University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 101: Computing Technologies and the Internet of Things

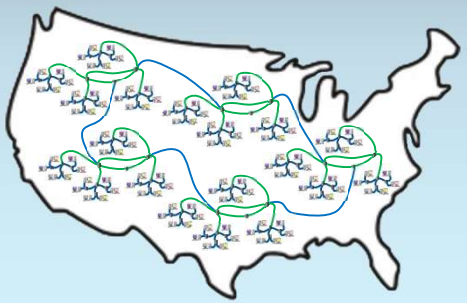
The Internet
(part 2 of 2)

1

If You Build It, They Will Come

The Internet plumbing is ready!

So now what?



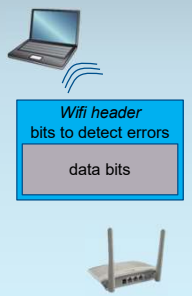
2

Packets Include a Header for Checking the Bits

Let's back up.

Before we talked
- about the Internet and IP, the Internet Protocol,
- we talked about some physical networks—
- ways to send bits as signals.
- For example, Wifi, Ethernet, and optical fibers.

When we send signals,
- we send some **bits**, but we **also**
- **send** some **extra bits in front—a header**—
- **to check for** bit **errors**.



3

Delivery of Packets is Not Guaranteed

If a bit error is detected, the packet is discarded.

That's all.

There are no guarantees.

You need to try again.

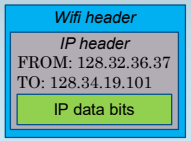
4

IP Packet Includes Source and Destination Addresses

IP operates on top of the physical layer.

Sender gives source and destination IP addresses.

Routers along the way **forward** the packet **using** a variety of **physical protocols**.



Where are the IP addresses?

In the IP header!

IP packets flow across the network (virtual connection)

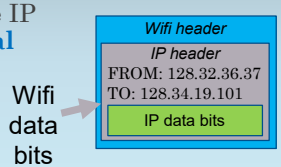
Each Protocol Packet is Encapsulated in the One Below

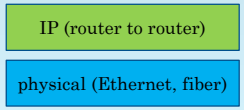
Notice that **the IP packet**—including the IP header—**is just data bits to the physical layer**.

This approach is called **encapsulation**:

- the **physical packet**
- is **wrapped around**
- the **IP packet** (like an envelope).

We draw a protocol stack with IP sitting on top of the physical layer.





IP Provides Only Best Effort Delivery

But **IP is still unreliable**!

IP does not guarantee that your packet is delivered.

The service is called **best effort**:

- **if the network can deliver** your packet, **it will**.
- **Otherwise**, if errors or congestion or failed systems affect your packet, **it won't** (deliver your packet).

When Reliability is Not Important, Use UDP

Sometimes that's ok!

Games:

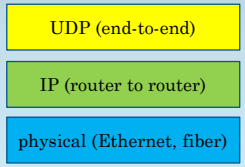
- "My player just moved to (10,3)."
- Next message has a different location.

Video conferencing:

- "here's what I look like now"
- In next video frame, I look different, so no point in re-sending old images!

Those applications

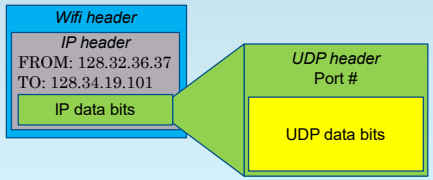
- use the **User Datagram Protocol (UDP),**
- **an unreliable end-to-end (transport) protocol**.

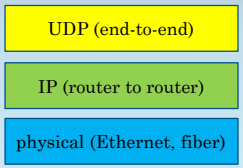


UDP Packets are Encapsulated within IP Packets

What about the UDP packets?

They are encapsulated within the IP packets!





TCP Provides Reliable Delivery of a Stream of Bytes

More often, we want reliable delivery.

Instead of UDP, we can use the **Transmission Control Protocol (TCP)**, which provides

- **connections**—a long-term conversation between two computers in the Internet,
- **an "infinite" stream of bytes**, and
- **reliable, in order delivery**.

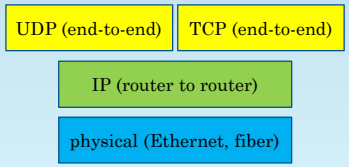
TCP Supports Thousands of Connections

Like UDP,

- **TCP sits on top of IP**, and
- TCP packets are encapsulated within IP packets.

TCP also **provides**

- **16-bit port numbers**
- to keep track of **thousands of different connections** per computer.

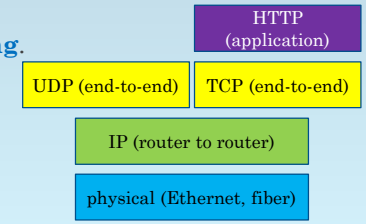


HTTP: An Application Protocol for Browsing the Web

Now we're ready to look at an application protocol!

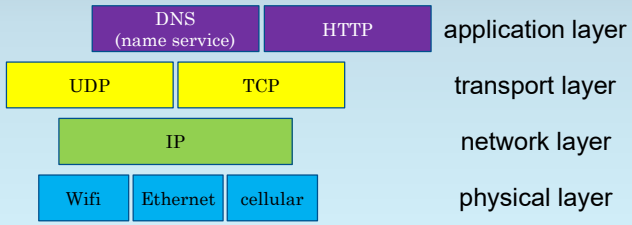
The Hyper-Text Transfer Protocol (**HTTP**) is **used for web browsing**.

It **uses TCP**, and its data are encapsulated in TCP packets.



Abstraction Layers in a Network Protocol Stack

Before diving in to HTTP, let's review …



How Can We Browse the Web?

You want to read the news.

So you open your browser and type "cnn.com" (for example).

Somewhere across the Internet is a server—probably in Atlanta—that has today's page.

Your browser needs
- to open a TCP connection
- in order to use HTTP and
- talk with that server.

But what is the server's IP address?

DNS Translates Human-Readable Names to IP Addresses

That's where DNS comes in!

DNS, the Domain Name Service.

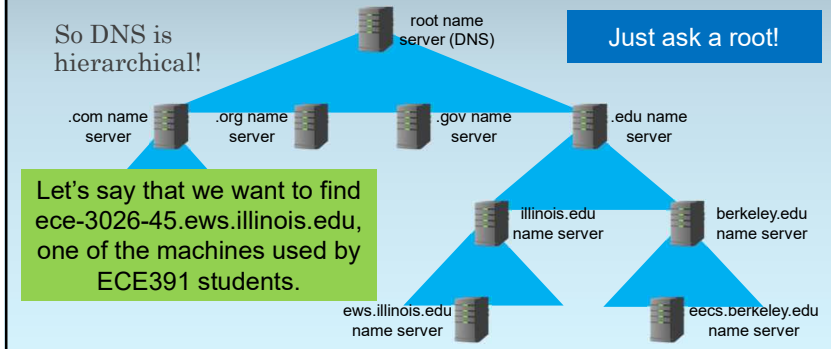
It **translates human-readable names** like cnn.com **into IP addresses**.

There are 13 DNS root servers around the world.

Billions of people using 13 servers…

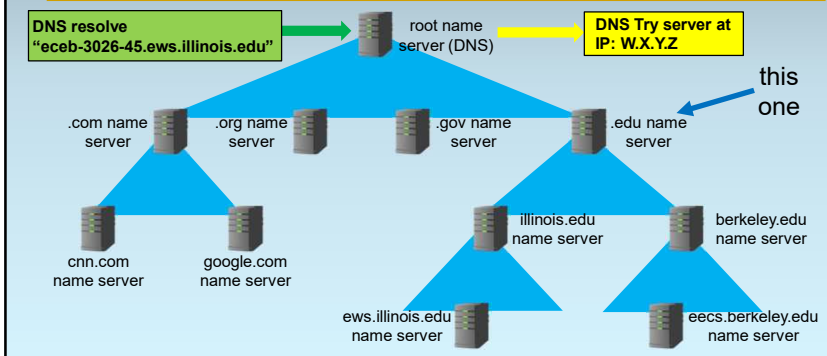
Domain Name Service (DNS) Server Hierarchy

So DNS is hierarchical!



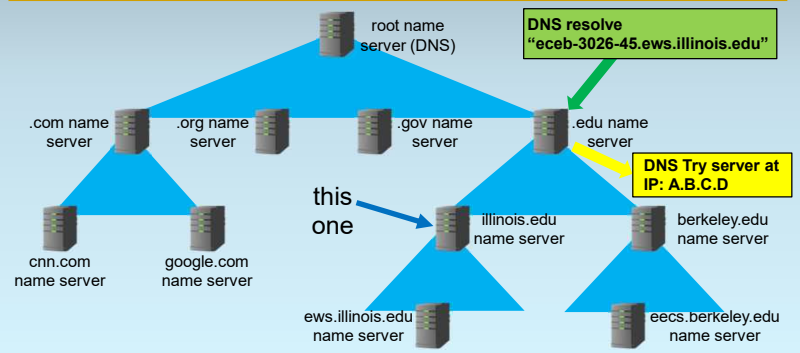
Let's say that we want to find ece-3026-45.ews.illinois.edu, one of the machines used by ECE391 students.

DNS Lookups Start Logically at the Root



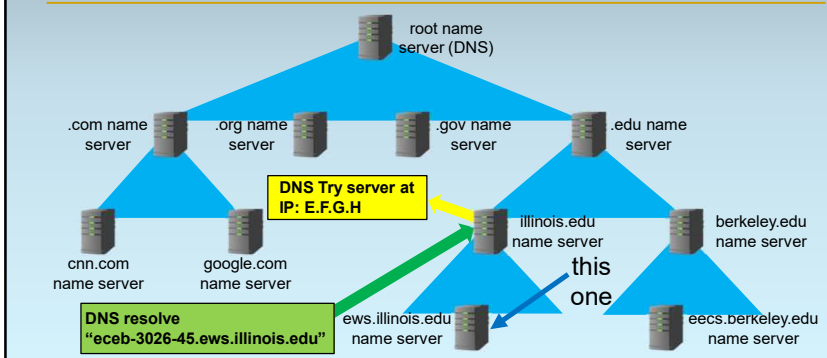
17

DNS Lookups Iterate Until an Answer is Found



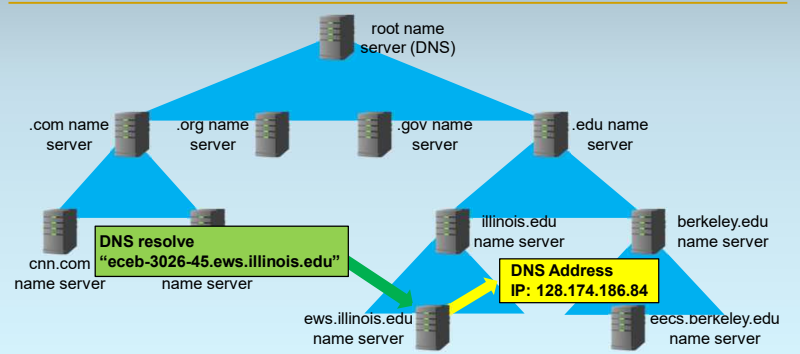
18

DNS Keeps Chugging Along



19

Until, Finally, an Answer!



20

DNS Provides the IP Address for CNN.com

Back to our example.

Your web browser wants to connect to "cnn.com"

It **queries DNS and obtains an IP address**: 151.101.67.5.

Browser Opens a TCP Connection to the Server

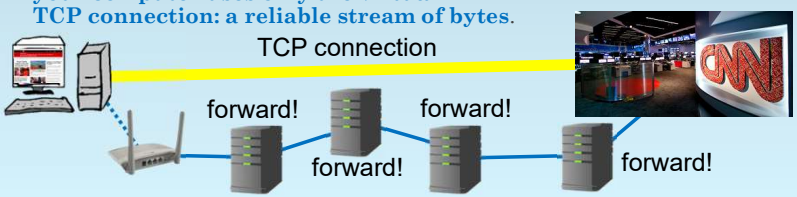
Once you have the IP address,
- your **computer can open**
- **a TCP connection**
- **to send an HTTP request**.

The web server is at TCP port 80.*

*See https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

Abstraction Layers Hide the Complexity of the Internet

Even though getting data from your computer to cnn.com requires
- many routing decisions and
- many physical connections,
- **your computer uses only the virtual TCP connection: a reliable stream of bytes**.



HTTP Requests and Responses Use Human-Readable Text

So what does your computer actually send?

English sentences?

"Please give me the web page at CNN.com."

No!

Like other protocols,
- **HTTP has rules**
- about how to request
- and receive information.

HTTP is more friendly, though—it does **use human-readable sequences of bytes** (text).

Contents of an HTTP Request

A HTTP request
- (like your browser sends) has
- a command (GET),
- a web page (/index.html)
- a protocol specification (HTTP/1.0),
- and some options.

```
GET /index.html HTTP/1.0
User-Agent: Mozilla/4.0
Host: cnn.com
(more options, ending blank line)
```


Contents of an HTTP Response (Successful)

A HTTP response
- (returned from a web server to your browser) has
- a protocol specification (HTTP/1.0),
- an error code (200 OK),
- some options,
- and the web page itself (in HyperText Markup Language, HTML).

```
HTTP/1.0 200 OK
Date: Fri, 9 Sep 2022 16:20:23 GMT
Server: Apache/2.2.15
(more info)

Bits for the web page (in HTML)!
```


Your Data Encapsulated by HTTP, Then TCP, Then…

Notice that
- your desired **web page name**
- **and** the **web page** itself
- are **just data bits to HTTP**!

In turn,
- all of the **HTTP information**
- is **just data** bits **to TCP**.

That's encapsulation!

When Browser Receives HTTP Response, It Renders Page

When your **browser**
- receives the web page,
- it **renders** it **on your monitor**.

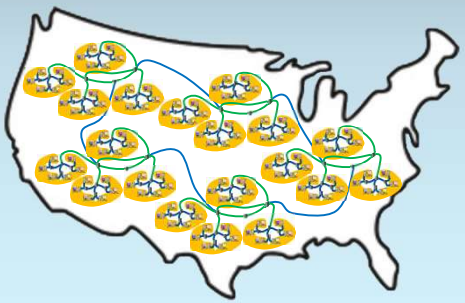
Usually, the browser
- **simultaneously makes requests**
- **for images, videos, and other things**
- embedded in the page.

Applications Operate on the Edges of the Internet

Applications mostly use TCP.

They view the Internet through end-to-end, reliable connections.

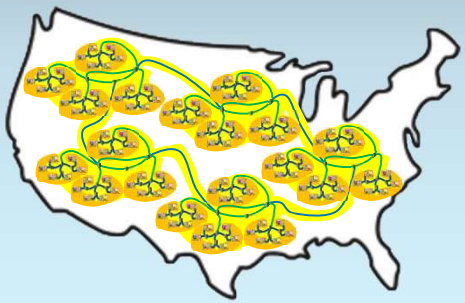
They **operate on the edge of the network**—the bottom of the Internet hierarchy.



The Rest of the Internet Forms the Core

Everything else
- all of the routers,
- all of the rest of the hierarchy,
- the plumbing of the Internet,

These are **called the core of the Internet**.



Applications Running on the "Edge"

And **applications are legion!**
- web browsing
- content distribution
- social networks
- file management
- banking
- medicine
- and on and on.

All running on the edge of the Internet.

Terminology You Should Know from These Slides

- packet of bits
- packet header
- Internet Protocol (IP)
- encapsulation
- protocol stack
- best effort delivery
- connection
- reliable delivery
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- port numbers
- HyperText Transfer Protocol (HTTP)
- Domain Name Service (DNS)
- HTTP request and response
- edge (of Internet)
- core (of Internet)

Concepts You Should Know from These Slides

- uses of packet header: checking for errors, IP addresses, port numbers, protocol commands (example: HTTP GET)
- network protocol layers and their roles in the Internet
- uses of UDP (end-to-end unreliable delivery)
- properties provided by IP, UDP, TCP
- what service DNS provides and how it works
- how a web browser finds and renders a page for you

