

Shivam Bhat
bhat41@purdue.edu

Instructions:

I was getting cookies delivered in some requests which was then found by server to not be in www and hence lead to the program crashing. For this, running in incognito solved the issue. Also this was being tested on Purdue's Cisco AnyConnect VPN.

Compilation commands are the same as required by the handout. They are nonetheless shown in the images

PART A

Server.c (localhost)

```
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > ./server 1234
The server is ready to receive
message-from-client: 127.0.0.1, 57616
TestMsg
close-client: 127.0.0.1, 57616
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > █
```

Client.c on (localhost)

```
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > ./client 127.0.0.1 1234 TestMsg
Response:TestMsg
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > █
```

Different Machines(amber02 and amber)

Server

```

● ● ● ~%1 -zsh %1 -zsh %2 ssh %3 ssh %4 -zsh %5
amber02 74 $ gcc server.c -o server
server.c: In function 'connectionHandler':
server.c:34:68: warning: '%s' directive writing up to 1023 bytes into a region of size 998 [-Wformat-overflow=]
  34 |     int retVal=sprintf(serverResponse,"message-from-client: %s, %d \n%s",client_ip,client_port,clientQuery);
                  ^~                                         ~~~~~
server.c:34:14: note: 'sprintf' output 27 or more bytes (assuming 1050) into a destination of size 1024
  34 |     int retVal=sprintf(serverResponse,"message-from-client: %s, %d \n%s",client_ip,client_port,clientQuery);
                  ^~~~~~                                         ~~~~~
amber02 75 $ gcc client.c -o client
amber02 76 $ ./server 12345
The server is ready to receive
message-from-client: 128.10.112.132, 57448
Hi

```

```

amber07 59 $ ./client 128.10.112.132 12354 hi
Response:hi

```

2. With Pthread Multi-threading

Server Mul on two different Terminals as stated in doc

```

~/Doc/P/PurduePrivate/l/c/l/s/1.b devCNS *5 !24 ?19 > gcc serverMul.c -o serverMul
~/Doc/P/PurduePrivate/l/c/l/s/1.b devCNS *5 !24 ?19 > gcc clientMul.c -o clientMul
~/Doc/P/PurduePrivate/l/c/l/s/1.b devCNS *5 !24 ?19 > ./serverMul 1234
The server is ready to receive
message-from-client: 127.0.0.1 57743
hi
message-from-client: 127.0.0.1 57745
jk
close-client: 127.0.0.1, 57743
close-client: 127.0.0.1, 57745

```

Clients

```

~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > ls
client      client.c      server      server.c
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > ./client 127.0.0.1 1234 hi
Response:hi
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 >

```

```
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > ./client 127.0.0.1 1234 jk  
Response:jk  
~/Doc/P/PurduePrivate/l/c/l/s/1.a devCNS *5 !24 ?19 > █
```

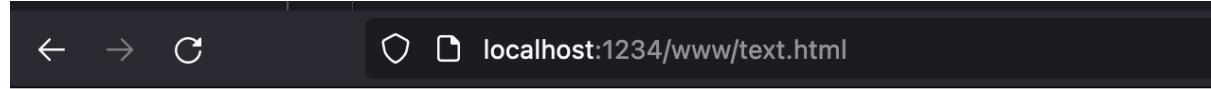
PART B

FIREFOX ,CHROME & SAFARI

www/text.html and [/text.html](http://text.html)

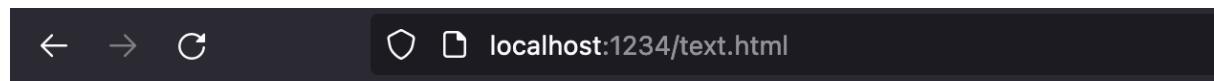
In some browser/OS config. If we reload the page with video.html displayed it breaks the pipe if we do the reloading before the video is completely sent.

FIREFOX



Test 1

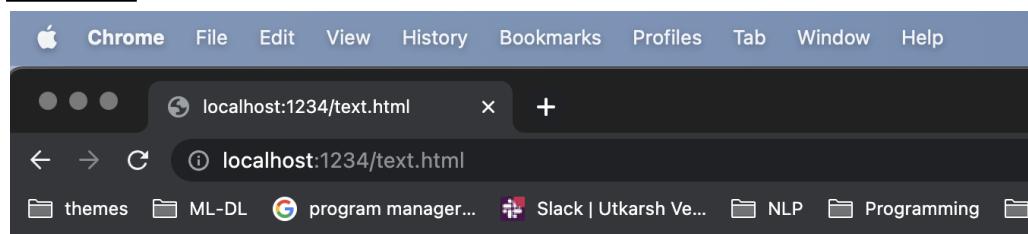
Hello World!



Test 1

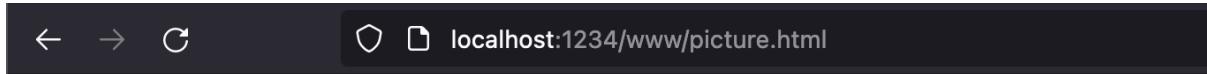
Hello World!

CHROME



Test 1

Hello World!

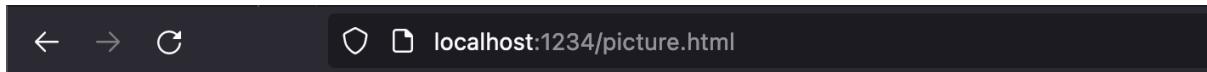


Test 2

Sample: A small picture.



FIREFOX

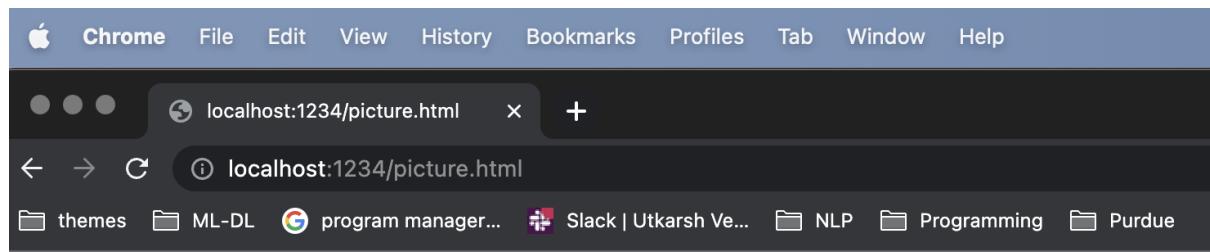


Test 2

Sample: A small picture.



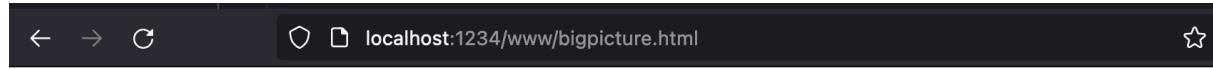
CHROME



Test 2

Sample: A small picture.



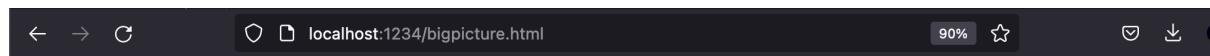


Test 3

Sample: A big picture.



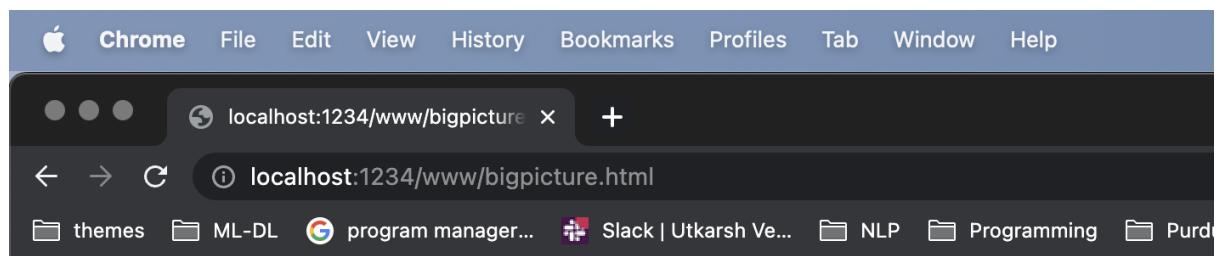
FIREFOX



Test 3

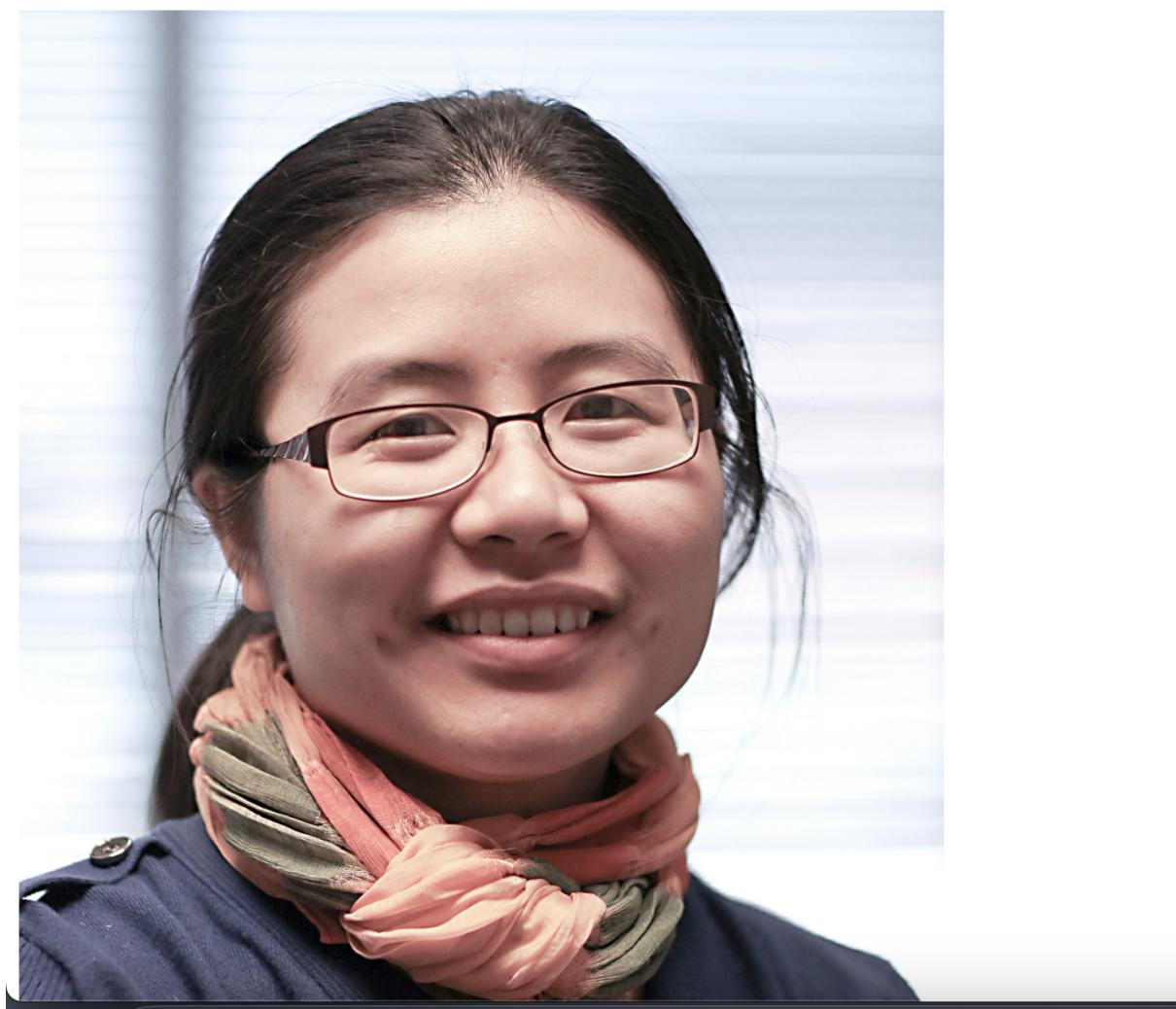
Sample: A big picture.



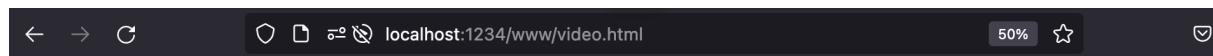


Test 3

Sample: A big picture.



CHROME



Video Test

The Future of Networking, and the Past of Protocols - Scott Shenker

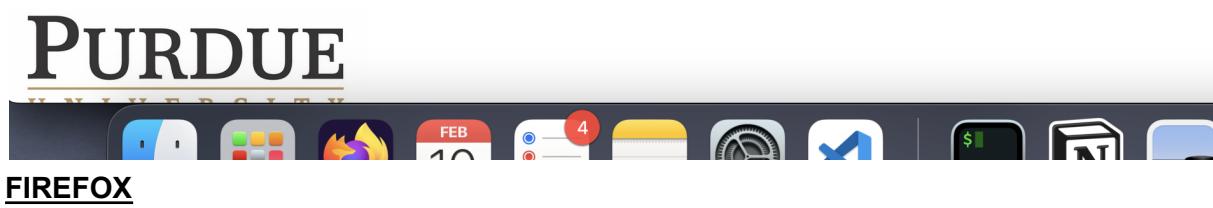
Note: the video is released at YouTube, https://www.youtube.com/watch?v=YHeyuD89nIY&t=123s&ab_channel=OpenNetworkingSummit



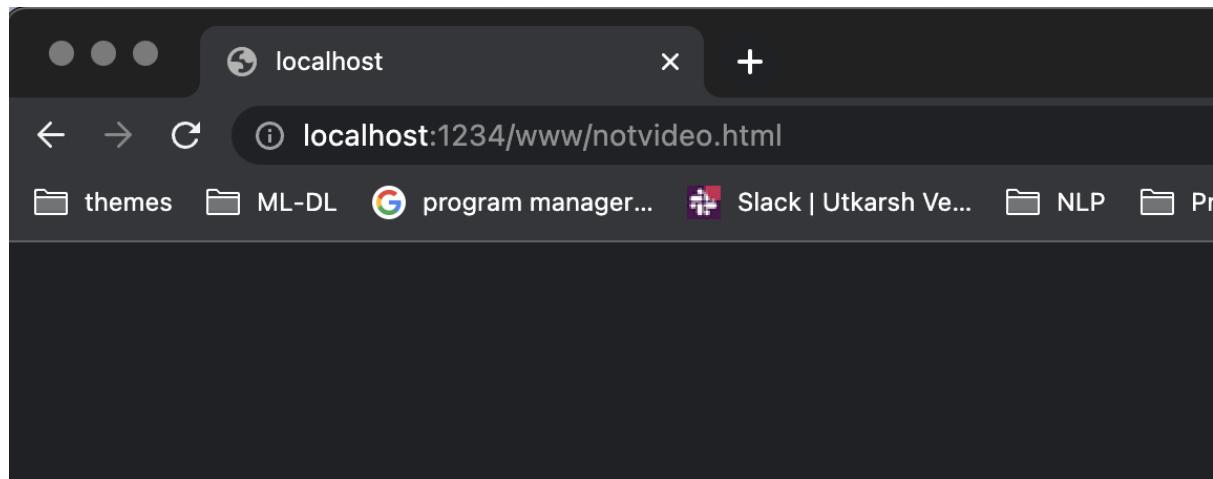
A big picture.



A small picture.



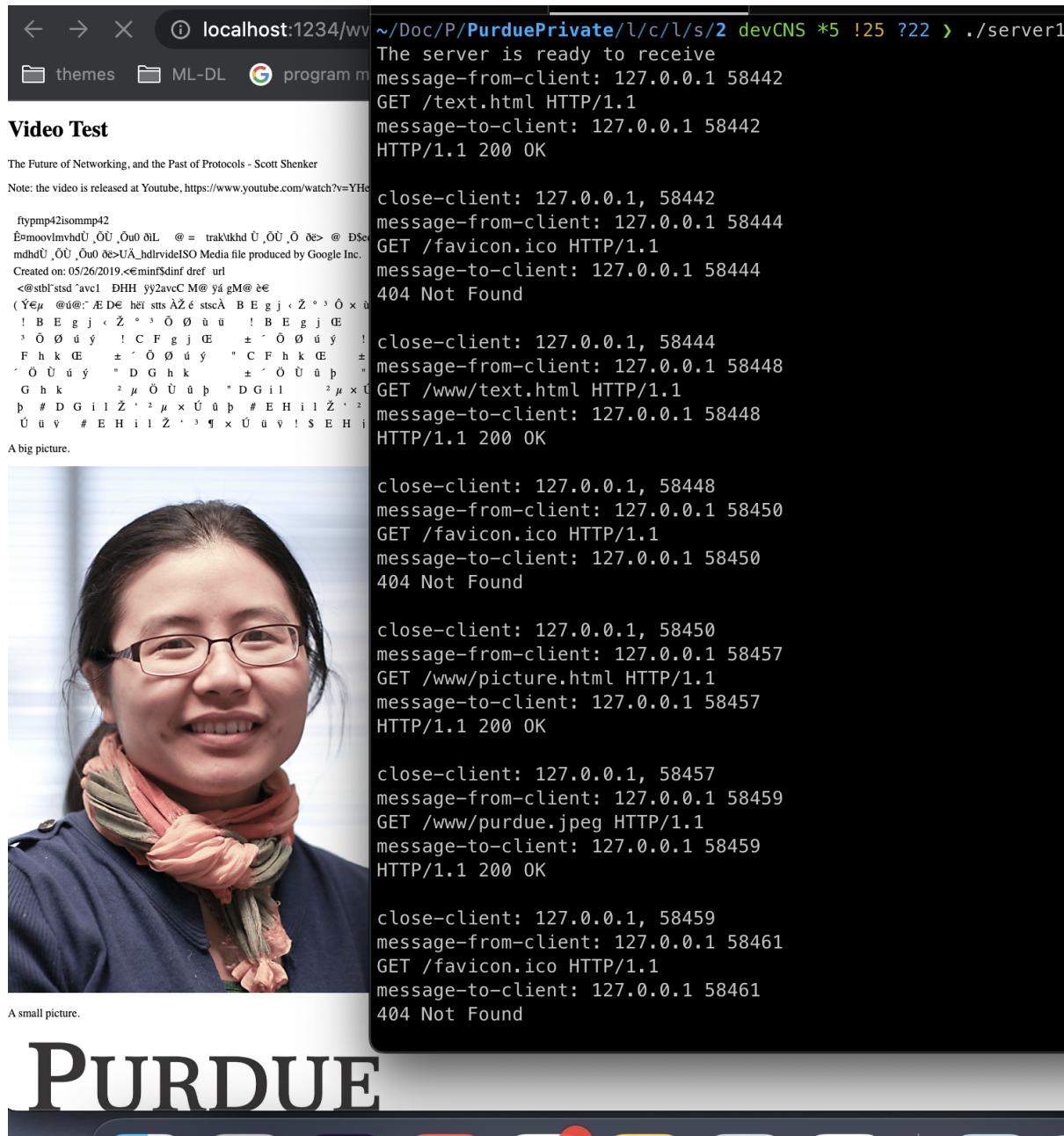
ERROR 404 - when searching for notvideo.html - NON EXISTING PAGE



```
~/Doc/P/PurduePrivate/l/c/l/s/2 devCNS *5 !25 ?22 > ./server1 1234
The server is ready to receive
message-from-client: 127.0.0.1 58512
GET /www/notvideo.html HTTP/1.1
message-to-client: 127.0.0.1 58512
404 Not Found

close-client: 127.0.0.1, 58512
```

LOGS



localhost:1234/www/video.html

localhost:1234/www/bigpicture.jpeg

localhost:1234/www/purdue.jpeg

localhost:1234/www/video.mp4

server1

-zsh

```

close-client: 127.0.0.1, 58461
message-from-client: 127.0.0.1 58463
GET /www/bigpicture.html HTTP/1.1
message-to-client: 127.0.0.1 58463
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58463
message-from-client: 127.0.0.1 58465
GET /www/bigpicture.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58465
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58465
message-from-client: 127.0.0.1 58467
GET /favicon.ico HTTP/1.1
message-to-client: 127.0.0.1 58467
404 Not Found

close-client: 127.0.0.1, 58467
message-from-client: 127.0.0.1 58469
GET /www/video.html HTTP/1.1
message-to-client: 127.0.0.1 58469
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58469
message-from-client: 127.0.0.1 58471
GET /www/bigpicture.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58471
HTTP/1.1 200 OK

message-from-client: 127.0.0.1 58473
GET /www/video.mp4 HTTP/1.1
message-to-client: 127.0.0.1 58473
HTTP/1.1 200 OK

message-from-client: 127.0.0.1 58475
GET /www/purdue.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58475
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58475
close-client: 127.0.0.1, 58471

```



URDUE

LOGS

Refreshed to load new pages

close-client: 127.0.0.1, 58448
message-from-client: 127.0.0.1 58450
GET /favicon.ico HTTP/1.1
message-to-client: 127.0.0.1 58450
404 Not Found

close-client: 127.0.0.1, 58450
message-from-client: 127.0.0.1 58457
GET /www/picture.html HTTP/1.1
message-to-client: 127.0.0.1 58457
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58457
message-from-client: 127.0.0.1 58459
GET /www/purdue.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58459
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58459
message-from-client: 127.0.0.1 58461
GET /favicon.ico HTTP/1.1
message-to-client: 127.0.0.1 58461
404 Not Found

close-client: 127.0.0.1, 58461
message-from-client: 127.0.0.1 58463
GET /www/bigpicture.html HTTP/1.1
message-to-client: 127.0.0.1 58463
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58463
message-from-client: 127.0.0.1 58465
GET /www/bigpicture.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58465
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58465
message-from-client: 127.0.0.1 58467
GET /favicon.ico HTTP/1.1
message-to-client: 127.0.0.1 58467
404 Not Found

close-client: 127.0.0.1, 58467
message-from-client: 127.0.0.1 58469
GET /www/video.html HTTP/1.1
message-to-client: 127.0.0.1 58469

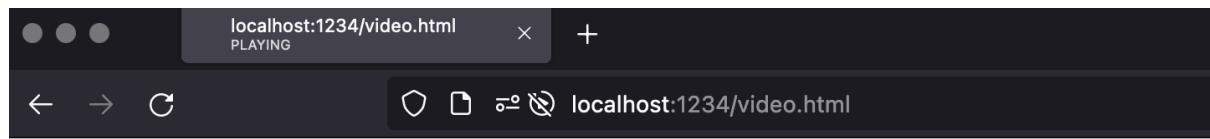
HTTP/1.1 200 OK

close-client: 127.0.0.1, 58469
message-from-client: 127.0.0.1 58471
GET /www/bigpicture.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58471
HTTP/1.1 200 OK

message-from-client: 127.0.0.1 58473
GET /www/video.mp4 HTTP/1.1
message-to-client: 127.0.0.1 58473
HTTP/1.1 200 OK

message-from-client: 127.0.0.1 58475
GET /www/purdue.jpeg HTTP/1.1
message-to-client: 127.0.0.1 58475
HTTP/1.1 200 OK

PART C



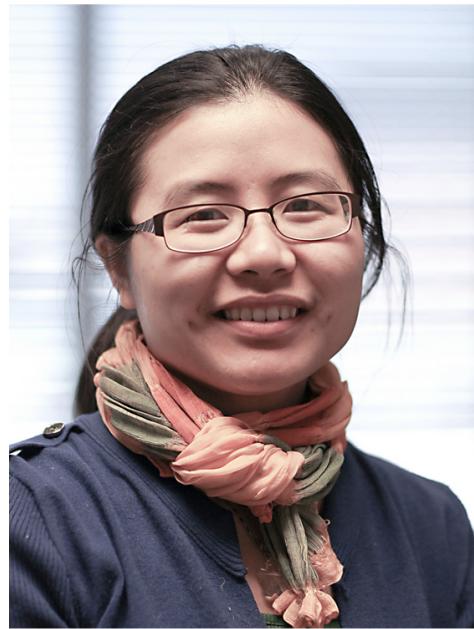
Video Test

The Future of Networking, and the Past of Protocols - Scott Shenker

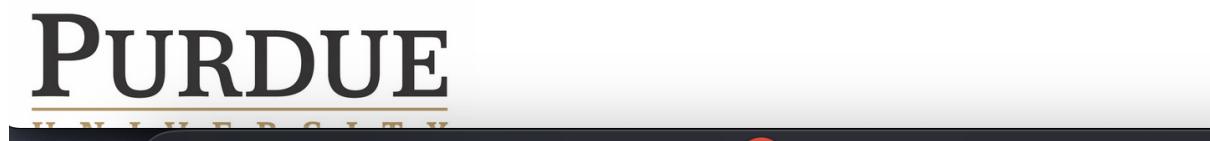
Note: the video is released at Youtube, https://www.youtube.com/watch?v=YHeyuD89nIY&t=123s&ab_channel=OpenNetworkingSummit

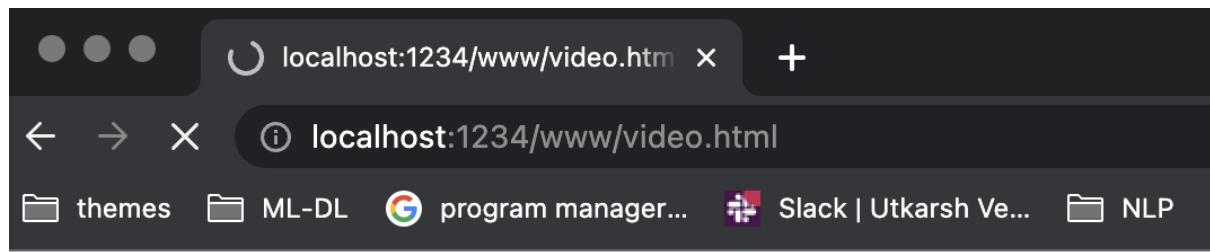


A big picture.



A small picture.





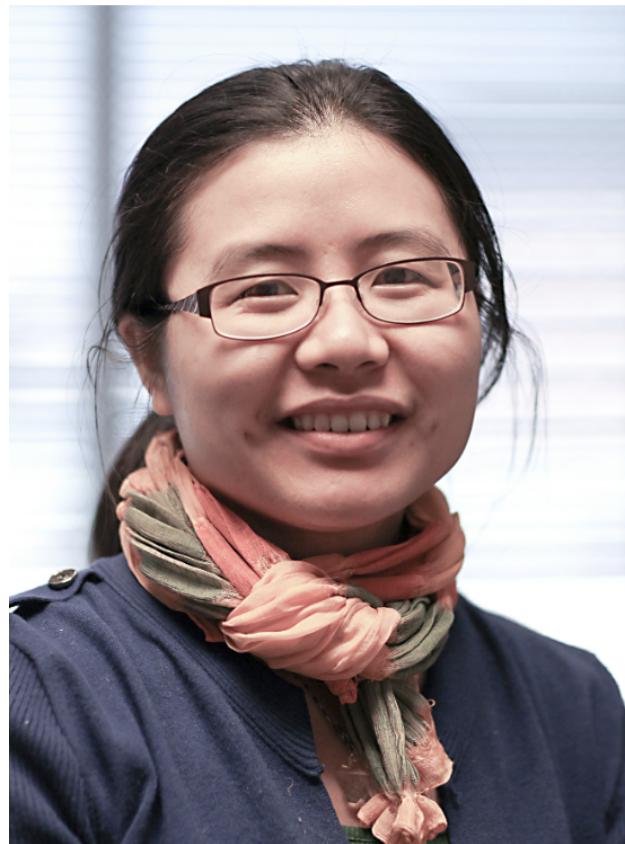
Video Test

The Future of Networking, and the Past of Protocols - Scott Shenker

Note: the video is released at Youtube, https://www.youtube.com/watch?v=YHeyuD89n1Y&t=123s&ab_channel=OpenNetworkingSummit

ftypmp42isommp42
Ê»moovImvhdÛ ,ÖÙ ,Öu0 ðiL @ = trak\khdÛ ,ÖÙ ,Ö ðë> @ Ð\$edts elst
mdhdÛ ,ÖÙ ,Öu0 ðë>UÄ _hdlrvideISO Media file produced by Google Inc.
Created on: 05/26/2019.<€minf\$dinf dref url
<@stbl^stds ^avc1 DHH ýy2avcC M@ ýá gM@ è€
(Ý€µ @í@.- AE D€ hëř sttA Z é stscA B E g j < Z ° ³ Ô x ù ü
! B E g j < Z ° ³ Ô Ø ù ü ! B E g j Ø °
³ Ô Ø ú ý ! C F g j Ø ± ' Ô Ø ú ý ! C
F h k Ø ± ' Ô Ø ú ý " C F h k Ø ±
' Ö Ù ú ý " D G h k ± ' Ö Ù û þ " D
G h k ² µ Ö Ù û þ " D G i l ² µ x Ú û
þ # D G i l Z ² µ x Ú û þ # E H i l Z ² µ x
Ú ü ý # E H i l Z ³ ¶ x Ú ü ý ! \$ E H i m

big picture.



small picture.

PURDUE

PART C

My code is able to parse for images and videos. It even handles the commented HTML code given in video.html. I use “\t\t<iframe” and “\t\t<img” for video and image search

I am serving the get requests in round robin fashion. I am nly sending the object and frame and not the actual data. But that would entails simple send and receive. For now my server is sending [OBJECT,FRAME] which is received by the Client.

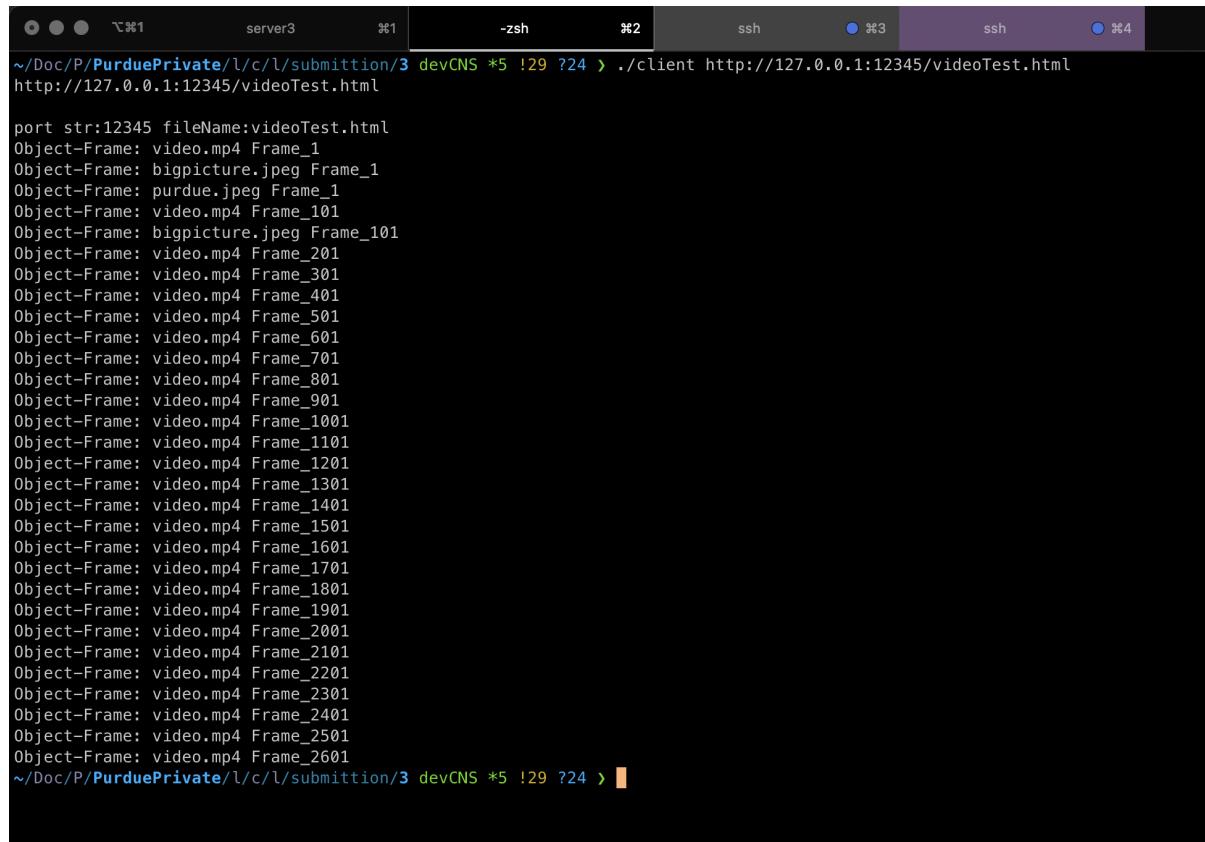
Server Receiving \$ HTTP/2.0 200 OK requests for video.html + purdue.jpeg + bigpicture.jpeg + video

```
● ● ●  ~%1      server3      %1      -zsh      %2      ssh
~/Doc/P/PurduePrivate/labs/cs536/l1/submittion/3 devCNS *5 !29 ?24 > ./server3 12345
The server is ready to receive

message-to-client: 127.0.0.1 61064
HTTP/2.0 200 OK

Object-Frame: video.mp4 Frame_1
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: purdue.jpeg Frame_1
Object-Frame: video.mp4 Frame_101
Object-Frame: bigpicture.jpeg Frame_101
Object-Frame: video.mp4 Frame_201
Object-Frame: video.mp4 Frame_301
Object-Frame: video.mp4 Frame_401
Object-Frame: video.mp4 Frame_501
Object-Frame: video.mp4 Frame_601
Object-Frame: video.mp4 Frame_701
Object-Frame: video.mp4 Frame_801
Object-Frame: video.mp4 Frame_901
Object-Frame: video.mp4 Frame_1001
Object-Frame: video.mp4 Frame_1101
Object-Frame: video.mp4 Frame_1201
Object-Frame: video.mp4 Frame_1301
Object-Frame: video.mp4 Frame_1401
Object-Frame: video.mp4 Frame_1501
Object-Frame: video.mp4 Frame_1601
Object-Frame: video.mp4 Frame_1701
Object-Frame: video.mp4 Frame_1801
Object-Frame: video.mp4 Frame_1901
Object-Frame: video.mp4 Frame_2001
Object-Frame: video.mp4 Frame_2101
Object-Frame: video.mp4 Frame_2201
Object-Frame: video.mp4 Frame_2301
Object-Frame: video.mp4 Frame_2401
Object-Frame: video.mp4 Frame_2501
Object-Frame: video.mp4 Frame_2601
close-client: 127.0.0.1, 61064
```

Client Side receiving data frames from server



The screenshot shows a terminal window with four tabs. The active tab (%2) displays the command `./client http://127.0.0.1:12345/videoTest.html`. The output of the command is a series of log entries indicating frame requests:

```
port str:12345 fileName:videoTest.html
Object-Frame: video.mp4 Frame_1
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: purdue.jpeg Frame_1
Object-Frame: video.mp4 Frame_101
Object-Frame: bigpicture.jpeg Frame_101
Object-Frame: video.mp4 Frame_201
Object-Frame: video.mp4 Frame_301
Object-Frame: video.mp4 Frame_401
Object-Frame: video.mp4 Frame_501
Object-Frame: video.mp4 Frame_601
Object-Frame: video.mp4 Frame_701
Object-Frame: video.mp4 Frame_801
Object-Frame: video.mp4 Frame_901
Object-Frame: video.mp4 Frame_1001
Object-Frame: video.mp4 Frame_1101
Object-Frame: video.mp4 Frame_1201
Object-Frame: video.mp4 Frame_1301
Object-Frame: video.mp4 Frame_1401
Object-Frame: video.mp4 Frame_1501
Object-Frame: video.mp4 Frame_1601
Object-Frame: video.mp4 Frame_1701
Object-Frame: video.mp4 Frame_1801
Object-Frame: video.mp4 Frame_1901
Object-Frame: video.mp4 Frame_2001
Object-Frame: video.mp4 Frame_2101
Object-Frame: video.mp4 Frame_2201
Object-Frame: video.mp4 Frame_2301
Object-Frame: video.mp4 Frame_2401
Object-Frame: video.mp4 Frame_2501
Object-Frame: video.mp4 Frame_2601
```

The other tabs are labeled %1, %3, and %4, and appear to be ssh sessions.

FRAMES FOR ALL THREE PAGES -> video.html + picture.html + bigpicture.html

```
~/Doc/P/PurduePrivate/l/c/l/submittion/3 devCNS *5 !29 ?24 > ./client3 http://127.0.0.1:12345/picture.html
Object-Frame: purdue.jpeg Frame_1
~/Doc/P/PurduePrivate/l/c/l/submittion/3 devCNS *5 !29 ?24 > ./client3 http://127.0.0.1:12345/bigpicture.html
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: bigpicture.jpeg Frame_101
~/Doc/P/PurduePrivate/l/c/l/submittion/3 devCNS *5 !28 ?24 > ./client3 http://127.0.0.1:12345/video.html
http://127.0.0.1:12345/video.html
Object-Frame: video.mp4 Frame_1
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: purdue.jpeg Frame_1
Object-Frame: video.mp4 Frame_101
Object-Frame: bigpicture.jpeg Frame_101
Object-Frame: video.mp4 Frame_201
Object-Frame: video.mp4 Frame_301
Object-Frame: video.mp4 Frame_401
Object-Frame: video.mp4 Frame_501
Object-Frame: video.mp4 Frame_601
Object-Frame: video.mp4 Frame_701
Object-Frame: video.mp4 Frame_801
Object-Frame: video.mp4 Frame_901
Object-Frame: video.mp4 Frame_1001
Object-Frame: video.mp4 Frame_1101
Object-Frame: video.mp4 Frame_1201
Object-Frame: video.mp4 Frame_1301
Object-Frame: video.mp4 Frame_1401
Object-Frame: video.mp4 Frame_1501
Object-Frame: video.mp4 Frame_1601
Object-Frame: video.mp4 Frame_1701
Object-Frame: video.mp4 Frame_1801
Object-Frame: video.mp4 Frame_1901
Object-Frame: video.mp4 Frame_2001
Object-Frame: video.mp4 Frame_2101
Object-Frame: video.mp4 Frame_2201
Object-Frame: video.mp4 Frame_2301
Object-Frame: video.mp4 Frame_2401
Object-Frame: video.mp4 Frame_2501
Object-Frame: video.mp4 Frame_2601
~/Doc/P/PurduePrivate/l/c/l/submittion/3 devCNS *5 !29 ?24 >
```

HW4_Bon Screenshot Screen Shot Screen
...kit-2.pdf 2022-12...0.51 PM 2022-12...2.51 PM 2022-11

```
server3 %1 -zsh %2 -zsh %3
```

```
~/Doc/P/PurduePrivate/l/c/l/s/3 devCNS *5 !25 ?22 > ./server3 1234
The server is ready to receive

message-to-client: 127.0.0.1 58766
HTTP/2.0 200 OK

message-to-client: 127.0.0.1 58766
HTTP/2.0 200 OK

Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: bigpicture.jpeg Frame_101
close-client: 127.0.0.1, 58766
```

```

~/Doc/P/PurduePrivate/l/c/l/s/3 devCNS *5 !25 ?22 > ./client3 http://localhost:1234/picture.html
http://localhost:1234/picture.html
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: bigpicture.jpeg Frame_101
~/Doc/P/PurduePrivate/l/c/l/s/3 devCNS *5 !25 ?22 >

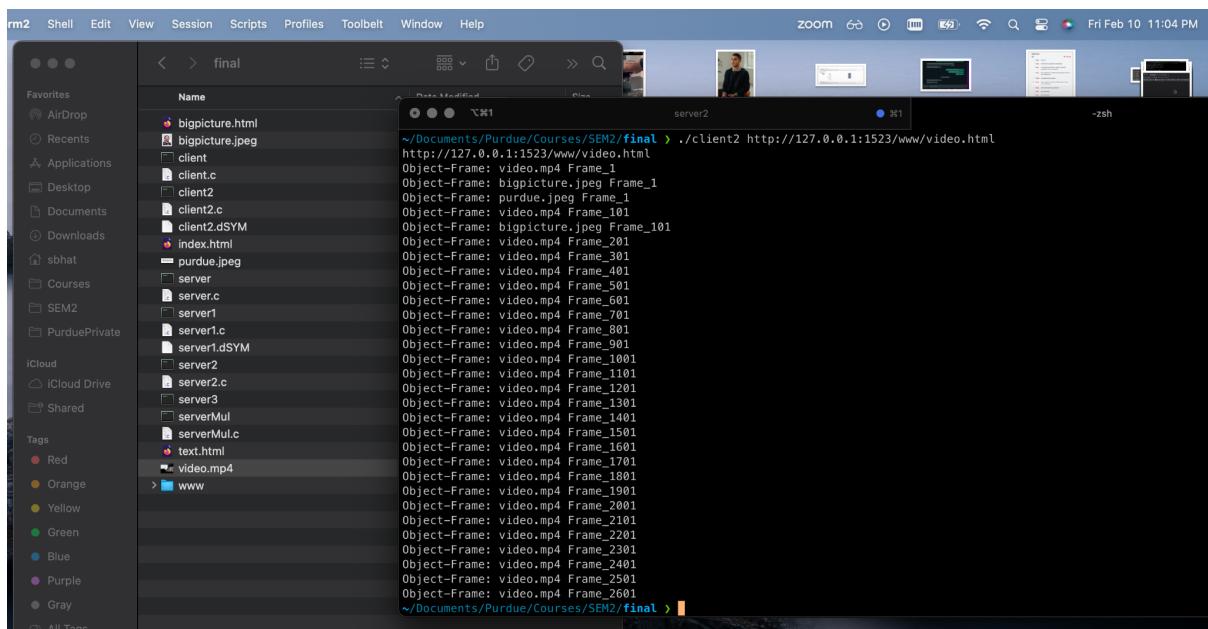
```

FOR A WEB PAGE WITH TWO IMAGES:

```

~/Doc/P/PurduePrivate/l/c/l/s/3 devCNS *5 !27 ?23 > ./client3 http://localhost:1234/video.html
http://localhost:1234/video.html
Object-Frame: bigpicture.jpeg Frame_1
Object-Frame: purdue.jpeg Frame_1
Object-Frame: bigpicture.jpeg Frame_101
~/Doc/P/PurduePrivate/l/c/l/s/3 devCNS *5 !27 ?23 >

```



- 1) Although HTTP2.0 is more efficient it takes a relatively more time due to the backend round-robin and reordering of files while reassembling
- 2) Only one request can be sent at a time and the client must wait for the response before deciding to send another request.
- 3) In HTTP 2.0 server push is more efficiently supported and allows the server to push resources to the client without a request

When run an HTTP/1.1 and HTTP/2 servers on the same host, but with different port numbers, and if then if clients are run on different host, we see similar behavior as if the server and clients were on the same host. The network latency between the client and the server though still play a role in the overall response time, but the underlying protocol used by the server (HTTP/1.1 or HTTP/2) still act as the main influencing factor and impact the performance of the request and response.

HTTP/2 server, we see improved response times due to the multiplexing of requests and responses over a single connection