

# Internet Technology

[http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

BUT WHEN SHE TRACED THE  
KILLER'S IP ADDRESS... IT WAS  
IN THE 192.168/16 BLOCK!



<http://xkcd.com/742/>



Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.  
<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2009-2011, Charles Severance, Martha Pollack, Paul Resnick

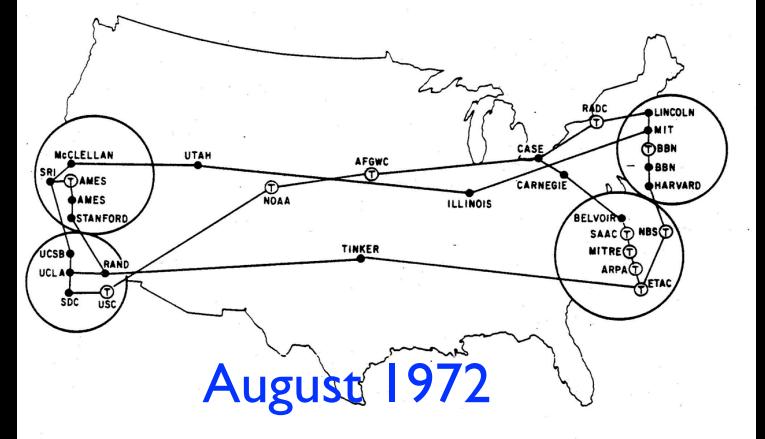
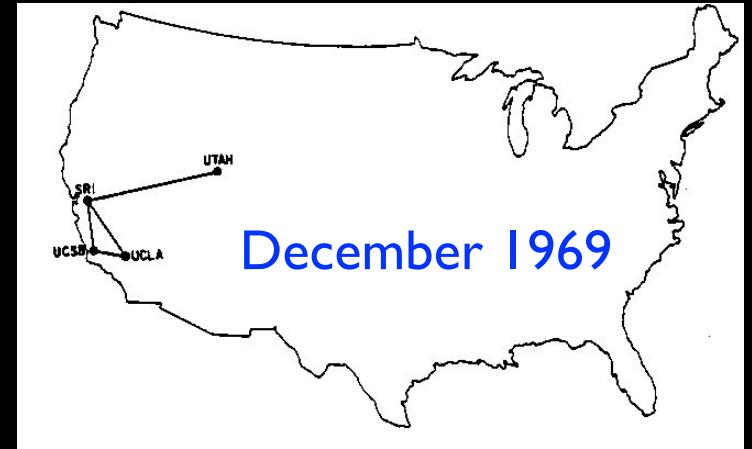
You assume all responsibility for use and potential liability associated with any use of the material. Material contains copyrighted content, used in accordance with U.S. law. Copyright holders of content included in this material should contact [open.michigan@umich.edu](mailto:open.michigan@umich.edu) with any questions, corrections, or clarifications regarding the use of content. The Regents of the University of Michigan do not license the use of third party content posted to this site unless such a license is specifically granted in connection with particular content. Users of content are responsible for their compliance with applicable law. Mention of specific products in this material solely represents the opinion of the speaker and does not represent an endorsement by the University of Michigan. For more information about how to cite these materials visit <http://michigan.educommons.net/about/terms-of-use>.

Any medical information in this material is intended to inform and educate and is not a tool for self-diagnosis or a replacement for medical evaluation, advice, diagnosis or treatment by a healthcare professional. You should speak to your physician or make an appointment to be seen if you have questions or concerns about this information or your medical condition. Viewer discretion is advised: Material may contain medical images that may be disturbing to some viewers.

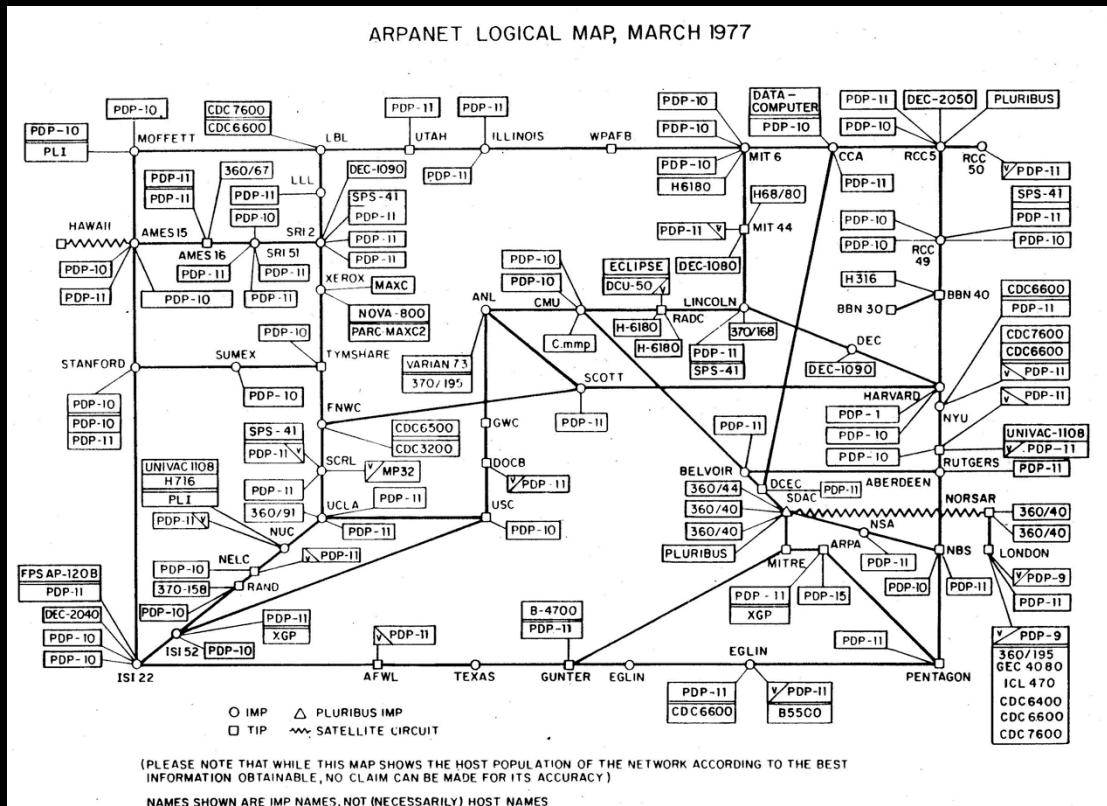


# Research Networks 1960-1980's

- How can we avoid having a direct connection between all pairs of computers?
- How to transport messages efficiently?
- How can we dynamically handle outages?



<http://som.csudh.edu/fac/lpress/history/arpamaps/>



Heart, F., McKenzie, A., McQuillian, J., and Walden, D., ARPANET Completion Report,  
 Bolt, Beranek and Newman, Burlington, MA, January 4, 1978.  
<http://som.csudh.edu/fac/lpress/history/arpamaps/arpanetmar77.jpg>

# Efficient Message Transmission: Packet Switching

- Challenge: in a simple approach, like store-and-forward, large messages block small ones
- Break each message into **packets**
- Can allow the packets from a single message to travel over different paths, dynamically adjusting for use
- Use special-purpose computers, called **routers**, for the traffic control

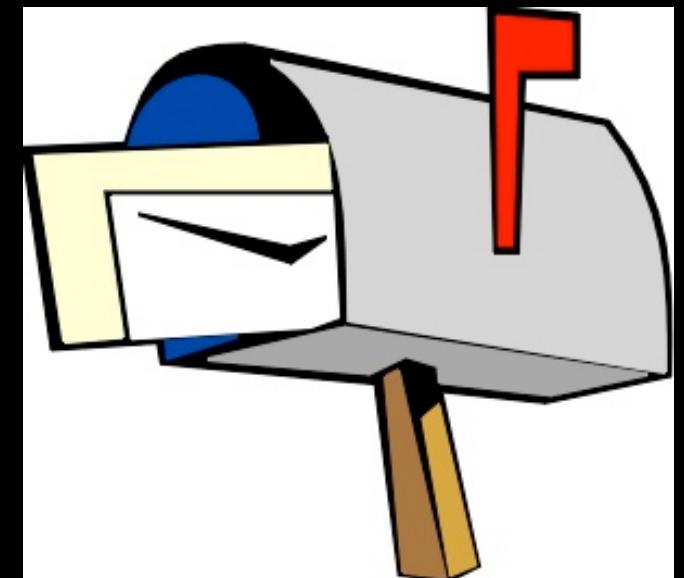
# Packet Switching - Postcards

Hello there, have a nice day.

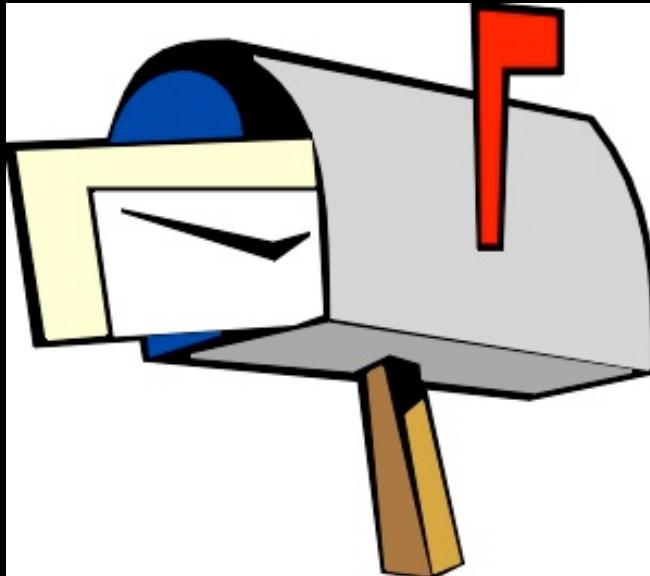
Hello ther (1, csev, glenn)

e, have a (2, csev, glenn)

nice day. (3, csev, glenn)

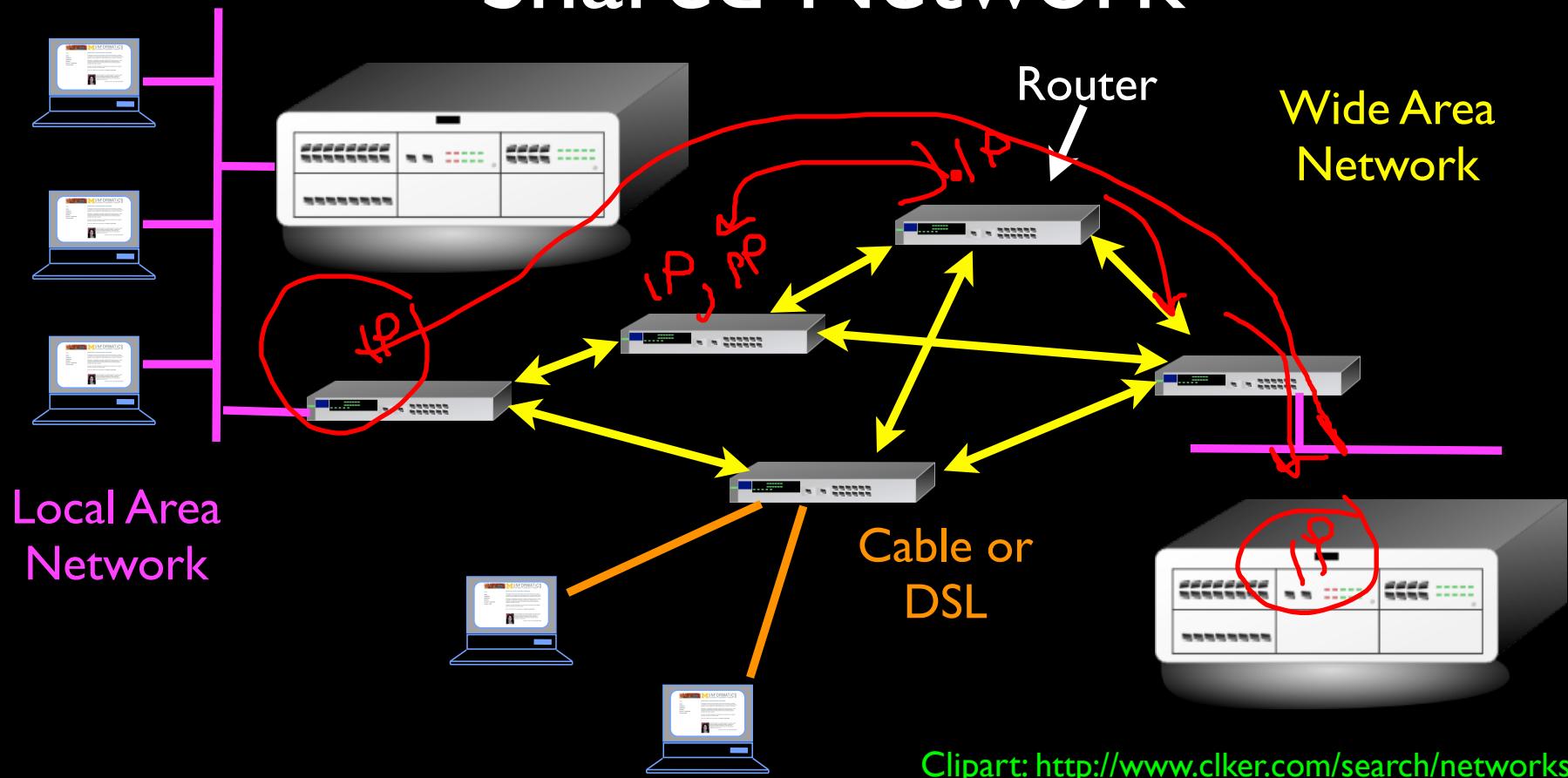


# Packet Switching - Postcards



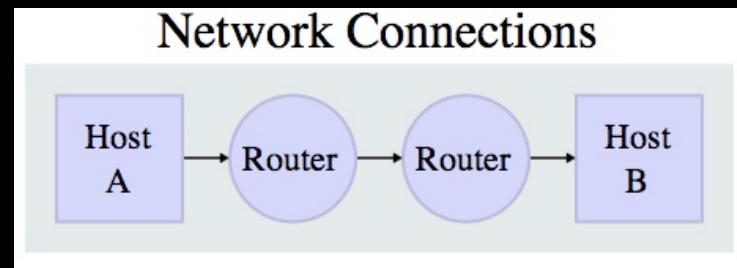
Hello there, have a nice day.

# Shared Network



# Shared Networks

- In order to keep cost low and the connections short geographically - data would be forwarded through several routers.
- Getting across the country usually takes about 10 “hops”
- Network designers continually add and remove links to “tune” their networks



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Layered Network Model

- A **layered** approach allows the problem of designing a network to be broken into more manageable sub problems
- Best-known model: **TCP/IP**—the “Internet Protocol Suite”
- There was also a 7 layer **OSI**: Open System Interconnection Model

Application Layer  
Web, E-Mail, File Transfer

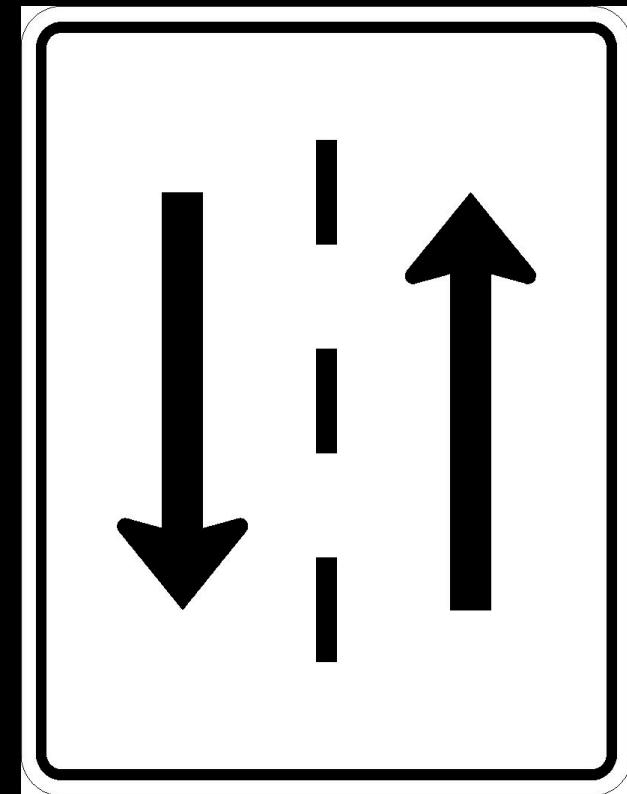
Transport Layer (TCP)  
Reliable Connections

Internetwork Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections

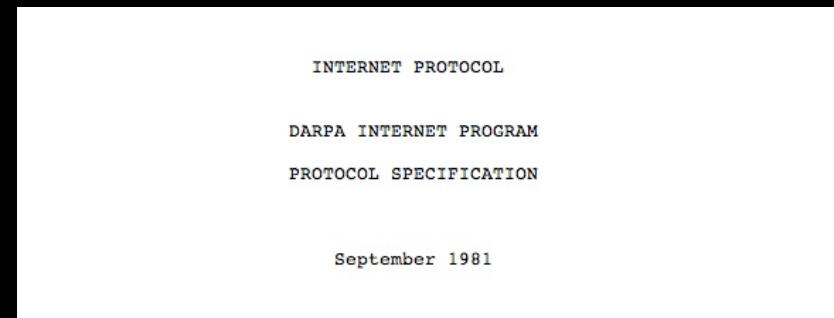
# What is a Protocol?

- A set of rules about how to behave



# Internet Standards

- The standards for all of the Internet protocols (inner workings) are developed by an organization
- Internet Engineering Task Force (IETF)
- [www.ietf.org](http://www.ietf.org)
- Standards are called “RFCs” - “Request for Comments”



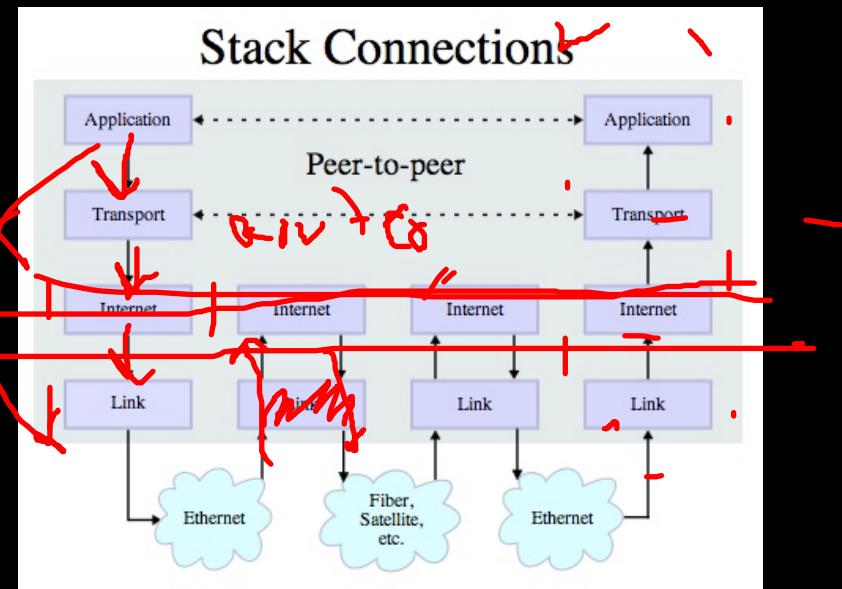
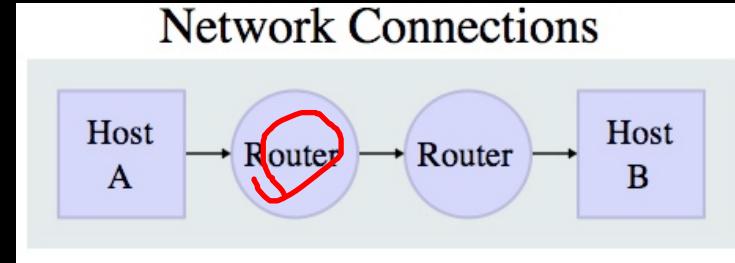
The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

Source: <http://tools.ietf.org/html/rfc791>

# Layered Architecture

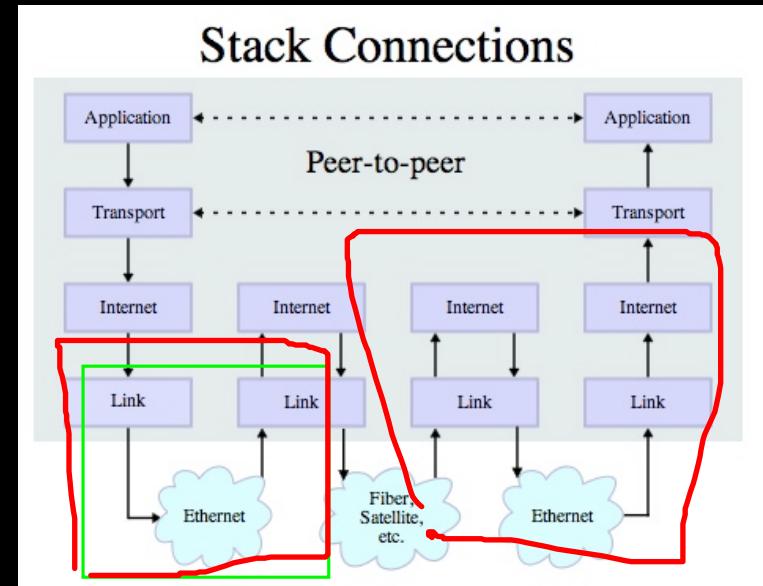
- The Physical and Internet Layers are like trucks and trains - they haul stuff and get it to the right loading dock - it takes multiple steps
- The Transport layer checks to see if the trucks made it and send the stuff again if necessary



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Link Layer (aka Physical Layer)

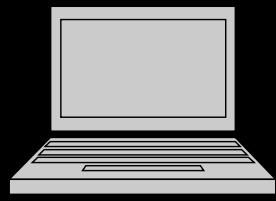
- As your data crosses the country may use a different physical medium for each “hop”
- Wire, Wireless, Fiber Optic, etc.
- The link is “one hop” - Is it up or down? Connected or not?
- Very narrow focus - no view at all of the “whole Internet”



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Problems solved by the Link Layer

- How does data get pushed onto a link?
  - How is the link shared?
- {Poe}, rbus +
- Common Link Technologies
    - Ethernet
    - WiFi ✓
    - Cable modem
    - DSL
    - Satellite ↴
    - Optical
- Mibile



Hello???

Here I  
am!



Hello???

# Link Layer Addresses



- Many physical layer devices have addresses built in to them by the manufacturer
  - Ethernet
  - Wireless Ethernet (Wifi)

The image shows two windows side-by-side. On the left is a Windows command prompt window titled 'C:\Windows\system32\cmd.exe' displaying network adapter information:

```
C:\Windows\system32\cmd.exe
Ethernet adapter Local Area Connection 2:

Connection-specific DNS Suffix . : umich.edu
Description . . . . . : Cisco Systems VPN Adapter
Physical Address . . . . . : 00-0C-9A-3C-78-88
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . . : Yes
```

On the right is a Mac OS X 'Network' preferences window for the 'AirPort' interface. The 'Ethernet' tab is selected. A yellow box highlights the 'Ethernet ID: 00:1f:5b:81:62:e7' field, which has been handwritten over with red ink. A red circle highlights the 'Configure:' dropdown set to 'Automatically'. The MTU setting is shown as 'Standard (1500)'.

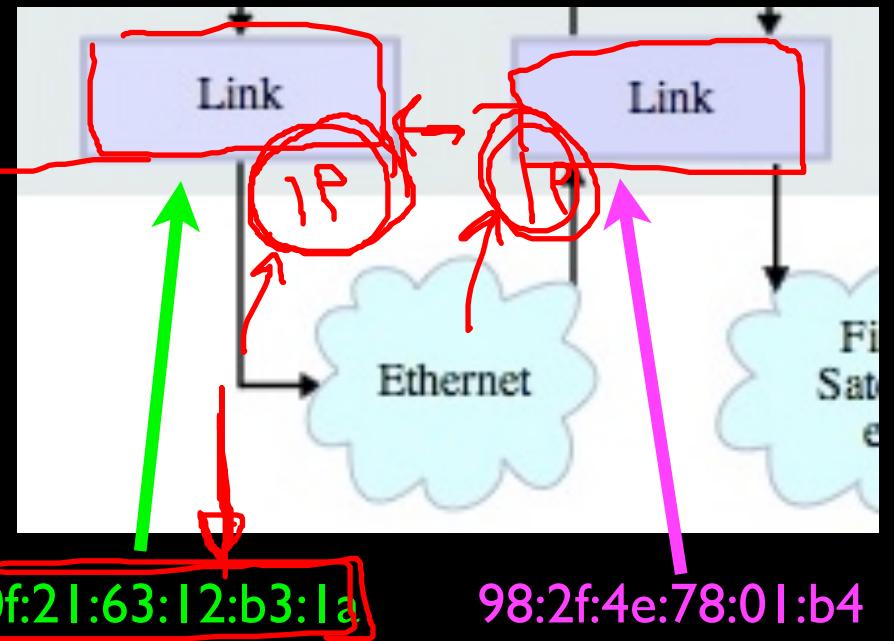
# How Many Possible MAC Addresses?

- 12 hexadecimal digits
- Hexadecimal = base 16: 0, 1, 2, . . . , 9, A, B, . . . F
- Total number of MAC address = ???

$$(16)^{12} = (2)^{48} = 281,474,976,710,656$$

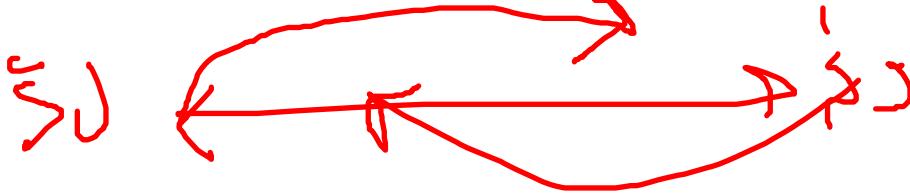
# Link Layer

- Physical addresses are to allow systems to identify themselves on the ends of a single link
- Physical addresses go no farther than one link
- Sometimes links like Wifi and Wired Ethernet are shared with multiple computers



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)





# Sharing Nicely - Avoiding Chaos

- **CSMA/CD Carrier Sense Media Access with Collision Detection**

- **To avoid garbled messages, systems must observe “rules” (Protocols)**
- **Ethernet rules are simple**

- Begin transmitting data
- Listen for your own data
- If you cannot hear your own data clearly, assume a collision, stop and wait before trying again
- Each system waits a different amount of time to avoid “too much politeness”

# Internet Layer



[http://en.wikipedia.org/wiki/Internet\\_Protocol](http://en.wikipedia.org/wiki/Internet_Protocol)

<http://en.wikipedia.org/wiki/Traceroute>

<http://en.wikipedia.org/wiki/Ping>

Application Layer  
Web, E-Mail, File Transfer

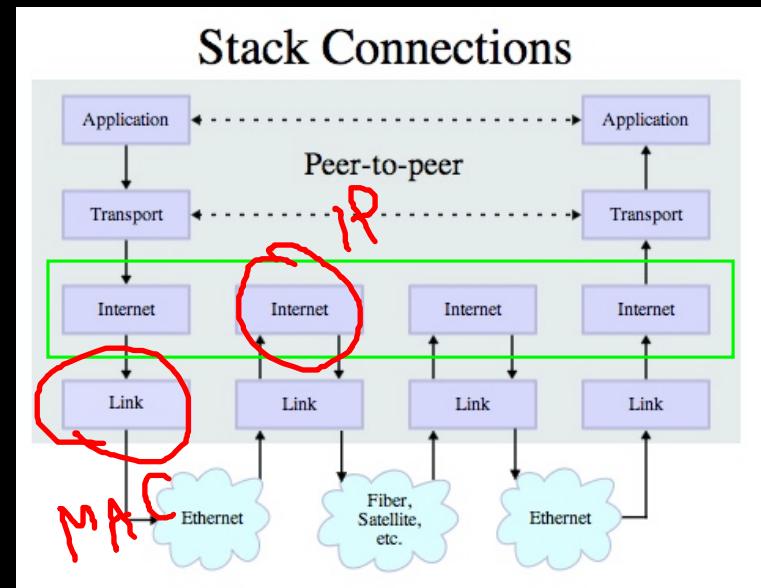
Transport Layer (TCP)  
Reliable Connections

Internet Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections

# Internet Protocol Layer

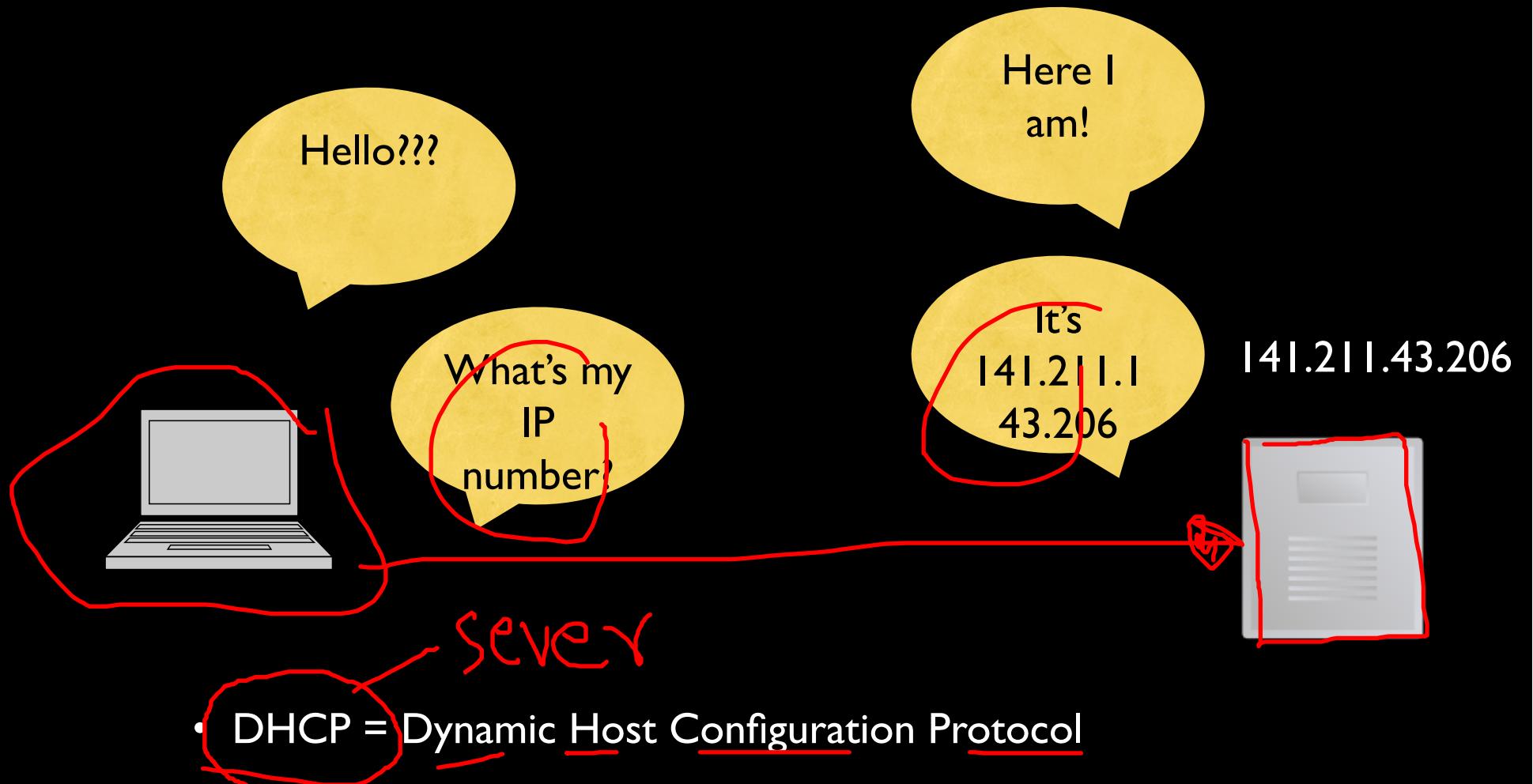
- Goal: Gets your data from this computer to the other computer half way across the world
- Each router knows about nearby routers
- IP Is best effort - it is OK to drop data if things go bad...



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# IP Addresses

- The IP address is the worldwide number which is associated with one particular workstation or server
- Every system which will send packets directly out across the Internet must have a unique IP address
- IP addresses are based on where station is connected
- IP addresses are not controlled by a single organization - address ranges are assigned
- They are like phone numbers – they get reorganized once in a great while



File Edit View History Bookmarks Tools Help

IP http://whatismyipaddress.com/ what's my IP address

Most Visited Getting Started Latest Headlines Readability

IP What Is My IP Address? - Lookup IP, ...

WhatIsMyIPAddress.com Dedicated to IP address discussion

My IP Change IP Hide IP IP Lookup Trace Email Blacklist Check IP Tools FAQs Resources

Ads by Google Search

What is my IP address?

Your IP address is **68.42.65.147**  
(Now detects many proxy servers)

Free Trial.

Tools

IP Lookup Blacklist Check Trace Email Visual Traceroute Traceroute

68.42.65.147 Trace Now

IP Lookup now shows ISP, Organization, Proxy Status, and Connection Type!

Map Satellite Hybrid Terrain

Wisconsin Michigan Ohio Pennsylvania West Virginia

Michigan

Grand Rapids Detroit Toledo Cleveland Akron Columbus Dayton Cincinnati Lima

Illinois Indiana

Peoria Elgin Rockford Racine Milwaukee Oshkosh Madison Davenport Fort Wayne

Ohio

Akron Hamilton Kitchener London Flint Erie Peterborough Oshawa Barrie

Pennsylvania

Harrisburg Pittsburgh Columbus

West Virginia

Map Data Terms of Use

Ann Arbor, Michigan United States

Read about [GeoLocation accuracy](#).

What is an IP address?

Every device connected to the public Internet is assigned a unique number known as an Internet Protocol (IP) address. IP addresses consist of four numbers separated by periods (also called a 'dotted-quad') and look something like 127.0.0.1.

Since these numbers are usually assigned to internet service providers within region-based blocks, an IP address can often be used to identify the region or country from which a computer is connecting to the Internet. An IP address can sometimes be used to show the

CLICK TO READ NOTES

# IP Address Format

- Four numbers with dots - each number 1-255 (32 bits)
- Kind of like phone numbers with an “area code”
- The prefix of the address is “which network”
- While the data is traversing the Internet - all that matters is the network number

(734) 764 1855

Area code

Network  
Number

141.211.144.188  
141.211.\*.\*

→ 10.133.0.0 Network Add  
10.133.0.255 Broadcast Add A-E

While in the network, all  
that matters is the  
Network number.

141.211.144.188



To: 67.149.\*.\*

To: 67.149.94.33

67.149.102.75



67.149.\*.\*

To: 67.149.94.33

67.149.94.33

Clipart: <http://www.clker.com/search/networksym/l>

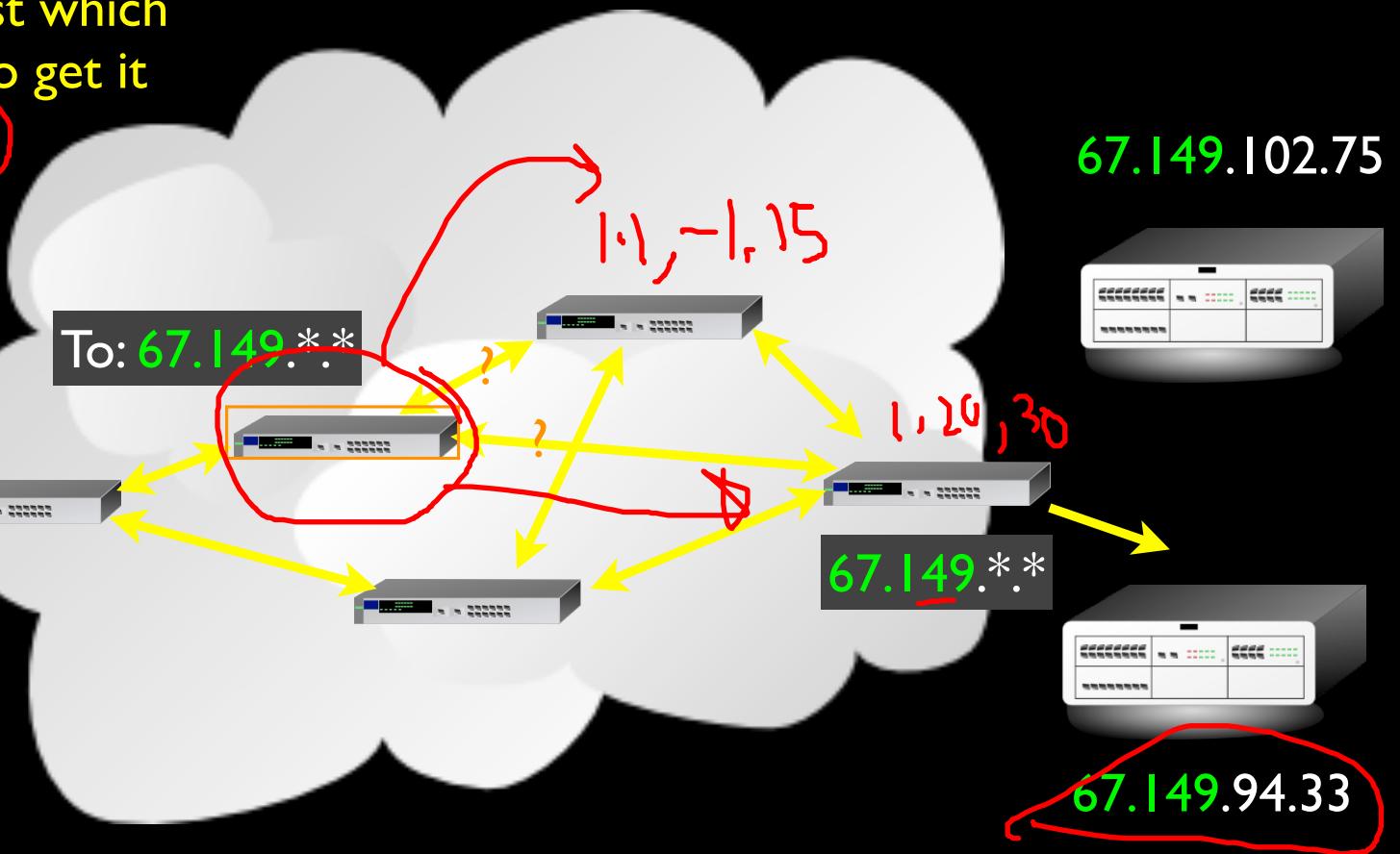
No single router knows the whole network - just which way to send data to get it

“closer”

141.211.144.188



To: 67.149.\*.\*

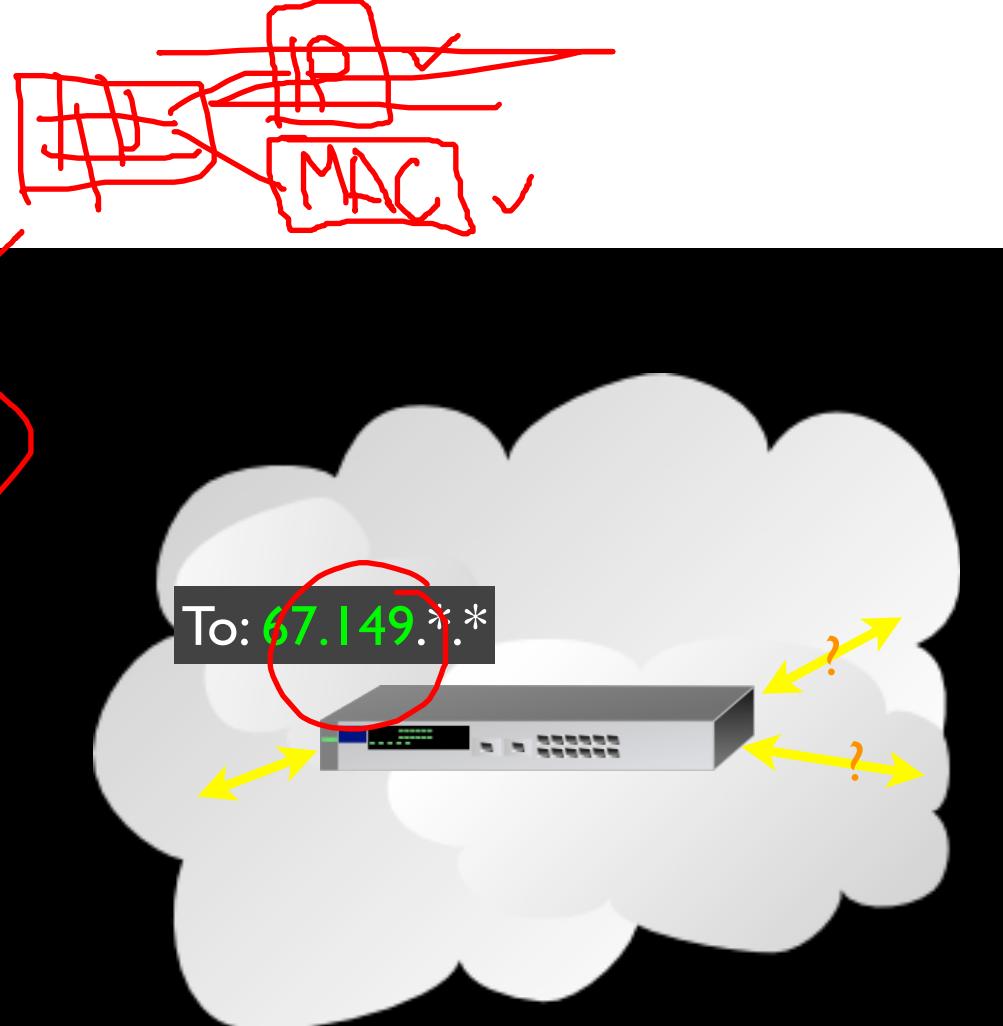


Clipart: <http://www.clker.com/search/networksym/>

# Router Tables

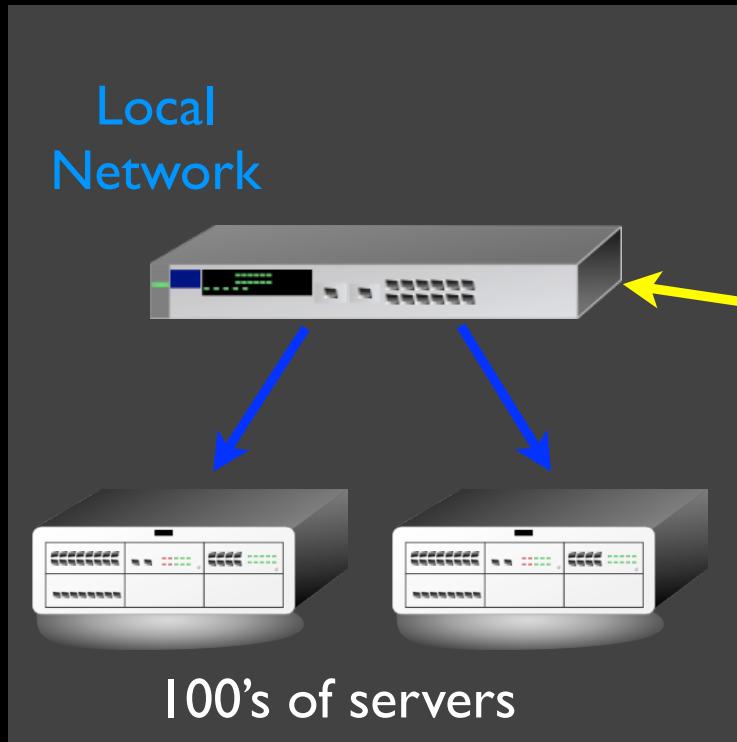
Lists of where to send packets, based on destination network address;  
bandwidth on adjacent links;  
traffic on adjacent links;  
state of neighbor nodes (up or not);  
...

Updated dynamically  
Routers “ask each other” for information



Clipart: <http://www.clker.com/search/networksym/>

# IP Is Simple



Thousands of network connections.  
Billions of bytes of data per seconds.



One “area code” to keep track of inside the Internet.



Clipart: <http://www.clker.com/search/networksym/l>

# Non-Routable Addresses

10.10.3.6

- A typical home router does Network Address Translation (NAT)
- Your ISP gives your home router a real global routable address
- Your router gives out local addresses in a special range (192.168.\*.\* )
- The router maps remote addresses for each connection you make from within your home network

[http://en.wikipedia.org/wiki/Network\\_address\\_translation](http://en.wikipedia.org/wiki/Network_address_translation)

BUT WHEN SHE TRACED THE  
KILLER'S IP ADDRESS... IT WAS  
IN THE 192.168/16 BLOCK!



<http://xkcd.com/742/>



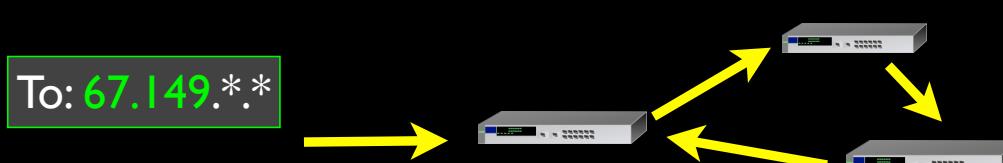
$$\frac{300}{3000} = \frac{1}{10}$$



3 ຂົງກີ່ນ → 10 ກົງ

# Peering into the Internet

- Most systems have a command that will reveal the route taken across the internet (traceroute on Mac and tracert on Windows)
- Each IP packet has a field called “Time to Live” - TTL
- The TTL is used to deal with loops in the network - normally if routers got confused and ended up with a loop - the network would clog up rapidly.



Clipart: <http://www.clker.com/search/networksym/>

# How Traceroute Works

- Normal packets are sent with a Time to Live (TTL) of 255 hops
- Trace route sends a packet with TTL=1, TTL=2, ...
- So each packet gets part-way there and then gets dropped and traceroute gets a notification of where the drop happens
- This builds a map of the nodes that a packet visits when crossing the Internet.

# Traceroute

\$ traceroute www.stanford.edu

```
traceroute to www5.stanford.edu (171.67.20.37), 64 hops max, 40 byte packets
1 141.211.203.252 (141.211.203.252) 1.390 ms 0.534 ms 0.490 ms
2 v-bin-seb.r-bin-seb.umnet.umich.edu (192.122.183.61) 0.591 ms 0.558 ms 0.570 ms
3 v-bin-seb-i2-aa.merit-aa2.umnet.umich.edu (192.12.80.33) 6.610 ms 6.545 ms 6.654 ms
4 192.122.183.30 (192.122.183.30) 7.919 ms 7.209 ms 7.122 ms
5 so-4-3-0.0.rtr.kans.net.internet2.edu (64.57.28.36) 17.672 ms 17.836 ms 17.673 ms
6 so-0-1-0.0.rtr.hous.net.internet2.edu (64.57.28.57) 31.800 ms 41.967 ms 31.787 ms
7 so-3-0-0.0.rtr.losa.net.internet2.edu (64.57.28.44) 63.478 ms 63.704 ms 63.710 ms
8 hpr-lax-hpr--i2-newnet.cenic.net (137.164.26.132) 63.093 ms 63.026 ms 63.384 ms
9 svl-hpr--lax-hpr-10ge.cenic.net (137.164.25.13) 71.242 ms 71.542 ms 76.282 ms
10 oak-hpr--svl-hpr-10ge.cenic.net (137.164.25.9) 72.744 ms 72.243 ms 72.556 ms
11 hpr-stan-ge--oak-hpr.cenic.net (137.164.27.158) 73.763 ms 73.396 ms 73.665 ms
12 bbra-rtr.Stanford.EDU (171.64.1.134) 73.577 ms 73.682 ms 73.492 ms
13 * * *
14 www5.Stanford.EDU (171.67.20.37) 77.317 ms 77.128 ms 77.648 ms
```

# Traceroute

```
$ traceroute www.msu.edu
```

```
traceroute to www.msu.edu (35.8.10.30), 64 hops max, 40 byte packets
```

```
1 141.211.203.252 (141.211.203.252) 2.644 ms 0.973 ms 14.162 ms
2 v-bin-seb.r-bin-seb.umnet.umich.edu (192.122.183.61) 1.847 ms 0.561 ms 0.496 ms
3 v-bin-seb-i2-aa.merit-aa2.umnet.umich.edu (192.12.80.33) 6.490 ms 6.499 ms 6.529 ms
4 lt-0-3-0x1.eq-chi2.mich.net (198.108.23.121) 8.096 ms 8.113 ms 8.103 ms
5 xe-0-0-0x23.msu6.mich.net (198.108.23.213) 7.831 ms 7.962 ms 7.965 ms
6 192.122.183.227 (192.122.183.227) 12.953 ms 12.339 ms 10.322 ms
7 cc-tl-gel-23.net.msu.edu (35.9.101.209) 9.522 ms 9.406 ms 9.817 ms
8 * * *
```

# Traceroute

```
$ traceroute www.pku.edu.cn
```

```
traceroute: Warning: www.pku.edu.cn has multiple addresses; using 162.105.129.104
```

```
traceroute to www.pku.edu.cn (162.105.129.104), 64 hops max, 40 byte packets
```

1	141.211.203.252 (141.211.203.252)	1.228 ms	0.584 ms	0.592 ms	
2	v-bin-seb.r-bin-seb.umnet.umich.edu (192.122.183.61)	0.604 ms	0.565 ms	0.466 ms	
3	v-bin-seb-i2-aa.merit-aa2.umnet.umich.edu (192.12.80.33)	7.511 ms	6.641 ms	6.588 ms	
4	192.122.183.30 (192.122.183.30)	12.078 ms	6.989 ms	7.619 ms	
5	192.31.99.133 (192.31.99.133)	7.666 ms	8.953 ms	17.861 ms	Michigan
6	192.31.99.170 (192.31.99.170)	59.275 ms	59.273 ms	59.108 ms	Tennessee
7	134.75.108.209 (134.75.108.209)	173.614 ms	173.552 ms	173.333 ms	
8	134.75.107.10 (134.75.107.10)	256.760 ms	134.75.107.18 (134.75.107.18)	256.574 ms	256.53
9	202.112.53.17 (202.112.53.17)	256.761 ms	256.801 ms	256.688 ms	Seoul
10	202.112.61.157 (202.112.61.157)	257.416 ms	257.960 ms	257.747 ms	
11	202.112.53.194 (202.112.53.194)	256.827 ms	257.068 ms	256.962 ms	
12	202.112.41.202 (202.112.41.202)	256.800 ms	257.053 ms	256.933 ms	Beijing

# The perfect is the enemy of the good

*Le mieux est l'ennemi du bien. –Voltaire*

- IP Does: Best effort to get data across bunch of hops from one network to another network
- IP Does Not: Guarantee delivery - if things go bad - the data can vanish
- Best effort to keep track of the good and bad paths for traffic - tries to pick better paths when possible
- This makes it fast and scalable to very large networks - and ultimately “reliable” because it does not try to do too much

10.10.0.15 → DNS → it-brach

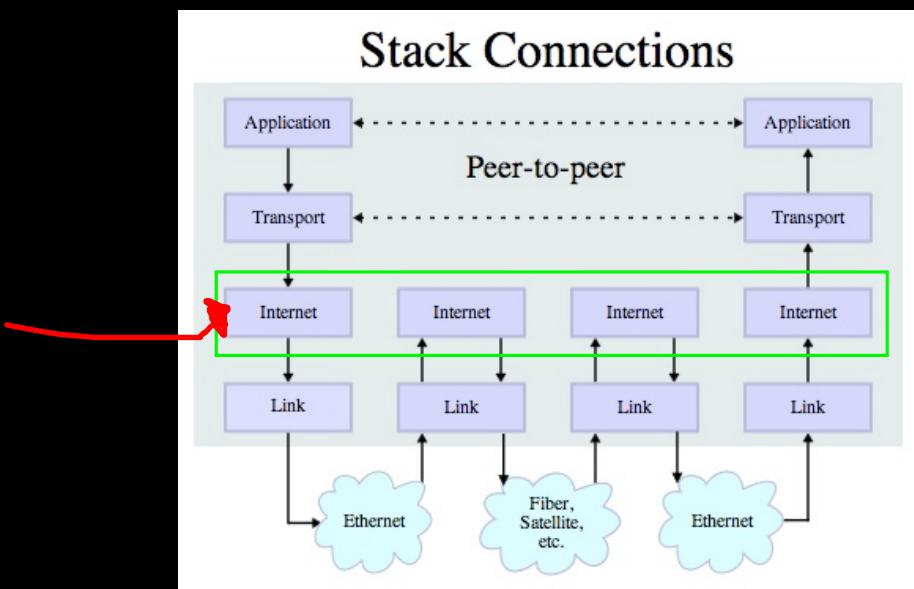
## Domain Name System

The Domain Name System  
convert user-friendly  
names, like

www.umich.edu

to network-friendly IP  
addresses, like

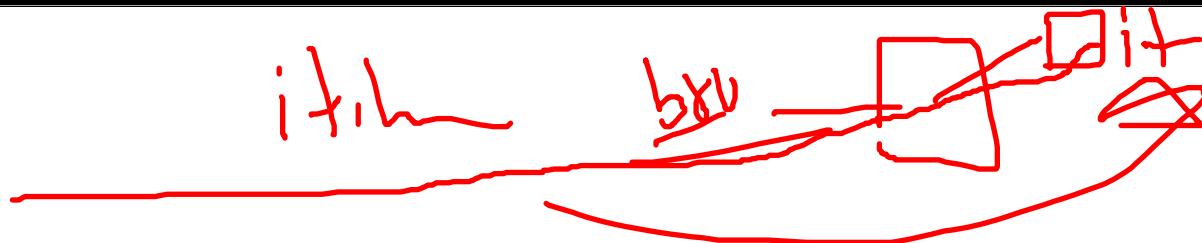
141.211.32.166



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Domain Name System

- Numeric addresses like 141.211.63.45 are great for Internet routers but lousy for people
- Each campus ends up with a lot of networks (141.211.\*.\* , 65.43.21.\*)
- Sometimes (rarely) the IP address numbers get reorganized
- When servers physically move they need new IP addresses



# DNS: Internet Address Book

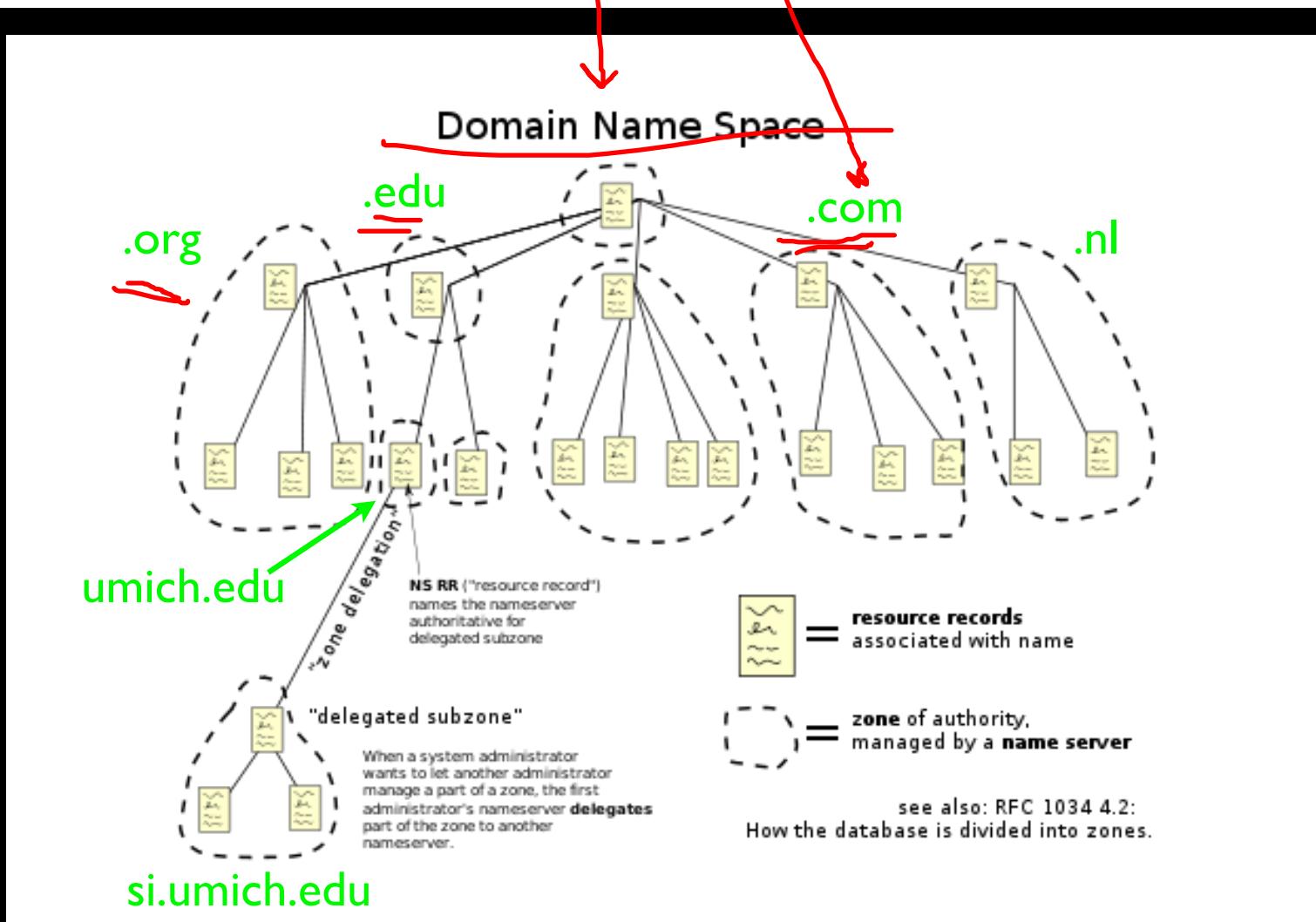
- The Domain Name System is a big fast distributed database of Internet names to Internet “phone numbers”
- IP Addresses reflect technical “geography”
  - **141.211.63.44** - read left to right like a phone number  

  - Domain names reflect organizational structure
    - **www.si.umich.edu** - read right to left like postal address  

    - **311 West Hall, Ann Arbor, MI 440109, USA, Earth**  


http://  .jav. ac. in

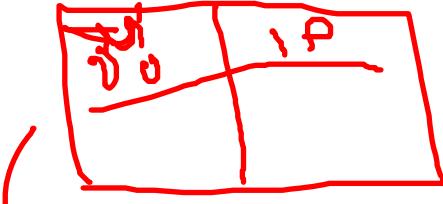
sub domain  
 https:// it.bm .co .ath  
 ac.h



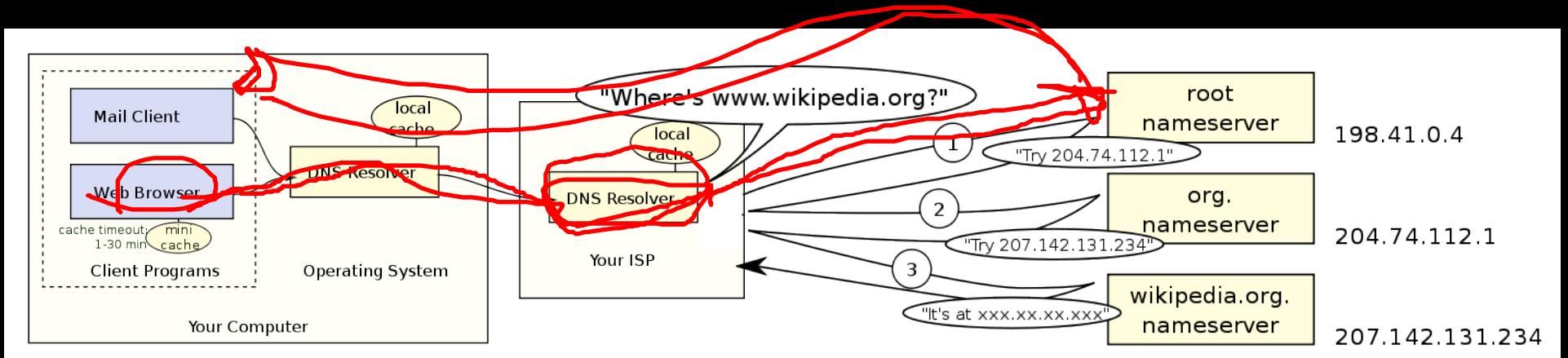
[http://en.wikipedia.org/wiki/File:Domain\\_name\\_space.svg](http://en.wikipedia.org/wiki/File:Domain_name_space.svg)

# Looking Up A Domain Name

- When a system sees a new domain name, it looks it up from its "closest" domain name server - this "close" domain name server is configured or handed out when the computer connects to the network
- If that DNS server does not know the answer, the server works its way up the tree and then back down to a server that knows the mapping.
- Then the local Domain Name server remembers the mapping so it only asks once per 24 hours (caching)



# Looking Up A Domain Name



[http://en.wikipedia.org/wiki/File:DNS\\_in\\_the\\_real\\_world.svg](http://en.wikipedia.org/wiki/File:DNS_in_the_real_world.svg)

[http://en.wikipedia.org/wiki/File:An\\_example\\_of\\_theoretical\\_DNS\\_recursion.svg](http://en.wikipedia.org/wiki/File:An_example_of_theoretical_DNS_recursion.svg)

# DNS Caching

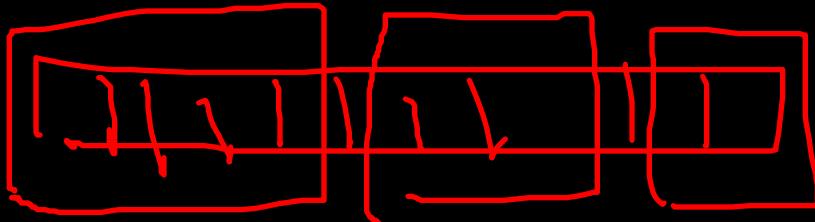
- In order to maintain speed and not waste too much network traffic on name lookups, each DNS server keeps a copy of any name to IP address mapping
- Typically a DNS will hold on to a mapping for up to 24 hours
- This means that a lookup will only generate backbone traffic once every 24 hours
- It also means if the mapping is wrong or changed - most servers will not see new information for up to 24 hours

<http://en.wikipedia.org/wiki/Cache>

# Getting your own .com or .org.

- You must choose an Internet Service Provider and get an account
  - [www.godaddy.com](http://www.godaddy.com)
  - [www.hostmonster.com](http://www.hostmonster.com) .....
- Choose a name that is not taken
  - Can be harder than you think
- Pay roughly \$10.00 per year

# Transport Layer



Session

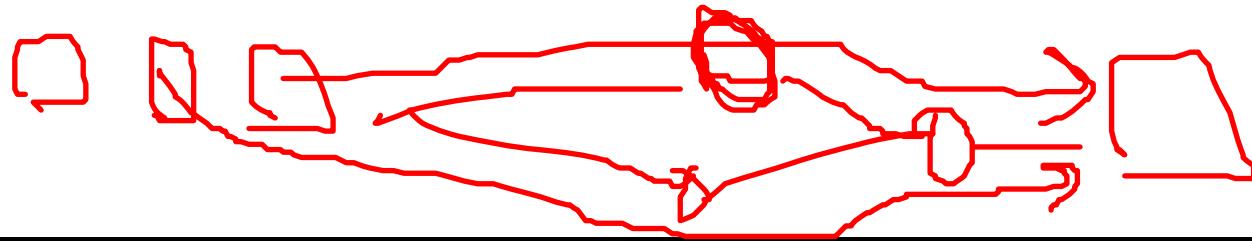
[http://en.wikipedia.org/wiki/  
Transmission\\_Control\\_Protocol](http://en.wikipedia.org/wiki/Transmission_Control_Protocol)

Application Layer  
Web, E-Mail, File Transfer

Transport Layer (TCP)  
Reliable Connections

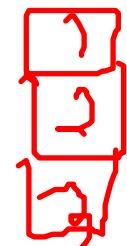
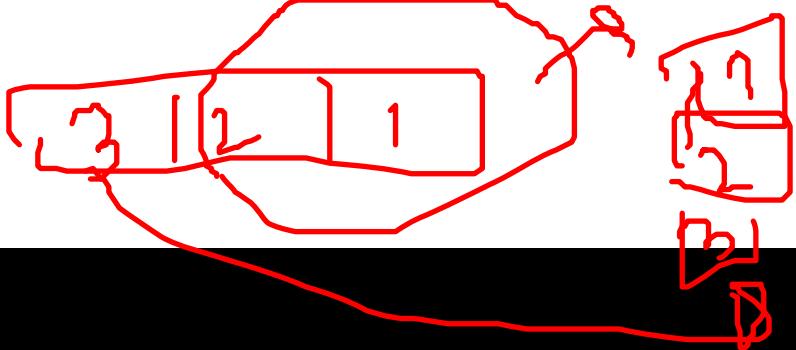
Internetwork Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections



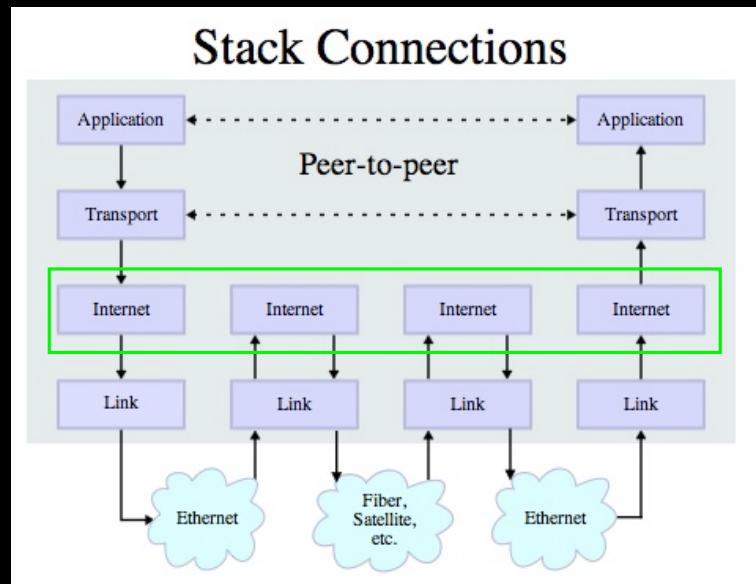
## Review: The Magic of IP

- What it does - Tries to get data a bunch of hops from one network to another network
- Keeps track of the good and bad paths for traffic - tries to pick better paths when possible
- But no guarantee of delivery - if things go bad - the data vanishes
- This makes it fast and big - and ultimately “reliable” because it does not try to do too much



# Internet Protocol

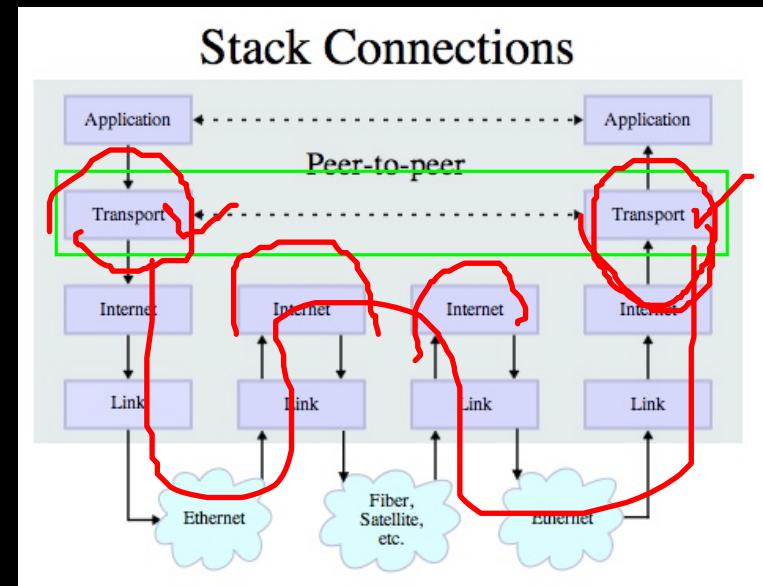
- So many links / hops
- So many routes
- Thinks can change dynamically and IP has to react (links up/down)
- IP can drop packets



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Transport Protocol (TCP)

- Built on top of IP
- Assumes IP might lose some data
- In case data gets lost - we keep a copy of the data and send until we get an acknowledgement
- If it takes “too long” - just send it again



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

Sender



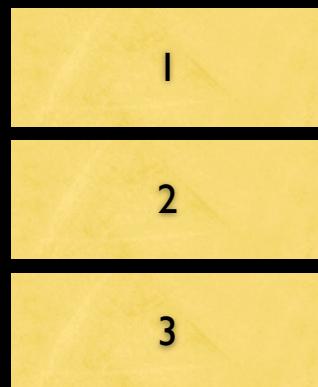
Receiver

Break Messages  
into Pieces

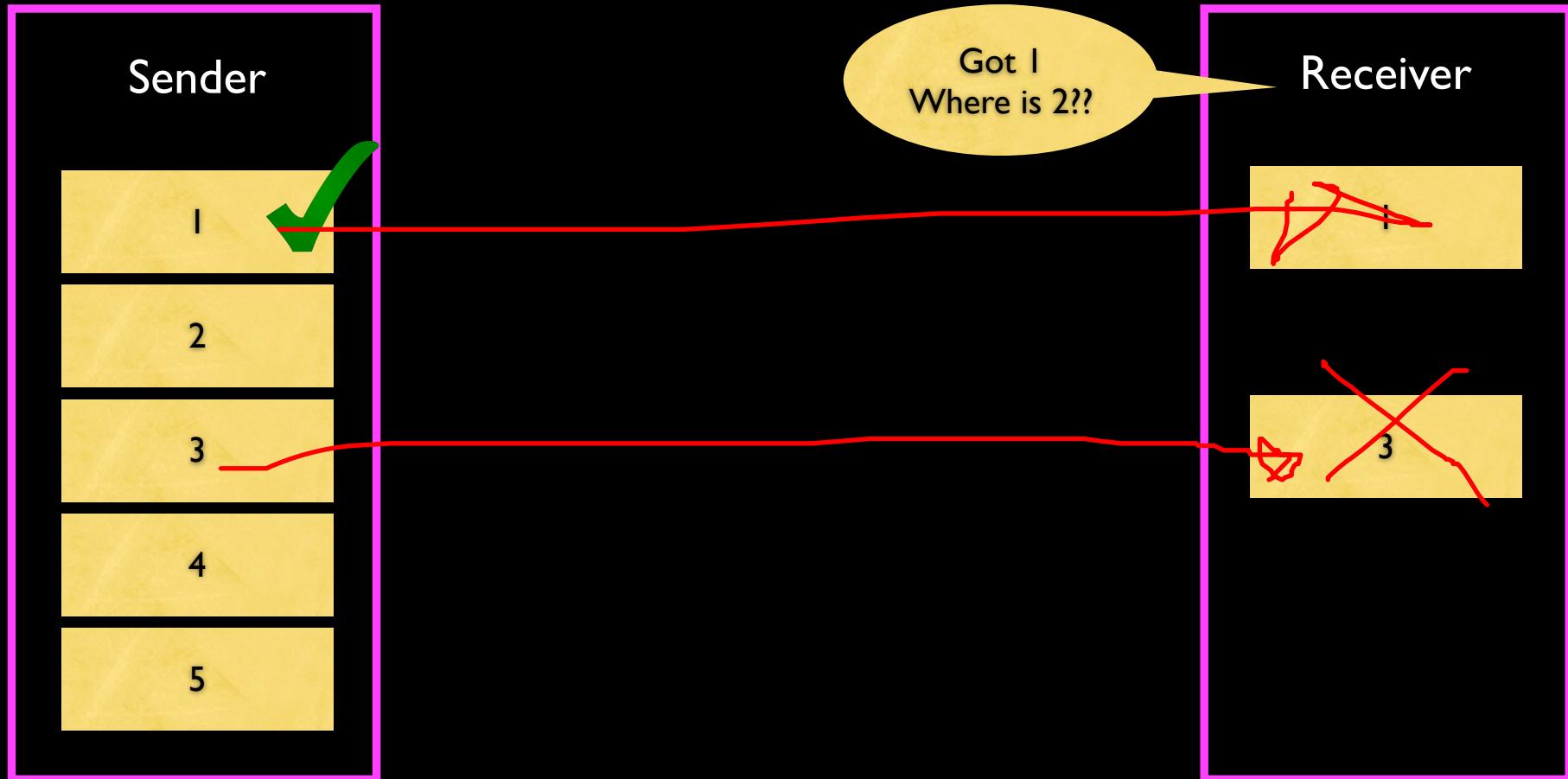
Sender



Receiver



Break Messages  
into Pieces



Break Messages  
into Pieces

Sender

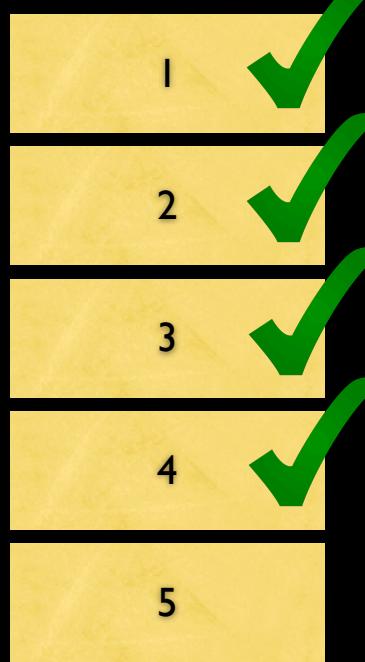


Receiver

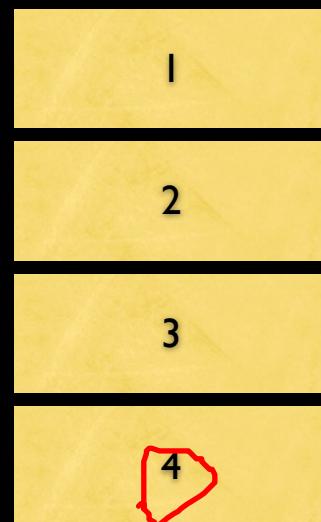


Break Messages  
into Pieces

Sender

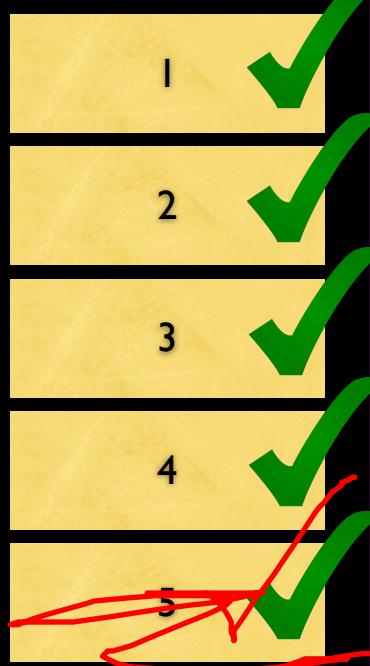


Receiver

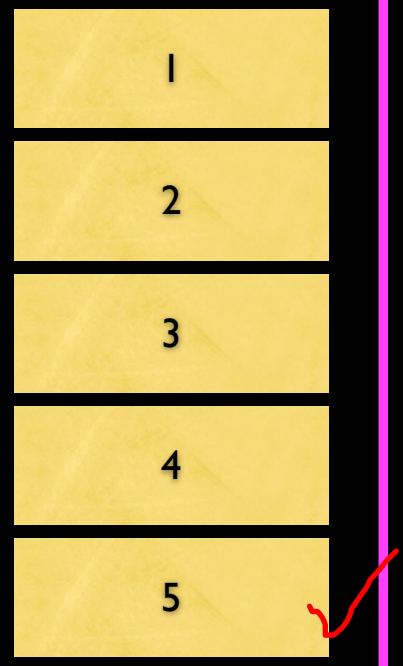


Break Messages  
into Pieces

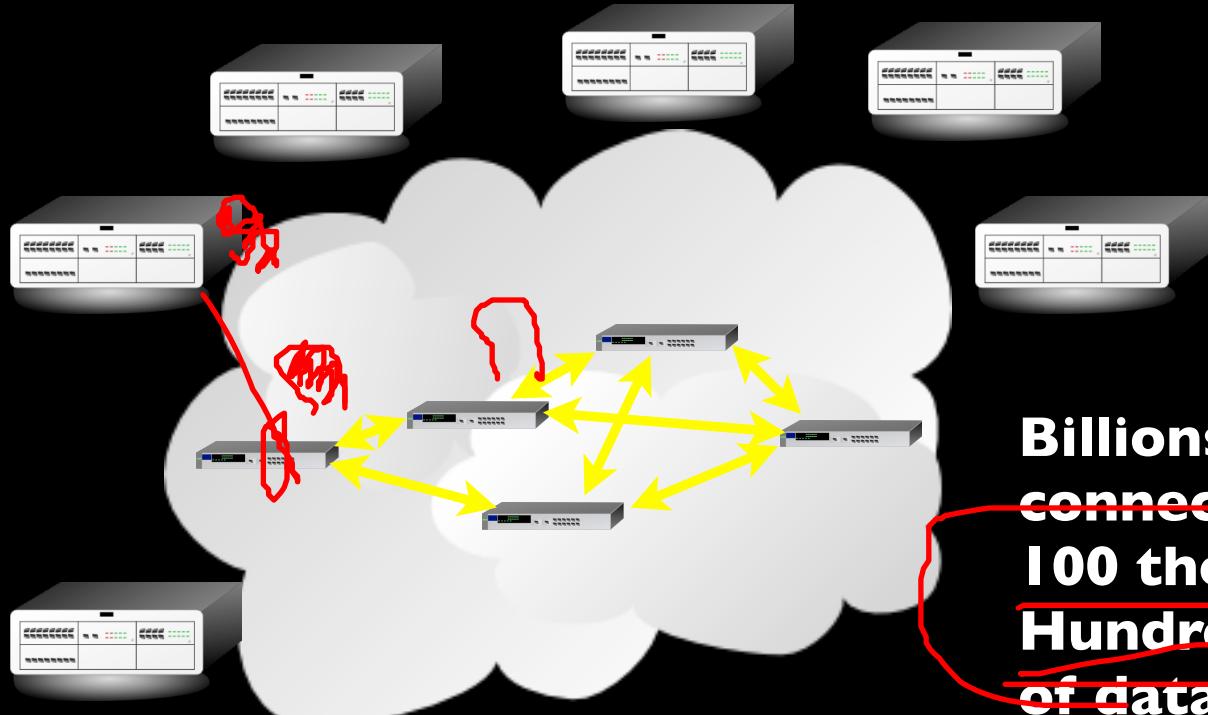
Sender



Receiver



Break Messages  
into Pieces



**Billions of computers  
connected to the internet;  
100 thousands of routers.  
Hundreds of billions bytes  
of data enroute at any  
moment.**

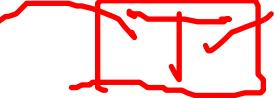
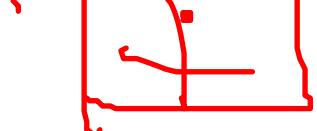
**Storage of enroute data  
done at the edges only!**

# Transport Protocol (TCP)

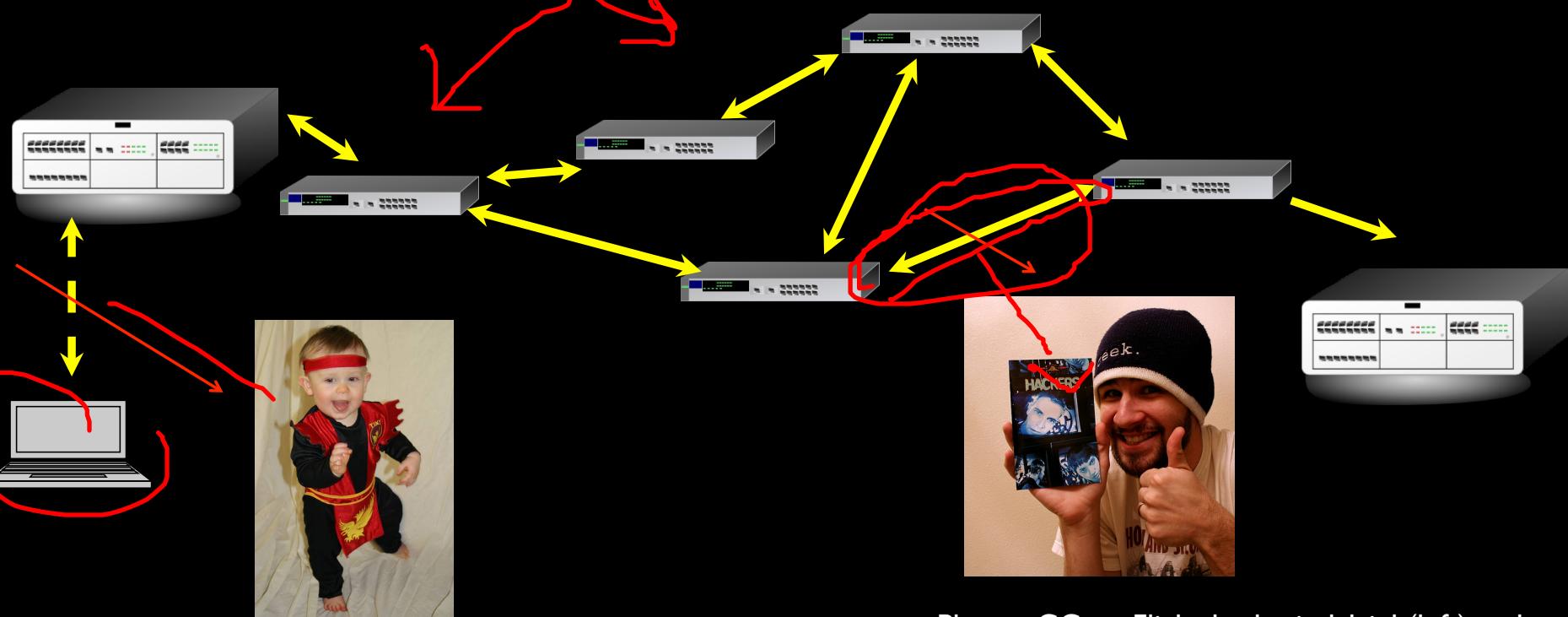
- The responsibility of the transport layer is to present a reliable end-to-end pipe to the application
- Data either arrives in the proper order or the connection is closed
- TCP keeps buffers in the sending and destination system to keep data which has arrived out of order or to retransmit if necessary
- TCP provides individual connections between applications

# Security for TCP

[http://en.wikipedia.org/wiki/Secure\\_Sockets\\_Layer](http://en.wikipedia.org/wiki/Secure_Sockets_Layer)

DNS                  MAC, IP

# Who's Listening as your Data Hops Along?



Photos CC on Flickr by: karindalziel (left) and ortizmj12 (right) ([flickr](#))



# Transport Layer Security (TLS)

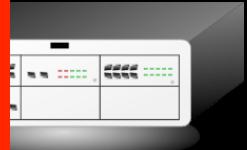
- Used to be called “Secure Sockets Layer” (SSL)
- Can view it as an extra layer “between” TCP and the application layer
- It is very difficult but not impossible to break this security - normal people do not have the necessary compute resources to break TLS
- Encrypting and decryption takes resources - so we use it for things when it is needed
- The IP and TCP are unaware whether data has been encrypted

# Secure Application Protocols

- There are often secure and unencrypted application protocols
  - <http://www.facebook.com>
  - <https://www.facebook.com>
- Your browser tells you when using a secure connection - you should never type passwords into a non-secure connection
- Especially over wireless - especially at a security conference...



## System to System Secure TCP/IP



Your local connection  
(particularly when  
wireless) is your greatest  
exposure.



Generally, the backbone of  
the Internet is pretty  
secure to prying eyes  
from generic baddies...

[http://en.wikipedia.org/wiki/Secure\\_Sockets\\_Layer](http://en.wikipedia.org/wiki/Secure_Sockets_Layer)

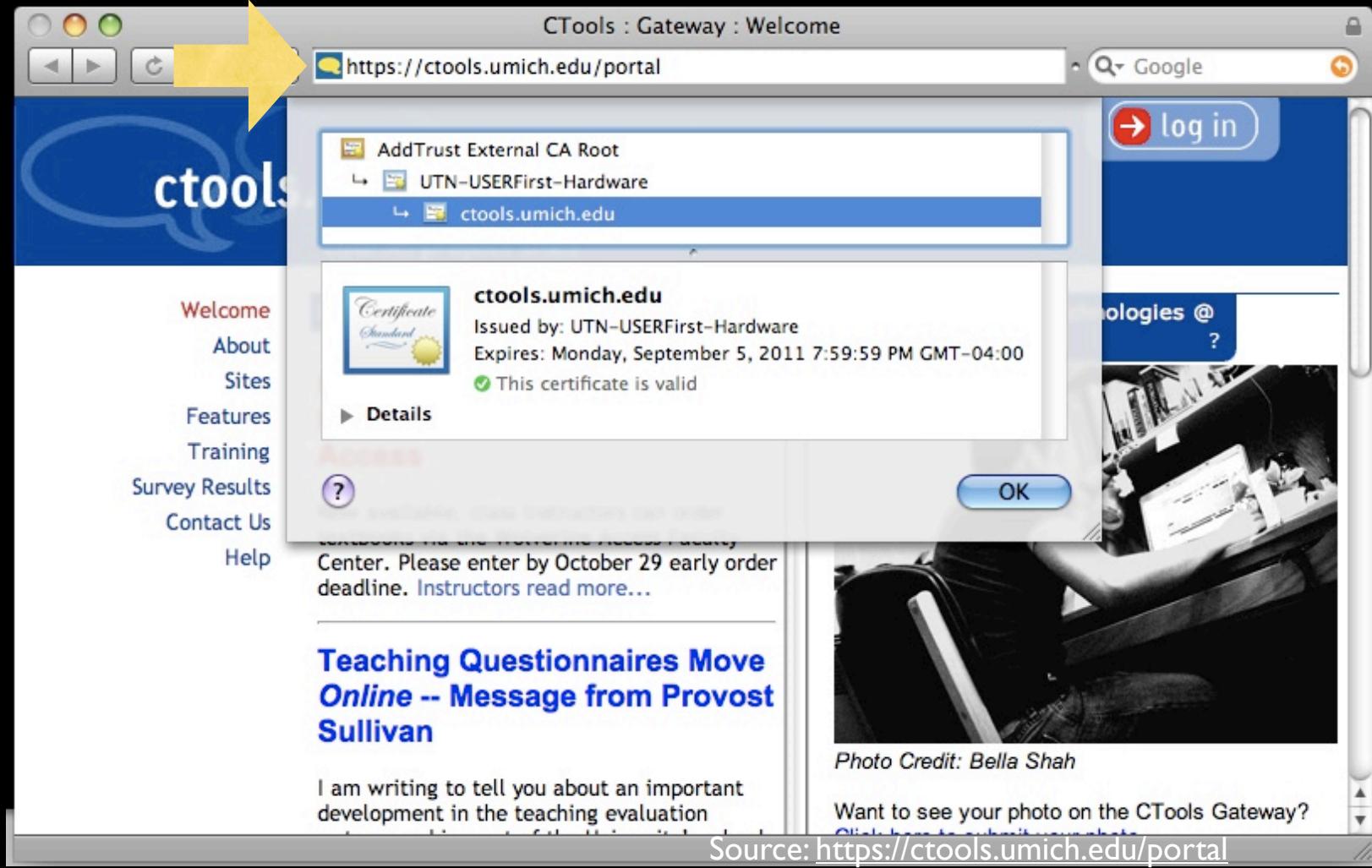
Clipart: <http://www.clker.com/search/networksym/>  
Photo CC BY: karindalziel ([flickr](#))  
<http://creativecommons.org/licenses/by/2.0/>

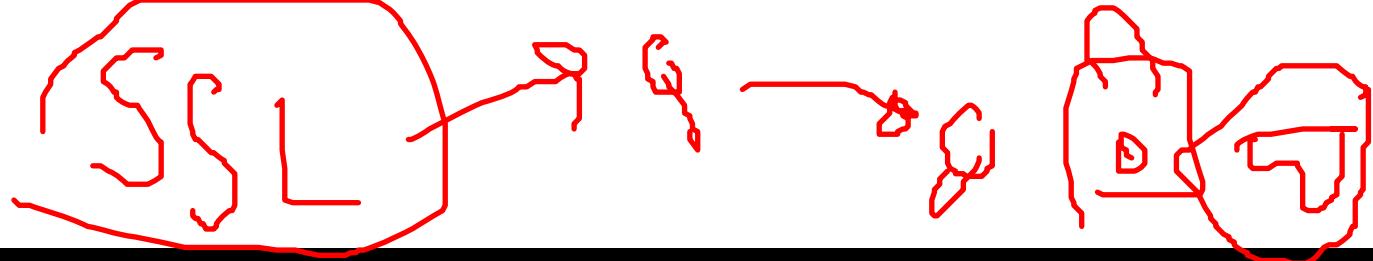
IP MAC

## Spoofing

- SSL (TLS) provides pretty good security
- As long as you know who you're talking to!!







# Digital Certificates

In cryptography, a **public key** certificate (also known as a digital certificate or identity certificate) is an **electronic document** which uses a digital signature to bind a public key with an identity — information such as the name of a person or an organization, their address, and so forth. **The certificate can be used to verify that a public key belongs to an individual.**

[http://en.wikipedia.org/wiki/Public\\_key\\_certificate](http://en.wikipedia.org/wiki/Public_key_certificate)

# Certificate Authority (CA)

A certificate authority is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. A CA is a trusted third party that is trusted by both the owner of the certificate and the party relying upon the certificate.

[http://en.wikipedia.org/wiki/Certificate\\_authority](http://en.wikipedia.org/wiki/Certificate_authority)

VeriSign Authentication Services – The leading provider of SSL. Products ...rotection, malware scan, code signing & public key infrastructure (PKI).

Now from **Symantec** VeriSign Authentication Services

Products & Services ▾ Partners ▾ Support ▾ My Account ▾

**Trust Means Business**

Everyone says their site is secure.  
Make sure your customers know it.

Learn more ➤

1 2 3 4

**BUY** SSL Certificates  
**BUY** VeriSign Trust Seal  
**BUY** Code Signing  
**TRY** Free Trial NEW!  
**RENEW** Renew SSL Certificates  
**SIGN IN** VeriSign Trust Center

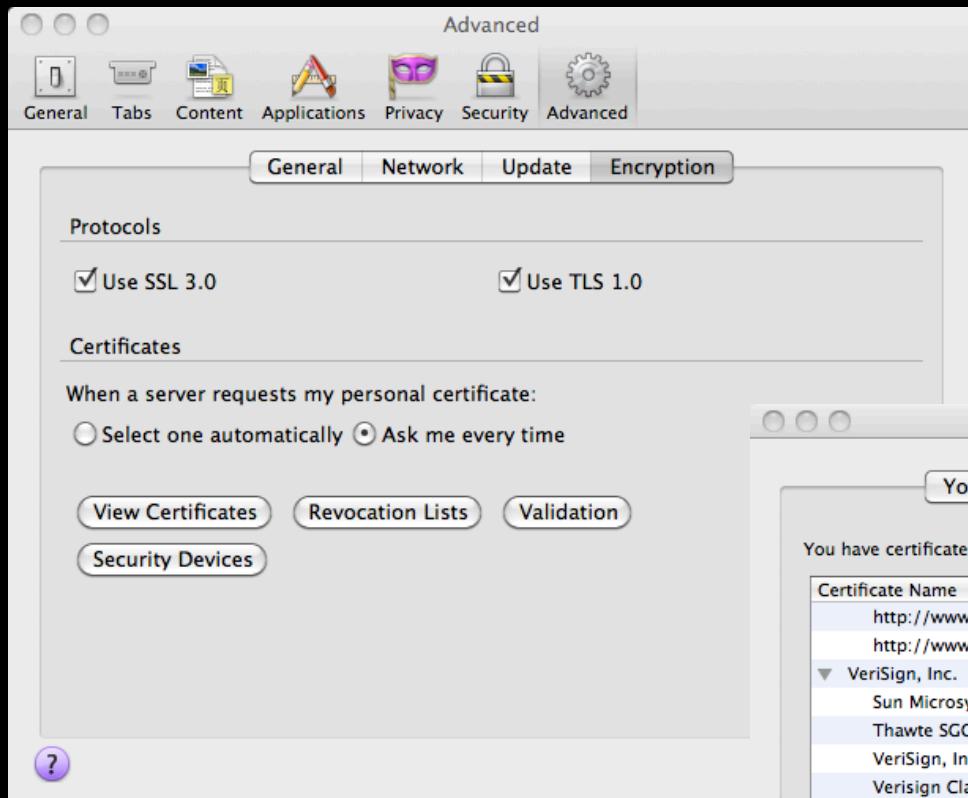
**Trust from Search to Browse to Buy**  
Boost your site traffic and conversions with powerful trust features. Free with every SSL Certificate.

Hobbs Travel Safaris Great prices & world class service to thousands of customers. Hawaiian Packages voted best by Travel in Style three years running. [Get quote](#) [Check - Similar](#)

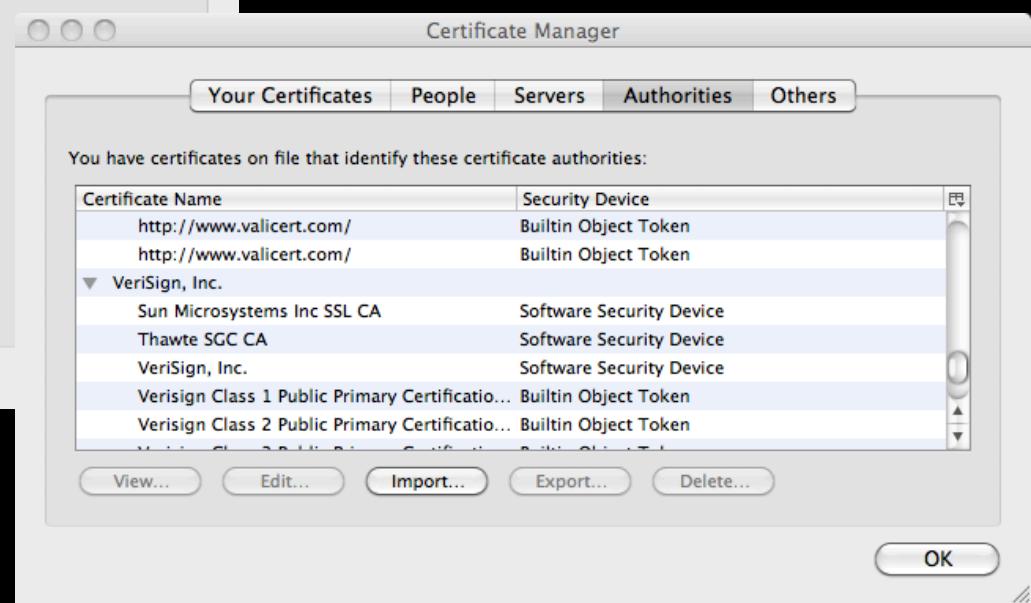
**Protect your Business from Online Threats**  
Find a Symantec solution to secure, backup and manage your valuable data.

VeriSign Protection Suite VeriSign Email Protection VeriSign Endpoint Protection

**VERISIGN**  
Find Whois, Registrar Information, Domain Name Services, Managed DNS, DDoS Protection and iDefense at



Your browser comes with certificates/public keys from some certificate authorities built in. Like Verisign.





# Application Layer

Application Layer  
Web, E-Mail, File Transfer

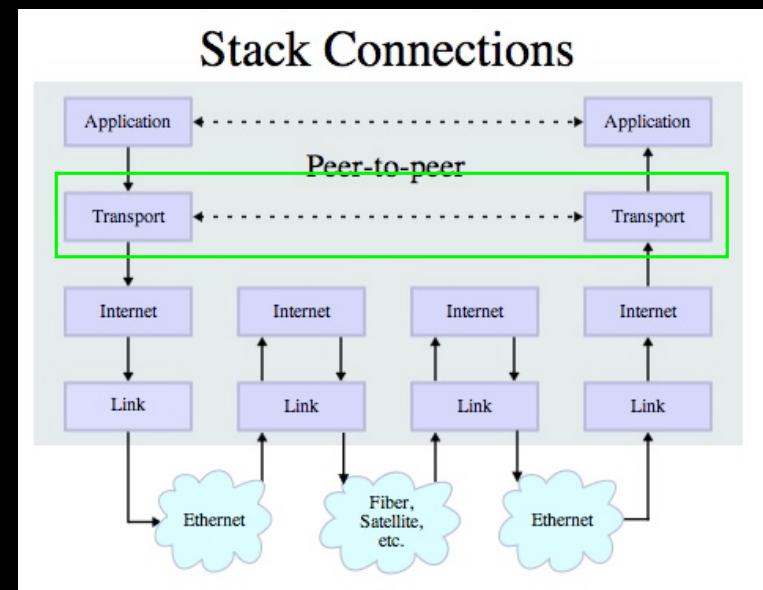
Transport Layer (TCP)  
Reliable Connections

Internetwork Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections

# Quick Review

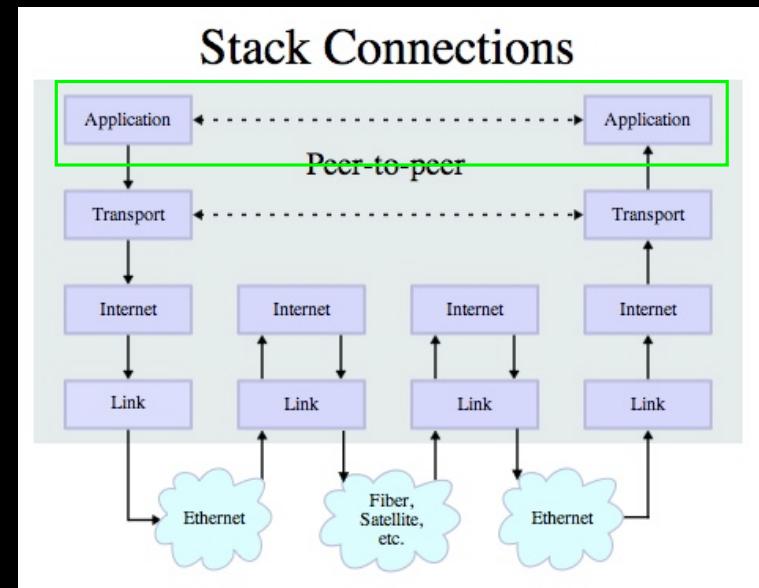
- Link layer: gets the data onto the link, and manages collisions on a single hop
- Internet layer: moves the data over one hop, trying to get it “closer” to its destination
- Transport layer: Assumes that the internet layer may lose data, so request retransmission when needed—provides a nice reliable pipe from source to destination



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Application Protocol

- Since TCP gives us a reliable pipe, what do we want to do with the pipe? What problem do we want to solve?
  - Mail ✓
  - World Wide Web ✓
  - Stream kitty videos ✓



Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# Two Questions for the Application Layer

- Which application gets the data?
  - Ports
- What are the rules for talking with that application?
- Protocols

HTTP : 80

FTP : 21

SSH : 22

SMTP : 25

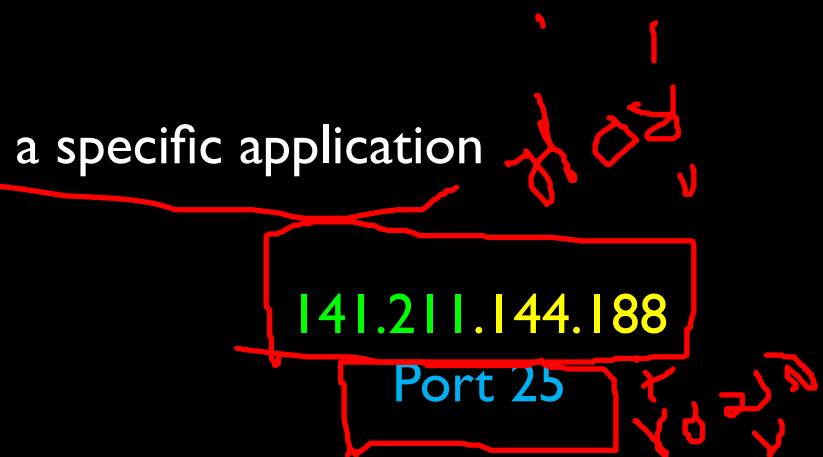
[http://en.wikipedia.org/wiki/TCP\\_and\\_UDP\\_port](http://en.wikipedia.org/wiki/TCP_and_UDP_port)

[http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)

# Ports

- Like extensions in a phone number
- The IP address network number (the area code) gets to the LAN
- The IP address host number (the telephone number) gets you to the destination machine
- The port (the extension) gets you to a specific application

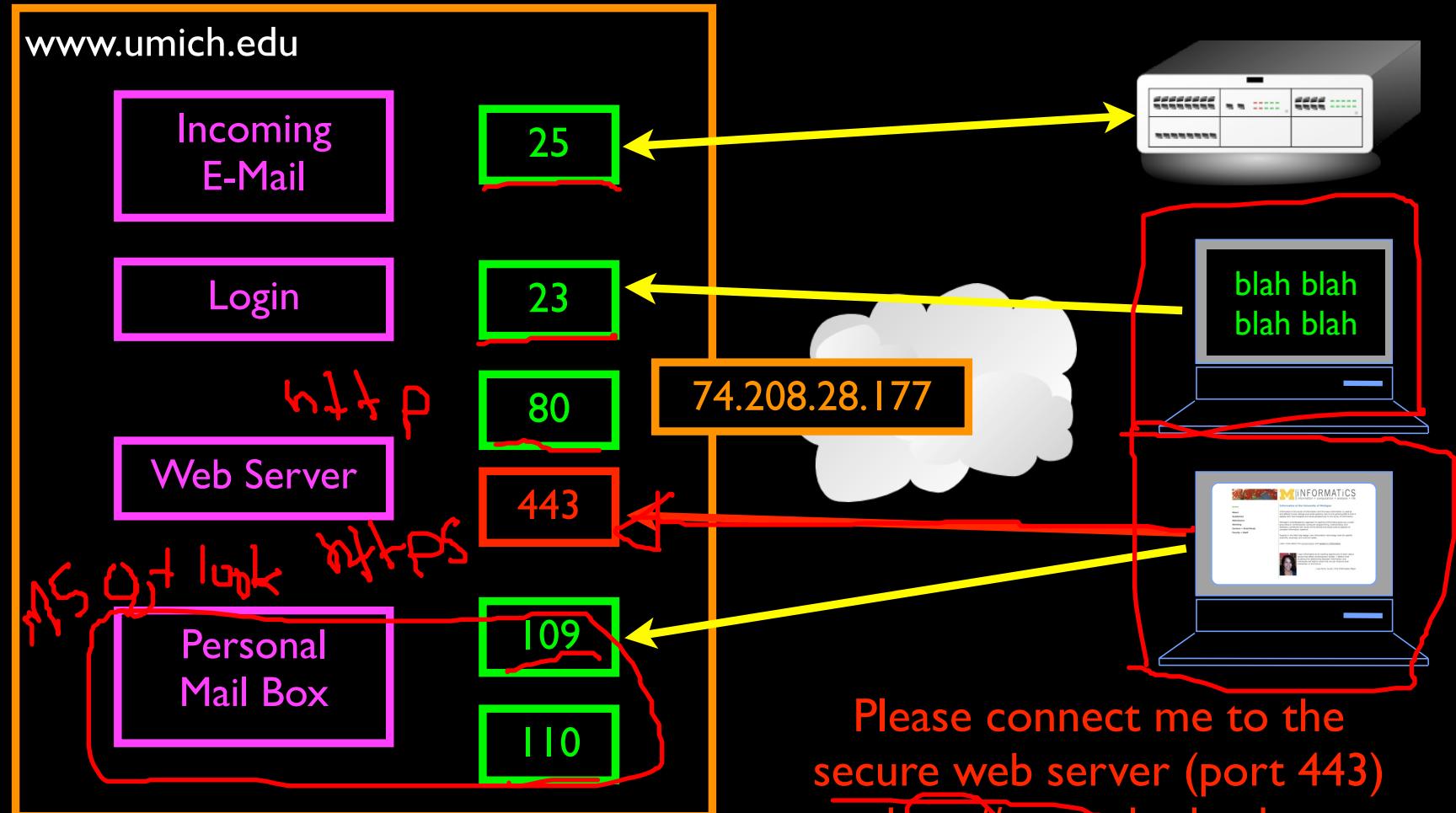
(734) 764 1855, ext. 27



# TCP, Ports, and Connections

[http://en.wikipedia.org/wiki/TCP\\_and\\_UDP\\_port](http://en.wikipedia.org/wiki/TCP_and_UDP_port)

[http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)



Clipart: <http://www.clker.com/search/networksym/>

# Common TCP Ports

- Telnet (23) - Login
- SSH (22) - Secure Login
- HTTP (80) ✓
- HTTPS (443) - Secure
- SMTP (25) (Mail)

- IMAP (143/220/993) - Mail Retrieval X

- POP (109/110) - Mail Retrieval X

- DNS (53) - Domain Name

- FTP (21) - File Transfer

[http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers)

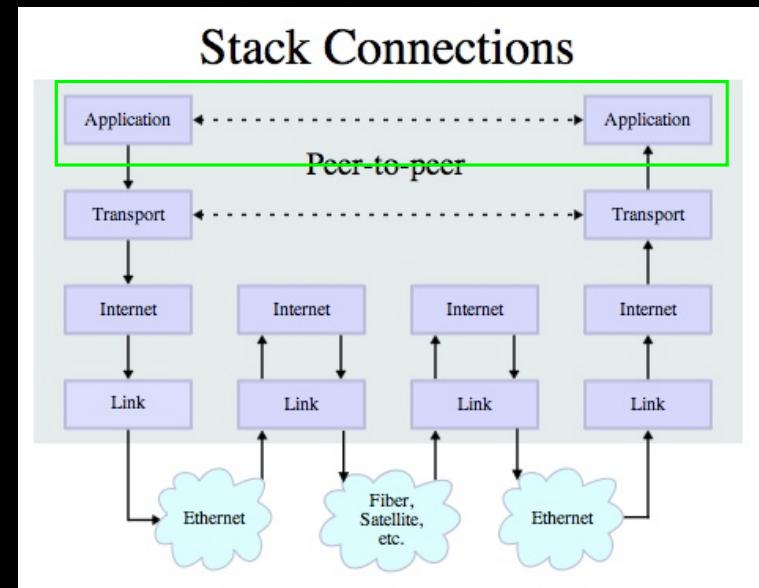
# Application Protocols

<http://en.wikipedia.org/wiki/Http>

<http://en.wikipedia.org/wiki/Pop3>

# Application Protocol

- Since TCP gives us a reliable pipe, what do we want to do with the pipe? What problem do we want to solve?
- Mail
- World Wide Web
- Stream kitty videos



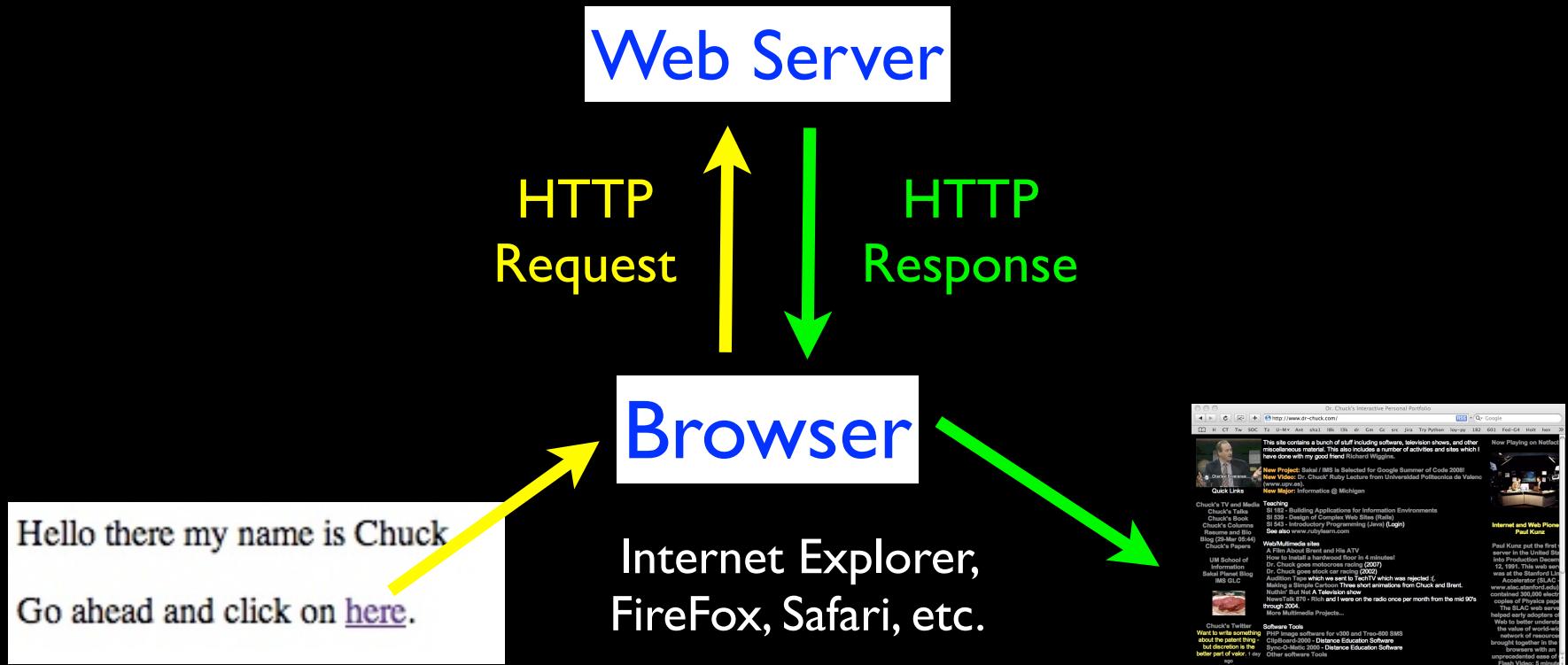
Source: [http://en.wikipedia.org/wiki/Internet\\_Protocol\\_Suite](http://en.wikipedia.org/wiki/Internet_Protocol_Suite)

# HTTP - Hypertext Transport Protocol

- The dominant Application Layer Protocol on the Internet
- Invented for the Web - to Retrieve HTML, Images, Documents etc
- Extended to be data in addition to documents - RSS, Web Services, etc..
- Basic Concept - Make a Connection - Request a document - Retrieve the Document - Close the Connection

<http://en.wikipedia.org/wiki/Http>

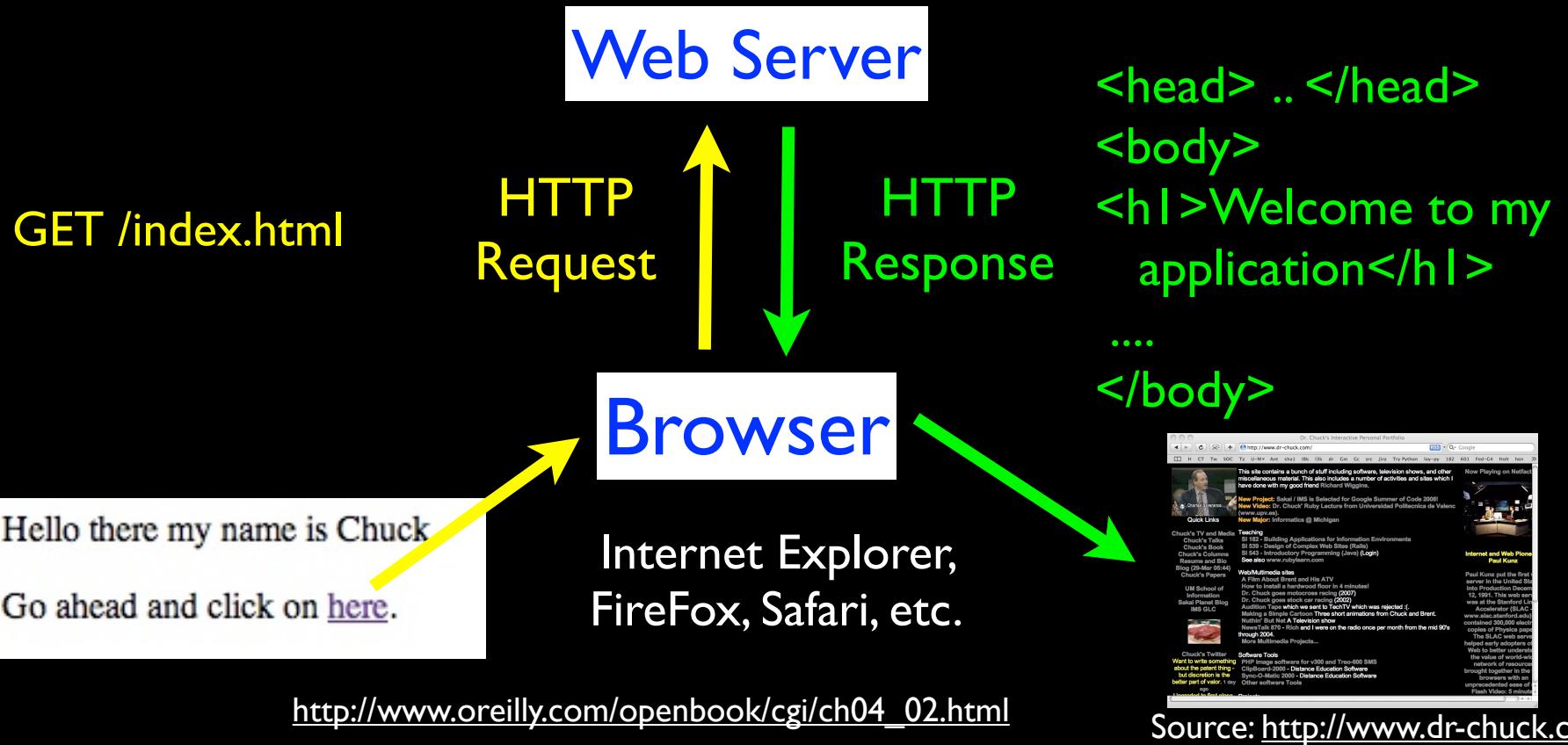
# HTTP Request / Response Cycle



[http://www.oreilly.com/openbook/cgi/ch04\\_02.html](http://www.oreilly.com/openbook/cgi/ch04_02.html)

Source: <http://www.dr-chuck.com/>

# HTTP Request / Response Cycle



Source: <http://www.dr-chuck.com/>

# Internet Standards

- The standards for all of the Internet protocols (inner workings) are developed by an organization
- Internet Engineering Task Force (IETF)
- [www.ietf.org](http://www.ietf.org)
- Standards are called “RFCs” - “Request for Comments”

Network Working Group  
Request for Comments: 1945  
Category: Informational

T. Berners-Lee  
MIT/LCS  
R. Fielding  
UC Irvine  
H. Frystyk  
MIT/LCS  
May 1996

## Hypertext Transfer Protocol -- HTTP/1.0

### Status of This Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

### IESG Note:

The IESG has concerns about this protocol, and expects this document to be replaced relatively soon by a standards track document.

### Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing of data representation, allowing systems to be built independently of the data being transferred.

Source: <http://www.ietf.org/rfc/rfc1945.txt>

### 5.1.2 Request-URI

The Request-URI is a Uniform Resource Identifier (Section 3.2) and identifies the resource upon which to apply the request.

```
Request-URI      = absoluteURI | abs_path
```

The two options for Request-URI are dependent on the nature of the request.

The absoluteURI form is only allowed when the request is being made to a proxy. The proxy is requested to forward the request and return the response. If the request is GET or HEAD and a prior response is cached, the proxy may use the cached message if it passes any restrictions in the Expires header field. Note that the proxy may forward the request on to another proxy or directly to the server specified by the absoluteURI. In order to avoid request loops, a proxy must be able to recognize all of its server names, including any aliases, local variations, and the numeric IP address. An example Request-Line would be:

```
GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.0
```

The most common form of Request-URI is that used to identify a resource on an origin server or gateway. In this case, only the absolute path of the URI is transmitted (see Section 3.2.1, `abs_path`). For example, a client wishing to retrieve the resource above directly from the origin server would create a TCP connection to port 80 of the host "`www.w3.org`" and send the line:

```
GET /pub/WWW/TheProject.html HTTP/1.0
```

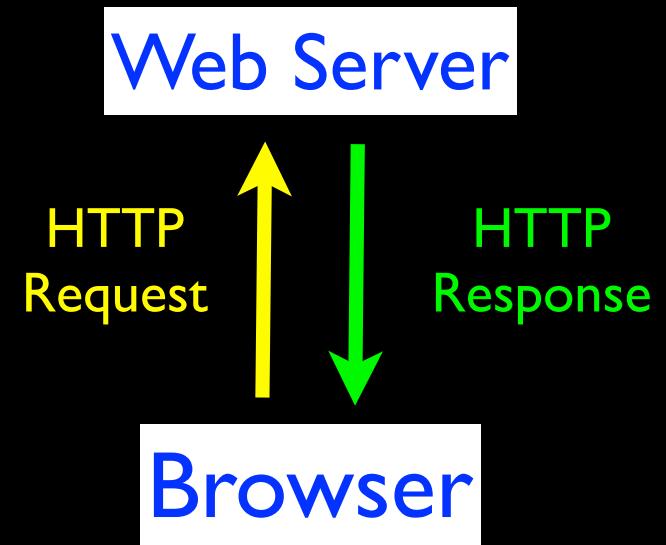
followed by the remainder of the Full-Request. Note that the absolute path cannot be empty; if none is present in the original URI, it must be given as "/" (the server root).

The Request-URI is transmitted as an encoded string, where some characters may be escaped using the "% HEX HEX" encoding defined by RFC 1738 [4]. The origin server must decode the Request-URI in order to properly interpret the request.

Source: <http://www.ietf.org/rfc/rfc1945.txt>

# “Hacking” HTTP

```
Last login: Wed Oct 10 04:20:19 on ttys2  
si-csev-mbp:~ csev$ telnet www.dr-chuck.com 80  
Trying 74.208.28.177...  
Connected to www.dr-chuck.com.  
Escape character is '^]'.  
GET http://www.dr-chuck.com/page1.htm  
<h1>The First Page</h1>  
<p>  
If you like, you can switch to the  
<a href="http://www.dr-chuck.com/page2.htm">  
Second Page</a>.  
</p>
```

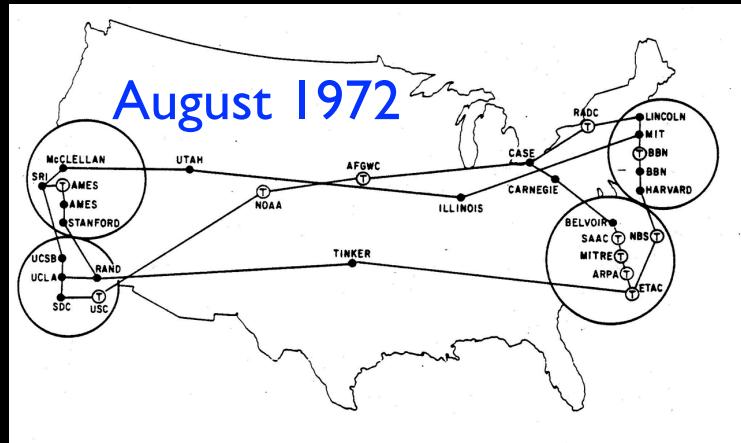


Port 80 is the non-encrypted HTTP port

# Application Layer Summary

- We start with a “pipe” abstraction - we can send and receive data on the same “socket”
- We can optionally add a security layer to TCP using SSL - Secure Socket Layer (aka TLS - Transport Layer Security)
- We use well known “port numbers” so that applications can find a particular application \*within\* a server such as a mail server, web service, etc

# The Architecture of the Internet



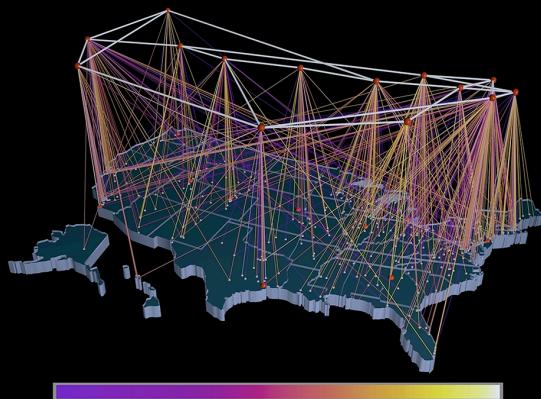
Application Layer  
Web, E-Mail, File Transfer

Transport Layer (TCP)  
Reliable Connections

Internetwork Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections

# The Architecture of the Internet

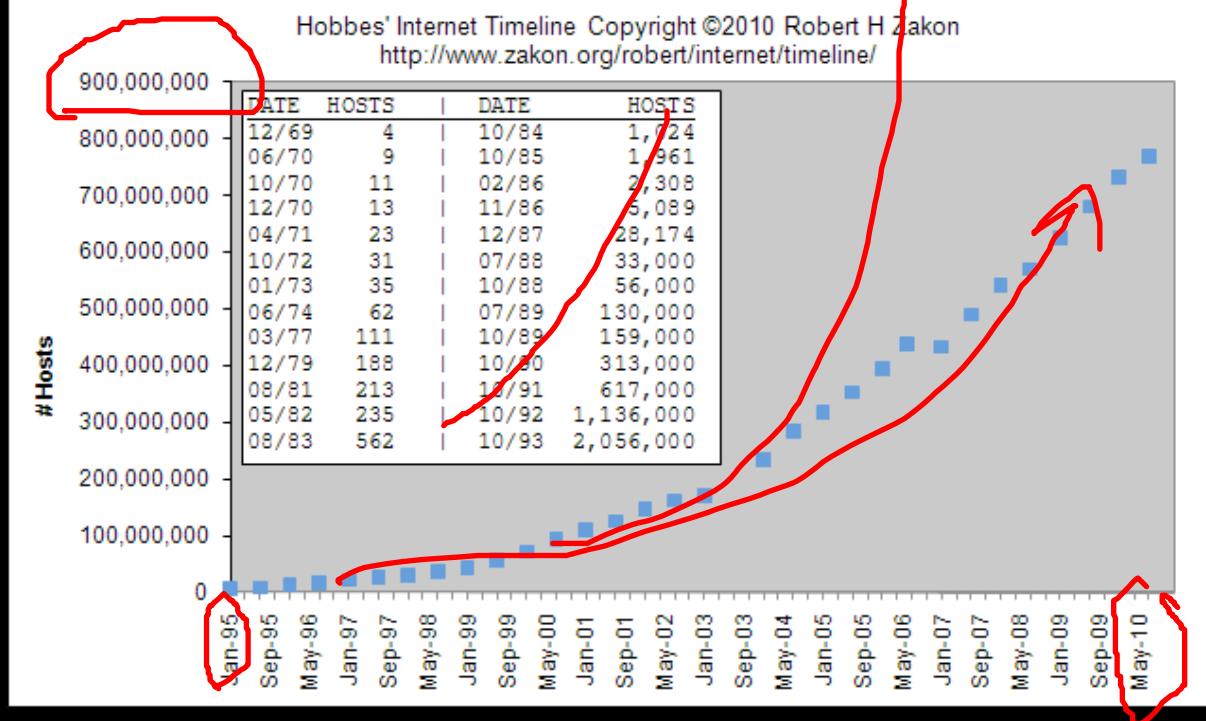


Application Layer  
Web, E-Mail, File Transfer

Transport Layer (TCP)  
Reliable Connections

Internetwork Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections



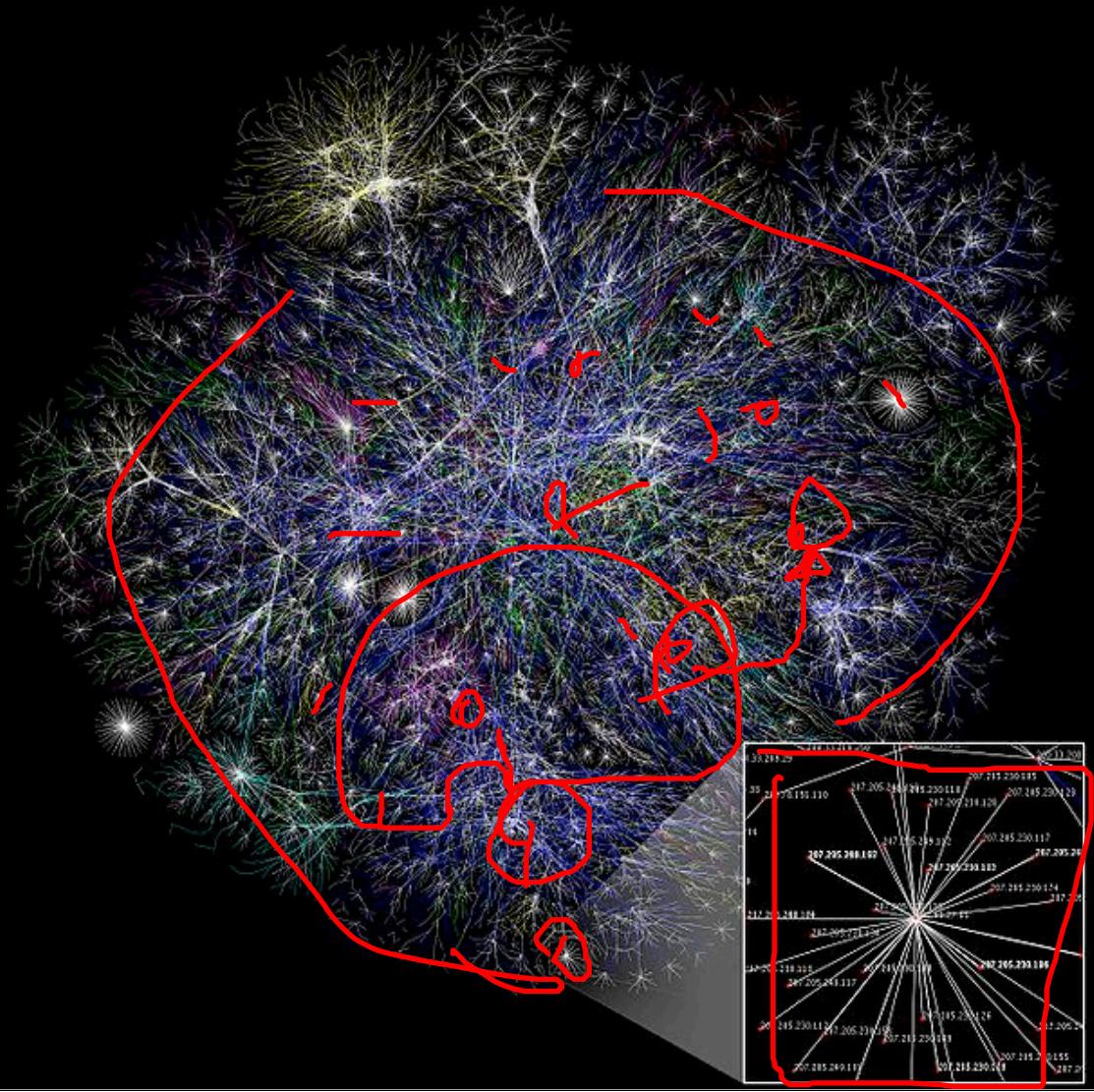
<http://www.zakon.org/robert/internet/timeline/>

Application Layer  
Web, E-Mail, File Transfer

Transport Layer (TCP)  
Reliable Connections

Internetes Layer (IP)  
Simple, Unreliable

Link Layer (Ethernet, WiFi)  
Physical Connections



**Application Layer**  
Web, E-Mail, File Transfer

**Transport Layer (TCP)**  
Reliable Connections

**Internetwork Layer (IP)**  
Simple, Unreliable

**Link Layer (Ethernet, WiFi)**  
Physical Connections

[http://en.wikipedia.org/wiki/File:Internet\\_map\\_1024.jpg](http://en.wikipedia.org/wiki/File:Internet_map_1024.jpg)

# The Internet:An Amazing Design

- Hundreds of millions of computers
- Thousands of routers inside the Internet
- Hundreds of millions of simultaneous connections
- Trillions of bytes of data moved per second around the world
- And it works

# The Internet

- It is said that “The Internet is the largest single engineering effort ever created by mankind”
- It was created to work in an organic way - to repair itself and automatically adjust when parts fail
- No one part of the Internet knows all of the Internet (like life)
- It is never 100% up - but it seems up all the time

Up time = Online time