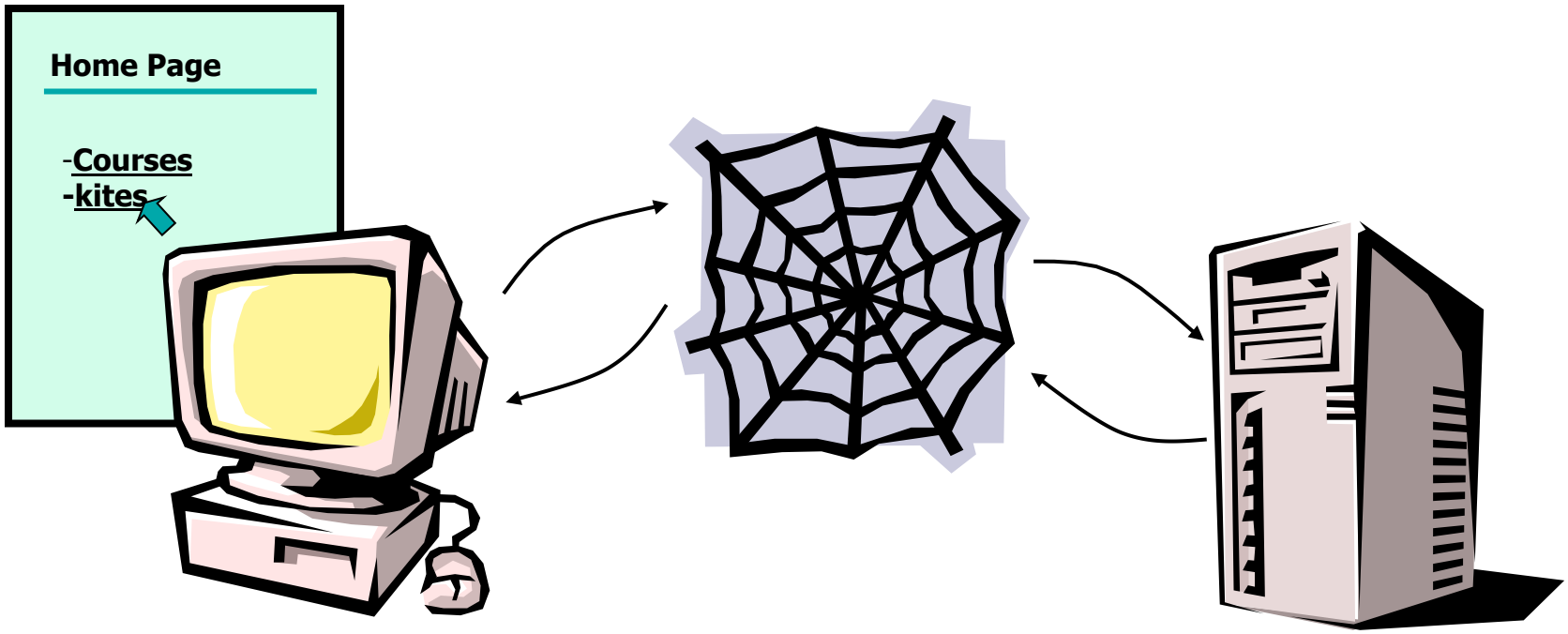
SIMPLE WEB REQUEST

Simple Web Request

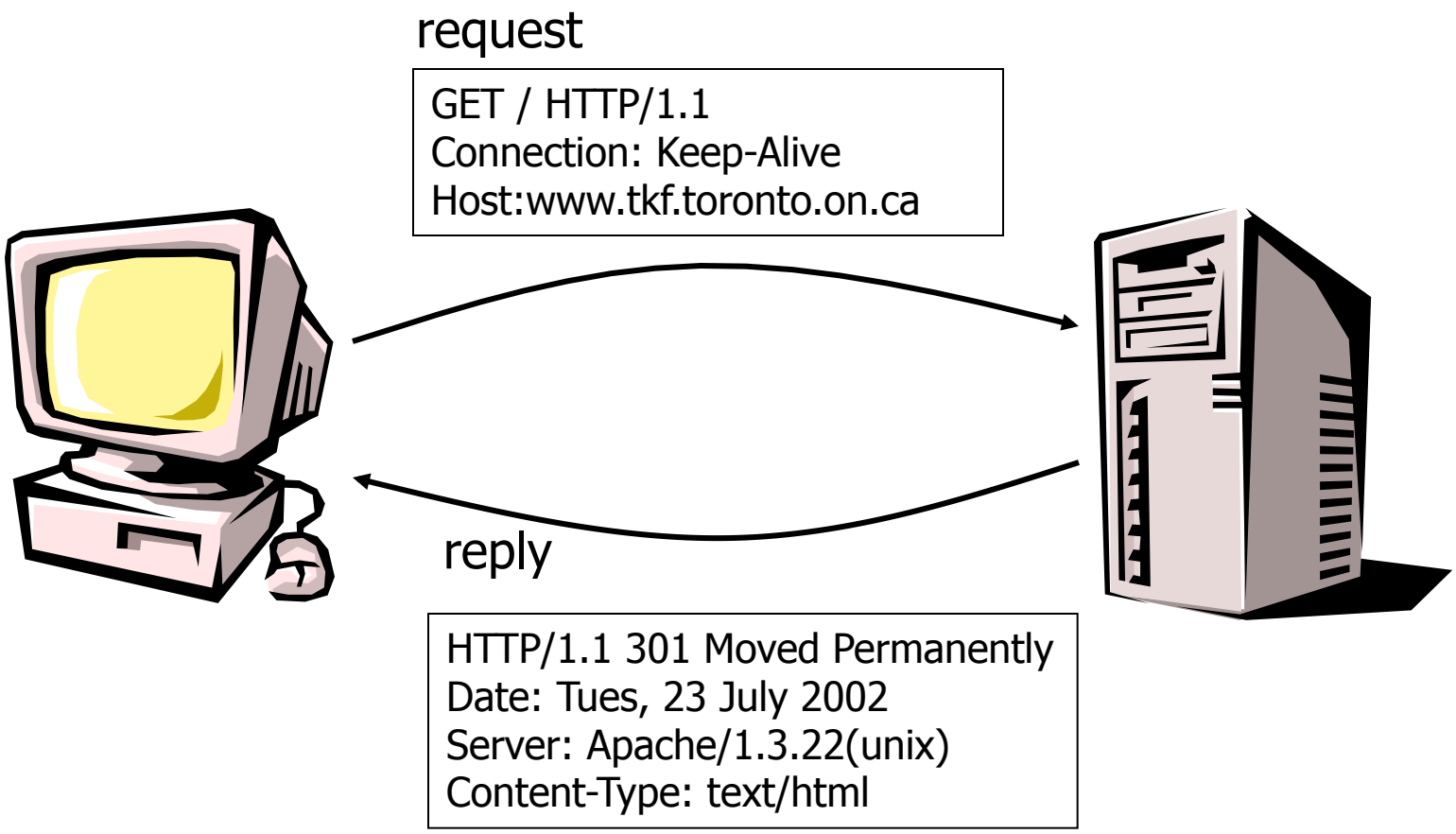


Thanks to Karen Reid for material in these slides

The request

- How do we tell the server what we want?
- How do we even find the server?
- How do the server and browser talk to each other?

HTTP Request

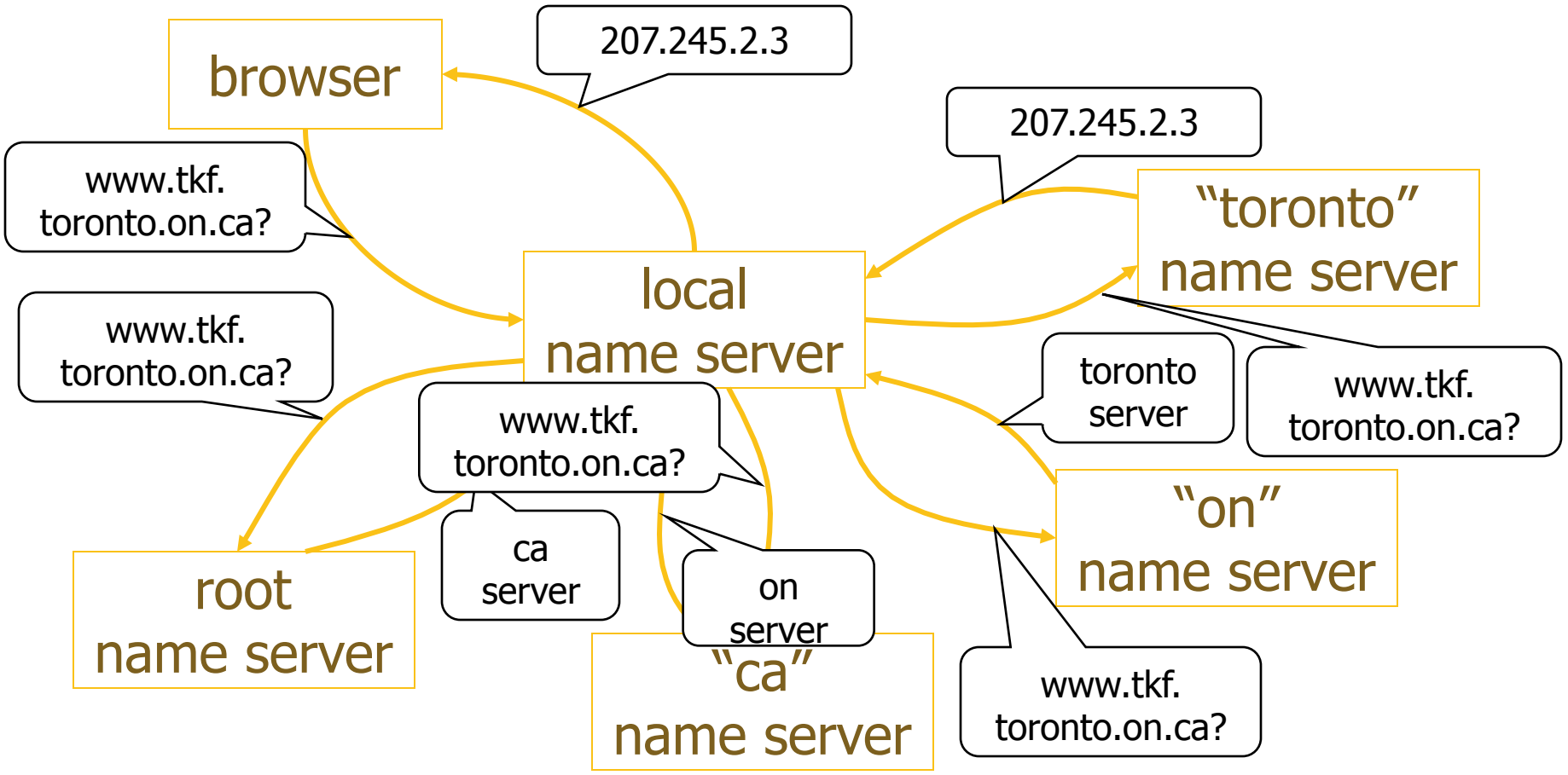


How do we find the server?

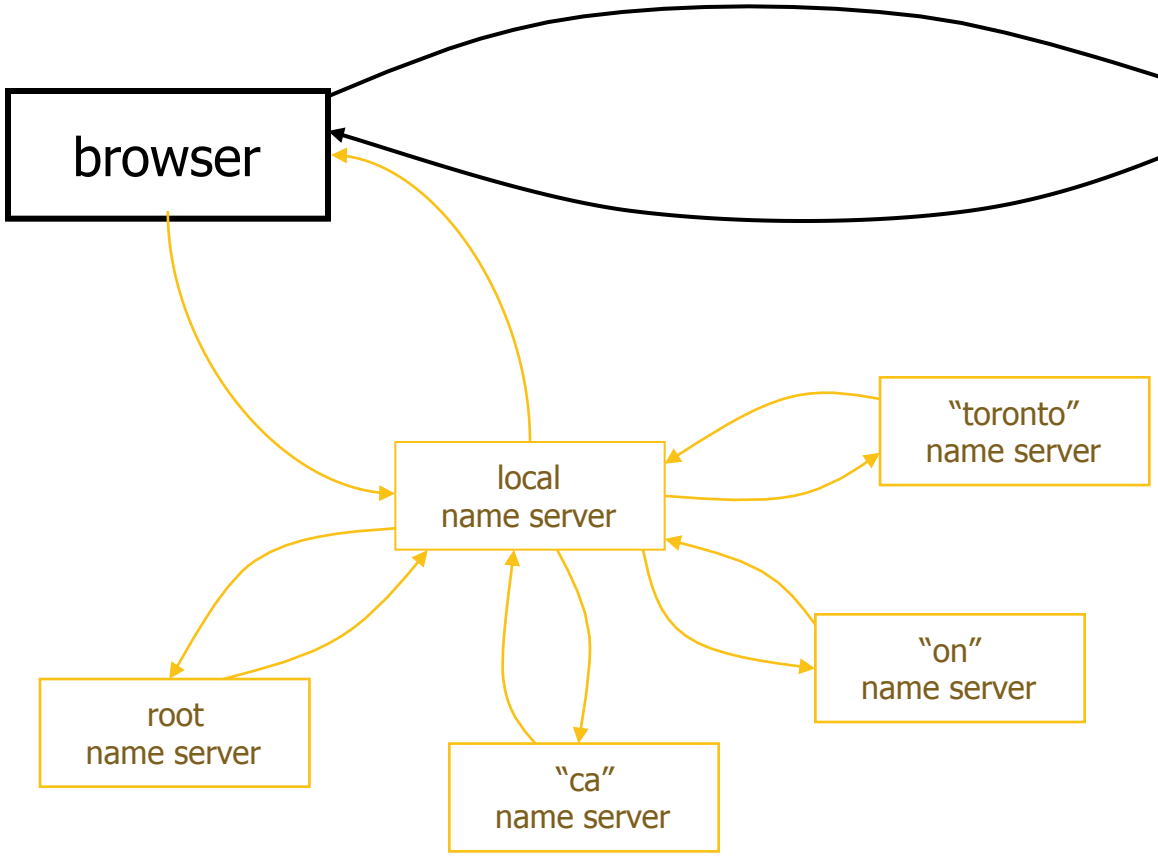
- Every computer on the Internet has an Internet address.
- Called an IP address (Internet Protocol)
- An IP address is 4 numbers separated by dots.

www.tkf.toronto.on.ca = 207.245.2.3

Domain Name Servers



This is getting complicated!



Number of messages?

10-12

Actually, it's worse than that

- The web page for www.tkf.toronto.on.ca doesn't really live at 207.245.2.3

request

```
GET /  HTTP/1.1
Connection: Keep-Alive
Host:www.tkf.toronto.on.ca
```

reply

```
HTTP/1.1 301 Moved Permanently
Date: Tues, 23 July 2002
Server: Apache/1.3.22(unix)
Location: http://www.kites.org
Content-Type: text/html
```

- So we need to go look up www.kites.org

Now what?

- Okay, we have the address.
- What do we do with it?
- Let's look at how two computers communicate.
- HTTP is a high-level protocol
- HTTP is specific to the web.
- Computers communicate for other reasons.

Protocols

- Computers use several layers of general protocols to communicate.
- To understand why these layers are important, think about how a company sends you an invoice for a purchase.

Protocols

Invoice:
Customer: Karen Reid
Order No: 5379

Qty:		Unit Price	Total
1	Athalon	219.00	219.00
2	128 MB	149.95	299.90
		Subtotal	518.90
		Tax	77.84
		TOTAL	596.74

Karen Reid Feb 18, 2001

Payable to: CPUS are us $596.74
Five hundred ninety six 74/100

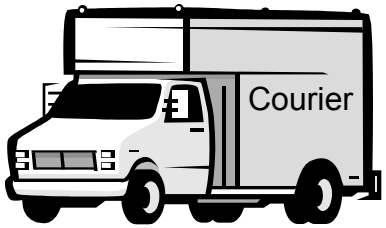
_____

CPUS are us

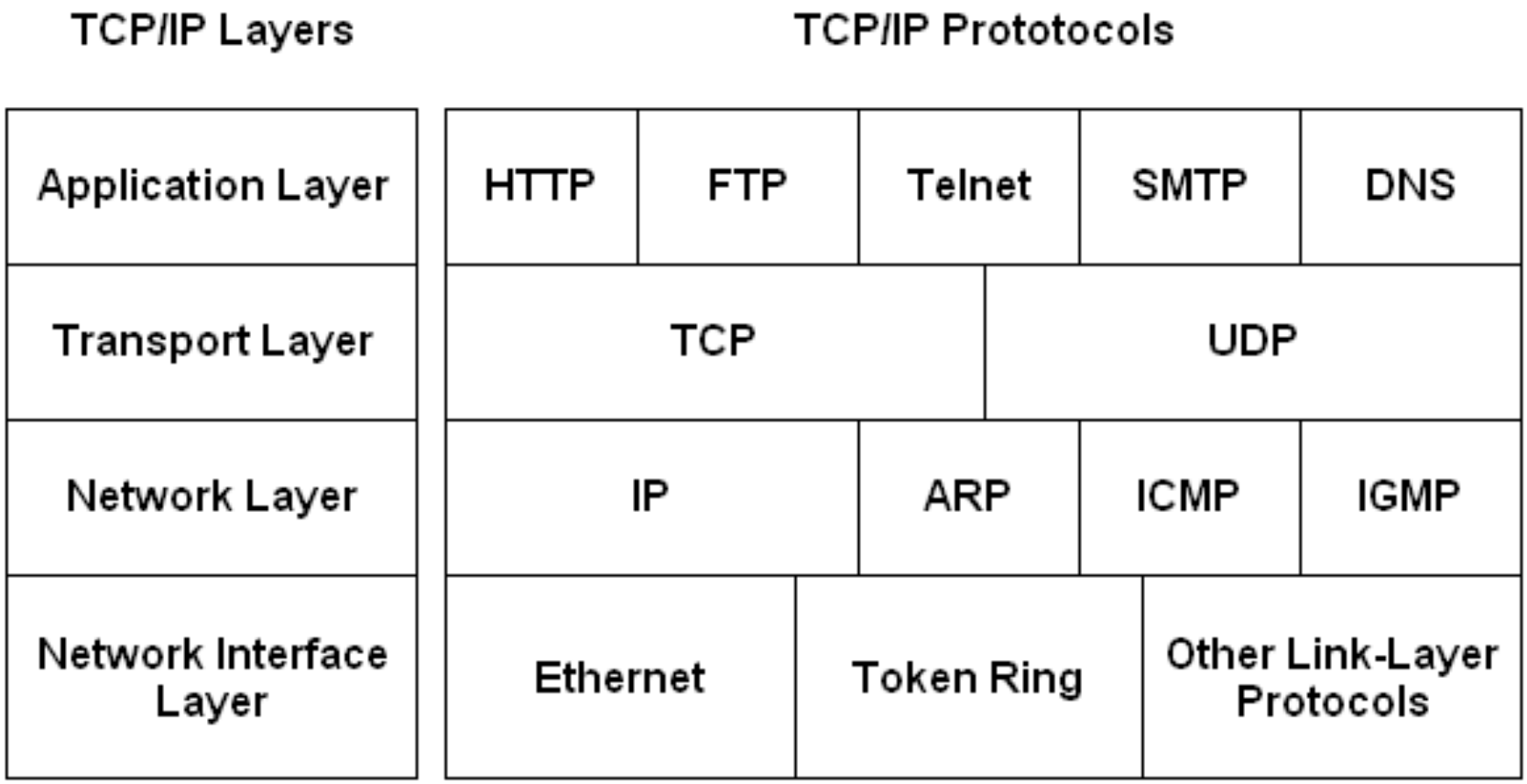
Karen Reid
Dept. of Computer Science
University of Toronto

Karen Reid

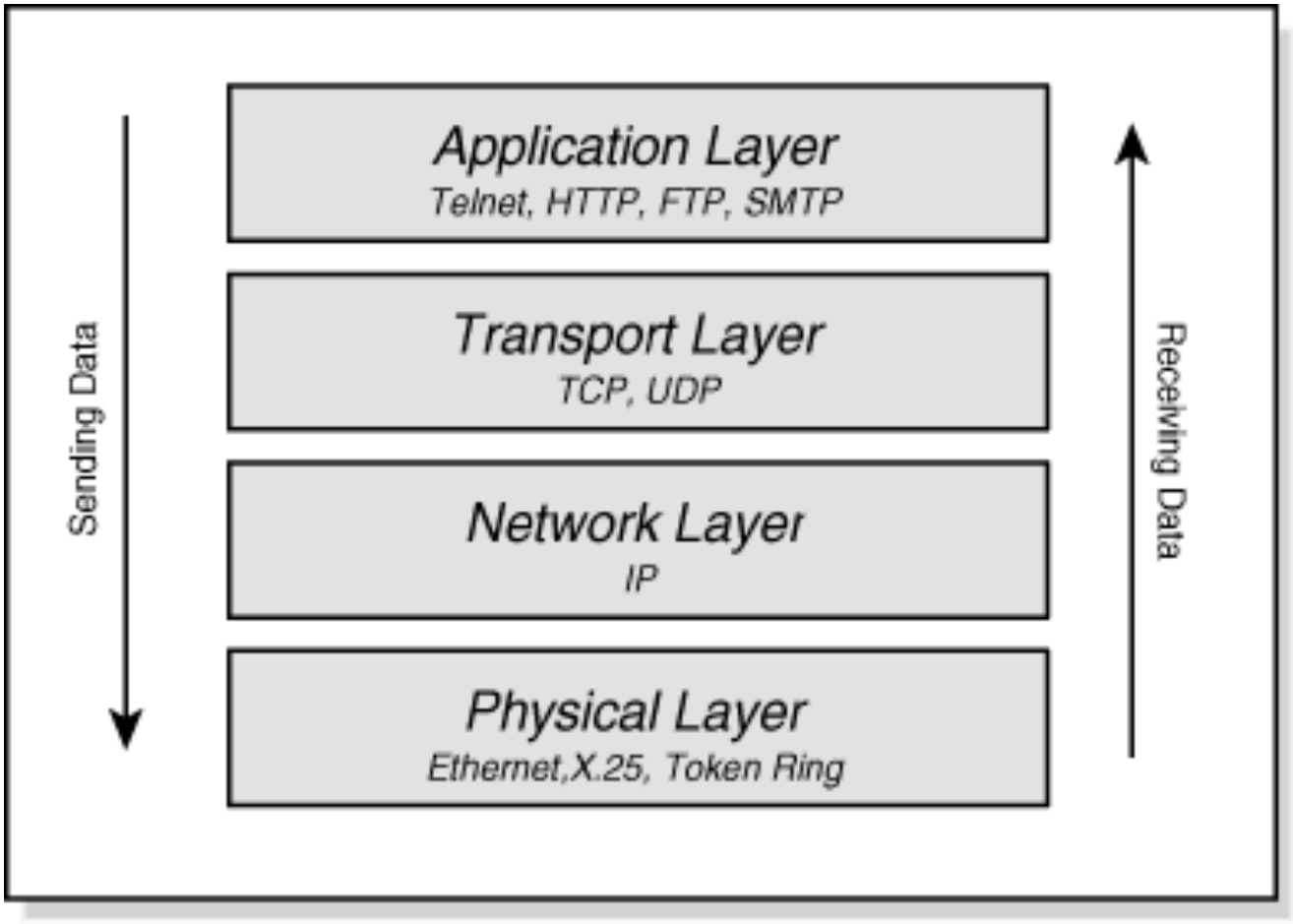
CPUS are us
0 College Street
Toronto Ontario M5S 3G4



TCP/IP: Protocol Framework

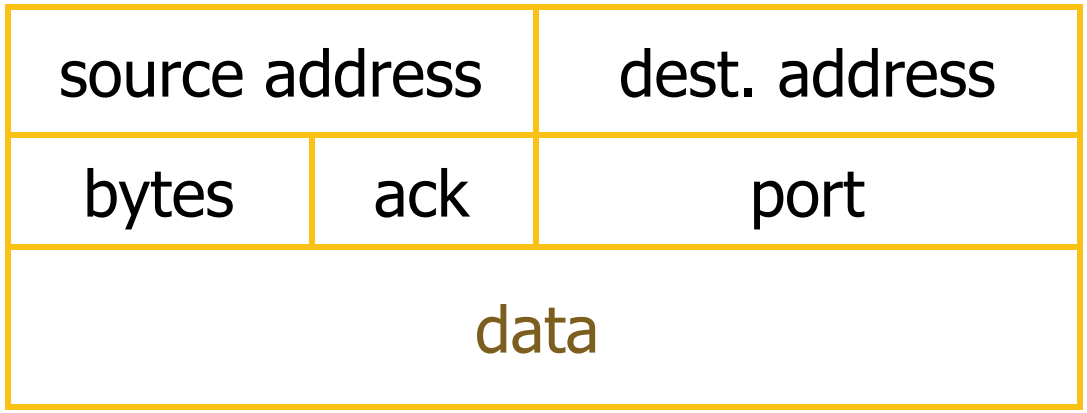


TCP/IP: Sending/Receiving Data



TCP/IP

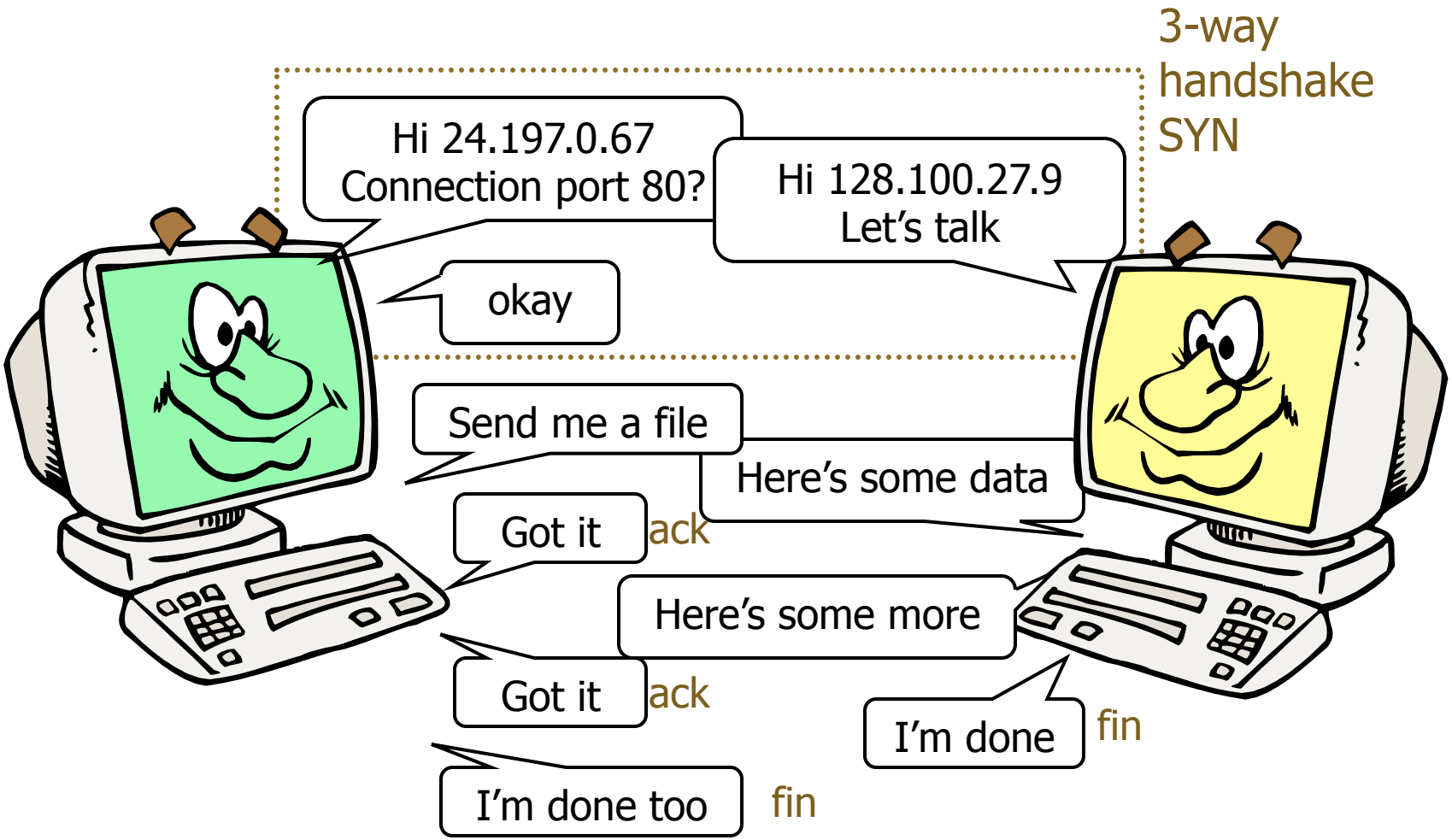
- Transmission Control Protocol.
- Tells us how to package up the data.



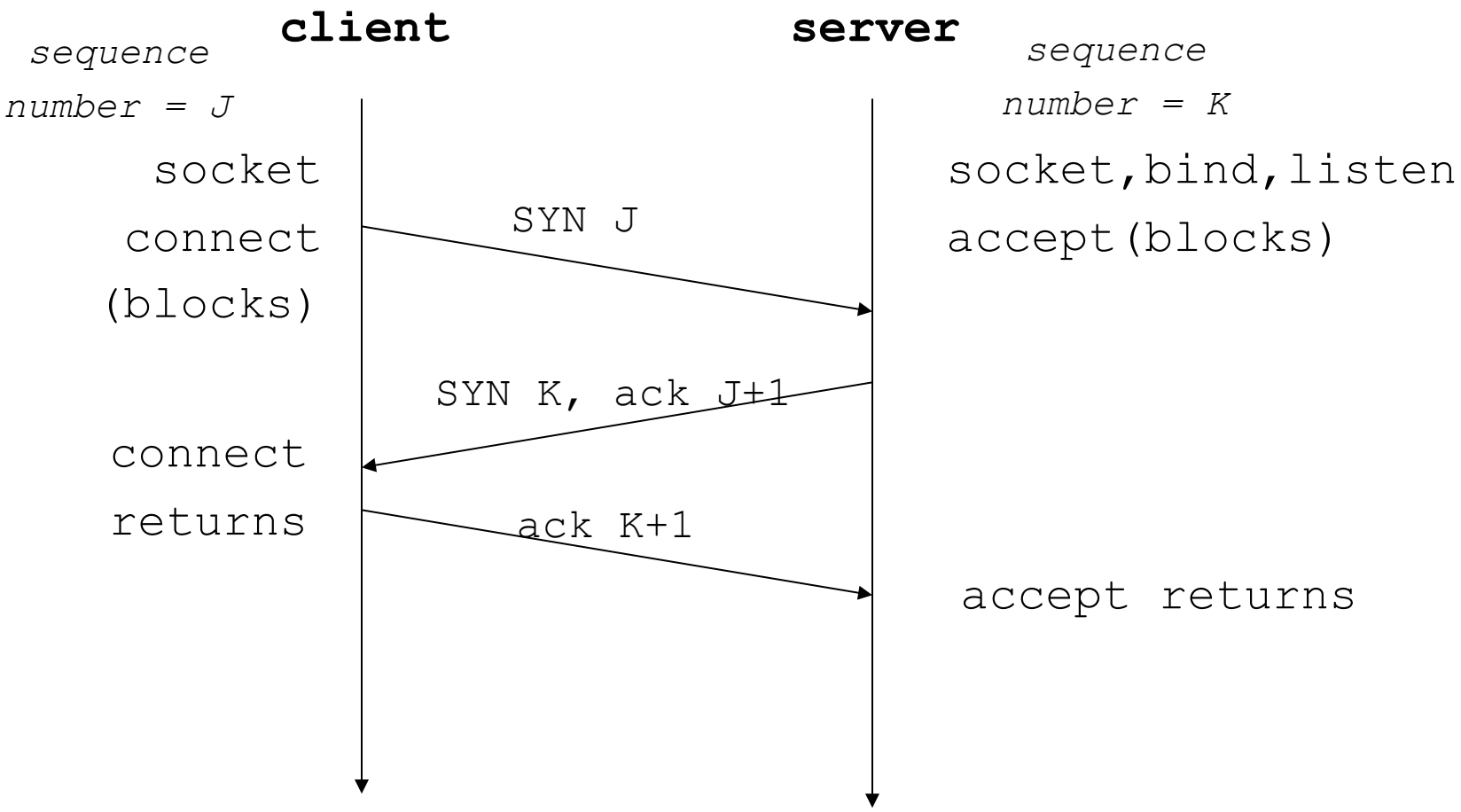
Example analogy:

You can think that HTTP is what goes in the envelop, TCP is the envelop, and IP is the truck.

TCP Connection

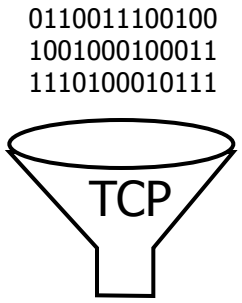


TCP: Three-way handshake



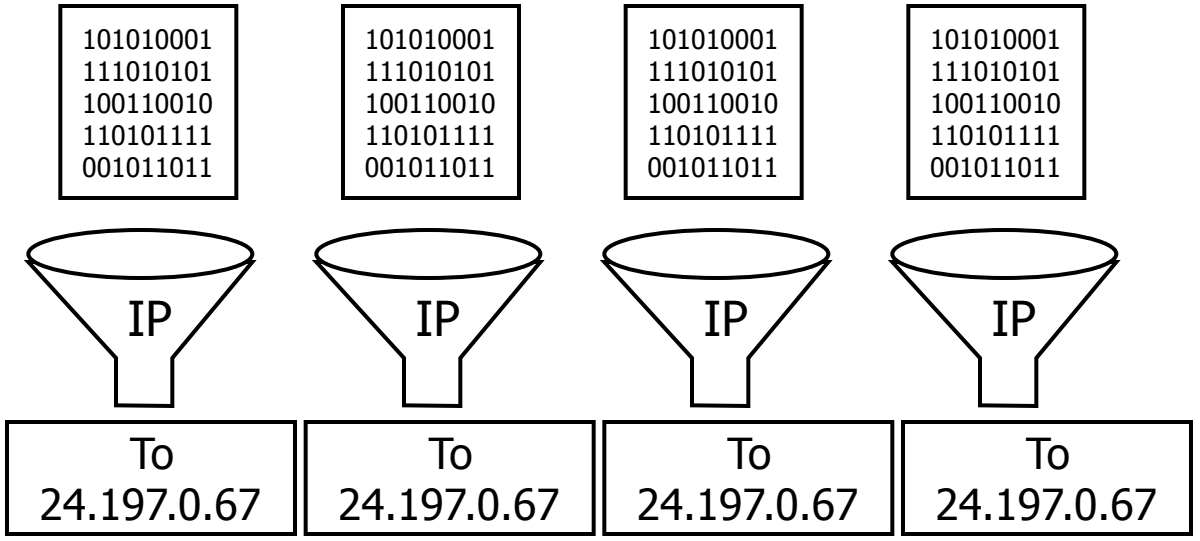
Details

make packets



- Each TCP packet is given a header
  - -sequence number
  - -checksum

put in an IP envelope with another header

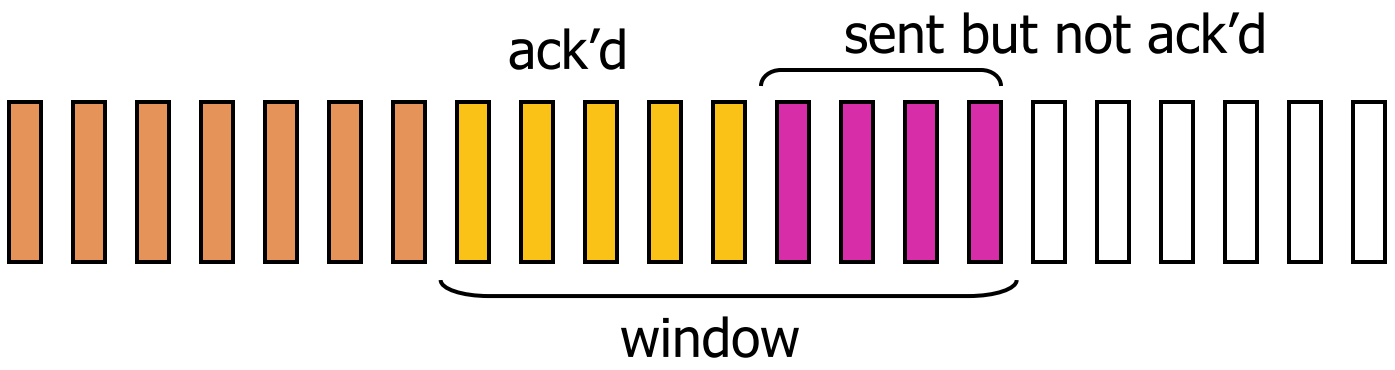


When something goes wrong

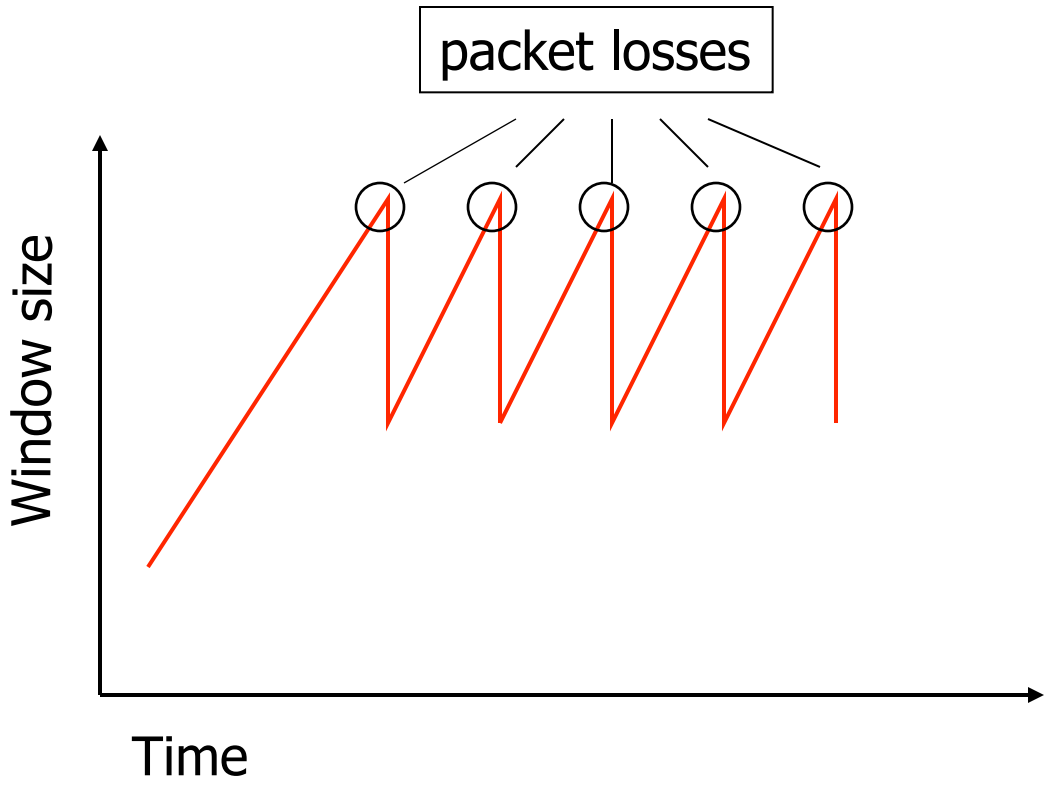
- A packet might not arrive
  - traffic overload
  - bit corruption
- Receiver asks for missing packets to be resent. Want to send data as fast as possible.
- Strategies:
  - Send packets as fast as possible (too many lost)
  - Send packets at a certain rate (can go faster)
  - Wait for the ack (too slow)
  - Window-based (adaptive)

TCP Congestion Control

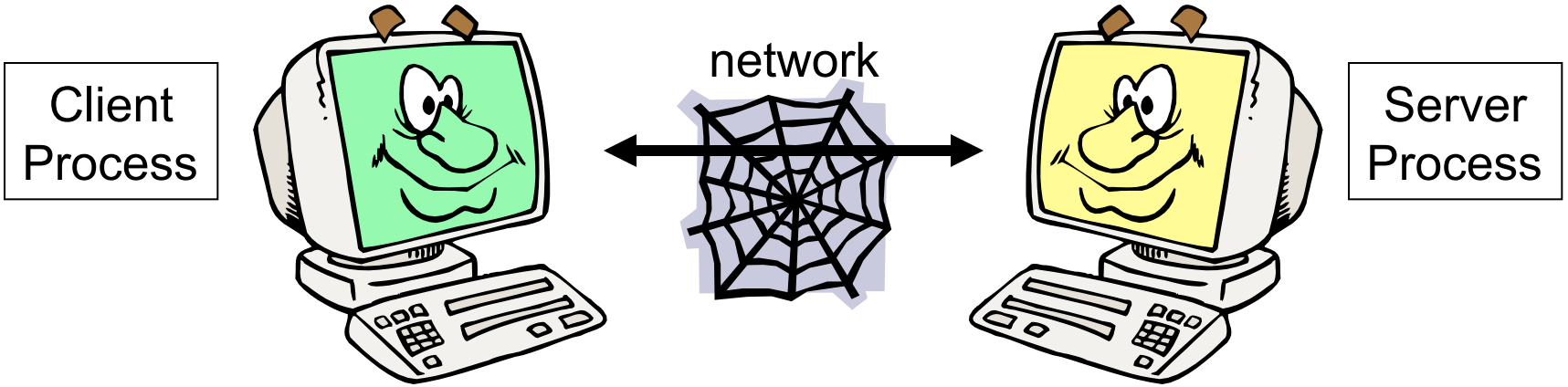
- Window-based:
  - some number of packets allowed to be sent and not ack'd
  - as successful ack's arrive, grow window
  - if packet loss is detected, cut window size



TCP Congestion Control

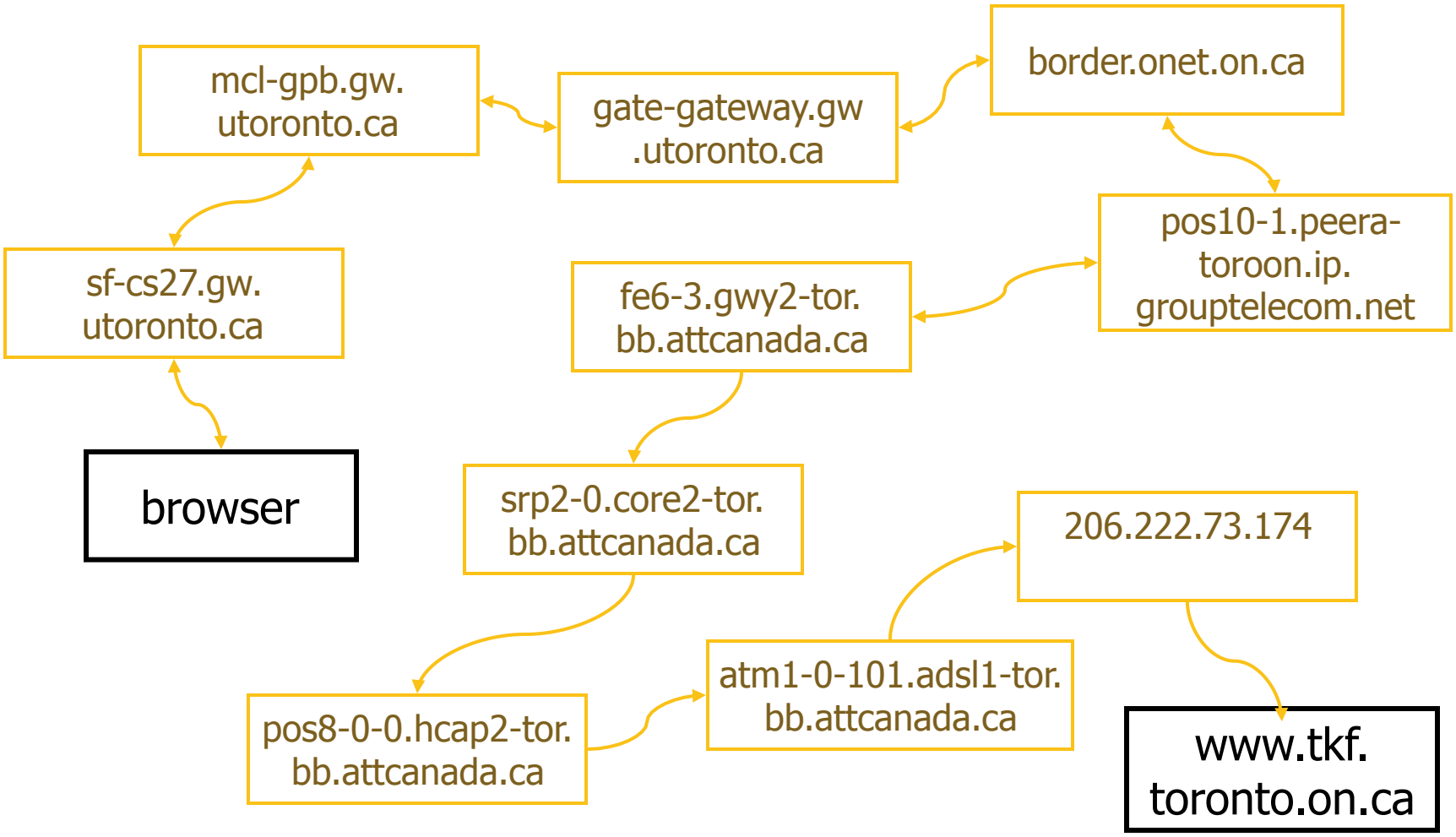


The Big Picture

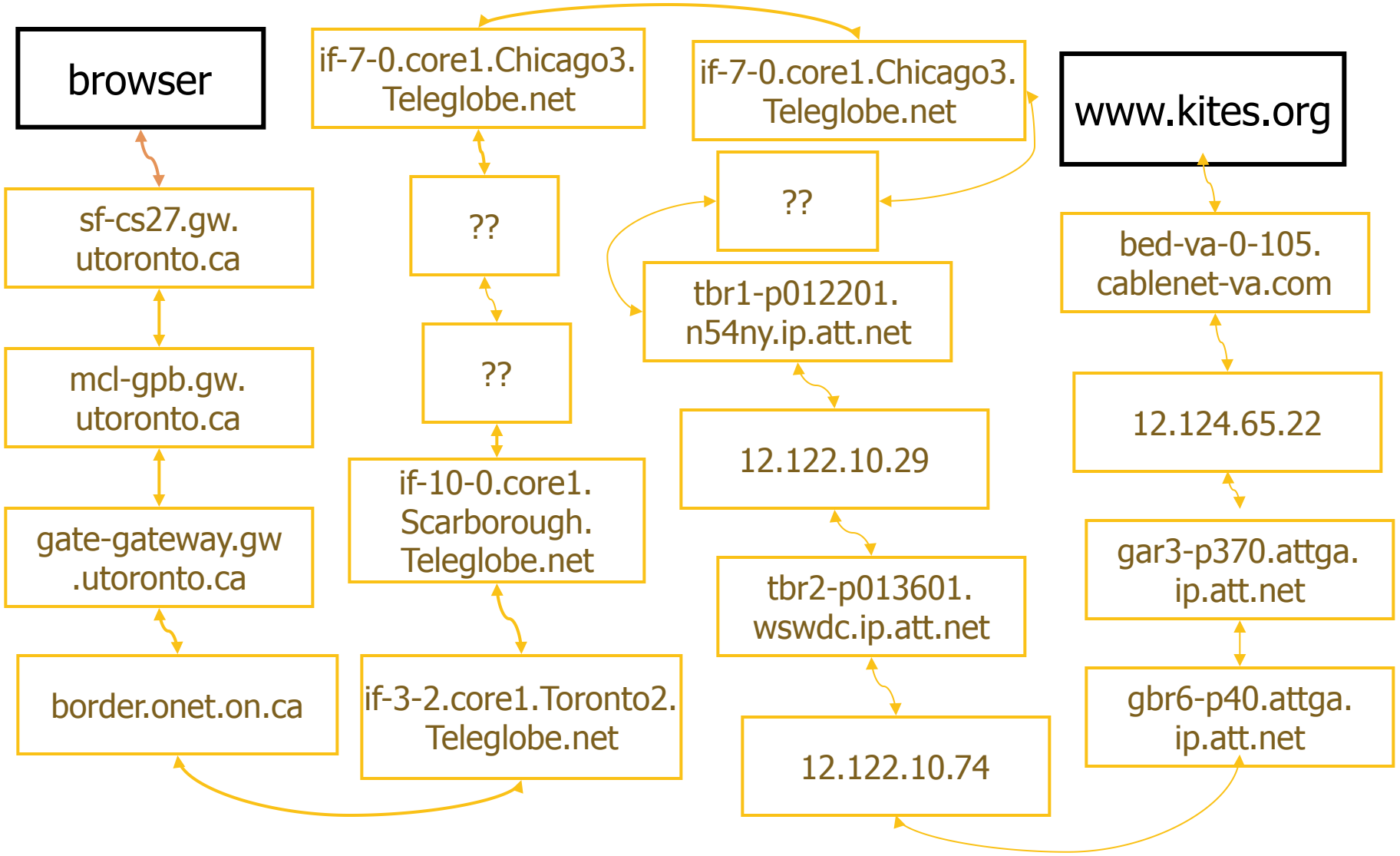


- **Client-Server model:** a client process wants to talk to a server process
- Client must find server - **DNS lookup**
- Client must find service on server - **ports**
- Finally **establish a connection** so they can talk

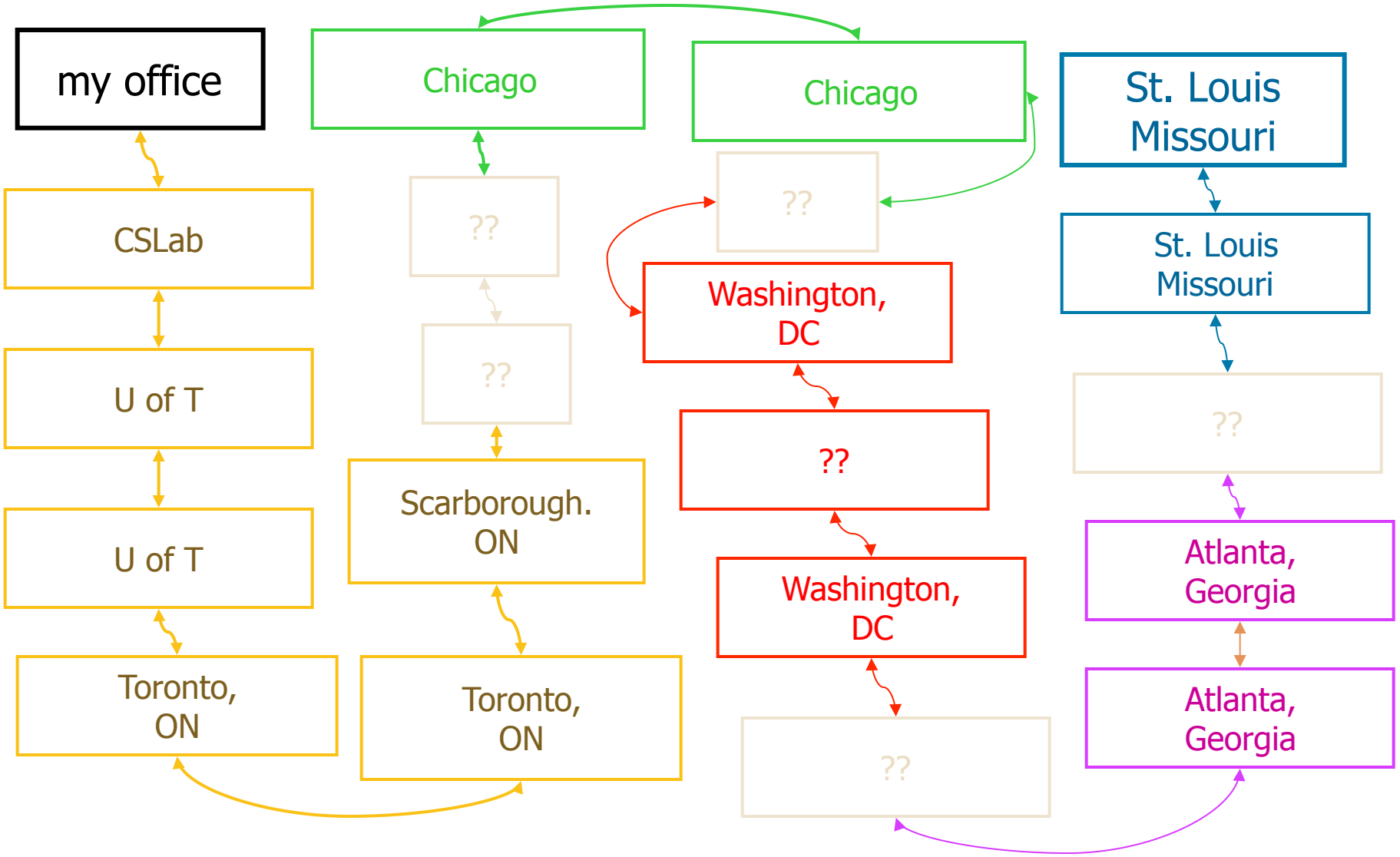
Routing


mcl-gpb.gw.
utoronto.ca
gate-gateway.gw
.utoronto.ca
border.onet.on.ca
sf-cs27.gw.
utoronto.ca
fe6-3.gwy2-tor.
bb.attcanada.ca
pos10-1.peera-
toroon.ip.
grouptelecom.net
browser
srp2-0.core2-tor.
bb.attcanada.ca
206.222.73.174
pos8-0-0.hcap2-tor.
bb.attcanada.ca
atm1-0-101.adsl1-tor.
bb.attcanada.ca
www.tkf.
toronto.on.ca

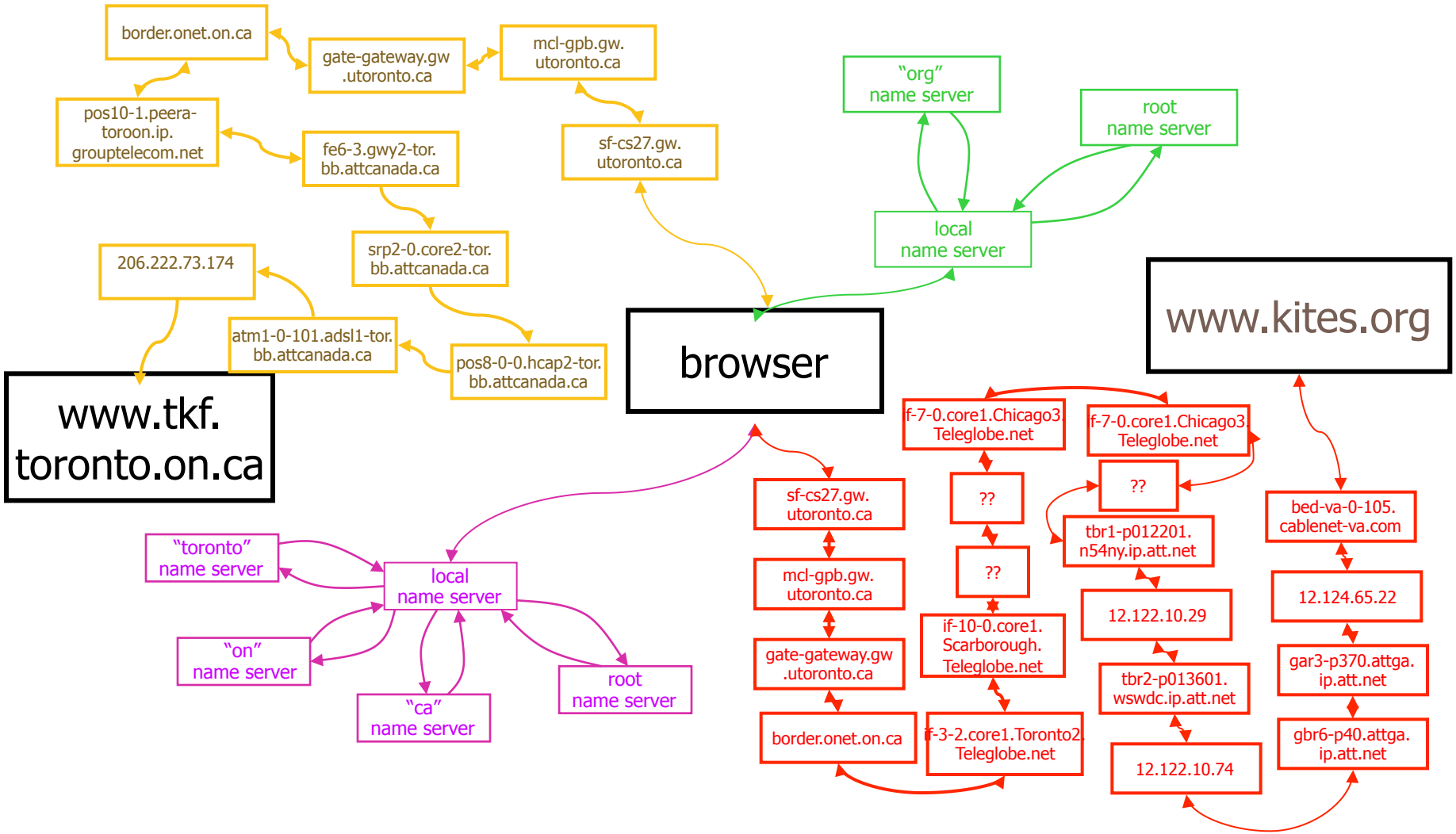
Plus the redirect (20 hops)



At least 5 different cities



Putting it together



How many messages?

- It depends on the size of the web page we retrieve.
- If the web page is 100 Kbytes (small!) it will be broken up into ~80 IP packets.

10 (DNS) +
22 (Connect with toronto.on.ca) +
6 (DNS) +
+ 80 * 20 hops = 1638 messages!

Types of Connection (TCP/UDP)

- Connection oriented model
  - Like phone calls
  - Uses Transmission Control Protocol (TCP)
  - Defined ordering of messages and acks
- Connectionless model
  - Like sending letters via postal service
  - Uses User Datagram Protocol (UDP)
  - More efficient and good for sending broadcasts to many machines