COMP 431 Internet Protocols & Services

Spring 2017  
Kevin Jeffay

Worksheet 8, February 14

1) The cute-pets website stores videos of, you guessed it, cute pets. To ensure users can watch videos with minimal delay, the website uses the “go-fast” CDN to deliver videos to users. Assume a user views a page on the cute-pets website and clicks on the link:

*http://snoring-cats.cute-pet-videos.com/videoXYZ*

to watch the video “videoXYZ.” Explain how the cute-pets and go-fast organizations can use the DNS to ensure that when the user clicks on the above link, the user is directed to a server in the go-fast CDN that has the requested content.

**Instead of cute-pets linking to content hosted on its own servers, it would place the content on the CDN and link to those files. Because CDN relies on having content hosted on servers to which clients can get information the fastest (often geographically closer,) the go-fast CDN DNS authoritative server could respond dynamically to each request by providing an A-record to the closest CDN server to the client, which the clients’ local and intermediate DNS servers could cache for the future based on their locations.**

Players: local DNS, cute pet server, go-fast dns

What will happen- user’s local DNS will query cute-pet’s authoritative DNS?

Option: return NS, not good

Option: Can return A record containing the IP address of a go-fast CDN server

Downside is that cute-pets has to do all the IP resolution

**Option: return a CNAME record with a hostname for go-fast**

**Cute-pets will return a CNAME to go-fast to the user’s local DNS. Client will contact go-fast authoritative DNS. Encoded host name will be analyzed for best server, return that IP.**

**CDN will use geolocation using the IP**

**There will be an extra DNS resolution**

Returning an A-record, cute-pets somehow had to know which go-fast CDN content server hosts the requested video. Removes the ability to shift load-balancing.

Idea: snoring-cats is not a real hostname. We encode the content into the hostname. Snoring-cats is equivalent to videoXYZ

2) A COMP 431 student has the Chrome and Firefox browsers installed on their laptop, and uses both of these browsers at the same time to access the COMP 431 site on Piazza. Assume that Chrome is used to access page “*c*” on the Piazza site and that Firefox is used to access page “*f*” on the site.

Explain how it is that the contents of the *c* page makes it back to the Chrome browser and the contents of the *f* page make it back to the Firefox browser as opposed to, for example, all the content coming back from, Piazza coming back to just one browser and not the other.

HTTP is encapsulated with TCP encapsulated with IP.

Source port is the OS’s identification of the application and socket. When the data arrives, the OS throws data into sockets (or queues,) and the application reads the data from its queues

**The two different browsers are running on different local ports since they are different processes, so their respective TCP connections to server would be separate because of the source port field, and the servers’ responses would be demultiplexed correctly.**

3) Consider distributing a file of *F* = 15 Gbits to *N* peers. The server has an upload rate of *us* = 30 Mbps, and each peer has a download rate of *di* = 2 Mbps and an upload rate of *u*. For *N* = 10, 100, and 1,000 and *u* = 300 Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of *N* and *u* for both client-server distribution and P2P distribution.

P2P:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | u (peer) bit/s | f bits | u (server) bit/s | d(peer) bit/s |
| 10 | 300000 | 15000000000 | 30000000 | 2000000 |
| 100 | 300000 | 15000000000 | 30000000 | 2000000 |
| 1000 | 300000 | 15000000000 | 30000000 | 2000000 |
| 10 | 700000 | 15000000000 | 30000000 | 2000000 |
| 100 | 700000 | 15000000000 | 30000000 | 2000000 |
| 1000 | 700000 | 15000000000 | 30000000 | 2000000 |
| 10 | 2000000 | 15000000000 | 30000000 | 2000000 |
| 100 | 2000000 | 15000000000 | 30000000 | 2000000 |
| 1000 | 2000000 | 15000000000 | 30000000 | 2000000 |
|  |  |  |  |  |
| f/u(s) seconds | f/d(min) seconds | nf/(u(s)+sum(u(peer))) seconds | | **(max) Seconds** |
| 500 | 7500 | 4545.454545 |  | **7500** |
| 500 | 7500 | 25000 |  | **25000** |
| 500 | 7500 | 45454.54545 |  | **45454.54545** |
| 500 | 7500 | 4054.054054 |  | **7500** |
| 500 | 7500 | 15000 |  | **15000** |
| 500 | 7500 | 20547.94521 |  | **20547.94521** |
| 500 | 7500 | 3000 |  | **7500** |
| 500 | 7500 | 6521.73913 |  | **7500** |
| 500 | 7500 | 7389.162562 |  | **7500** |

Client-Server:

|  |  |  |  |
| --- | --- | --- | --- |
| N | f bits | u (server) bit/s | d(peer) bit/s |
| 10 | 15000000000 | 30000000 | 2000000 |
| 100 | 15000000000 | 30000000 | 2000000 |
| 1000 | 15000000000 | 30000000 | 2000000 |
| 10 | 15000000000 | 30000000 | 2000000 |
| 100 | 15000000000 | 30000000 | 2000000 |
| 1000 | 15000000000 | 30000000 | 2000000 |
| 10 | 15000000000 | 30000000 | 2000000 |
| 100 | 15000000000 | 30000000 | 2000000 |
| 1000 | 15000000000 | 30000000 | 2000000 |
|  |  |  |  |
| NF/u\_s | F/d\_min | **Max** |  |
| 5000 | 7500 | **7500** |  |
| 50000 | 7500 | **50000** |  |
| 500000 | 7500 | **500000** |  |
| 5000 | 7500 | **7500** |  |
| 50000 | 7500 | **50000** |  |
| 500000 | 7500 | **500000** |  |
| 5000 | 7500 | **7500** |  |
| 50000 | 7500 | **50000** |  |
| 500000 | 7500 | **500000** |  |

4) Consider distributing a file of *F* bits to *N* peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed *us*.

*a*) Suppose that *us*/N ≤ *dmin*. Specify a distribution scheme that has a distribution time of *NF*/*us*.

Split the upload N ways with each connection having equal distribution and each client has the same download rate. Clients can download faster than the server uploads.

*b*) Suppose that *us*/N ≥ *dmin*. Specify a distribution scheme that has a distribution time of *F*/*dmin*.

Split the connection N ways, the total time is still limited by the slowest connection. Clients have different download times. Match the servers’ upload speeds to clients’ download speeds.

*c*) Show that the minimum distribution time is in general given by max{*NF*/*us*, *F*/*dmin* }.

Since these are the only two possible situations with the client-server model (either all the clients have the same download rate or they don’t,) we will be constrained to the slower of the two, which would occur with uneven rates.

5) Consider distributing a file of *F* bits to *N* peers using a P2P architecture. Assume a fluid model. For simplicity assume that *dmin* is very large, so that peer download bandwidth is never a bottleneck.

*a*) Suppose that *us* ≤ (*us* + *u*1 + ... + *uN*)/N. Specify a distribution scheme that has a distribution time of *F*/*us*.

In this case, the server upload speed has a slower speed than the average speed of the server and all the peers. A scheme with this distribution time would be one where peers with faster upload speeds are sharing data with each other as they receive data from the server. Peers send data in parallel. The server would have to upload the file once.

*b*) Suppose that *us* ≥ (*us* + *u*1 + ... + *uN*)/*N*. Specify a distribution scheme that has a distribution time of   
*NF* /(*us* + *u*1 + ... + *uN*).

In this case, the server can upload faster than the average client can share data. Such a distribution scheme could consist of the server taking up more of a share of the distribution and file uploads to individual peers since it is faster than the average peer. The server would have to upload multiple copies of the same data to multiple peers. The distribution time would scale depending on the upload speeds of new peers.

*c*) Show that the minimum distribution time is in general given by max{*F*/*us*, *NF* /(*us* + *u*1 + ... + *uN*)}.

The server’s upload speed will either be faster, slower, or equal to the average upload speed in the network because of numbers work. In the case of the server being faster, the second term will be greater because we are constrained by the peers. In the case of the server being slower, the first term will be greater because we are constrained by the server. In the case that the server is equal to the average, the two terms will be equal as well. Therefore, the minimum distribution time is given by the max of the two terms.