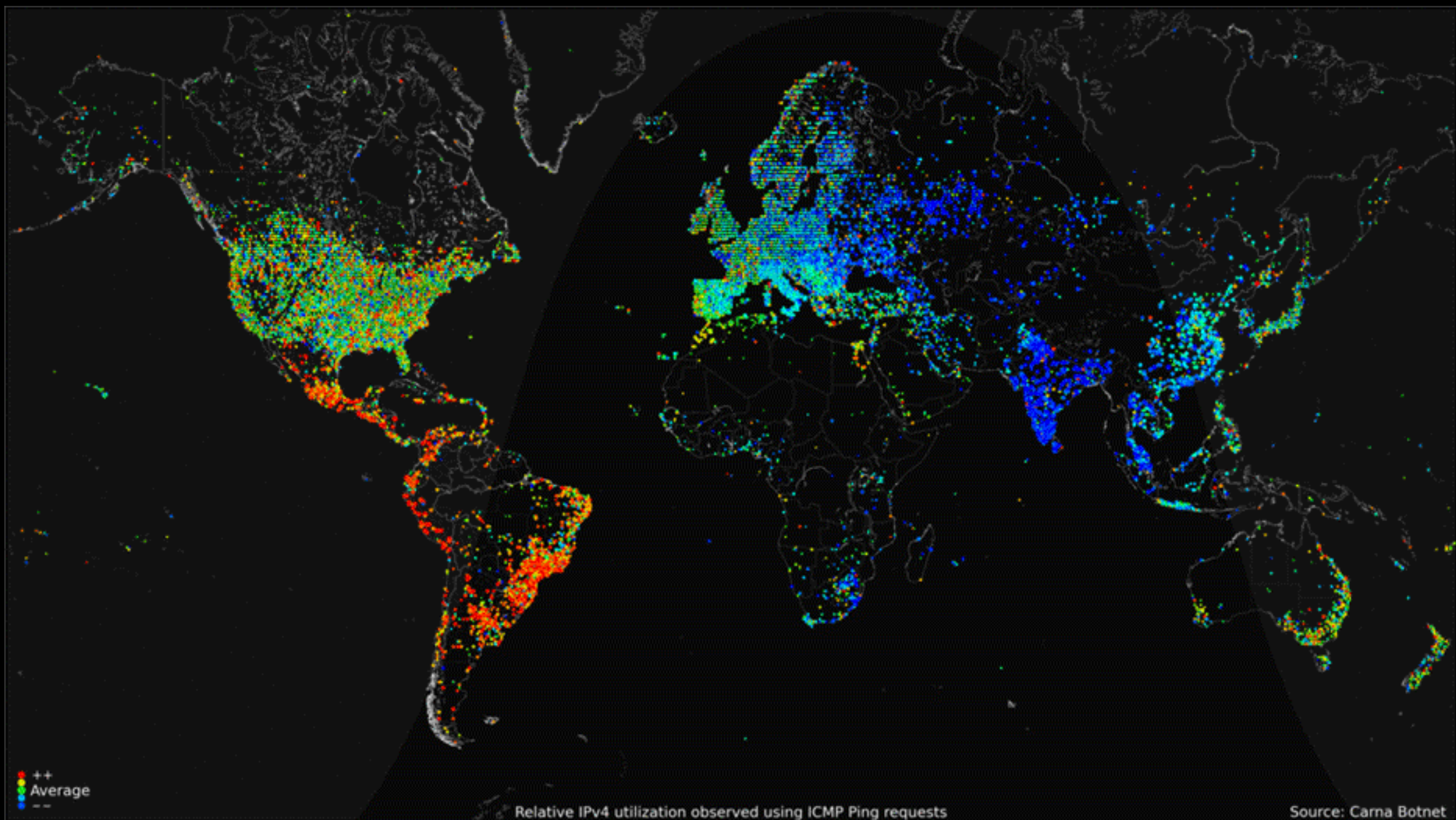


CS6421: Distributed Systems

Networks and Sockets



Prof. Tim Wood

Before we start...

Participation: 1 contributions every 3 weeks

- Ask/answer a question in class
- Post to Piazza forum
- Come of my office hours (arrange by email)

Homework 1: Message Board

- Easy/hard?

How does this work?



Me

———— ?????????????????????? ———→

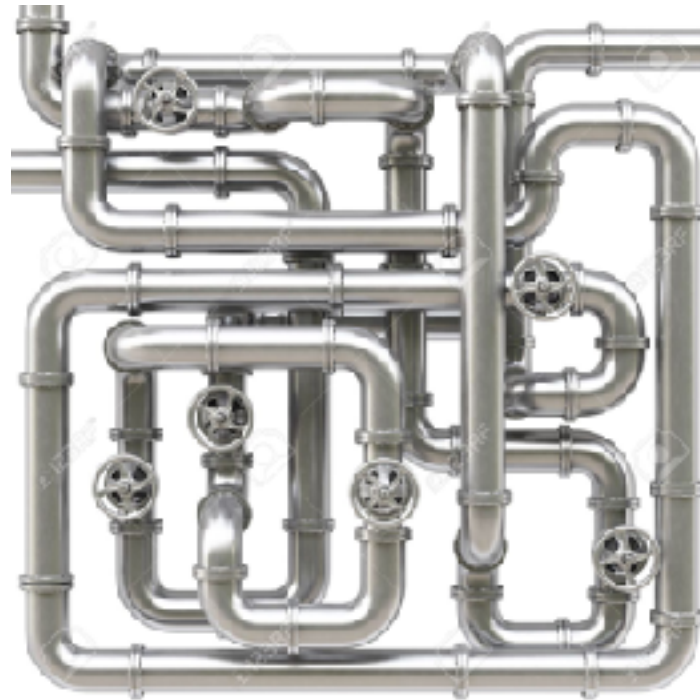


a cat

A series of tubes?



Me



a cat

*"...the Internet is not something that you just dump something on. **It's not a big truck. It's a series of tubes...**"*

-- United States Senator Ted Stevens (R-Alaska)

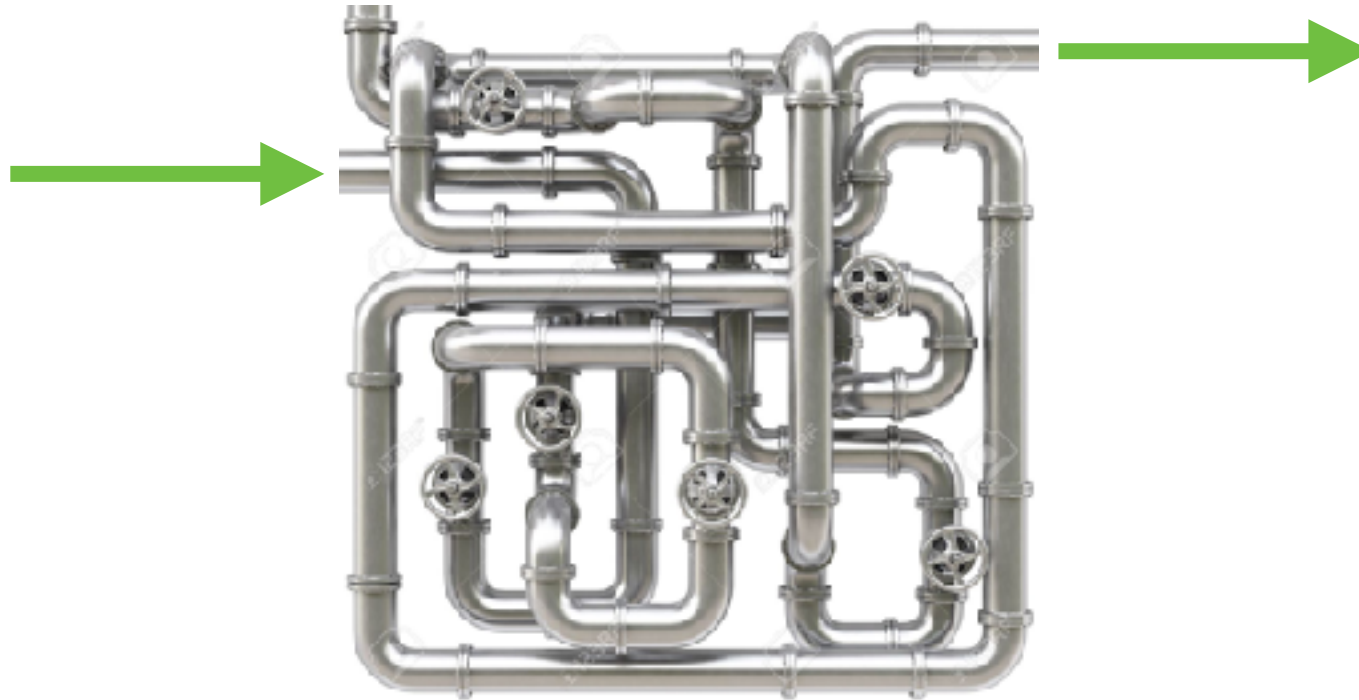
Not a truck!



Not a truck! A series of tubes!



Me



a cat

Actually, it's not really tubes either...

How will this work???

What do we need to figure out?

Find a path

- Where is our destination? How do we get there?

Traffic control

- We aren't the only ones looking for cats!

Language to communicate

- How can I tell them what I want?

Physical way of sending data

- How do you make information a tangible thing you can move?



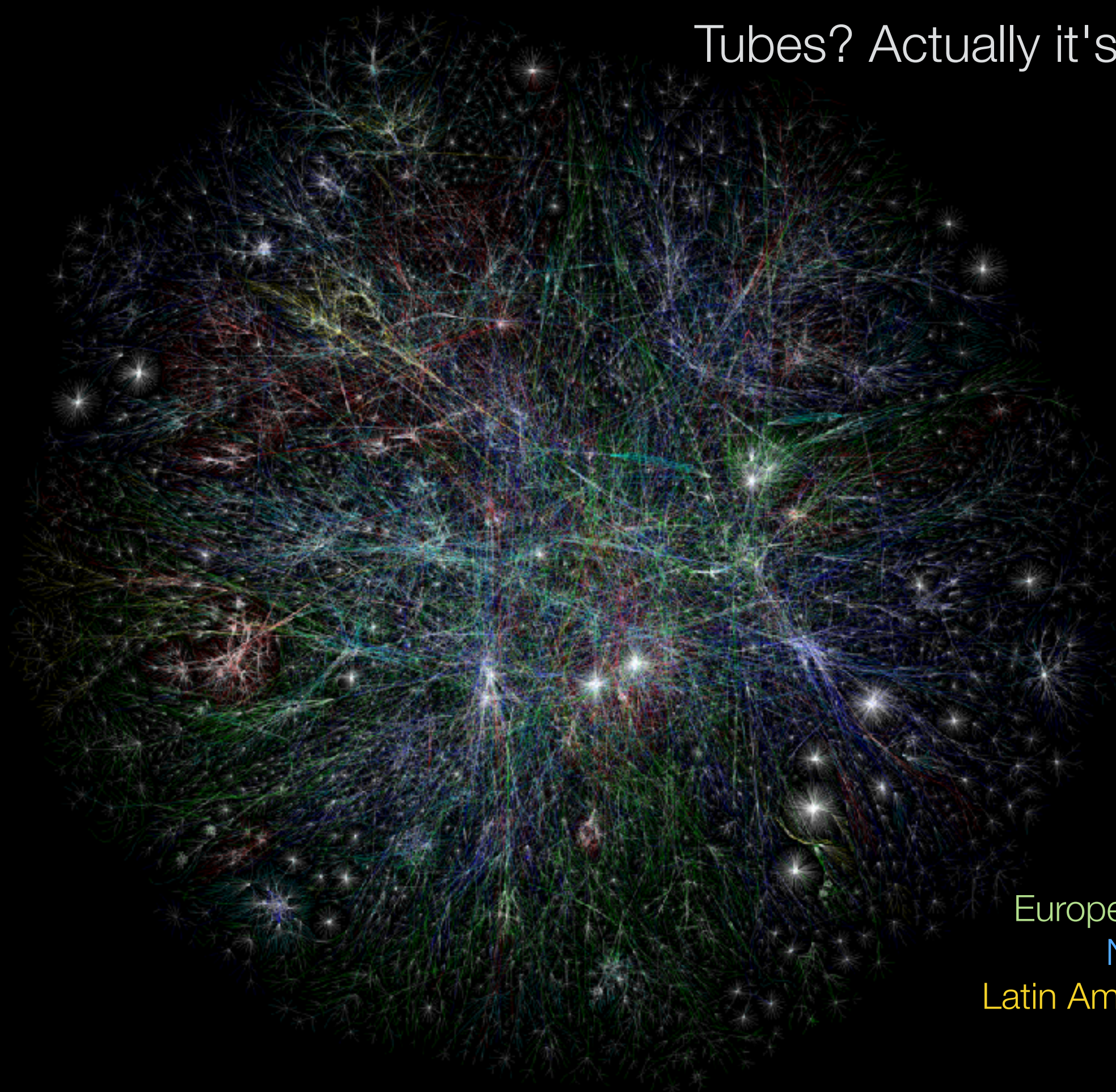
Finding our way through the tubes



Tubes? Actually it's more like a web.

and it is
really
really BIG

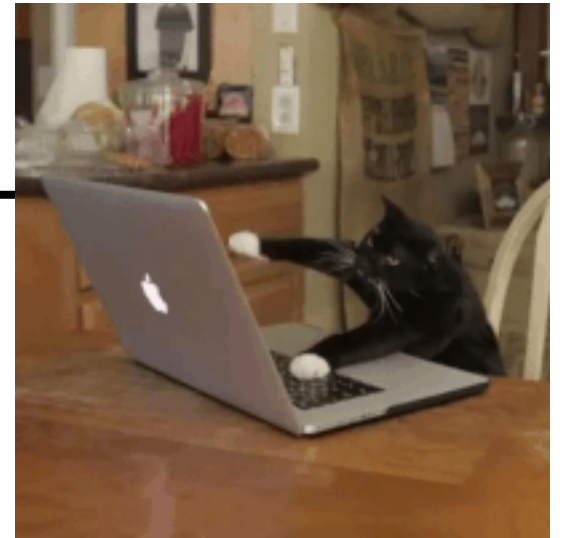
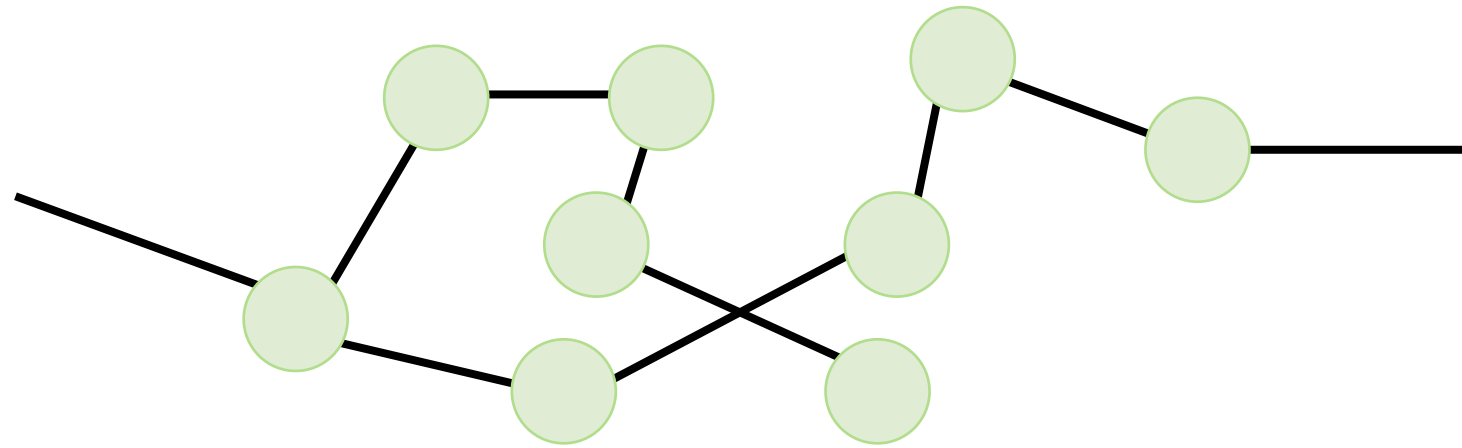
Asia Pacific
Europe/Central Asia/Africa
North America
Latin American and Caribbean
Unknown



Finding a Path



Me



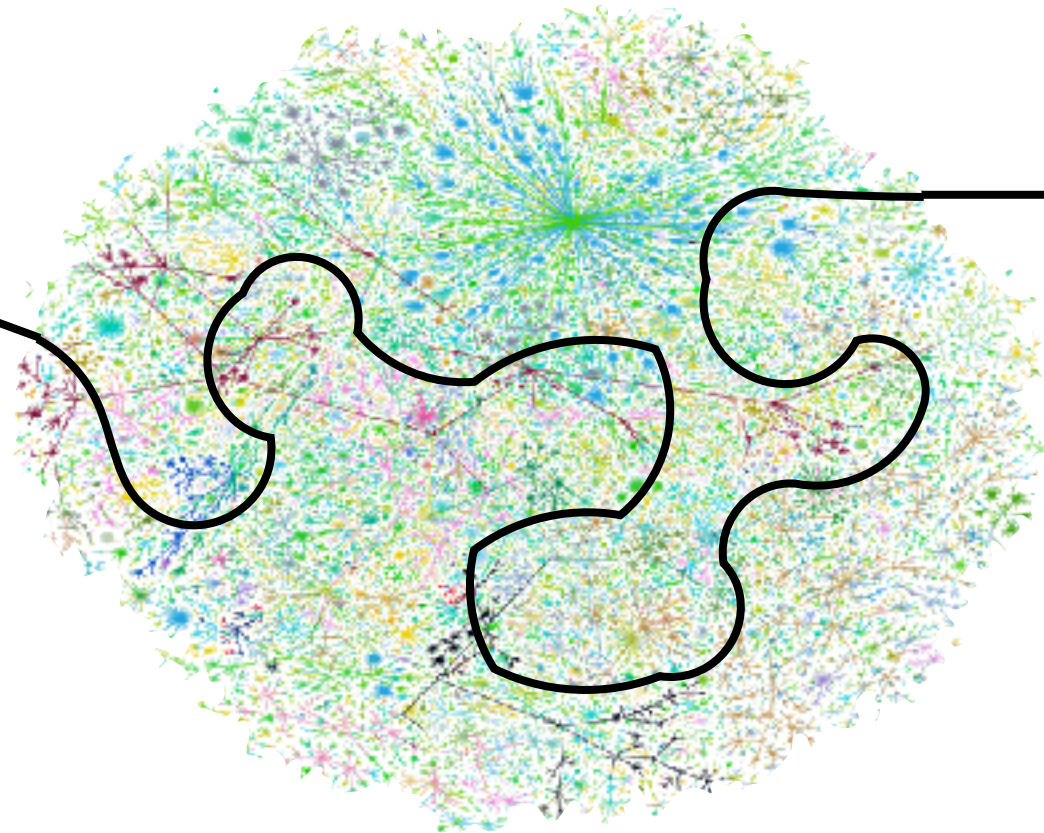
a cat

Can you solve this?

Finding a Path



Me



a cat

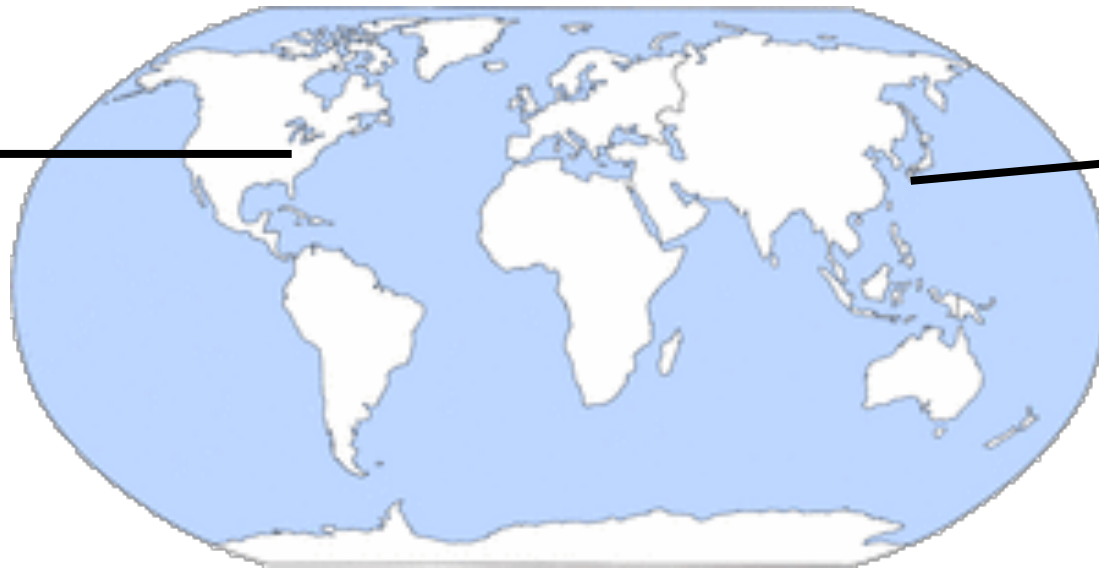
Can you solve this?

- What information do you need?
- Is this practical?
- How does it scale?

Finding a Path



Me
800 22nd St NW
Washington, DC 20052
USA



Neko
1-10-5 Akasaka
Minato-ku, Tokyo 107-8420
JAPAN

Can you solve this?

- What information do you need?
- Is this practical?
- How does it scale?

Let's try to load a web page



<http://faculty.cs.gwu.edu/timwood/simple.html>

Steps?

1. Type `http://faculty.cs.gwu.edu/timwood/simple.html` and hit enter
2. ??



Break?

until 2:05

Internet Design Principles

Addresses should have **meaning**

Work should be **distributed**

Names

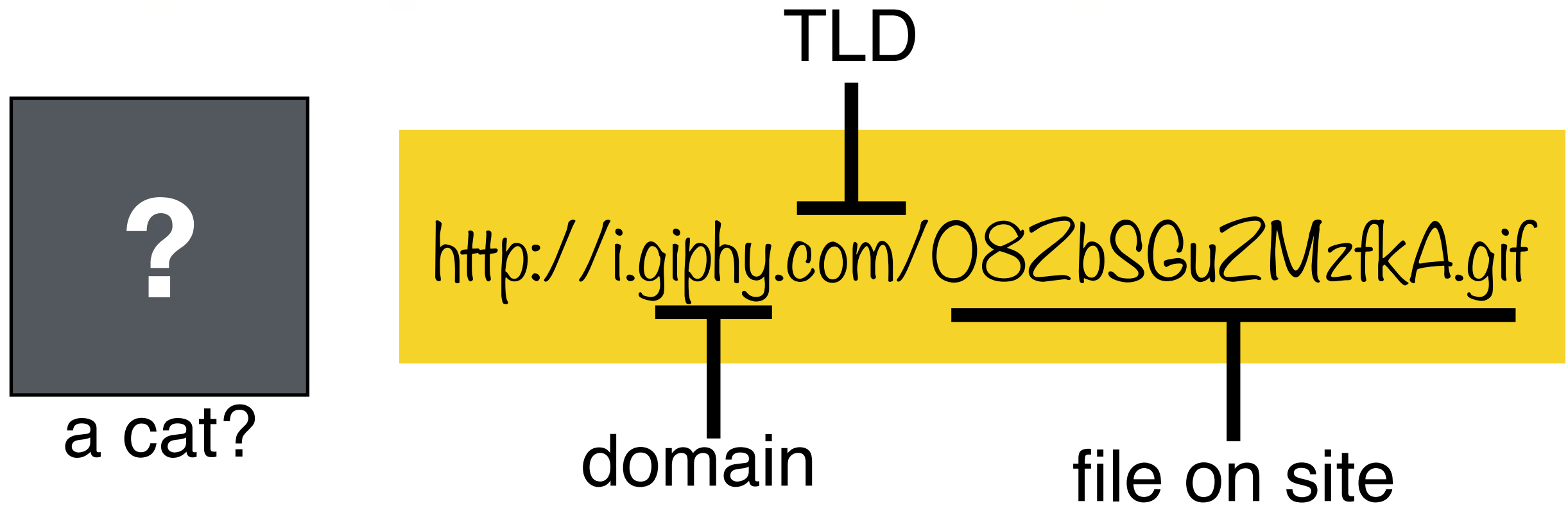
What does this URL tell me?



a cat?

<http://i.giphy.com/O8ZbSGuZMzfkA.gif>

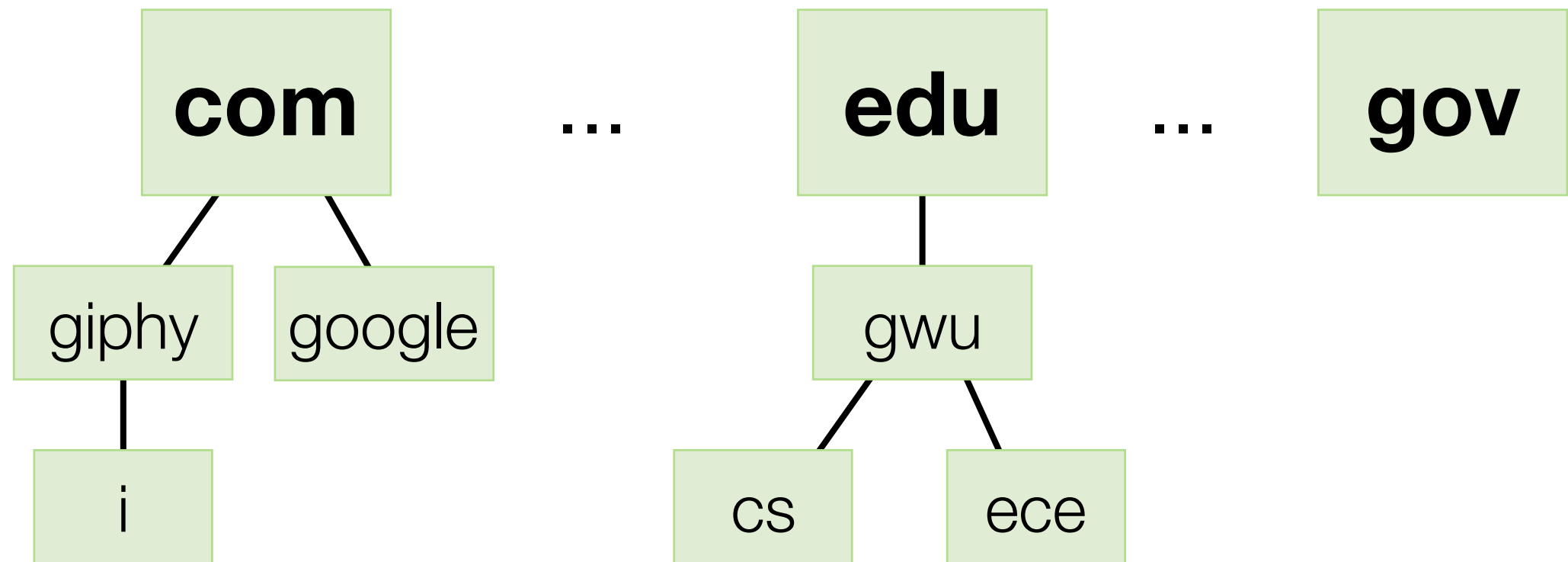
Names



What does this URL tell me?

- O8ZbSGuZMzfkA.gif = file
- i = subdomain
- giphy = domain
- com = top level domain

DNS: Names to Addresses



Converts human readable name to machine address



What good are addresses?

Me



? a cat?

161.253.78.16

104.16.12.250



Me

800 22nd St NW
Washington, DC 20052
USA

Neko

1-10-5 Akasaka
Minato-ku, Tokyo 107-8420
JAPAN

What good are addresses?

A hierarchy of information

? a cat?

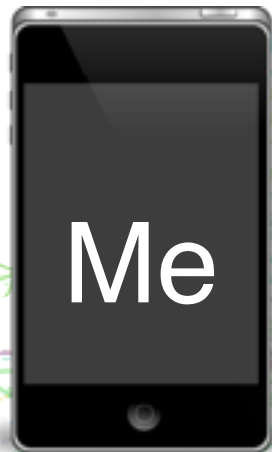
104.16.12.250

A hierarchy of information
Relative location information
Well defined names



Neko
1-10-5 Akasaka
Minato-ku, Tokyo 107-8420
JAPAN

Finding a Path



161...

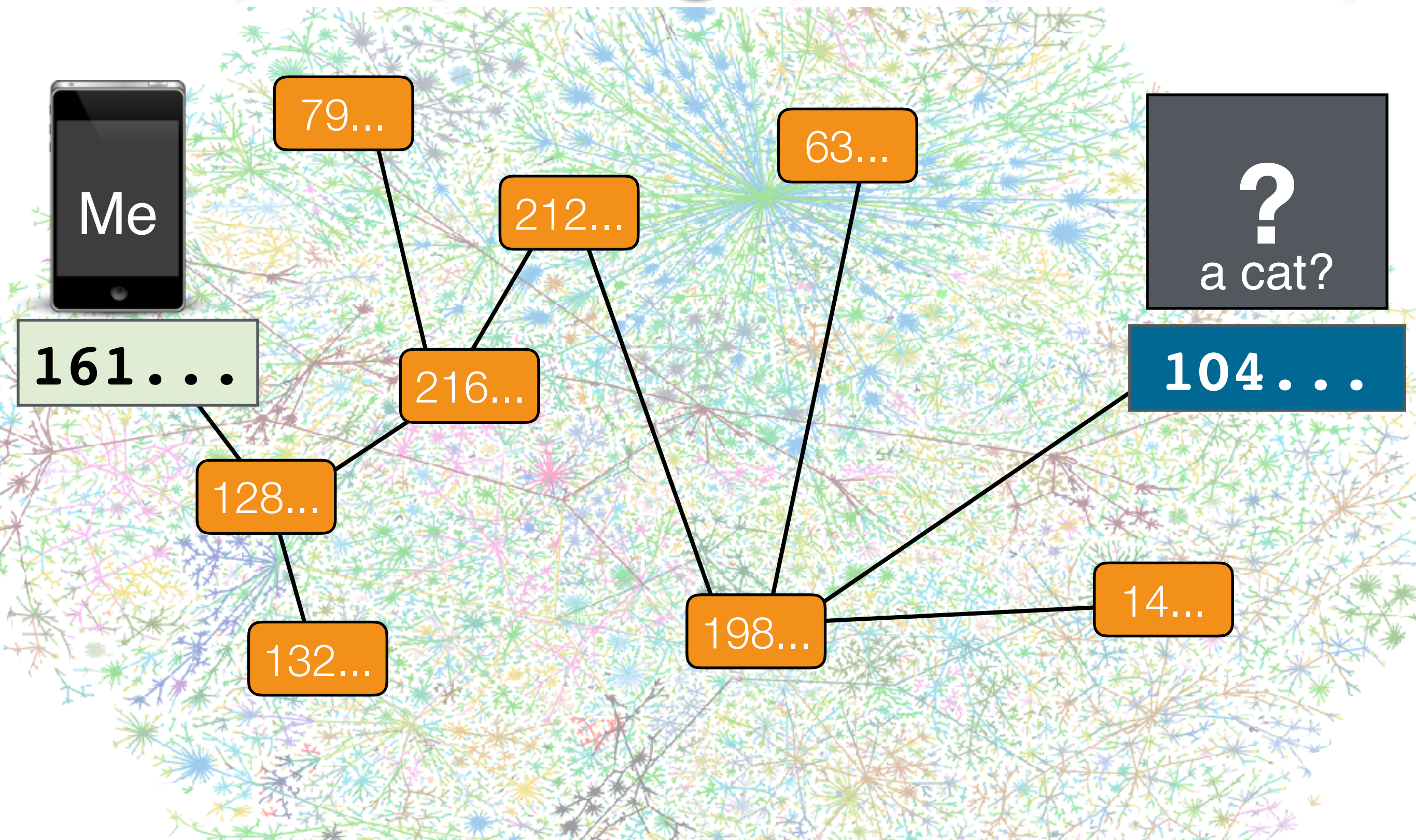
Can you solve this?

- What information do you need?
- Is this practical?
- How does it scale?

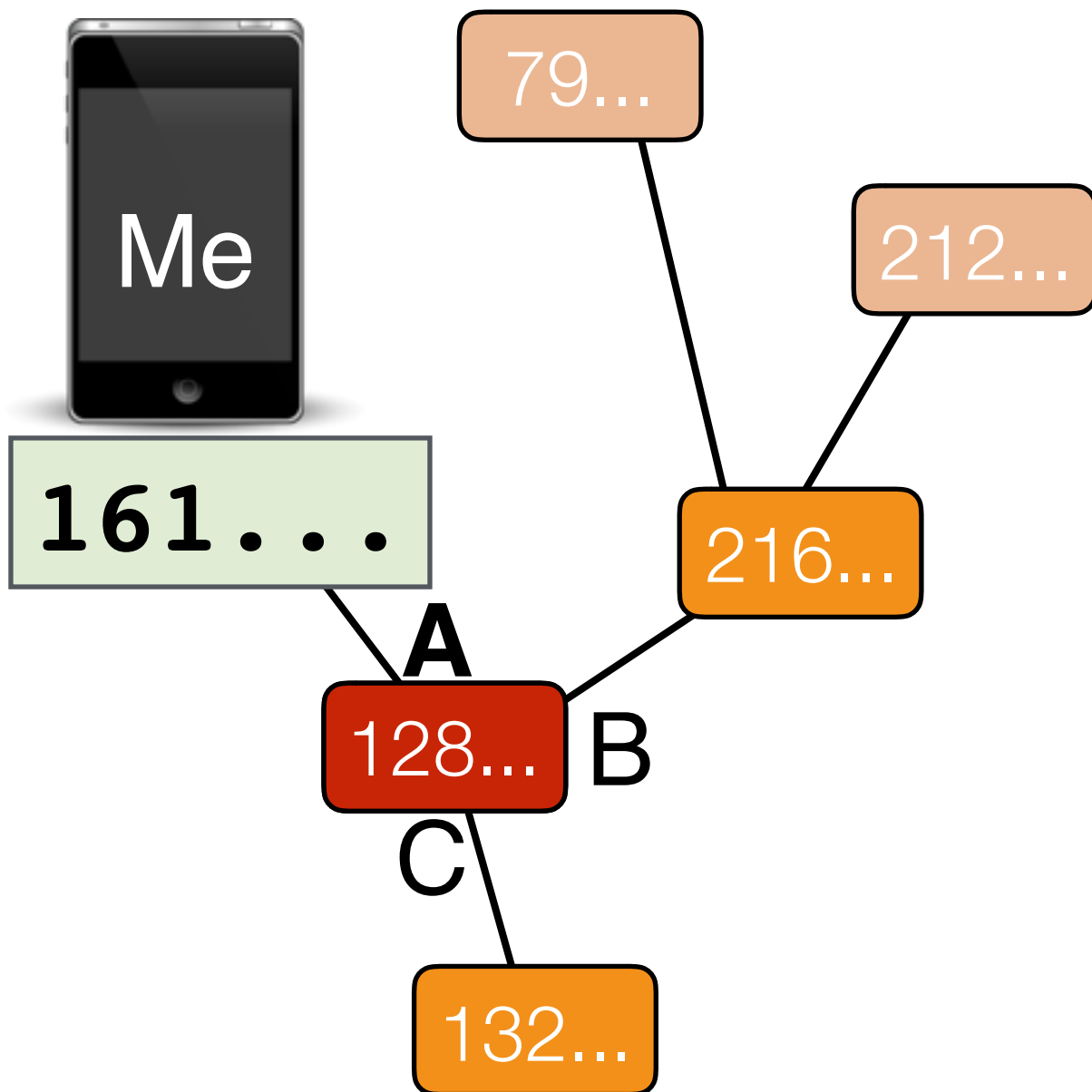
?
a cat?

104...

Finding a Path



Routers Know About...



Directly connected networks

- A-> 161
- B-> 216
- C-> 132

Next hop for some IP ranges

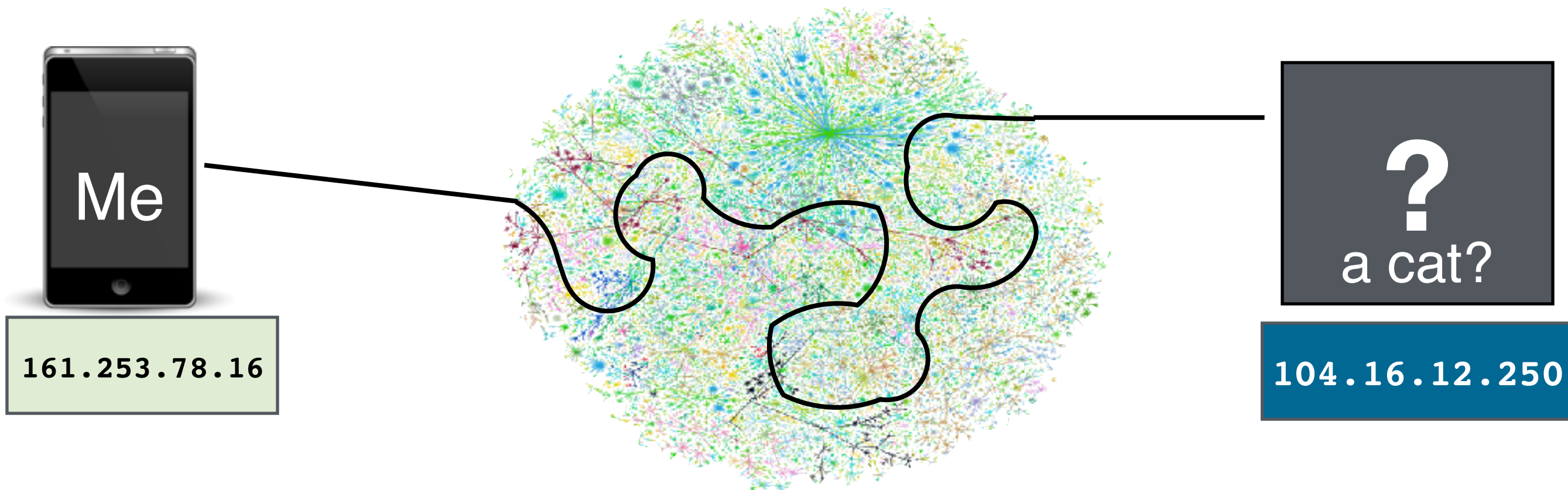
- B-> 216, 79, 212, ...

Default gateway

- Typically a more central router

Exchange information with adjacent nodes to discover more routes

Found a Path!

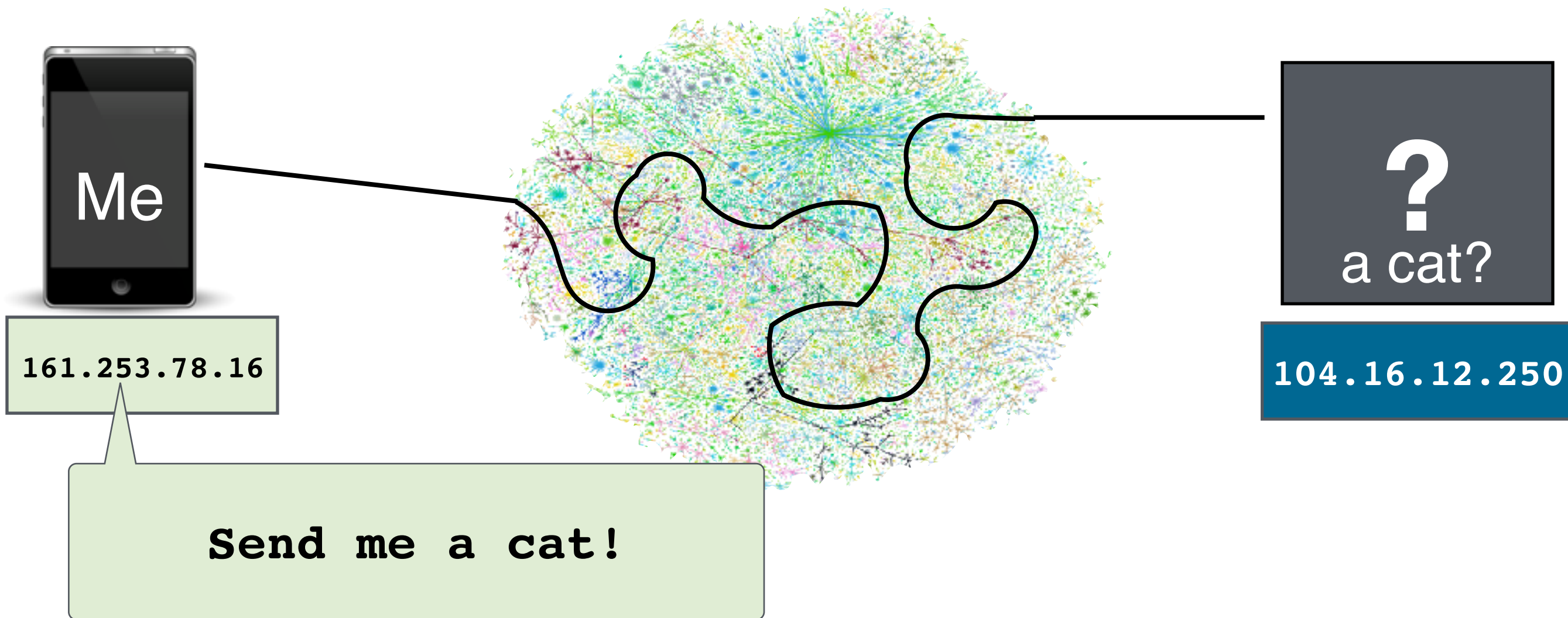


Use IP address hierarchy to simplify path finding
Distributed routing tables, no centralized knowledge

Communicating through the tubes

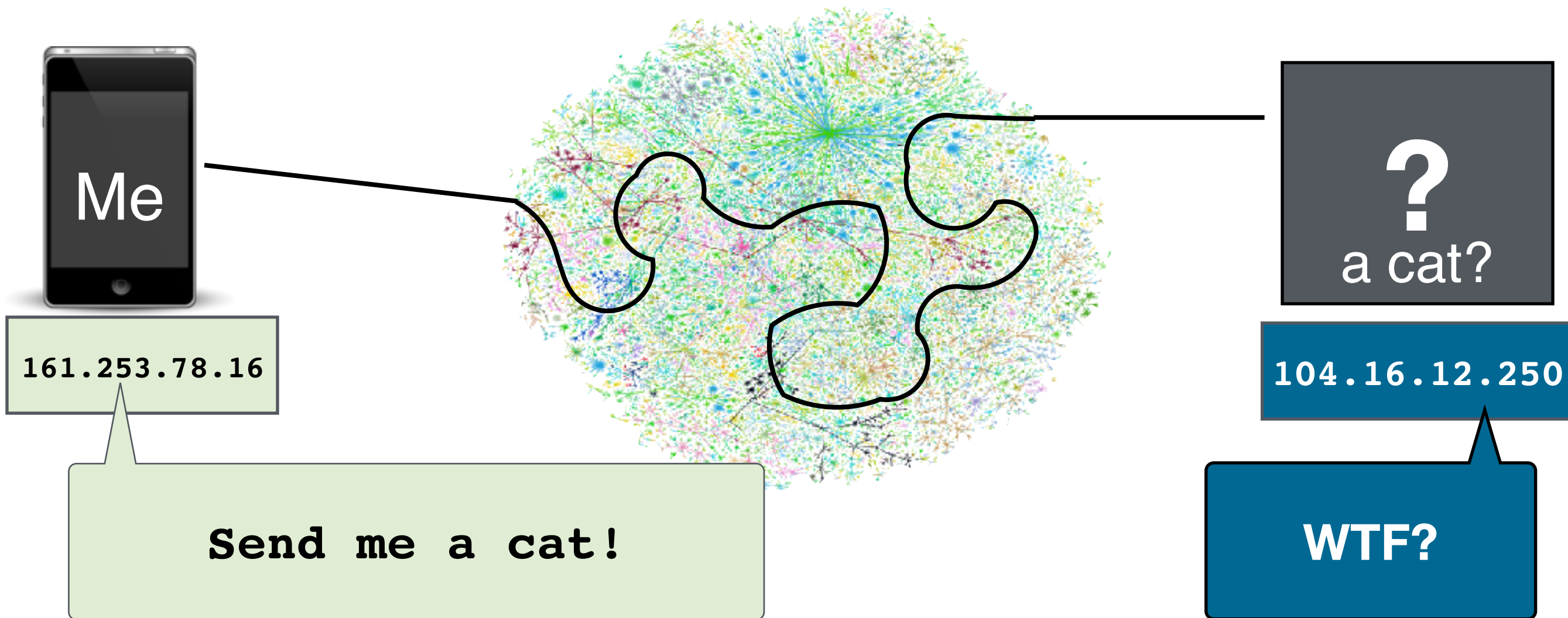


Asking for data



Use IP address hierarchy to simplify path finding
Distributed routing tables, no centralized knowledge
How do we get our data through the tubes???

Asking for data



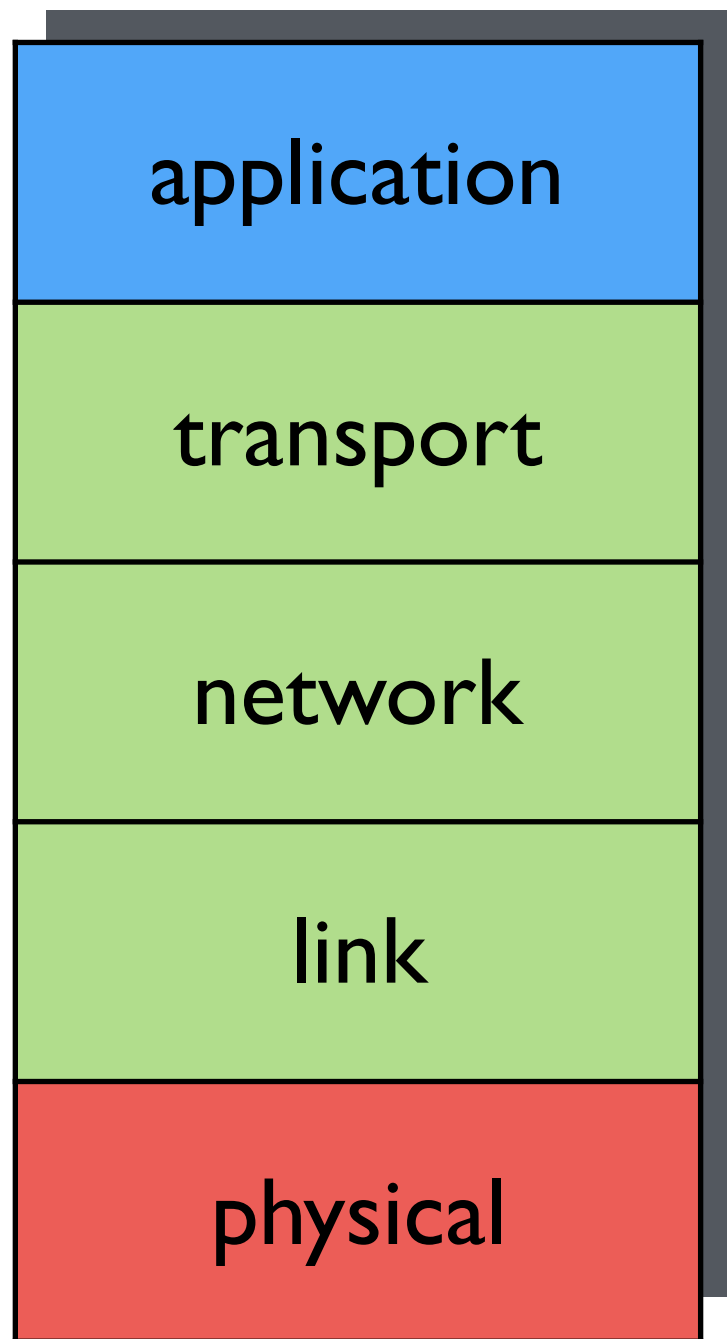
Use IP address hierarchy to simplify path finding
Distributed routing tables, no centralized knowledge
How do we get our data through the tubes???

Internet Design Principles

Protocols define how to
communicate

Protocols can be **layered** for
complexity

Protocol Layers



application:

- FTP, SMTP, HTTP

transport: data transfer

- TCP, UDP

network: finding routes

- IP, routing protocols

link: adjacent nodes

- Ethernet, 802.111 (WiFi), PPP

physical:

- bits on the wire or in the air

How to speak _____?

Notice something about these addresses?

www.google.com

ftp.cs.gwu.edu

smtp.gmail.com

How to speak *website*?

HTTP: HyperText Transfer Protocol



161.253.78.16

```
GET /timwood/simple.html HTTP/1.1
Host: faculty.cs.gwu.edu
(blank line)
```



185.199.109.153

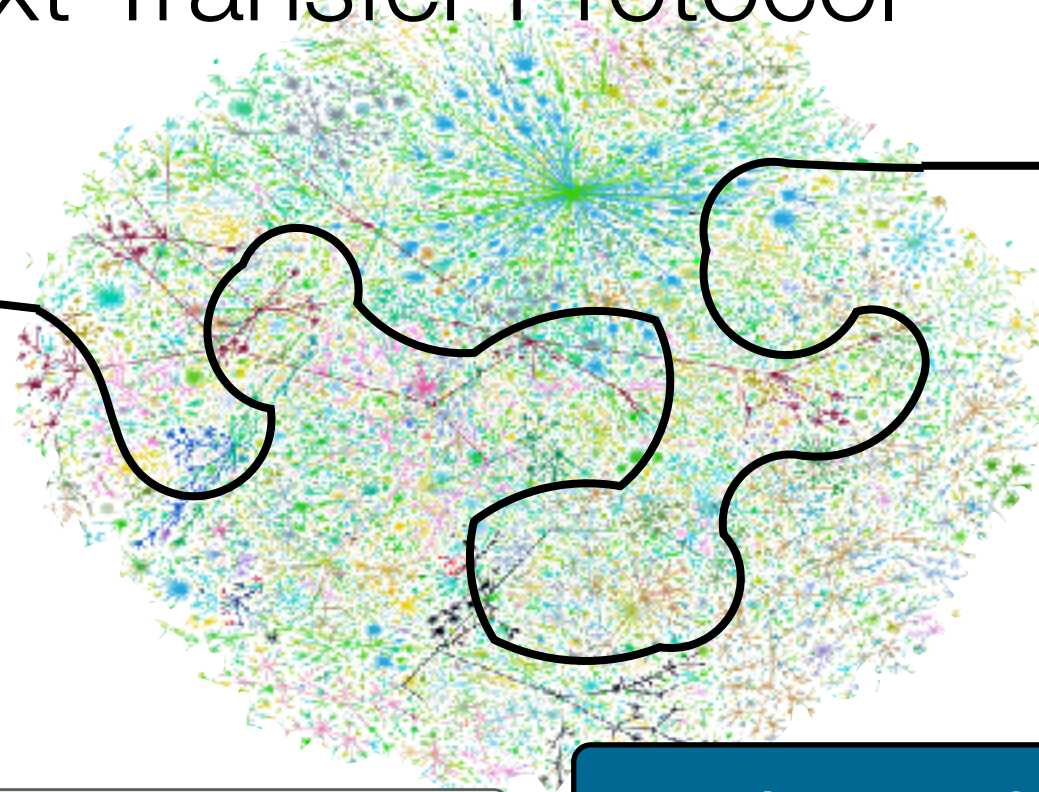
How to speak *website*?

HTTP: HyperText Transfer Protocol



161.253.78.16

```
GET /timwood/simple.html HTTP/1.1
Host: faculty.cs.gwu.edu
(blank line)
```



185.199.109.153

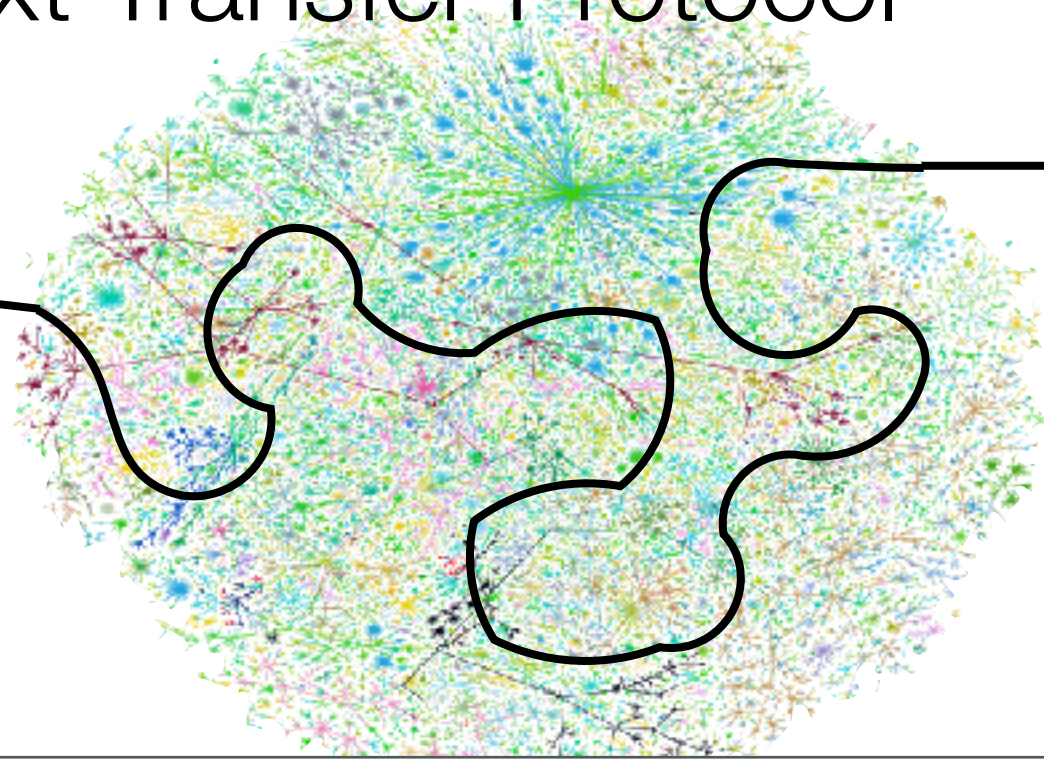
```
HTTP/1.1 200 OK
Server: GitHub.com
Content-Type: text/html; charset=utf-8
Last-Modified: Thu, 06 Sep 2018 17:57:20 GMT
ETag: "5b916a80-b6"
Access-Control-Allow-Origin: *
Expires: Thu, 06 Sep 2018 18:09:00 GMT
...
```

How to speak *website*?

HTTP: HyperText Transfer Protocol



161.253.78.16



i.giphy.com

```
GET /O8ZbSGuZMzfkA.gif HTTP/1.1
Host: i.giphy.com
(blank line)
```

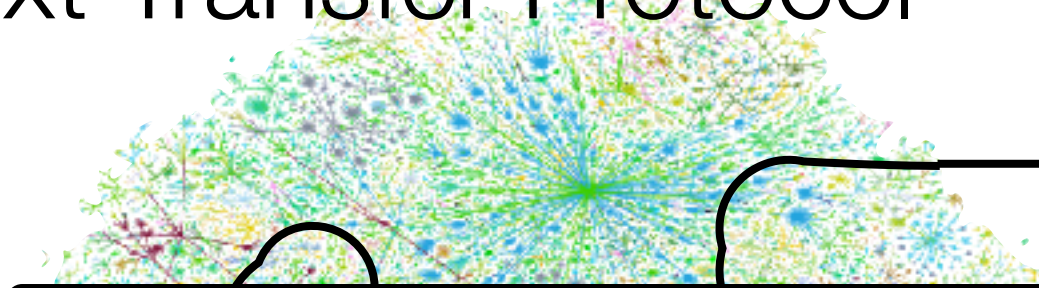
<http://i.giphy.com/O8ZbSGuZMzfkA.gif>

How to speak *website*?

HTTP: HyperText Transfer Protocol



161.253.78.16



104.16.12.250



Steps?

1. Type URL and hit enter
2. DNS lookup: hostname->IP
3. ARP lookup: IP->MAC address
4. Socket setup
5. Send call moves data from browser to OS
6. TCP Handshake
7. Routing lookups along path
8. HTTP request issued
9. Parse HTTP response: HTML->DOM tree
10. Make additional requests for other resources



Problems?

What are possible points of failure?

- ???

What are possible performance bottlenecks?

- ???

Sockets and Protocols

Network Layers

Network Interface Card (NIC)

- Reads “bytes on wire”

Driver

- Loads network data into memory

Internet Protocol (IP)

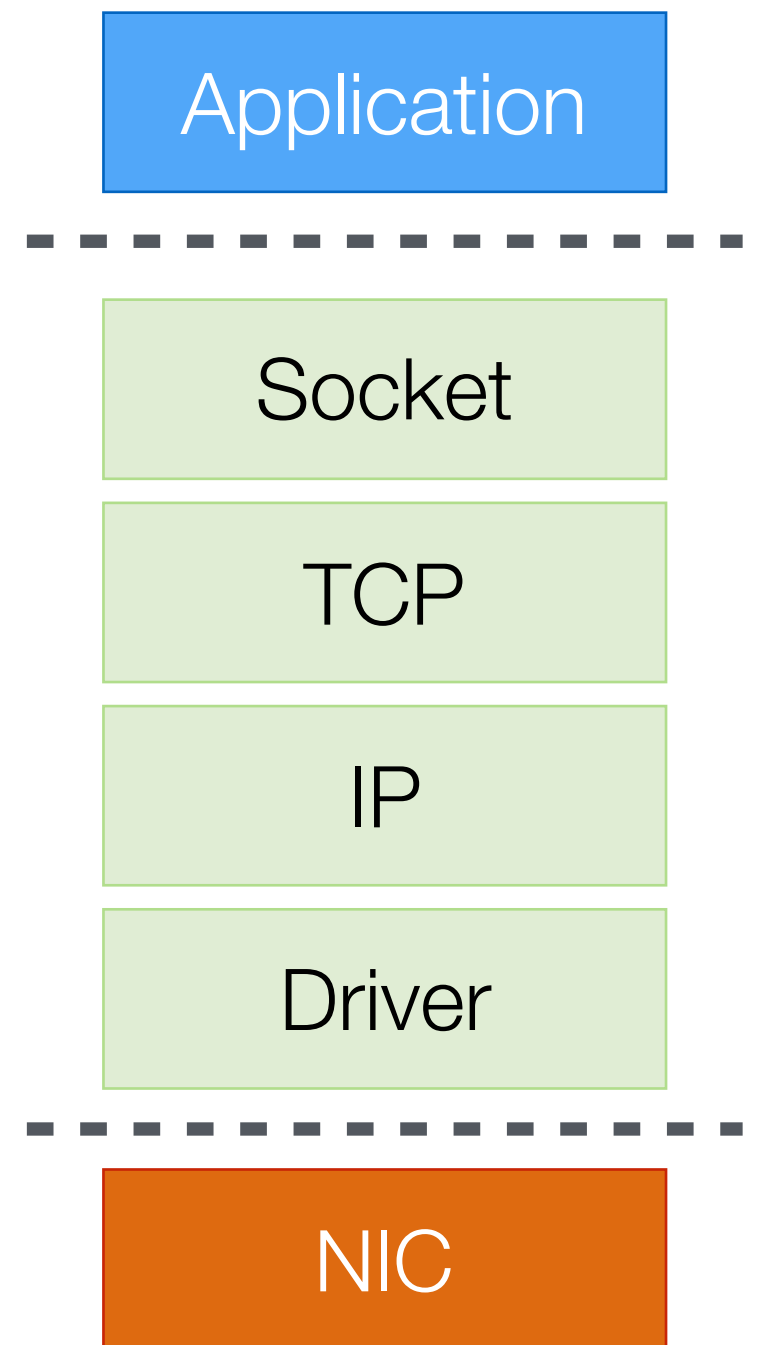
- Handles addressing and routing

Transmission Control Protocol (TCP)

- Ensures reliable, ordered transmission of packets and manages congestion

Socket

- Provides interface between OS and App



What can a socket do?

What can a socket do?

Send

Receive

What can a socket do?

Send

Receive

Blocking

Non-blocking

Protocols

How are protocols developed?

- TCP/IP
- HTTP
- Message board / chat server

What fundamentally is a protocol?

Protocol Design

Algorithms

- A sequence of actions performed by each communicating party
- *Chat server?*
- *TCP?*

Data structures

- Message formats used by each communicating party
- *Chat server?*
- *TCP?*

Example: TCP/IP

Data Structures:

- TCP Header
- TCP state machine

Algorithms:

- 3-step handshake
- Congestion control
- Reliable transport
- Slow start

source port #					dest port #				
sequence number									
acknowledgement number									
head len	not used	U	A	P	R	S	F	receive window	
checksum					Urg data pointer				
options (variable length)									
application data (variable length)									

TCP format

Example: GWcached

Stores data values in memory and allows them to be looked up by their key

Algorithm

- Command: Get, Search, Delete, Add, Modify
- Policy: eviction, request ordering/consistency, security, memory management, fault detection

Data Structure

- Hash Table (key, value) —> What is a “value”? What is a “key”?

- - checksum -> hash(full message contents)
 - UserID -> GWNetID
 - Password -> hash(pass)
 - Operation - > lower case string (get, search)
 - key -> string with no newlines
 - length -> 32 bit int in ASCII
 - data -> can be binary or ascii data at most <length> bytes