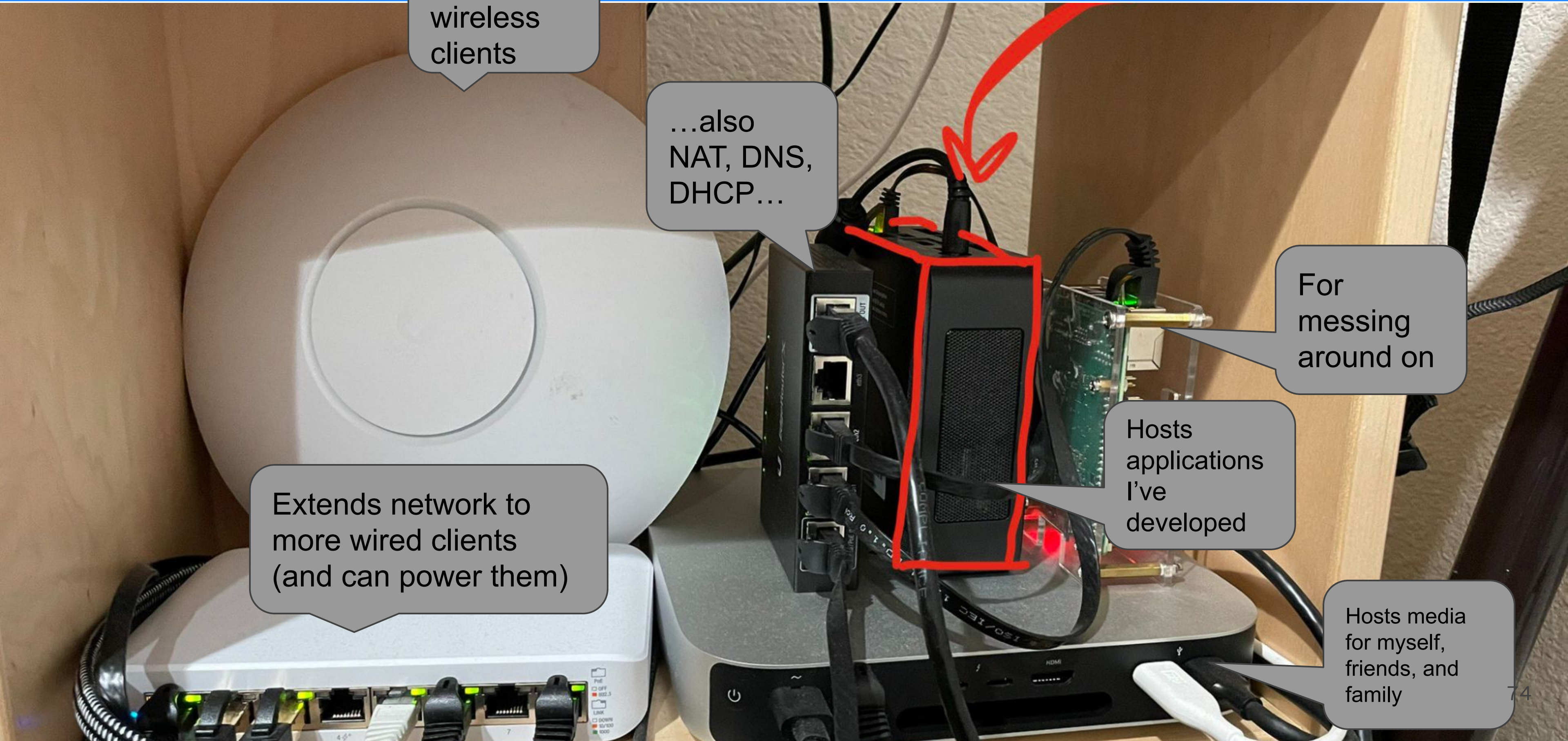


Visiting My Dorm Room: III

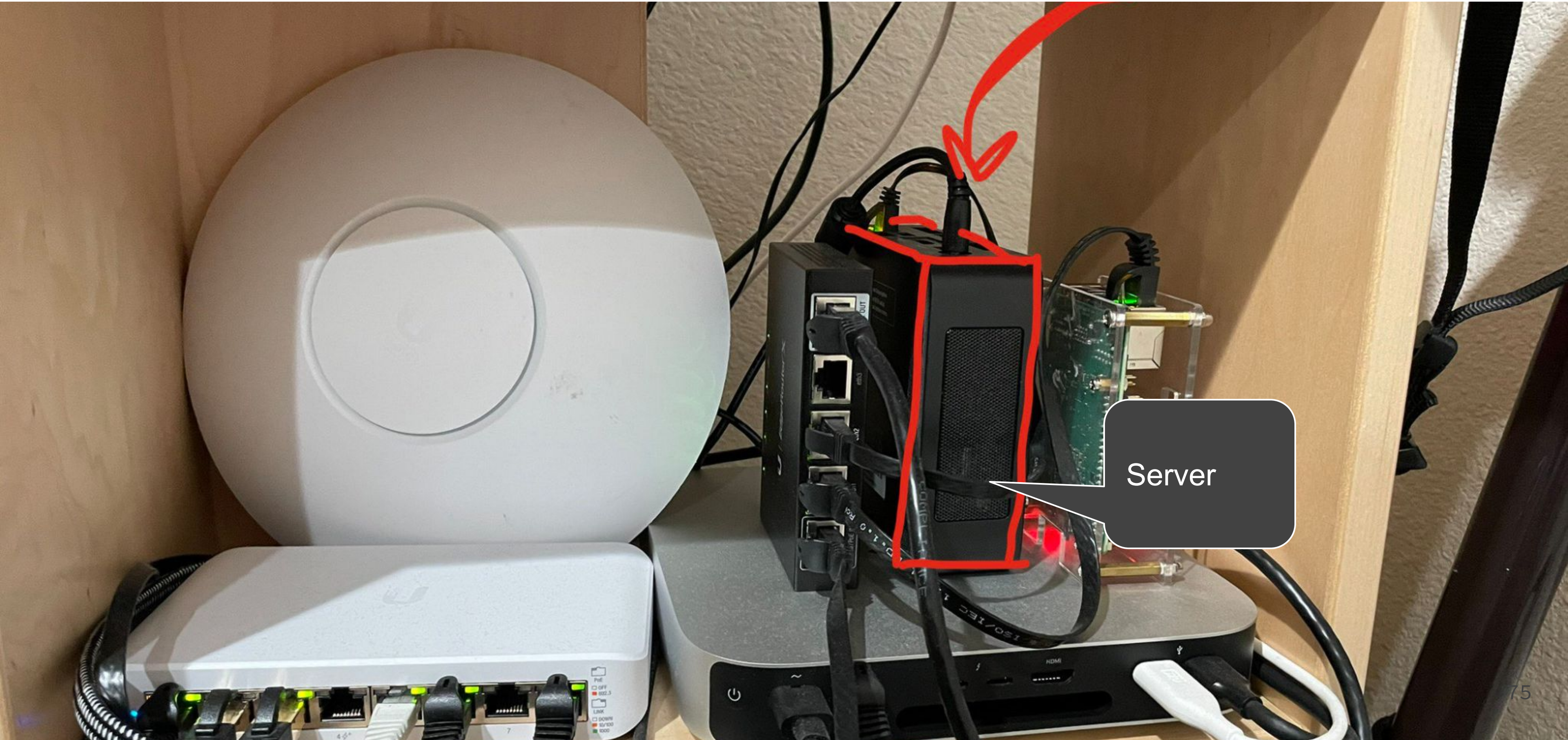
Visiting My Dorm Room: III



Visiting My Dorm Room: III

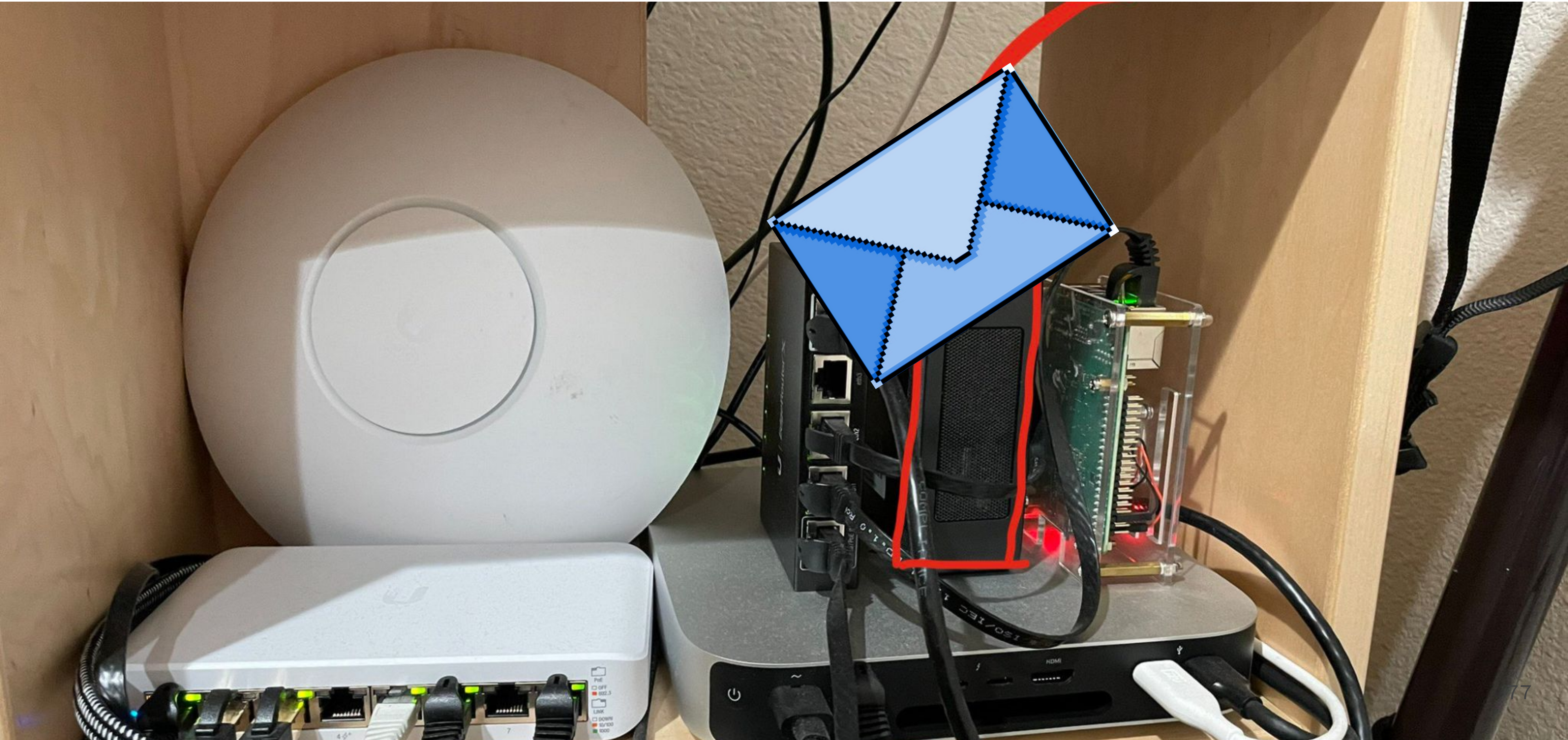


Visiting My Dorm Room: III

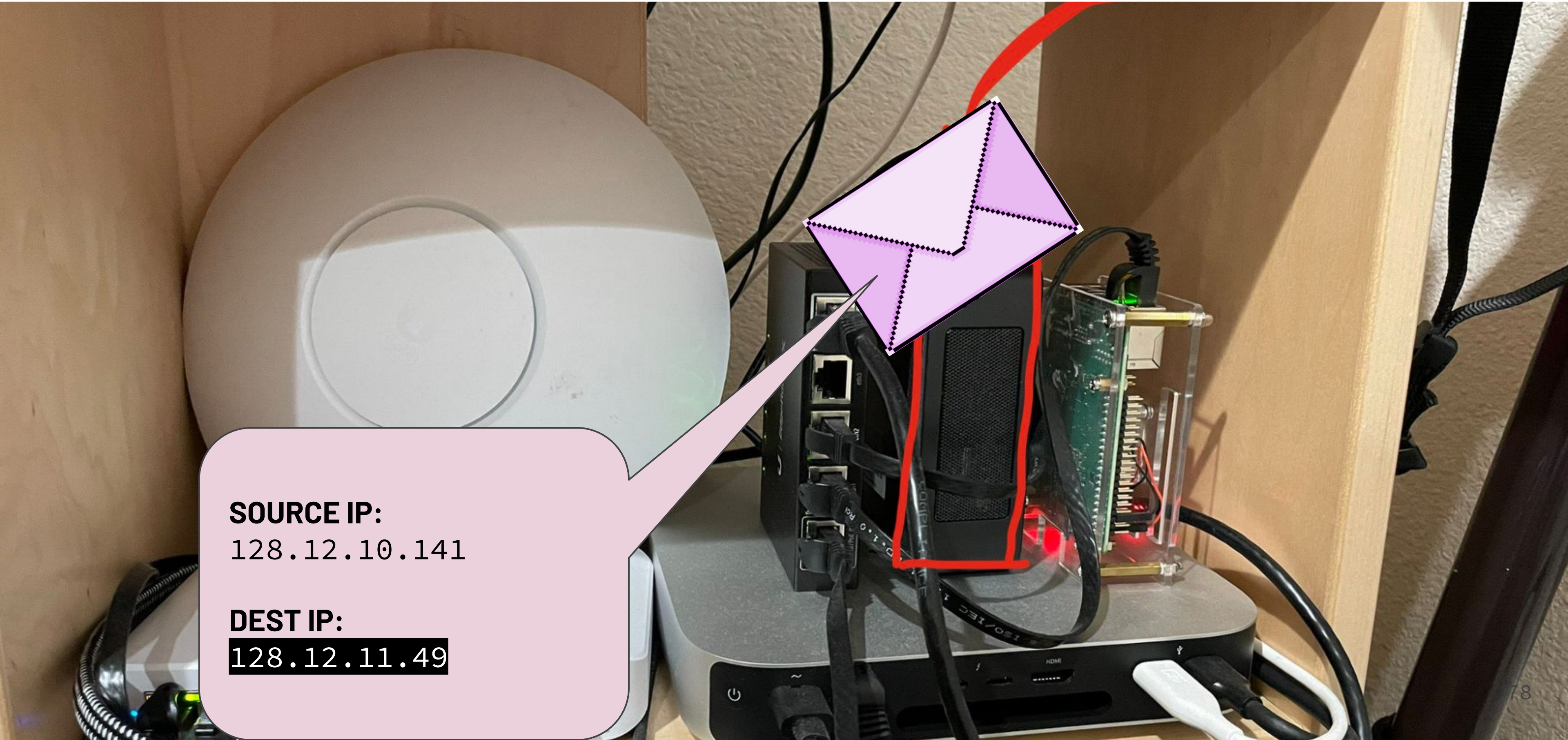


Visiting My Dorm Room: III

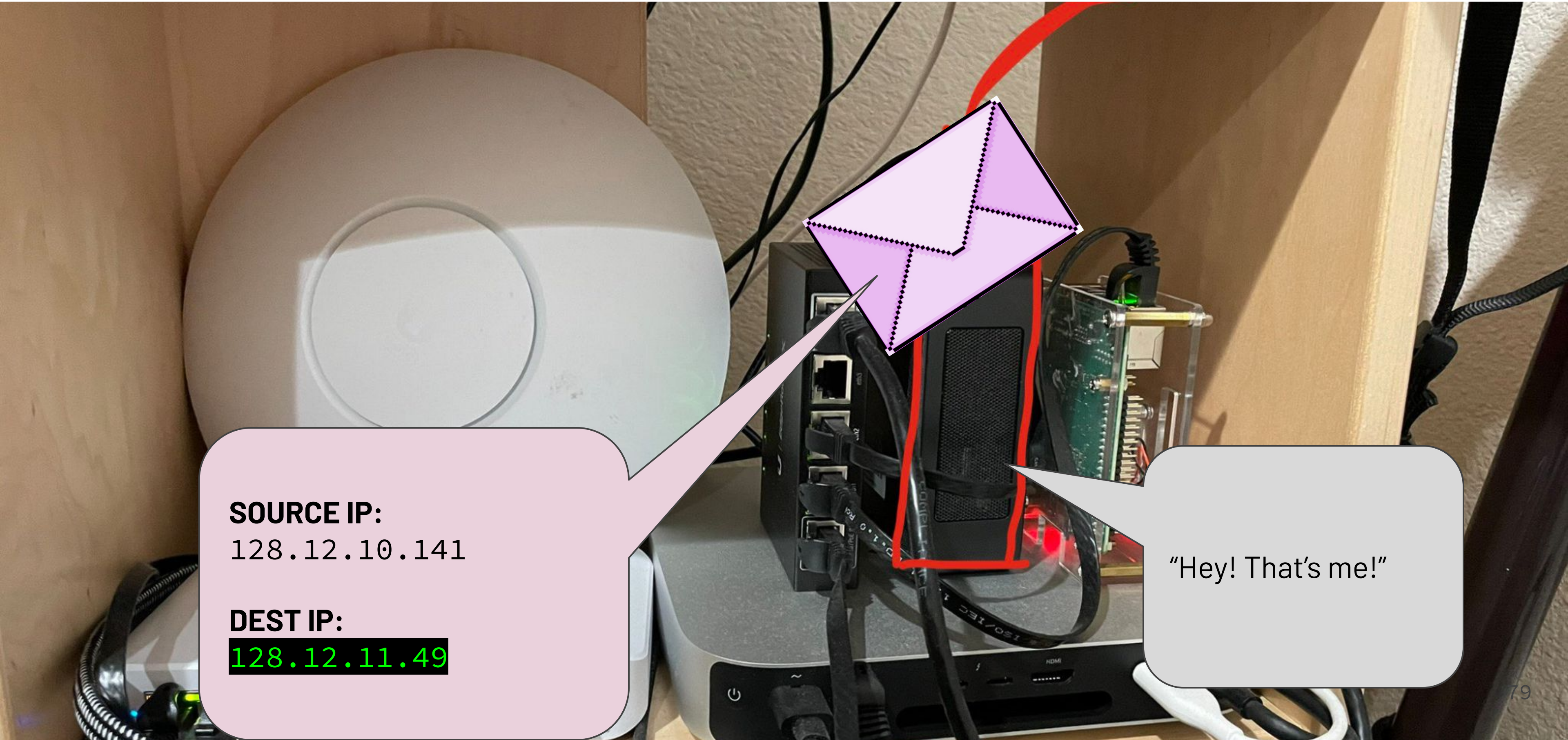
Visiting My Dorm Room: III



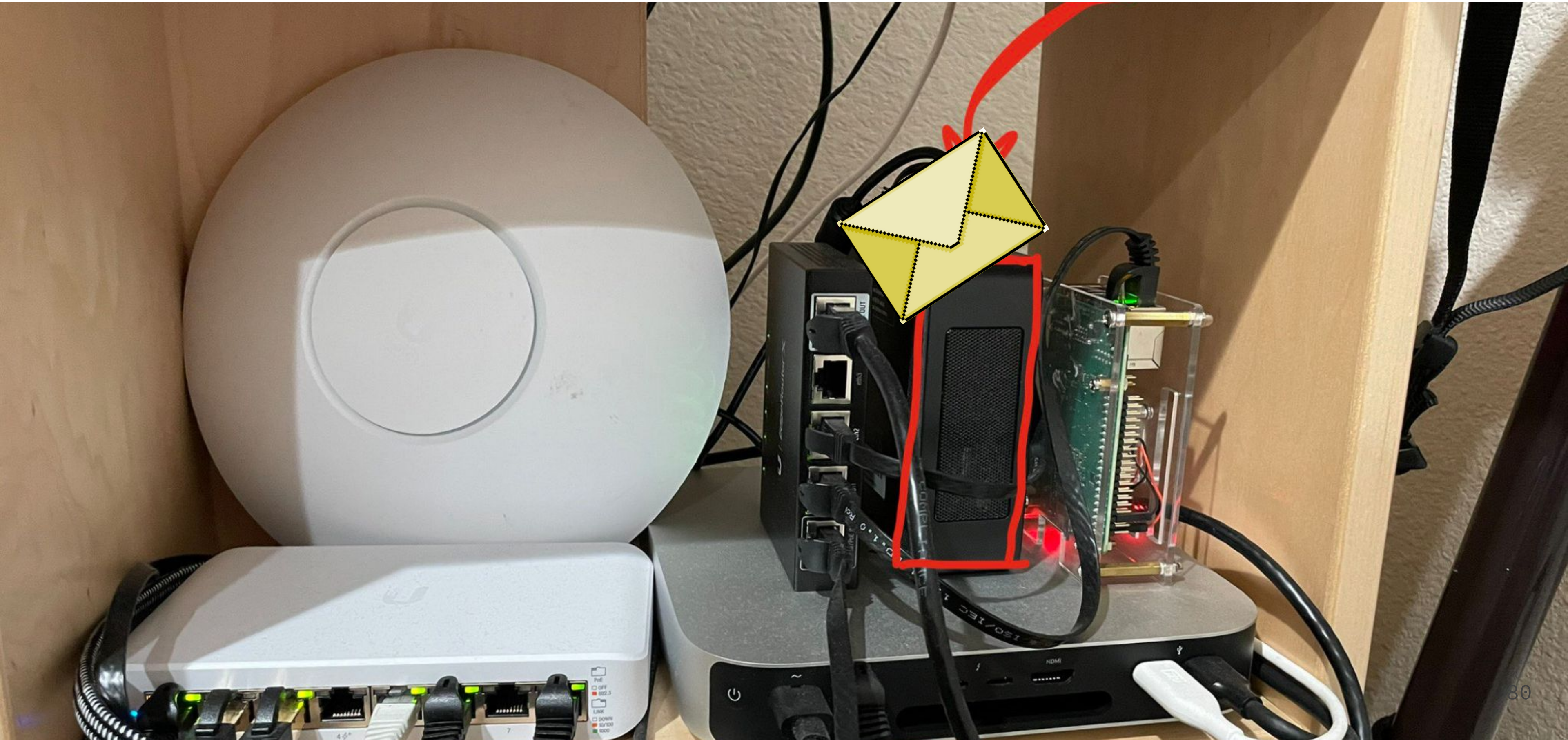
Visiting My Dorm Room: III



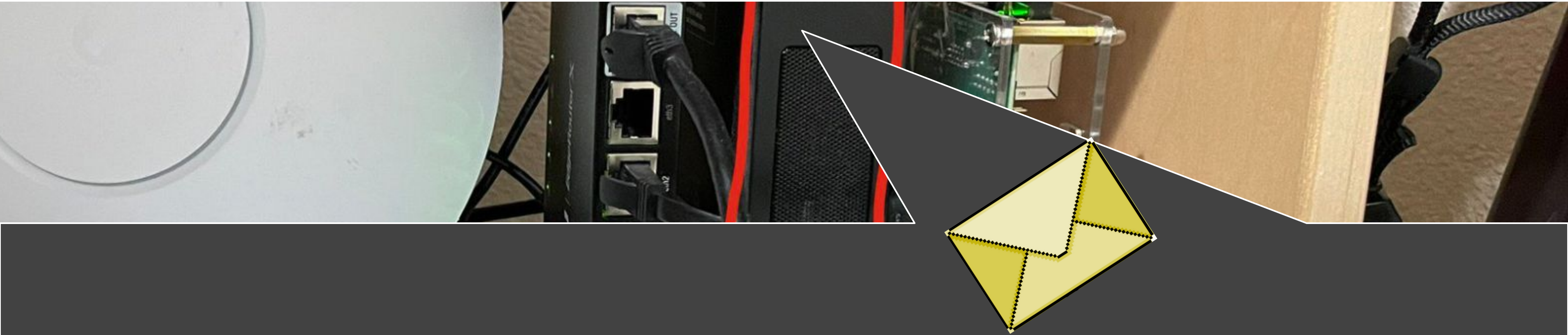
Visiting My Dorm Room: III



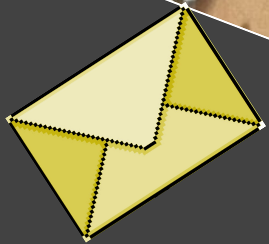
Visiting My Dorm Room: III



Visiting My Dorm Room: III



Visiting My Dorm Room: III

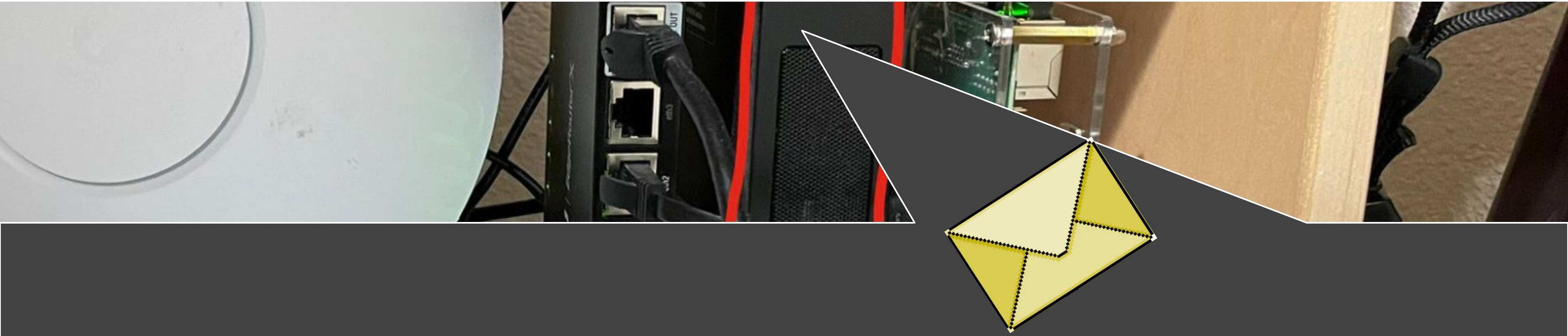


WEB SERVER

MINECRAFT SERVER

SSH SERVER

Visiting My Dorm Room: III



WEB SERVER
LISTEN: TCP PORT 80
LISTEN: TCP PORT 443

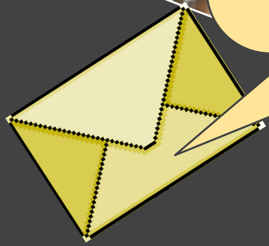
MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

Visiting My Dorm Room: III

SOURCE PORT:
43224

DEST PORT:
80

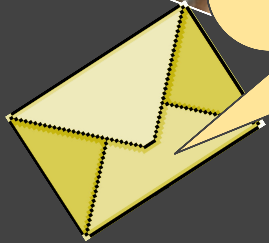


WEB SERVER
LISTEN: TCP PORT 80
LISTEN: TCP PORT 443

MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

Visiting My Dorm Room: III



SOURCE PORT:
43224

DEST PORT:
80

WEB SERVER
LISTEN: TCP PORT 80
LISTEN: TCP PORT 443

MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

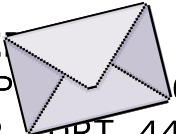
Visiting My Dorm Room: III

WEB SERVER
LISTEN: TCP PORT 80
LISTEN: TCP PORT 443

MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

Visiting My Dorm Room: III



WEB SE[illegible]
LISTEN: TCP [illegible]0
LISTEN: TCP [illegible]ORT 443

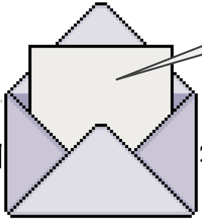
MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

Visiting My Dorm Room: III

"Hello World!"

WEB
LISTEN: 80
LISTEN: T 443



MINECRAFT SERVER
LISTEN: TCP PORT 25565

SSH SERVER
LISTEN: TCP PORT 22

Now You Try

Ping

The **ping** tool is useful for checking if a host is alive and reachable.

```
$ ping <destination>
$ ping 128.12.11.49
```

Traceroute

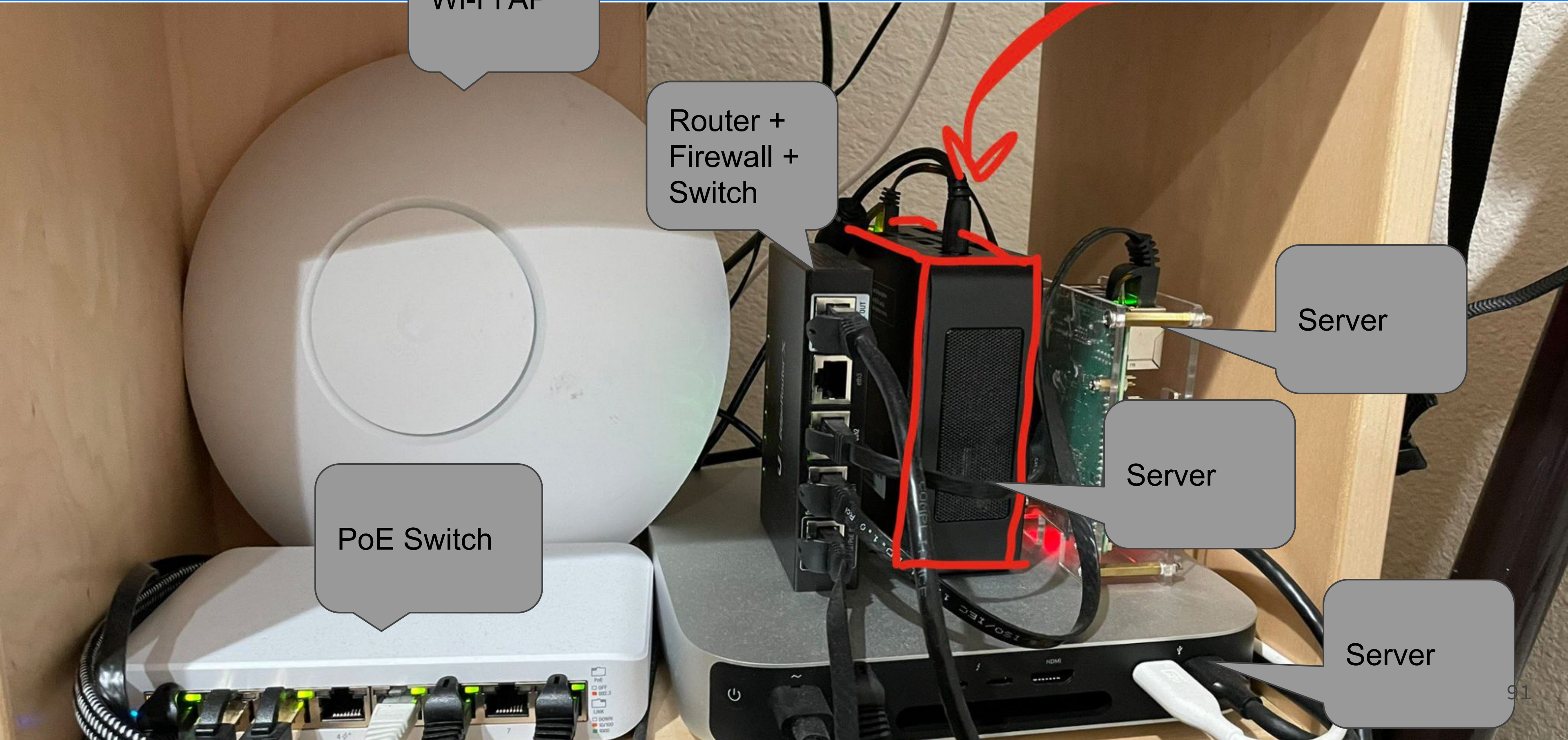
The **traceroute** (**tracert** on Windows) tool is useful for checking the path/**route** to a given host.

```
$ traceroute <destination>
$ traceroute 128.12.11.49
```

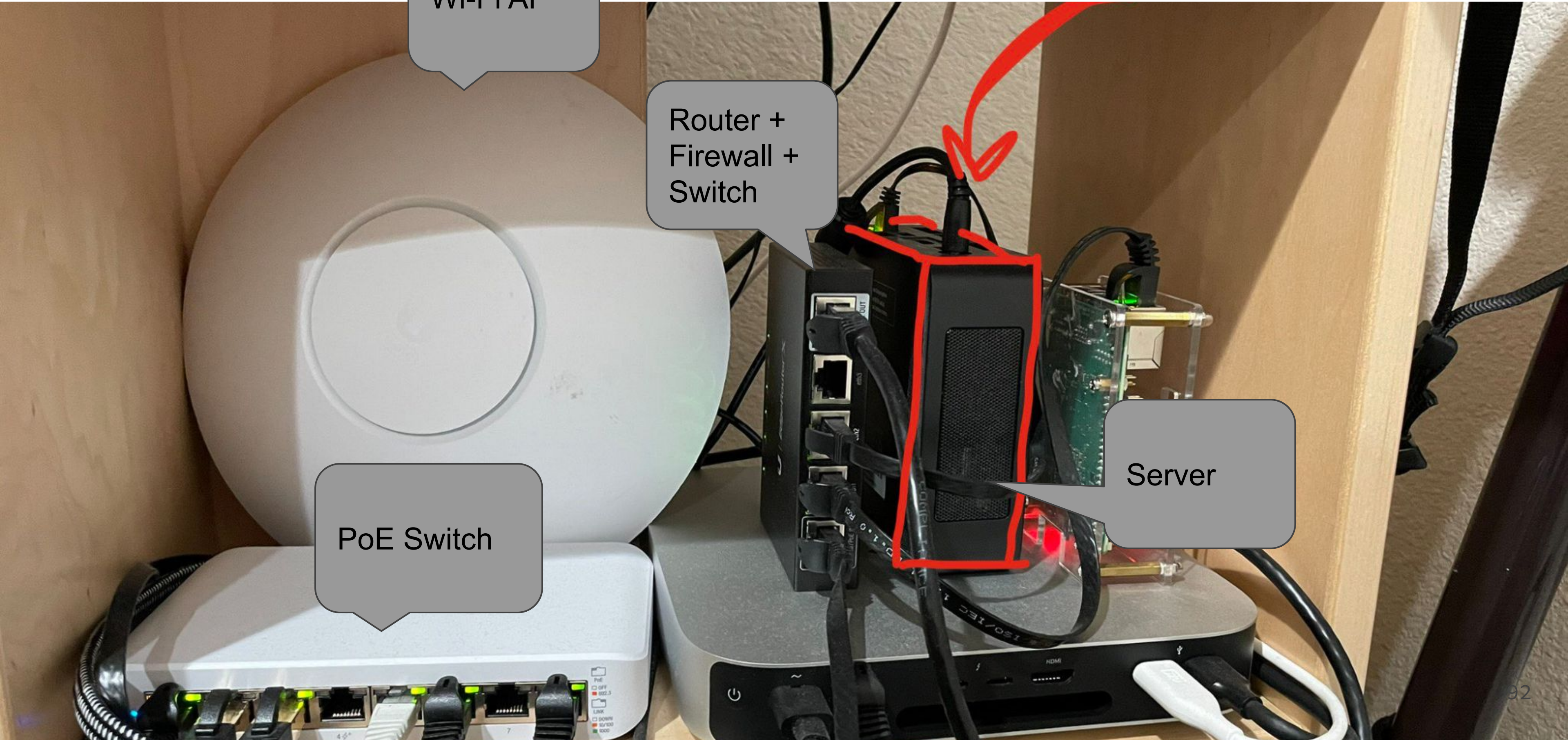
Delving Deeper

[demo wireshark]

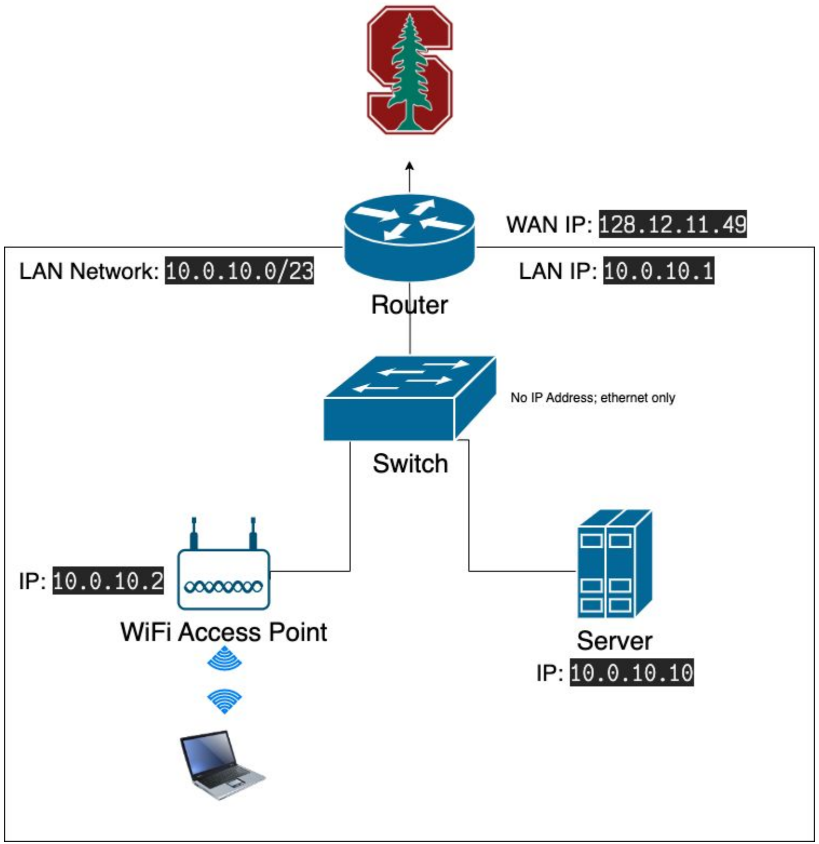
Going Deeper On My Network



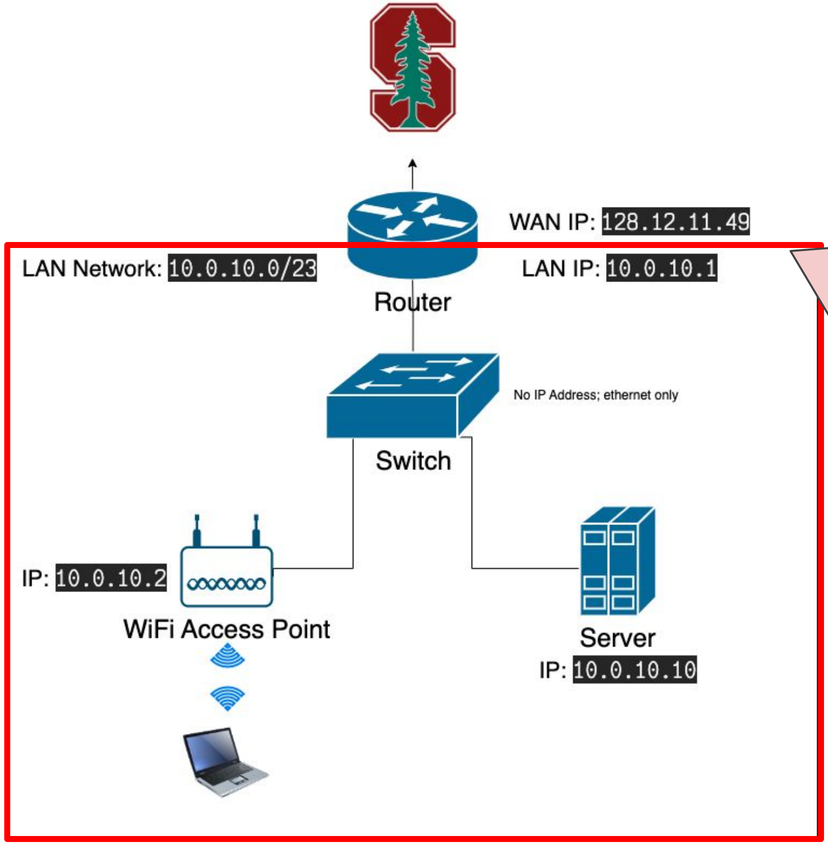
Going Deeper On My Network



My Network


WAN IP: 128.12.11.49
LAN Network: 10.0.10.0/23
Router
LAN IP: 10.0.10.1
No IP Address; ethernet only
Switch
IP: 10.0.10.2
WiFi Access Point
Server
IP: 10.0.10.10

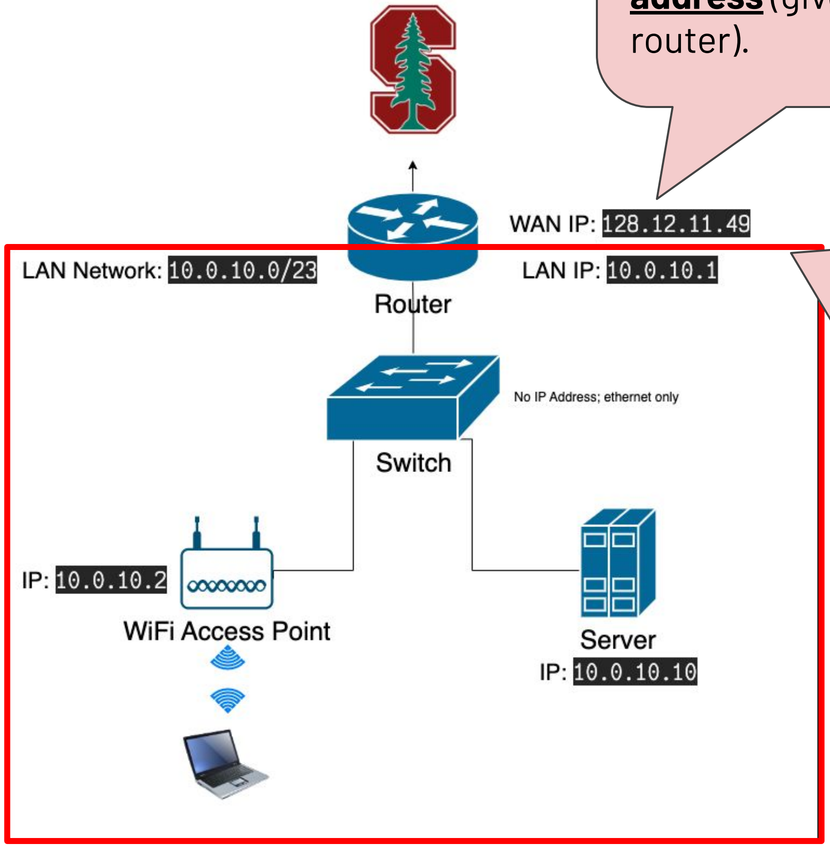
My Network



Note that the **inside of this box is a <u>different network</u> than outside (it is <u>off the public internet</u>).** This means that the router must use **NAT** to translate between them.

My Network

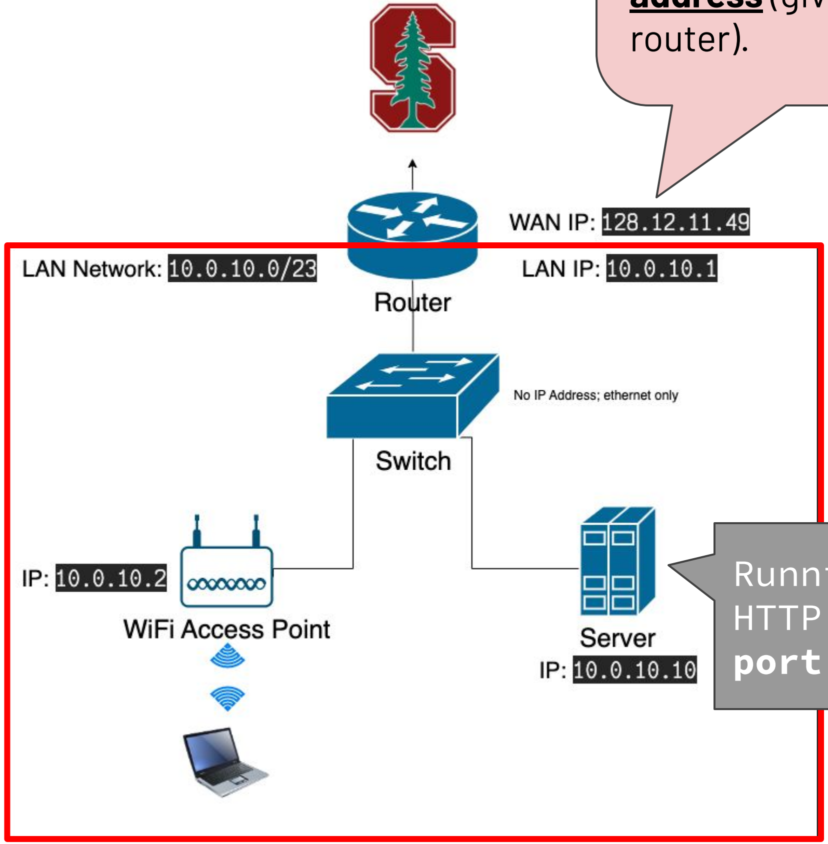
Also note that the **entire inner network** has **<u>only one Internet IP address</u>** (given to the router).



Note that the **inside of this box is a <u>different network</u> than outside (it is <u>off the public internet</u>).** This means that the router must use **NAT** to translate between them.

Port Forwarding

Also note that the **entire inner network** has **<u>only one Internet IP address</u>** (given to the router).

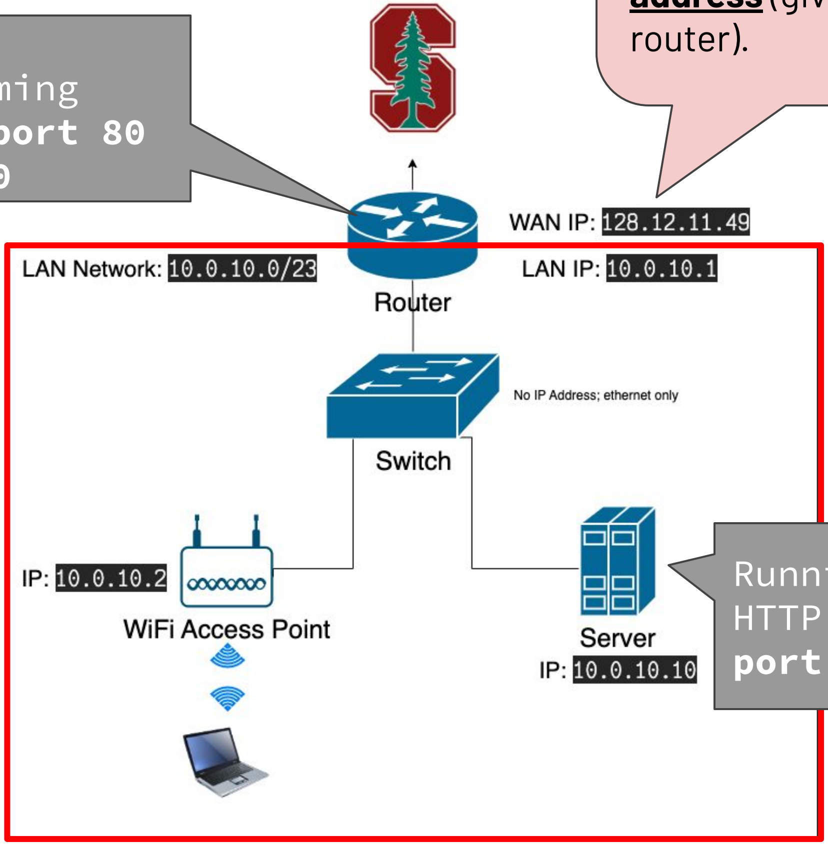


Running HTTP on **port 80**

Port Forwarding

Also note that the **entire inner network** has **<u>only one Internet IP address</u>** (given to the router).

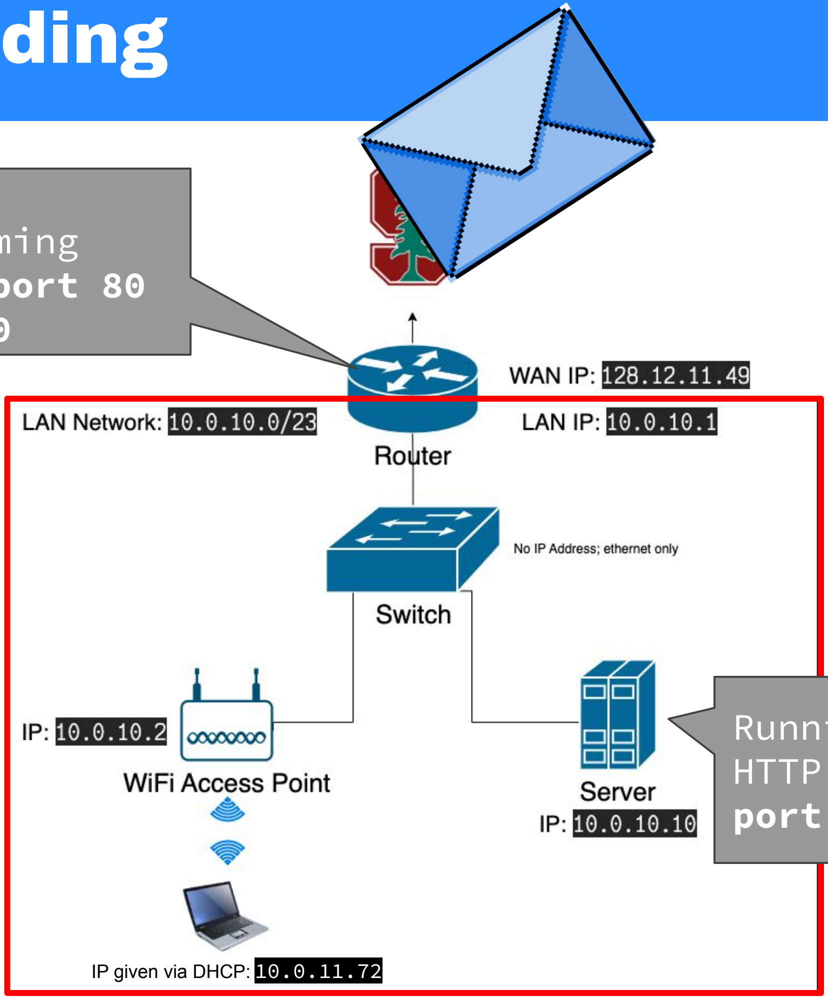
Port Forwarding:
`Forward incoming data on TCP port 80 to 10.0.10.10`



`Running HTTP on port 80`

Port Forwarding

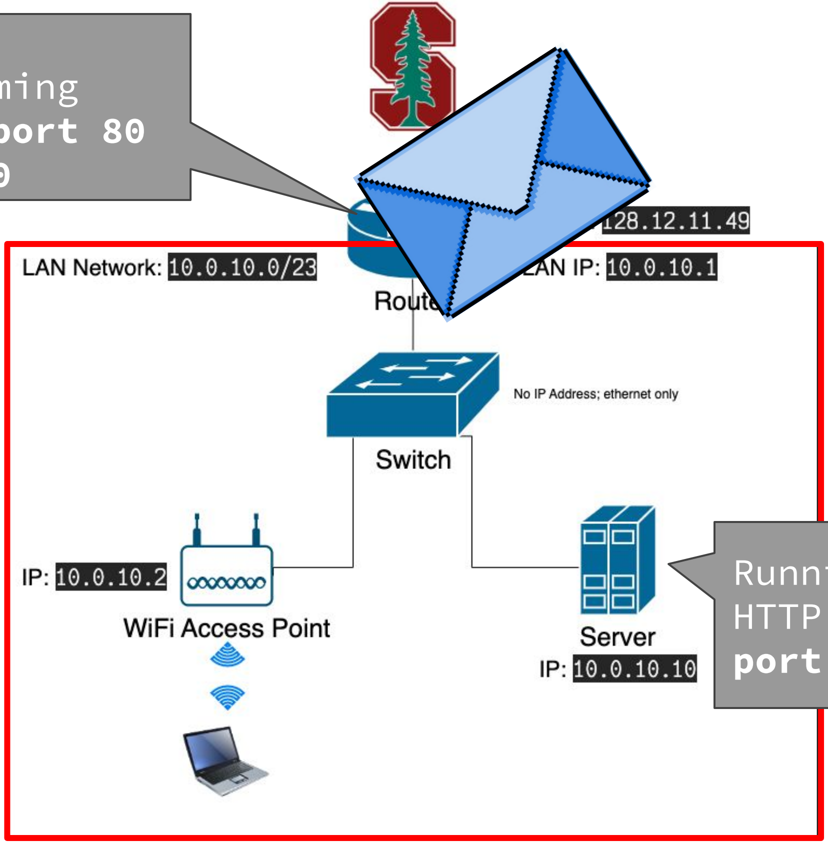
Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**



Running HTTP on **port 80**

Port Forwarding

Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**



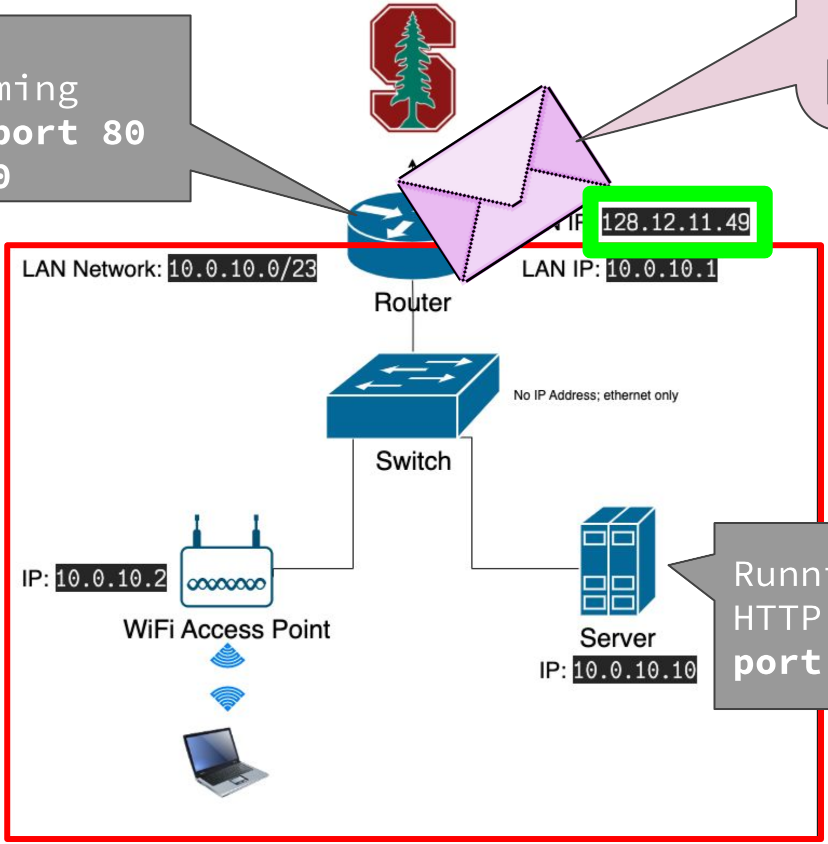
Running HTTP on **port 80**

Port Forwarding

Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**

SOURCE IP:
128.12.10.141

DEST IP:
128.12.11.49



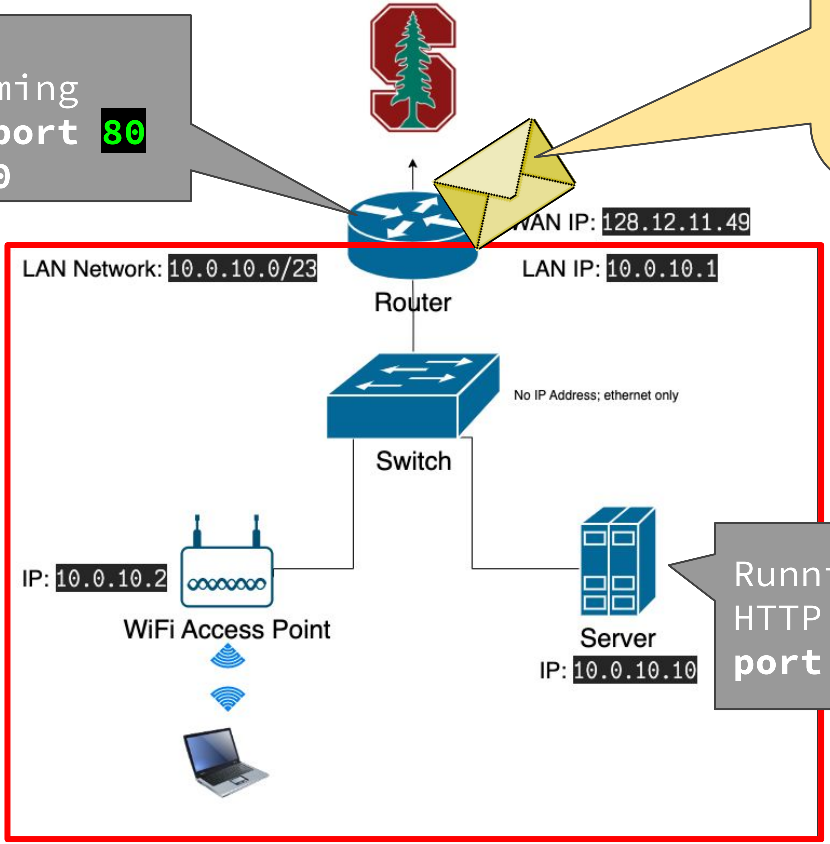
Running HTTP on **port 80**

Port Forwarding

SOURCE PORT:
43224

DEST PORT:
80

Port Forwarding:
`Forward incoming data on TCP port 80 to 10.0.10.10`



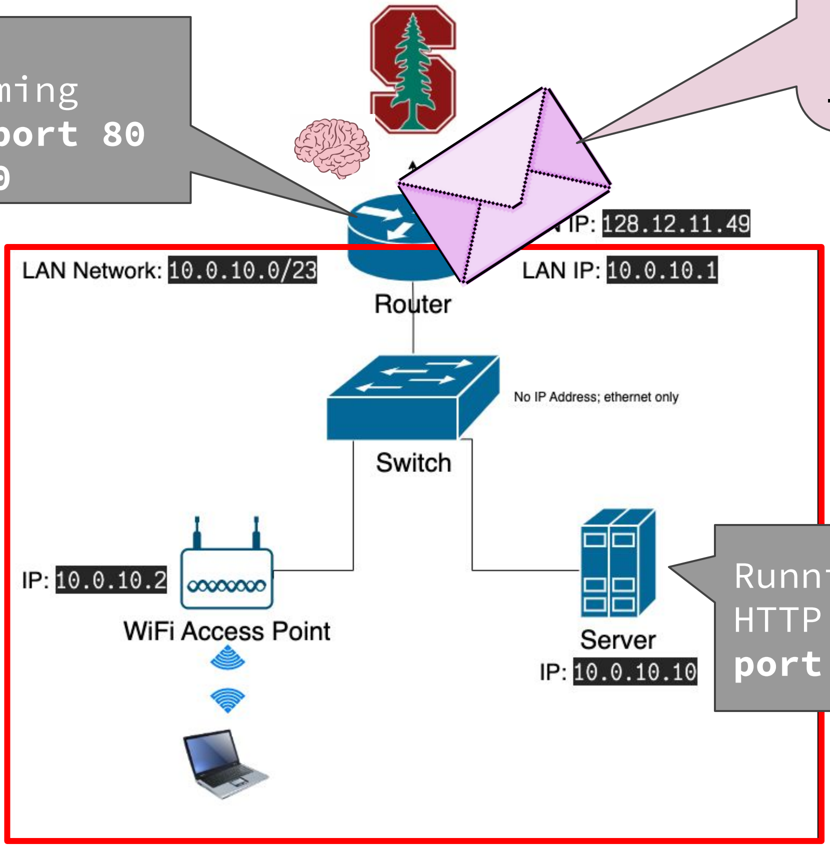
`Running HTTP on port 80`

Port Forwarding

Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**

SOURCE IP:
<u>10.0.10.1</u>

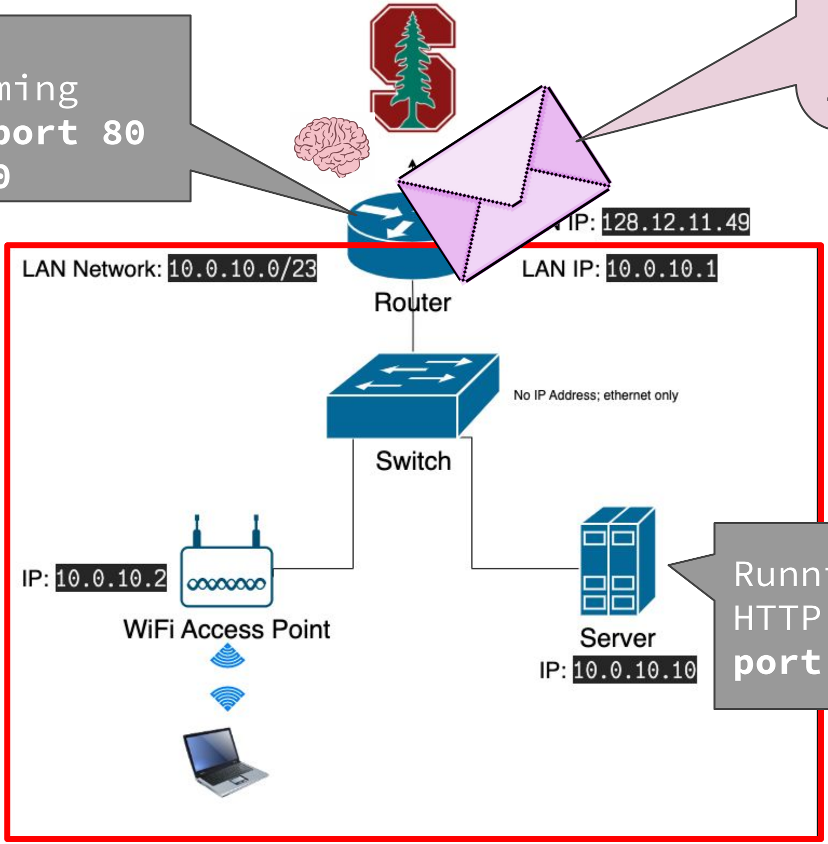
DEST IP:
<u>10.0.10.10</u>



Running HTTP on **port 80**

Port Forwarding

Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**



SOURCE IP:
10.0.10.1

DEST IP:
10.0.10.10

Running HTTP on **port 80**

NAT:
Note how the source and destination have been rewritten for the local network!

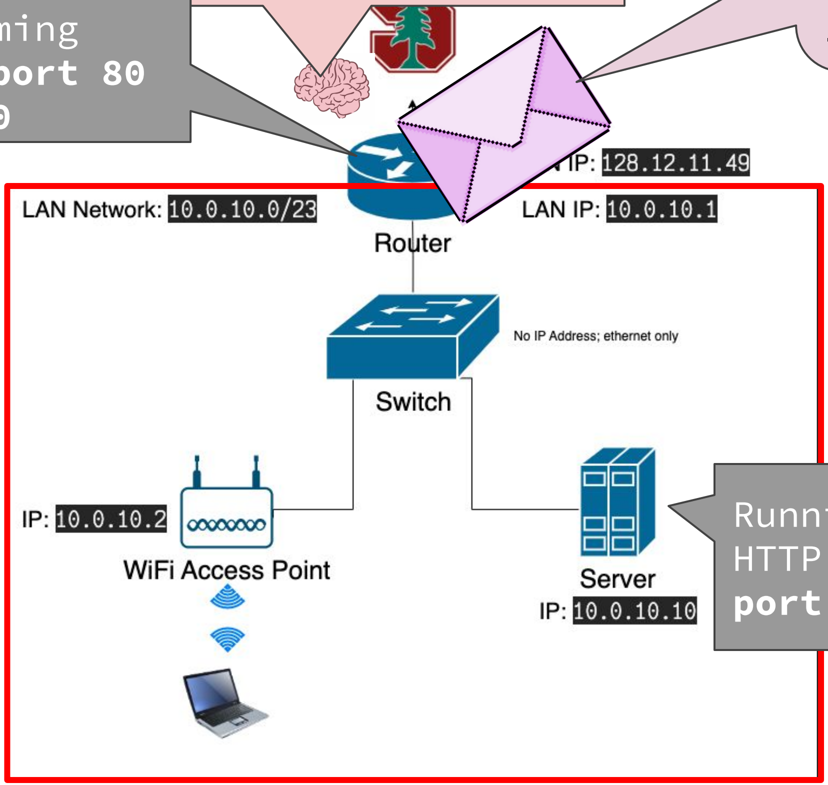
Port Forwarding

NAT:
The router must remember this translation in order for both parties to communicate

SOURCE IP:
<u>`10.0.10.1`</u>

DEST IP:
<u>`10.0.10.10`</u>

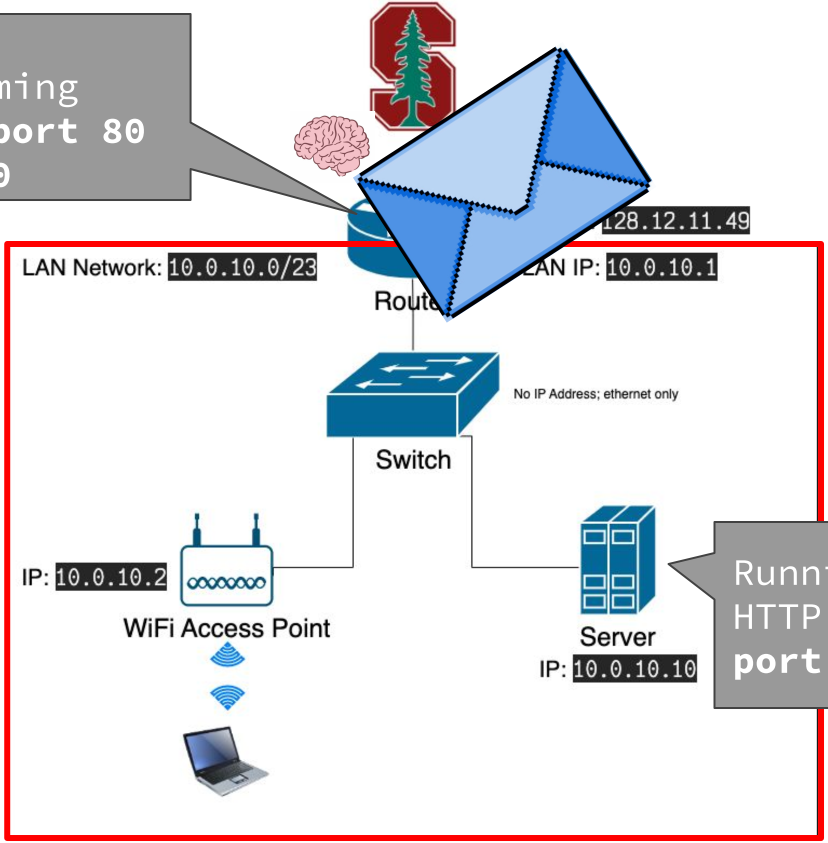
Port Forwarding:
`Forward incoming data on TCP port 80 to 10.0.10.10`



`Running HTTP on port 80`

Port Forwarding

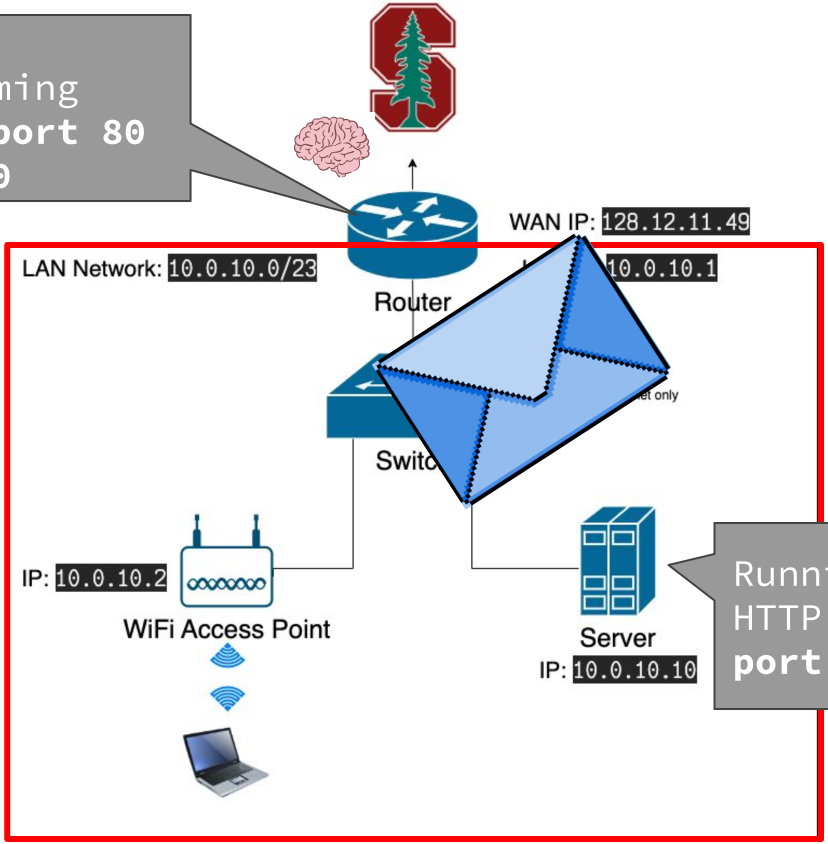
Port Forwarding:
`Forward` `incoming data on` **`TCP port 80`** `to` **`10.0.10.10`**



`Running HTTP on` **`port 80`**

Port Forwarding

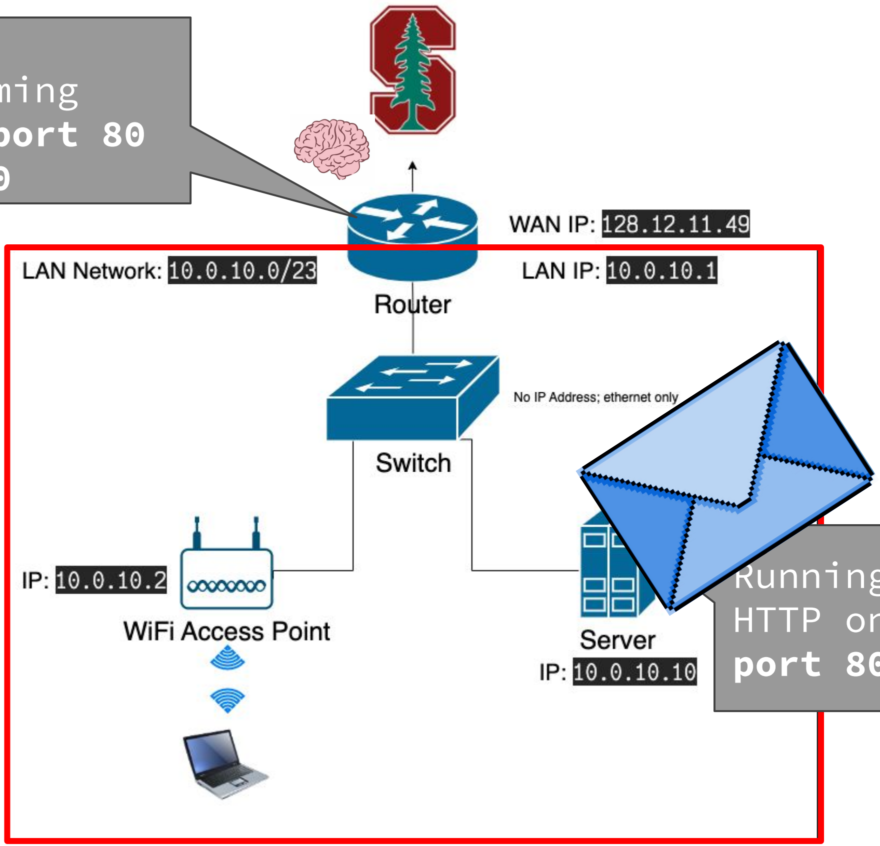
Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**



Running HTTP on **port 80**

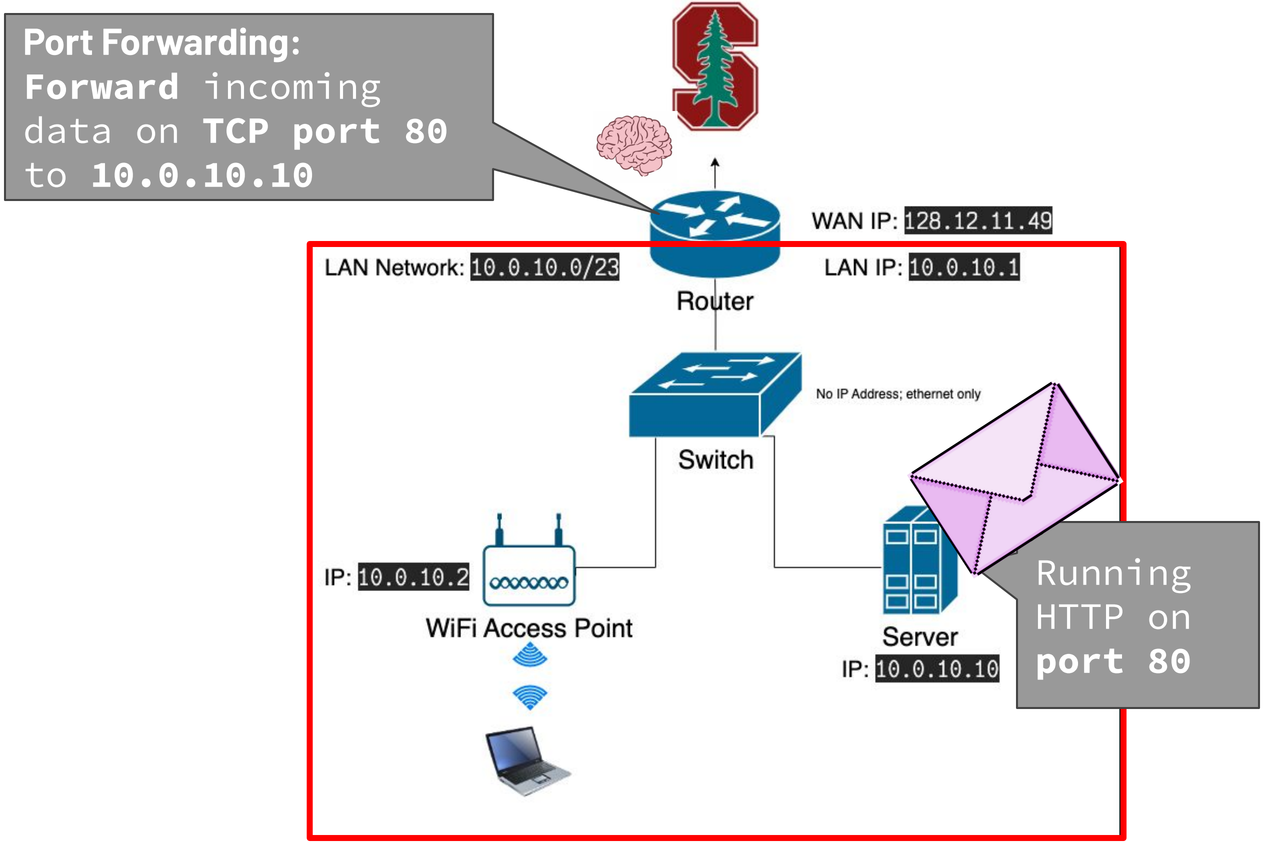
Port Forwarding

Port Forwarding:
Forward incoming data on **TCP port 80** to **10.0.10.10**

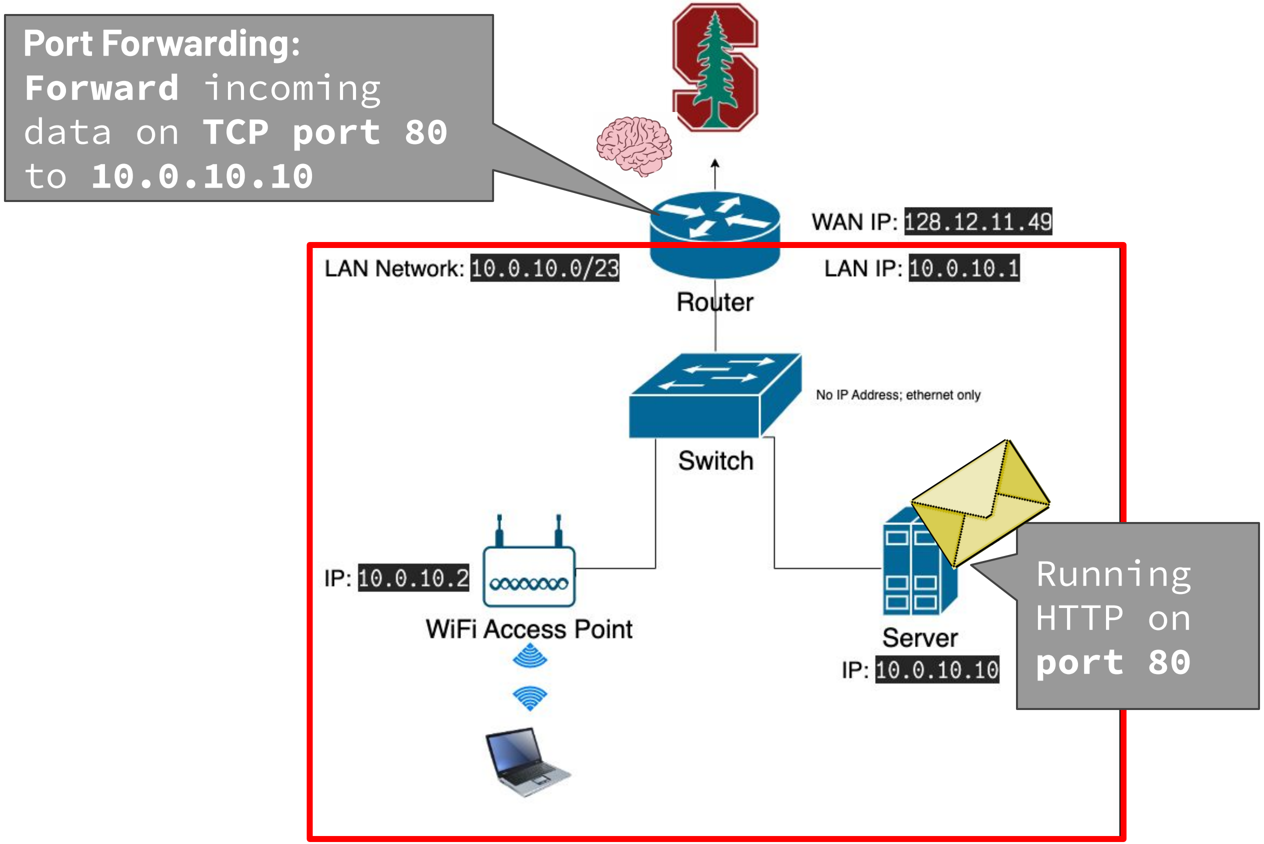


Running HTTP on **port 80**

Port Forwarding

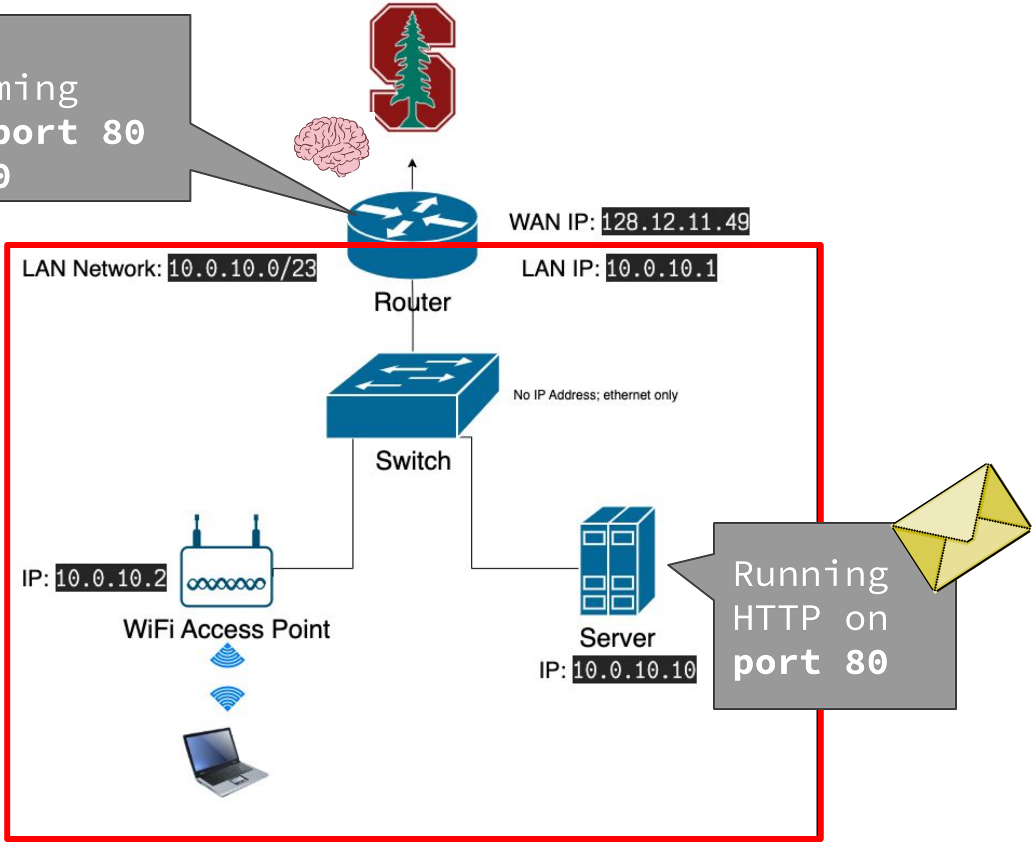


Port Forwarding

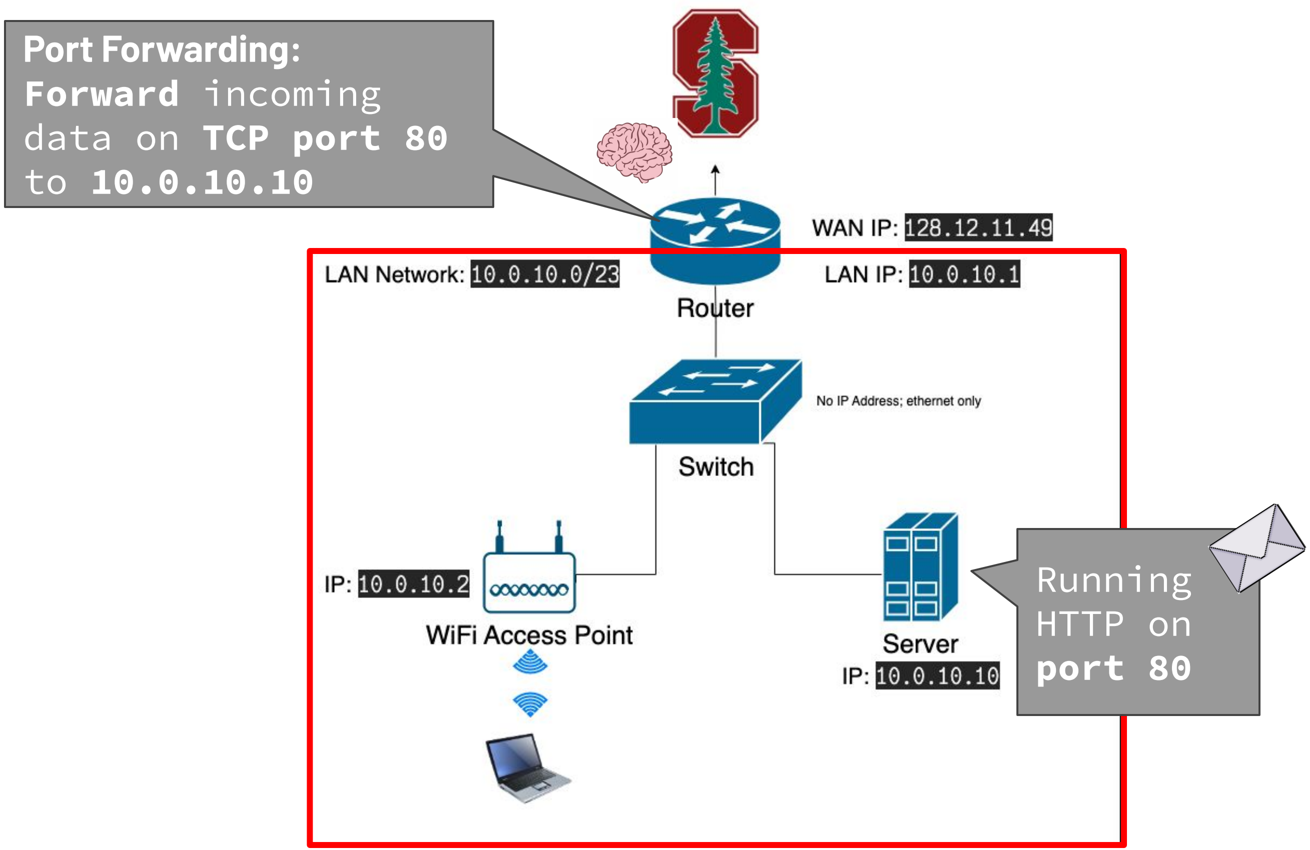


Port Forwarding

Port Forwarding:
Forward incoming
data on **TCP port 80**
to **10.0.10.10**



Port Forwarding



Servers & Clients

Resources For All

Most interactions over the network have **two sides:**

(1) The **Server**, which has resources or services that you want to access and use.
(2) The **Client**, which accesses/uses those resources or services.

Servers Aren’t Special

Servers are just computers.

Anything can be a server! The only thing it needs to do to qualify as a server is:

(1) **Run** an application that **listens** for connections on a particular port

That’s it!

Demo time!

[demo hosting and port forwarding]

Now you try!

[demo hosting and connecting locally]

Names & Addresses

From IP Addresses to Domain Names

Okay, but you never type in **`128.12.11.49`** when accessing websites, for example. You'd prefer to type in, say, **`jonak.link`**!

The system that allows this to happen is called **the Domain Name System,** or **DNS**

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
```

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
```

^Name

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
            ^TTL (how long you can cache this record)
```

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
                  ^for the internet
```

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
                     ^A-type record
```

(An **A** record contains IPv4 addresses)

The Internet's Yellowpages

Basically Just A Table

Originally, all hostnames were in a file *manually* maintained by Stanford Research Institute (SRI). This, obviously, didn't scale.

Now it's a fancy table! You have **records** that map **names** to **data**, within a **zone** (i.e. a domain). For example:

```
jonak.link. 300IN A  76.76.21.21
                        ^value
```

The Internet’s Yellowpages

[demo nslookup and dig]

The Internet's Yellowpages

[demo Cloudflare DNS]

Protocols

The Standards By Which Computers Speak

Computers aren't flexible. They need **specific rules** to be able to talk with each other. These are called **protocols**.

Protocols

The Standards By Which Computers Speak

Computers aren't flexible. They need **specific rules** to be able to talk with each other. These are called **protocols**.

One common protocol is **HTTP**, which we use for loading webpages. By convention, HTTP servers listen on port **80**.

Bonus: ssh

You can use the **ssh** ("**s**ecure **sh**ell") command to securely create a remote terminal. *That means you're connected and running all your commands on the remote computer!*

- This is the #1 way to connect to remote servers/etc
- It is secure– you can create secure proxies and port-forwards between your computer and a server.
- It's pretty lightweight
- You can also transfer files using the protocol (**scp**)

Administrivia

- Assignment 3 has been released and is due next Wednesday, May 3rd!
- Reminder that our office hours are listed below our faces on our website, https://cs45.stanford.edu