Lab Exercise 3: DNS & Socket Programming

There are 6 labs during this course. For each student, the 5 best performing labs will contribute to your final lab mark.

Objectives:

- Gain insights into the operation of the DNS protocol
- Dig deep into DNS server organisation
- Get familiar with the socket programming

Prerequisites & Links:

- Week 3 Lectures
- Relevant Parts of Chapter 2 of the textbook (Sections 2.4 on DNS and 2.7 on Socket programming)
- Introduction to Tools of the Trade (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/44919)
- Basic understanding of Linux. A good resource is here (http://www.ee.surrey.ac.uk/Teaching/Unix/) but there are several other resources online.
- Several socket programming resources and sample code for all 3 programming languages are available here (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/45218) .
- dns-wireshark-trace-2 (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/45191)
- index.html (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/45182)

Marks: 10 marks.

- This lab comprises of a number of exercises. Pl note that not all the exercises for this lab are marked. You have to submit a report containing answers to selected exercises only (Exercises 3 & 4)
- Please attend the lab in your allocated lab time slot
- We expect the students to go through as much of the lab exercises as they can at home and come to the lab for clarifying any doubts in procedure/specifications

Deadline:

09:45 am 30/06/2020 **.** You can submit as many times as you wish before the deadline. A later submission will override the earlier submission, so make sure you submit the correct file. Do not leave until the last moment to submit, as there may be technical or communications error and you will not have time to rectify it.

Late Report Submission Penalty:

Late penalty will be applied as follows:

- 1 day after deadline: 20% reduction
- 2 days after deadline: 40% reduction
- 3 or more days late: NOT accepted

Note that the above penalty is applied to your final mark for your lab report. For example, if you submit your lab work report 2 days late and your score on the lab report is 8, then your final mark will be 8 - 3.2 (40% penalty) = 4.8.

Submission Instructions:

Create a PDF document **Lab3.pdf** with answers to Exercise 3. You also need to submit your source code file for Exercise 4. Your server should be named WebServer.c or WebServer.java or WebServer.py. Create a tar archive of all the files called **Lab3.tar** . Submit the archive using give or use WebCMS3 interface. You can submit from a lab machine or ssh into the CSE login server. Instructions to ssh into CSE login servers are here (https://taggi.cse.unsw.edu.au/FAQ/Logging_In_With_SSH/) .

If you are using Python, then please add a comment at the top of your code to indicate the correct version of Python to use for testing.

Original Work Only:

You are strongly encouraged to discuss the questions with other students in your lab. However, each student must submit his or her own work. You may need to refer to the material indicated above (particularly Tools of the Trade document) and also conduct your own research to answer the questions.

OS Compatibility:

We will test your code on CSE machines via the command line interface. Note that CSE machines support the following: **gcc version 8.2, Java 11, Python 2.7 and 3.7.** It is thus imperative that you test your code on CSE machines before submitting. This is particularly important if you write your code on your own machine and use an IDE. You MUST test your code on CSE machines (via command line) before submitting. If we cannot get your code to work on CSE machines, then we can't mark it and unfortunately, you won't receive any marks.

Exercise 1: Explore DNS records (Not marked, No need to submit)

DNS servers use different record types for different purposes. For each type of DNS record, there is an associated type of DNS query. Check the following page (https://en.wikipedia.org/wiki/List_of_DNS_record_types (https://en.wikipedia.org/wiki/List_of_DNS_record_types)) and find out what the following resource record types are used for:

- A
- CNAME
- MX
- NS
- PTR

- SOA

Exercise 2: Tracing DNS with Wireshark (Not marked, No need to submit)

For this particular experiment download the dns trace file: dns-wireshark-trace-2 (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/45191) .

Step 1: Open an xterm and run Wireshark.

Step 2: Load the trace file *dns-wireshark-trace-2* by using the *File* pull down menu, choosing *Open* and selecting the appropriate trace file. This file captures the sequence of messages exchanged between a host and its default DNS server while using the *nslookup* (this tool is similar to the *dig* tool that we used earlier in the lab) utility for obtaining the canonical name (type A record) of www.mit.edu (http://www.mit.edu) . The IP address of the default DNS server for the host is 128.238.29.22. Now filter out all non-DNS packets by typing “dns” (without quotes) in the filter field. Also click the right arrow for DNS in the packet-header detail window. Now focus on the **last two DNS messages** from the 6 listed and answer the following questions:

Question 1: What transport layer protocol is being used by the DNS messages?

Question 2: What is the source and destination port for the DNS query message and the corresponding response?

Question 3: To what IP address is the DNS query message sent? Is this the same as the default local DNS server?

Question 4: How many “questions” are contained in the DNS query message? What “Type” of DNS queries are they? Does the query message also contain any “answers”?

Question 5: Examine the DNS response message. Provide details of the contents of the “Answers”, “Authority” and “Additional Information” fields. What can you infer from these?

Exercise 3: Digging into DNS (marked, include in the lab report)

In order to answer the following questions, you will make DNS queries using some of the query types you have encountered in the above exercise. Some questions require you to make multiple DNS queries. Before you proceed, read the manpage of dig (type man dig in the terminal). Make sure you understand how you can explicitly specify the following:

- nameserver to query
- type of DNS query to make (the default query types are those you saw in exercise 1)
- performing reverse queries

Note: Include the output of all the dig commands you have used in your answers.

To send a query to a particular name server (say x.x.x.x) you should use the following command:

```
dig @x.x.x.x hostname
```

Question 1. What is the IP address of www.eecs.berkeley.edu (https://eecs.berkeley.edu/) . What type of DNS query is sent to get this answer?

Question 2. What is the canonical name for the eecs.berkeley web server? Suggest a reason for having an alias for this server.

Question 3. What can you make of the rest of the response (i.e. the details available in the Authority and Additional sections)?

Question 4. What is the IP address of the local nameserver for your machine?

Question 5. What are the DNS nameservers for the " www.eecs.berkeley.edu (https://eecs.berkeley.edu/) ." domain (note: the domain name is eecs.berkeley.edu (https://eecs.berkeley.edu/) and not www.eecs.berkeley.edu (https://eecs.berkeley.edu/))? Find out their IP addresses? What type of DNS query is sent to obtain this information?

Question 6. What is the DNS name associated with the IP address 111.68.101.54? What type of DNS query is sent to obtain this information?

Question 7. Run dig and query the CSE nameserver (129.94.242.33) for the mail servers for Yahoo! Mail (again the domain name is yahoo.com, not www.yahoo.com (http://www.yahoo.com)). Did you get an authoritative answer? Why? (HINT: Just because a response contains information in the authoritative part of the DNS response message does not mean it came from an authoritative name server. You should examine the flags in the response to determine the answer)

Question 8. Repeat the above (i.e. Question 7) but use one of the nameservers obtained in Question 5. What is the result?

Question 9. Obtain the authoritative answer for the mail servers for Yahoo! mail. What type of DNS query is sent to obtain this information?

Question 10. In this exercise you simulate the iterative DNS query process to find the IP address of your machine (e.g. lyre00.cse.unsw.edu.au). First, find the name server (query type NS) of the "." domain (root domain). Query this nameserver to find the authoritative name server for the "au." domain. Query this second server to find the authoritative nameserver for the "edu.au." domain. Now query this nameserver to find the authoritative nameserver for "unsw.edu.au". Next query the nameserver of unsw.edu.au to find the authoritative name server of cse.unsw.edu.au. Now query the nameserver of cse.unsw.edu.au to find the IP address of your host. How many DNS servers do you have to query to get the authoritative answer?

Question 11. Can one physical machine have several names and/or IP addresses associated with it?

Exercise 4: A Simple Web Server (Marked, submit your code)

Please submit the source code as a separate file. The tutor will run the code to check if the output is as expected.

In this exercise, you will learn the basics of TCP socket programming: how to create a socket, bind it to a specific address and port, as well as send and receive an HTTP packet. You will also learn some basics of HTTP header format. You will develop a web server that handles one HTTP request at a time. Specifically, your web server should do the following:

(i) create a connection socket when contacted by a client (browser).

(ii) receive HTTP request from this connection. Your server should only process GET request. You may assume that only GET requests will be received.

(iii) parse the request to determine the specific file being requested.

(iv) get the requested file from the server's file system.

(v) create an HTTP response message consisting of the requested file preceded by header lines.

(vi) send the response over the TCP connection to the requesting browser.

(vii) If the requested file is not present on the server, the server should send an HTTP "404 Not Found" message back to the client.

(viii) the server should listen in a loop, waiting for next request from the browser.

You don't have to deal with any other error conditions.

Your program should be called WebServer.c or WebServer.java or WebServer.py.

You should write the server so that it executes with the following command:

```
$java WebServer port (for Java)
```

```
$WebServer port (for C)
```

```
$python WebServer.py port (for Python)
```

where port is the port No your Web server will be listening on. Specify a non-standard port No (other than 80 and 8080, and > 1024). We will use this port No in the URL while issuing requests from the web browser.

1. Place a simple HTML file (index.html, without any hyperlinks) in the same directory as the server program. A sample index.html file is provided here (https://webcms3.cse.unsw.edu.au/COMP3331/20T2/resources/45182). Run the server program as indicated above. Open a web browser on the same machine. Type the following url:

```
http://127.0.0.1:port/index.html
```

where *port* is the port number your server is listening on. Note that if you forget to include the port number, the browser will assume the default port of 80 while we are running our web server on a non-standard port. The browser should display the content of index.html.

2. Place multiple image files (.png) in the same directory as the server program. Run the server program as indicated above. Open a web browser on the same machine. Type

```
http://127.0.0.1:port/myimage.png
```

where *port* is the port number the server listens on and *myimage.png* is one of the image files present in the server's directory. The browser should display the image.

3. Now try and request for an object that does not exist in the server directory, e.g.:

```
http://127.0.0.1:port/bio.html
```

The browser should display the 404 error message. Note that showing a specific error message depends on the browser. If you are sending the correct 404 error message and the browser is still not displaying it, you may consider sending a custom error message as text/html in the body of your response.

Note that you cannot use any of the pre-made web servers available in different programming languages.

Resource created 4 months ago (Wednesday 29 April 2020, 03:56:57 PM), last modified 2 months ago (Sunday 28 June 2020, 03:15:47 PM).

Comments

(/COMP3331/20T2/forums/search?forum_choice=resource/44924)
(/COMP3331/20T2/forums/resource/44924)

Add a comment

Tong Zhang (/users/z5235242) 19 days ago (Mon Aug 03 2020 17:38:36 GMT+0800 (China Standard Time))

Hi, the mark of my Lab3 and Lab5 is not shown here. Could you please check if there's anything wrong with it?

Reply Edit Delete

Isura Nirmal (/users/z5262974) 19 days ago (Mon Aug 03 2020 19:09:41 GMT+0800 (China Standard Time))

Could you please contact the tutor ?

Reply

Dongzhu Huang (/users/z5257526) 2 months ago (Mon Jun 29 2020 22:49:24 GMT+0800 (China Standard Time)), last modified 2 months ago (Tue Jun 30 2020 00:50:26 GMT+0800 (China Standard Time))

Hello, is there a Python module we should use to parse the HTTP request or do we have to do it manually?

Edit: I just used `string.split()` to get the file name.

Reply

Ayda Valinezhad Orang (/users/z5195667) 2 months ago (Tue Jun 30 2020 11:26:20 GMT+0800 (China Standard Time))

Dear Huang,

You need to do it manually. It seems fine.

Regards,

Ayda

Reply

Michael Gysel (/users/z5251938) 2 months ago (Mon Jun 29 2020 15:00:35 GMT+0800 (China Standard Time)), last modified 2 months ago (Mon Jun 29 2020 20:39:56 GMT+0800 (China Standard Time))

[deleted]

Reply

Wen Hu (/users/z2260468) 2 months ago (Mon Jun 29 2020 19:07:56 GMT+0800 (China Standard Time))

Hi Michael,

Does your server implementation waiting for the second GET (image) from the browser? A good way to understand the problem is to run wireshark and see the exchanging of messages between the client and the server (and see what is missing/went wrong).

Reply

Michael Gysel (/users/z5251938) 2 months ago (Mon Jun 29 2020 20:41:08 GMT+0800 (China Standard Time))

I ended up figuring it out. Thank you, I will run wireshark moving forward to determine any errors.

Reply

Xuan Truong (/users/z5200695) 2 months ago (Mon Jun 29 2020 13:08:41 GMT+0800 (China Standard Time)), last modified 2 months ago (Mon Jun 29 2020 13:10:33 GMT+0800 (China Standard Time))

Hello,

Can I use TCPServer.c code provided in the lecture resources as a starter code to work on the programming question or I have to write everything from scratch?

Thanks

Reply

Isura Nirmal (/users/z5262974) 2 months ago (Mon Jun 29 2020 13:56:18 GMT+0800 (China Standard Time))

HI Xuan, Yes you can usethis as a skeleton code to start with.

Reply

Swapnil Yadav (/users/z5118150) 2 months ago (Mon Jun 29 2020 10:41:02 GMT+0800 (China Standard Time))

Hi,

For exercise 4 it says you cannot use any pre-made web servers. I was just wondering is using the http.server module in python counted as a pre-made web server?

Thanks!

Reply

Wei Song (/users/z5198433) 2 months ago (Mon Jun 29 2020 11:39:08 GMT+0800 (China Standard Time))

Hi

You cannot use http.server this moduler, you need to implement your own http server which is the purpose we set this exercise.

Reply

Dongzhu Huang (/users/z5257526) 2 months ago (Mon Jun 29 2020 02:22:58 GMT+0800 (China Standard Time))

Hi, for exercise 3.9, I don't really understand the question. I've already ran `dig yahoo.com MX`, so am I supposed to run it again for `mt7.am0.yahoodns.net`? Since the `AUTHORITY` section's DNS type is always `NS`, the 'type of DNS query sent to obtain this information' should be `NS` right?

Reply

Wei Song (/users/z5198433) 2 months ago (Mon Jun 29 2020 11:42:39 GMT+0800 (China Standard Time))

Hi Dongzhu

If you didn't quey the authoritative name server then you cannot get the authoritative answer. We can check it by 'flags', if aa can be seen in flags we think the answer is authoritative otherwise it's not.

Please review my lab2 recording where I explained very clear https://unsw.zoom.us/rec/share/2JFXI5ShrH1JRrf28G7... (https://unsw.zoom.us/rec/share/2JFXI5ShrH1JRrf28G7AXKMiIpb4X6a80SZL8qZczDe5PsfIEiWRGLpvvQRWWw0)

Reply

I Su Park (/users/z5163307) 2 months ago (Mon Jun 29 2020 00:38:55 GMT+0800 (China Standard Time))

Hi, just to confirm, our WebServers are only required to parse/handle the request-lines of the requests received right? I see that a GET request could have a more complex structure, but I just wanted to confirm that our implementations do not have to expect/account for those kinds.

For your reference, below is the format of a valid Request from https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html (https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html) :

```
Request       = Request-Line              ;
                *(( general-header        ;
```

Thanks!

Reply

Wei Song (/users/z5198433) 2 months ago (Mon Jun 29 2020 11:45:58 GMT+0800 (China Standard Time))

Hi

The actually GET request header has a lot of information, but on our exercise you only need to parse which file the brower is requesting as well as the file type, after you confirm this information then you can make the corresponding response.

Reply

I Su Park (/users/z5163307) 2 months ago (Mon Jun 29 2020 11:52:25 GMT+0800 (China Standard Time))

Cool, thanks a lot :)

Reply

Michael Gysel (/users/z5251938) 2 months ago (Sun Jun 28 2020 20:11:47 GMT+0800 (China Standard Time))

Hello,

For question 2, when finding the CNAME for eecs.berkeley, two names come up, per the screenclip below. It appears that one is for a nameserver and one is for a host? Is there a meaning behind these two names, or were they just previously used url's that forward to the shorter eecs.berkeley? Thanks!

Reply

Wen Hu (/users/z2260468) 2 months ago (Mon Jun 29 2020 13:12:38 GMT+0800 (China Standard Time))

Hi Michael,

SOA in the response indicates the "Start of [a zone of] authority record", which specifies authoritative information about a DNS zone (e.g., eecs.berkeley.edu), including the primary name server (part of your answer), the email of the domain administrator (part of your answer), the domain serial number, and several timers relating to refreshing the zone (part of your answer).

Reply

Michael Gysel (/users/z5251938) 2 months ago (Mon Jun 29 2020 15:01:40 GMT+0800 (China Standard Time))

Great, thank you so much!

Reply

Yu Liu (/users/z5261162) 2 months ago (Sun Jun 28 2020 17:58:08 GMT+0800 (China Standard Time))

hi,

Do we have to include the png file and html file in the archive? one tutor tells me do not need to include them but another tells me I should include them

Reply

Xuehua Ma (/users/z5262637) 2 months ago (Sun Jun 28 2020 14:10:04 GMT+0800 (China Standard Time))

hello,

I want to know if the lab3 tar file just need to include the report, code file, index as well as png picture? And for the image file, can I just place one picture called myimage.png or we should place multiple? When we run the code, for the terminal emulation window, does it need to show something like "success" or "fail" or we don't have to? Thank you.

Reply

Mohammadali Yaghoubzadehfard (/users/z5138589) 2 months ago (Sun Jun 28 2020 16:48:20 GMT+0800 (China Standard Time))

There is no need for submitting any image; the image will be provided by the tutor who is marking the submission.
You do not need to output anything in the terminal. You will be marked by the output in the browser

Reply

Xuehua Ma (/users/z5262637) 2 months ago (Sun Jun 28 2020 17:30:28 GMT+0800 (China Standard Time))

for the index html file，I also don't need to include it in my tar file, right? Thank you

Reply

Load More Comments