What makes the world wide web work?

Karen Reid
Simple Web Request

MarkUs

MarkUs is a web application for giving high-quality feedback to students. The administrator...
Response
The Request

• How do we tell the web server what we want?
• How do we even find the web server?
• How do the web server and browser talk to each other?
HTTP Request

request
GET / HTTP/1.1
Host: markusproject.org
...

reply
HTTP/1.1 200 OK
Date: Tue, 13 Mar 2017
Server: Apache/2.2.22(Debian)
Content-Type: text/html
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.

markusproject.org = 69.164.221.145
Domain Name Servers

- Browser
- 69.164.221.145
- Root name server
- Delegate server
- "org" name server
- Org server
- Delegate servers
- Markusproject.org?
This is getting complicated!

Number of messages? 8 (or so)
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 many 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

<table>
<thead>
<tr>
<th>Qty</th>
<th>Description</th>
<th>Unit Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Athalon</td>
<td>219.00</td>
<td>219.00</td>
</tr>
<tr>
<td>2</td>
<td>128 MB</td>
<td>149.95</td>
<td>299.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Subtotal</strong></td>
<td><strong>518.90</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Tax</strong></td>
<td>77.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>596.74</strong></td>
</tr>
</tbody>
</table>

**Payable to:** CPUS are us  
Fifty nine six hundred $74/100

---

**CPUS are us**
Karen Reid  
Dept. of Computer Science  
University of Toronto

---

**Courier**

---

**We deliver!**
**TCP/IP**

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

<table>
<thead>
<tr>
<th>source address</th>
<th>dest. address</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes</td>
<td>ack</td>
</tr>
<tr>
<td></td>
<td>port</td>
</tr>
</tbody>
</table>

**data**
TCP Connection

Hi 207.99.53.46
Connection port 80?
okay
Send me a file
Got it
Here’s some data
Got it
Here’s some more
Got it
I’m done
I’m done too

Hi 192.168.1.1
Let’s talk
okay
Send me a file
Got it
Here’s some data
Got it
Here’s some more
Got it
I’m done
I’m done too
Packaging up the data

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

- make packets

- put in an IP envelope with another header

To 207.99.53.46

To 207.99.53.46

To 207.99.53.46

To 207.99.53.46
**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 process on server - **ports**
- Finally **establish a connection** so two processes can talk.
7 cities, 5 states/prov, 2 countries

- browser
  - home network
    - Rogers cable
      - Toronto
        - Teksavvy
          - Richmond Hill
            - Teksavvy
              - Richmond Hill
              - Tempe, Arizona
                - Teksavvy
                  - Toronto
                    - Tempe, Arizona
              - Tempe, Arizona
              - Marina Del Rey, California
                - Parsippany, New Jersey
                  - Newark, New Jersey
                    - Greendell, New Jersey
                      - Parsippany, New Jersey
Putting it together

Browser

Markusproject.org

192.168.1.1
10.126.51.129
67.231.222.81

Bloor6.cable.teksavvy.com

Tge11-3.fr4.yyz.llnw.net

Bloor1.cable.teksavvy.com

Tge8-1.fr3.lga.llnw.net

Bloor1.cable.teksavvy.com

Eqix.e-2-3.tbr2.ewr.nac.net

Bloor1.cable.teksavvy.com

0.e1-2.tbr2.mmu.nac.net

Li136-145.members.linode.com

207.99.53.42

Vlan804.esd2.mmu.nac.net

Root name server

Local name server

"Org" name server

Delegate server

Markusproject.org

192.168.1.1
10.126.51.129
67.231.222.81

Bloor6.cable.teksavvy.com

Tge11-3.fr4.yyz.llnw.net

Bloor1.cable.teksavvy.com

Tge8-1.fr3.lga.llnw.net

Bloor1.cable.teksavvy.com

Eqix.e-2-3.tbr2.ewr.nac.net

Bloor1.cable.teksavvy.com

0.e1-2.tbr2.mmu.nac.net

Li136-145.members.linode.com

207.99.53.42

Vlan804.esd2.mmu.nac.net

Root name server

Local name server

"Org" name server

Delegate server

Markusproject.org
How many messages?

- It depends on the size of the web page.
- The web page that appears for markusproject.org is less than 30 Kbytes.
- If the web page is 30 Kbytes (small!) it will likely be broken up into ~20 IP packets.

\[
\text{8 (DNS) + 20 \times 15 \text{ hops}} \quad = \quad 308 \text{ messages}
\]
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.
• But sending too fast wastes resources.
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

![Graph showing time, window size, and packet losses]
All we did was click on a link...
Take aways

• The web today is made up of complex layers of software
• No one person, organization, or company could have created it in isolation
• We can understand it because we can study one layer at a time
• We can create new things by building on top of existing layers