



دانشگاه فردوسی مشهد

دانشکده مهندسی

گروه مهندسی کامپیوتر

گزارش پروژه درس شبکه های کامپیوتری

عنوان

نگارش

الهه متقین

استاد درس

دکتر یغمایی

بهمن ماه 1399

1. مقدمه

در انجام این پروژه از زبان C استفاده شده است. دلیل انتخاب این زبان برنامه نویسی آشنایی قبلی من با این زبان بوده است. همچنین این موضوع که به کمک زبان C یادگیری پایه ای تر انجام می شود و منابع معرفی شده در کلاس حل پروژه نیز مطابق با زبان C است در انتخاب من موثر بوده است. از طرفی زبان C برای برنامه نویسی در لایه های پایین شبکه نیز توصیه شده است.

منابع، کدها و گزارش پروژه به تدریج در آدرس زیر قرار می گیرد.

https://github.com/elahemotaghin/Network_project.git

2. پینگ(فاز دوم)

اگر از طرفداران بازی های آنلاین هستید و یا پایداری و کیفیت سرویس اینترنت برای شما اهمیت دارد، احتمالاً عبارت "پینگ" را شنیده اید و ممکن است این را نیز بدانید که هرچقدر پینگ کمتر باشد سرویس اینترنت شما از کیفیت بهتری برخوردار است. به همین جهت برای راهنمایی شما در انتخاب مناسب ترین سرویس در این نوشته سعی کردیم همه چیز درباره پینگ سرویس اینترنت را بررسی کنیم.

2.1. پینگ چیست؟

Ping معیاری برای احتساب فاصله ی زمانی اتصال بین دو دستگاه در شبکه است. از طریق میزان پینگ یک سرویس اینترنتی، می توان دریافت که چقدر طول خواهد کشید که یک بسته اطلاعات از سمت دستگاه هوشمند مانند کامپیوتر به سمت سرور رفته و دوباره از آن مسیر برگردد. پینگ درواقع تأخیر زمانی است که در هنگام انجام هر کار Online اتفاق می افتد، از کلیک بر روی لینک، تا استریم کردن یک ویدئو.

اما Ping صرفاً ابزاری برای اندازه گیری «تأخیر» نیست. هر شبکه از تعداد متعددی دستگاه و سرور تشکیل شده است. هرکدام از این دستگاه ها می توانند بسته ای از داده را که آن را Packet می نامند به دستگاه هوشمندی که مقصد است ارسال کند. دستگاه مقصد بسته ی ارسالی که نیاز به پاسخ دارد را دریافت می کند و در جواب یک Packet به دستگاه اول ارسال می کند.

این مراحل برای هر پینگ تکرار می شود. پینگ دستوری است که این امکان را فراهم می کند مدت زمان رفت و برگشت بسته ها را بین دستگاه ها، اندازه بگیرد. استفاده از دستور پینگ برای چک کردن وضعیت سرورها بسیار مفید است. اگر یکی از سرورها down شود و یا کند عمل کند، می تواند مانع اتصال دو دستگاه به یکدیگر شود. در این وضعیت Packet Loss اتفاق خواهد افتاد و همان طور که از نامش مشخص است، Packet Loss زمانی اتفاق می افتد که بسته یا بسته هایی از داده به مقصدشان نمی رسند.

Ping همچنین کمک می‌کند تا بسته‌هایی که از بین رفته‌اند دیده شوند. برای مثال فرض کنید دستور پینگ را اجرا کرده‌اید و با پاسخ‌های مختلف روبرو شده‌اید. مثلاً Request timed out یعنی Packet توسط رایانه مقصد دریافت نشده است، یا پاسخ‌های سرور به دستگاه اول ارسال نشده است. یا ممکن است در طول مسیر ارسال و دریافت از بین رفته باشد.

اجرای دستور پینگ و شناسایی از دست رفتن پکت، برای بررسی مشکلات اینترنت کاربرد دارد. وقتی پکت‌ها از دست بروند، یعنی مشکلی یا در شبکه‌ی اینترنت خانگی و ISP یا در اتصال کلی اینترنت و وبسایت مقصد وجود دارد. برای بررسی عمیق‌تر، بهتر است پکت لاس را برای وبسایت‌های گوناگون بررسی کنید. به عنوان مثال، اگر دسترسی اینترنت به وبسایت‌های خارجی قطع شود، با اجرای پینگ برای وبسایت‌هایی با سرور داخل پکت لاس ندارید؛ اما برای وبسایت‌هایی همچون google.com، با پیغام خطای Request timed out روبرو می‌شوید.

درباره‌ی دستیابی به پینگ صفر نیز با توجه به توضیحات گفته‌شده، می‌توان ناممکن بودن آن را ادعا کرد. به هر حال، ارسال و دریافت پکت حتی به روتر خانگی هم به زمان نیاز دارد و با اتصال کابلی به روتر نیز نمی‌توان به پینگ صفر رسید. شاید زمانی با درهم‌تنیدگی کوانتومی بتوان پینگ را به صفر رساند. اگر زمانی به فناوری استفاده از درهم‌تنیدگی کوانتومی در اینترنت دست پیدا کنیم، شاید پینگ صفر هم ممکن شود. البته در یک صورت می‌توان پینگ کمتر از یک میلی‌ثانیه را تجربه کرد و آن هم با ارسال و دریافت پکت با کامپیوتر اصلی، یعنی اجرای دستور ping localhost، ممکن می‌شود.

2.2. پیاده سازی پینگ در پروژه

در این فاز از پروژه امکان بدست آوردن پینگ با کمک آدرس IP و یا دامنه سایت‌ها بوجود آمده است. به این صورت که کاربر آدرس IP و یا دامنه سایت مورد نظر خود را وارد می‌کند و همچنین اندازه پکت ارسالی و مدت زمان انتظار نیز از کاربر دریافت می‌شود و سپس برنامه شروع به اجرا می‌کند. فرایند اجرا به این صورت است که ابتدا آدرس وارد شده توسط کاربر اعتبار سنجی می‌شود و به کمک DNS Server (در صورت ورود دامنه) آدرس IP متناظر با آن بدست می‌آید. سپس به کمک یک raw socket به سرور مورد نظر پکت‌هایی با سایز تعیین شده توسط کاربر می‌فرستیم و مدت زمان صرف شده تا بازگشت پاسخ را نگه داری می‌کنیم. در نهایت تعداد پکت از سال شده، تعداد پکتی که به مقصد رسیده، مدت زمان صرف شده برای ارسال این پکت‌ها و درصد سالم به مقصد رسیدن بسته‌ها را نمایش می‌دهیم.

برای کامپایل برنامه از دستور زیر استفاده می‌کنیم.

```
gcc -o ping <file name> ---> gcc -o ping ping.c
```

```
gcc -o ping <file name> -l pthread ---> gcc -o ping parallel_ping.c -l pthread
```

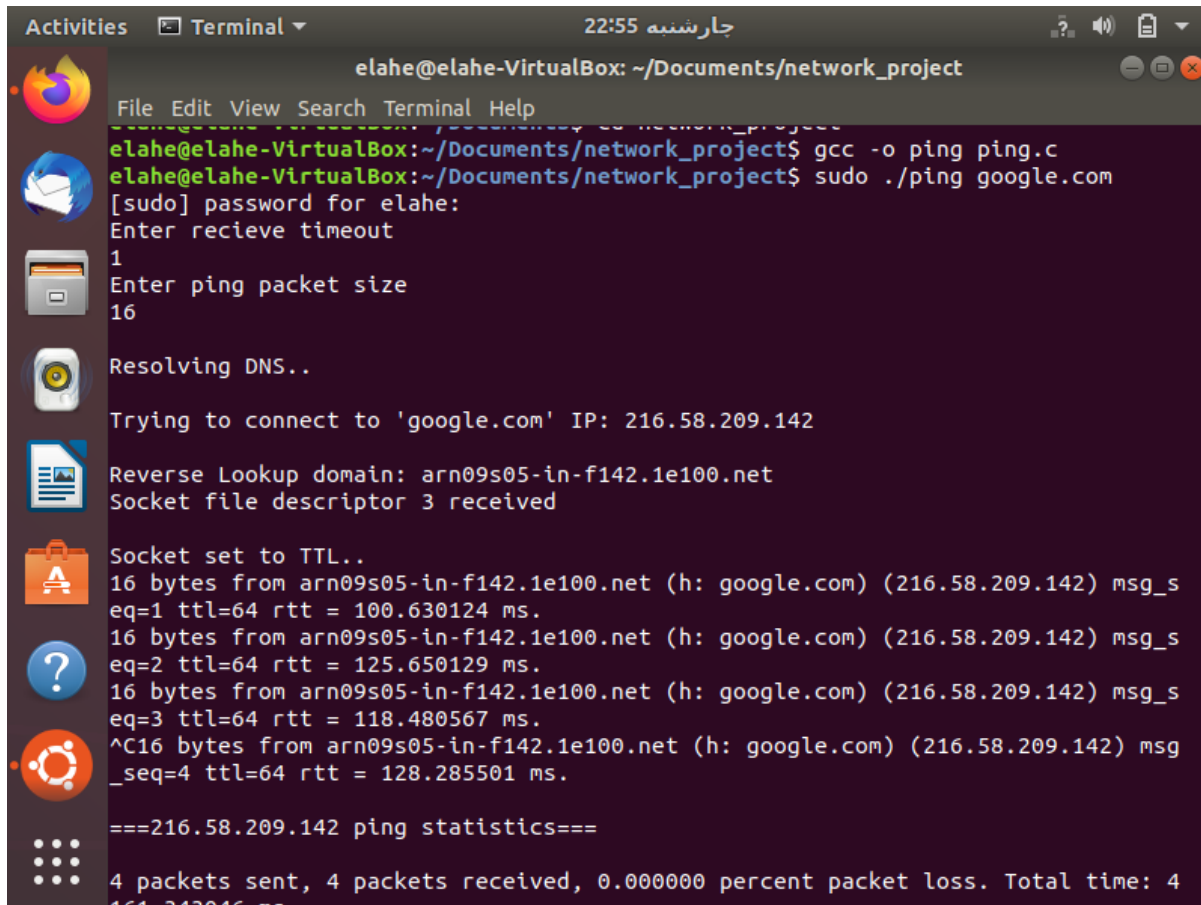
برای اجرای برنامه از دستور زیر استفاده می‌کنیم.

```
Sudo ./ping <ip or domain> --> sudo ./ping google.com
```

```
Sudo ./ping <ip or domain> --> sudo ./ping google.com 8.8.8.8
```

با فشردن **ctrl+c** خروجی را مشاهده می کنیم.

در ادامه نمونه ای از اجرای برنامه آورده شده است.



```
elahe@elahe-VirtualBox: ~/Documents/network_project
File Edit View Search Terminal Help
elahe@elahe-VirtualBox:~/Documents/network_project$ gcc -o ping ping.c
elahe@elahe-VirtualBox:~/Documents/network_project$ sudo ./ping google.com
[sudo] password for elahe:
Enter receive timeout
1
Enter ping packet size
16
Resolving DNS..
Trying to connect to 'google.com' IP: 216.58.209.142
Reverse Lookup domain: arn09s05-in-f142.1e100.net
Socket file descriptor 3 received
Socket set to TTL..
16 bytes from arn09s05-in-f142.1e100.net (h: google.com) (216.58.209.142) msg_s
eq=1 ttl=64 rtt = 100.630124 ms.
16 bytes from arn09s05-in-f142.1e100.net (h: google.com) (216.58.209.142) msg_s
eq=2 ttl=64 rtt = 125.650129 ms.
16 bytes from arn09s05-in-f142.1e100.net (h: google.com) (216.58.209.142) msg_s
eq=3 ttl=64 rtt = 118.480567 ms.
^C16 bytes from arn09s05-in-f142.1e100.net (h: google.com) (216.58.209.142) msg
_seq=4 ttl=64 rtt = 128.285501 ms.
===216.58.209.142 ping statistics===
4 packets sent, 4 packets received, 0.000000 percent packet loss. Total time: 4
161.343046 ms
```

3. Traceroute

اگر شما در هنگام بازکردن یک سایت در اینترنت و یا در هنگام ارتباط با یک سیستم مشکل دارید، با استفاده از دستور **Traceroute** می توانید متوجه شوید که مشکل ارتباط شما در کدام قسمت است. همچنین این دستور می تواند نقشه مسیر ارتباطی از کامپیوتر شما تا وب سروری که وب سایت مورد نظر بر روی آن قرار دارد را نشان دهد. ابزار **Traceroute** یک دستور خط فرمانی می باشد که در ویندوز و سیستم عامل های دیگر قابل استفاده است. مثل دستور **Ping** دستور **Traceroute** هم یک ابزار مفید برای عیب یابی و فهمیدن مشکل در اتصال به شبکه اینترنت است. این دستور نیز شامل **High Latency** و **Packet Loss** می باشد.

زمانیکه شما به یک وب سایت مانند **www.datispro.com** متصل می شوید، ترافیک ارسالی از مسیرها و واسطه های مختلفی عبور می کند تا به مقصد برسد. همچنین شما می توانید با استفاده از این دستور میزان تاخیر بوجود آمده در هر توقف را نیز مشاهده کنید. اگر گاهی اوقات مشکلی در رسیدن به وب سایت مورد نظر دارید اما می دانید که آن وب سایت

بدرستی کار می کند، قطعا در مسیر مشکلی وجود دارد، دستور Traceroute به شما نشان می دهد که مشکل در کدام قسمت از مسیر است.

از منظر فنی دستور Traceroute یک ترتیب متوالی از بسته ها را با استفاده از پروتکل ICMP ارسال می کند. هر کدام از این بسته ها مقداری را بررسی می کنند و دارای یک زمان مشخص می باشند. هرگاه زمان هر بسته ای به صفر برسد، روتر مورد نظر آن را برگشت داده و پیام خطا نمایش داده می شود. با ارسال بسته ها به این شیوه، Traceroute مطمئن می شود که هر روتر در مسیر فعال هست یا نه.

3.1. TTL چیست؟

به زبان ساده TTL یا همان Time To Live، عمر یک پکت را در شبکه، محدود می کند. این مقدار برای هر بسته توسط فرستنده آن مشخص می شود و با رسیدن به هر HOP در شبکه، یک واحد از آن کاسته می شود. یعنی اگر TTL یک بسته 100 باشد، نهایتا از 100 عدد HOP می تواند عبور کند تا به مقصد مورد نظر برسد در غیر ای صورت بسته ارسال شده، دور انداخته می شود (از بین می رود) و یک پیام خطا از طریق ICMP برای فرستنده ارسال می شود. HOP همان روترهایی هستند که بر سر راه این پکت قرار دارند.

3.2. RTT چیست؟

به زبان ساده RTT یا همان Round-Trip Delay، مدت زمانی است که طول می کشد تا یک پکت ارسال شود و پاسخ آن دریافت شود. RTT به ما کمک می کند تا بفهمیم که به طور میانگین چه فاصله زمانی بین یک فرستنده و گیرنده وجود دارد و همچنین به ما کمک می کند تا تاخیرهای موجود در ارتباط بین دو سیستم را ارزیابی کنیم.

3.3. عملکرد Traceroute

Traceroute در ابتدا شروع به ارسال پکت هایی می کند که TTL آن ها 1 می باشد. این پکت ها بعد از اینکه به اولین HOP رسیدند، از مقدار TTL یک واحد کم می کنند، از این رو، مقدار TTL برای این پکت ها به صفر می رسد. همان طور که بالاتر گفتیم، وقتی مقدار TTL به صفر می رسد، اولاً پکت DROP می شود. دوماً یک پیام خطا به سمت فرستنده ارسال می شود. حالا تا به اینجا RTT نشان دهنده فاصله زمانی بین فرستنده (کسی که پکت ها را ارسال کرد و در اصل همان کسی که دستور Traceroute را اجرا کرده است) و اولین HOP می باشد، چرا که TTL را برابر 1 قرار داده بودیم.

در گام بعدی، Traceroute شروع به ارسال پکت هایی می کند که TTL آن ها 2 می باشد. هدف از این کار این است که بسته های ما تنها تا HOP دوم حرکت کنند. پس بسته ها پس از اینکه شروع به حرکت کردند، ابتدا از HOP اول عبور می کنند و TTL آن ها از عدد 2 به عدد 1 کاهش پیدا می کند. سپس با رسیدن پکت ها به HOP دوم، TTL آن ها از یک به صفر کاهش پیدا می کند و در نتیجه یک پیام خطا به سمت فرستنده ارسال می شود. بدین صورت فرستنده متوجه می شود

که وضعیت RTT تا HOP دوم به چه صورت است. Traceroute این کار مجدداً ادامه می‌دهد و هر بار TTL را یک واحد افزایش می‌دهد تا وضعیت ارتباط و RTT هر یک از HOP ها را متوجه شود.

3.4. پیاده سازی traceroute در پروژه

در این فاز از پروژه لازم بوده است تا عملکرد دستور traceroute پیاده سازی شود. برای این کار از پروتکل icmp استفاده شده است. به این صورت که کاربر آدرس IP و یا دامنه سایت مورد نظر خود را وارد می‌کند و همچنین اندازه پکت ارسالی، مدت زمان انتظار، حداکثر تعداد تلاش برای پیدا کردن هر hop و مقدار آغازین و حداکثر TTL نیز از کاربر دریافت می‌شود و سپس برنامه شروع به اجرا می‌کند. فرایند اجرا به این صورت است که ابتدا آدرس وارد شده توسط کاربر اعتبار سنجی می‌شود و به کمک DNS Server (در صورت ورود دامنه) آدرس IP متناظر با آن بدست می‌آید. سپس به کمک یک raw socket به سرور مورد نظر پکتی با سایز تعیین شده توسط کاربر می‌فرستیم و مدت زمان صرف شده تا بازگشت پاسخ را به عنوان خروجی چاپ می‌کنیم. لازم به ذکر است در هر مرحله عمل reverse DNS برای هر hop انجام می‌شود. برای کامپایل برنامه از دستور زیر استفاده می‌شود.

```
gcc -o traceroute traceroute.c
```

همچنین برای اجرای برنامه از دستور زیر استفاده می‌شود.

```
Sudo ./traceroute google.com
```

در ادامه نمونه ای از اجرای برنامه آورده شده است.

```
Activities Terminal 18:18 جارشنه
elahe@elahe-VirtualBox: ~/Documents/network_project
File Edit View Search Terminal Help
DNS lookup failed: could not resolve hostname.
elahe@elahe-VirtualBox:~/Documents/network_project$ sudo ./traceroute instagram
.COM
Enter receive timeout
1
Enter traceroute packet size
16
Enter TTL start point
10
Enter max TTL value
20
Enter max try number value
3
Resolving DNS..
Trying to connect to 'instagram.com' IP: 54.225.149.138
Reverse Lookup domain: ec2-54-225-149-138.compute-1.amazonaws.com
Socket file descriptor 3 received

*** TTL is 10 ***
Socket set to TTL..
16 bytes from ec2-54-225-149-138.compute-1.amazonaws.com (h: ec2-54-225-149-138
.compute-1.amazonaws.com) (54.225.149.138) msg_seq=1 ttl=10 rtt = 58.640439 ms
and number of try is 1.
saddr = 2, 10.138.98.6: 0

===54.225.149.138 traceroute statistics===
```

```
Activities Terminal 18:19 جارشنه
elahe@elahe-VirtualBox: ~/Documents/network_project
File Edit View Search Terminal Help
1 packets sent, 1 packets received, 0.000000 percent packet loss. Total time: 1
330.388590 ms.

*** TTL is 19 ***
Socket set to TTL..
16 bytes from ec2-54-225-149-138.compute-1.amazonaws.com (h: ec2-54-225-149-138
.compute-1.amazonaws.com) (54.225.149.138) msg_seq=1 ttl=19 rtt = 330.776867 ms
and number of try is 1.
saddr = 2, 213.248.68.39: 0

===54.225.149.138 traceroute statistics===
1 packets sent, 1 packets received, 0.000000 percent packet loss. Total time: 1
331.305310 ms.

*** TTL is 20 ***
Socket set to TTL..
16 bytes from ec2-54-225-149-138.compute-1.amazonaws.com (h: ec2-54-225-149-138
.compute-1.amazonaws.com) (54.225.149.138) msg_seq=1 ttl=20 rtt = 1013.032527 m
s and number of try is 1.
saddr = 2, 213.248.68.39: 0

===54.225.149.138 traceroute statistics===
1 packets sent, 1 packets received, 0.000000 percent packet loss. Total time: 2
014.129337 ms.
```

Activities Terminal 18:19 جارشنه

elahe@elahe-VirtualBox: ~/Documents/network_project

File Edit View Search Terminal Help

```
*** TLL is 16 ***
Socket set to TTL..
16 bytes from ec2-54-225-149-138.compute-1.amazonaws.com (h: ec2-54-225-149-138
.compute-1.amazonaws.com) (54.225.149.138) msg_seq=1 ttl=16 rtt = 267.663240 ms
and number of try is 1.
saddr = 2, 62.115.122.138: 0

===54.225.149.138 traceroute statistics===

1 packets sent, 1 packets received, 0.000000 percent packet loss. Total time: 1
268.750069 ms.

*** TLL is 17 ***
Socket set to TTL..
16 bytes from ec2-54-225-149-138.compute-1.amazonaws.com (h: ec2-54-225-149-138
.compute-1.amazonaws.com) (54.225.149.138) msg_seq=1 ttl=17 rtt = 280.095057 ms
and number of try is 1.
saddr = 2, 62.115.122.159: 0

===54.225.149.138 traceroute statistics===

1 packets sent, 1 packets received, 0.000000 percent packet loss. Total time: 1
281.298557 ms.

*** TLL is 18 ***
```


4. مراجع

- [1] <https://www.geeksforgeeks.org/ping-in-c/>
- [2] <https://vigiato.net/p/105454>
- [3] <https://www.zoomit.ir/howto/347187-what-is-ping-is-zero-ping-possible/>