نام و نام خانوادگی: محمد کربلایی شعبانی

شماره دانشجویی: ۹۹۲۲۲۰۸۵

شماره تمرین:۴ **تاریخ تحویل: ۱۶ خرداد ۱۴۰۳**

عنوان تمرین: UDP Pinger

چکیده

این پیاده‌سازی شامل ایجاد یک مدل ارتباطی client-server با استفاده از UDP[1] است. تمرکز بر درک رفتار UDP است که یک پروتکل غیرقابل اعتماد و بدون اتصال است و اغلب برای سناریوهایی که سرعت مهم است و از دست دادن گاه‌به‌گاه بسته‌ها قابل قبول است، استفاده می‌شود. سرور به گونه‌ای طراحی شده است که بسته‌ها را به صورت تصادفی drop کند تا غیرقابل اعتمادی شبکه را شبیه‌سازی کند. مشتری ده پیام پینگ به سرور ارسال می‌کند، زمان رفت و برگشت RTT[2] هر پیام را اندازه‌گیری می‌کند و در مواردی که پاسخ‌ها در بازه زمانی یک ثانیه‌ای دریافت نمی‌شوند، پاسخ می‌دهد.

پیاده‌سازی client شامل ارسال بسته‌های UDP به سرور، انتظار برای دریافت پاسخ‌ها و محاسبه RTT برای بسته‌های دریافت شده است. اگر پاسخی در بازه زمانی یک ثانیه ای دریافت نشود، کلاینت فرض می‌کند که بسته از دست رفته است و تعداد بسته‌های lost را افزایش می‌دهد. نتایج شامل چاپ هر پاسخ سرور، RTT برای پینگ‌های موفق و خلاصه‌ای از حداقل، حداکثر و میانگین RTT ها است. علاوه بر این، کلاینت نرخ از دست دادن بسته‌ها را به عنوان درصدی محاسبه و نمایش می‌دهد.

این پیاده سازی چالش‌ها و ویژگی‌های عملی استفاده از UDP برای ارتباطات، از جمله مدیریت از دست دادن بسته‌ها، اندازه‌گیری عملکرد شبکه از طریق RTT و برخورد با زمان‌بندی‌ها را نشان می‌دهد. پیاده‌سازی پیشرو بینشی درباره‌ی نحوه‌ی عملکرد پروتکل‌های شبکه در سناریوهای واقعی که یکپارچگی داده‌ها و زمان‌بندی عوامل حیاتی هستند، ارائه می‌دهد.

پیاده سازی client

این برنامه، یک پیاده‌سازی از برنامه ping با استفاده از پروتکل UDP است. این برنامه ۱۰ پیغام ping را به یک سروری که روی localhost در پورت ۱۲۰۰۰ اجرا می‌شود ارسال می‌کند. کلاینت زمان رفت و برگشت (RTT) برای هر ping را اندازه‌گیری می‌کند و آمارهایی از جمله حداقل، حداکثر و میانگین RTT و نرخ از دست رفتن بسته ها را محاسبه می‌کند.

در ادامه مراحل توسعه داده شده در برنامه را تشریح می کنیم:

[1] User Datagram Protocol
[2] Round-Trip Time

نام و نام خانوادگی: محمد کربلایی شعبانی
شماره دانشجویی: ۹۹۲۲۲۰۸۵
شماره تمرین:۴ تاریخ تحویل: ۱۶ خرداد ۱۴۰۳
عنوان تمرین: UDP Pinger

---

۱. import کردن ماژول‌ها:

```
import time
import socket
```

کد با import ماژول‌های لازم "time" و "socket" شروع می‌شود.

"time" - برای اندازه‌گیری فواصل زمانی استفاده می‌شود.

"socket" - دسترسی به رابط سوکت را فراهم می‌کند.

۲. آدرس سرور:

```
server_address = ('localhost', 12000)
```

آدرس سرور به عنوان localhost:۱۲۰۰ تعریف میشود.

۳. مقداردهی اولیه سوکت:

```
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

یک سوکت UDP ایجاد می‌کند.

```
client_socket.settimeout(1)
```

محدودیت یک ثانیه ای برای عملیات سوکت تنظیم می‌کند. این timeout در ادامه برای تشخصی packet loss استفاده شده است.

۴. مقداردهی اولیه متغیرها:

```
rtt_list = []
packet_loss_count = 0
```

"rtt_list" - لیستی برای ذخیره زمان رفت و برگشت برایping های موفق.

"packet_loss_count" - شمارنده برای پیگیری تعداد بسته‌های گمشده.

نام و نام خانوادگی: محمد کربلایی شعبانی
شماره دانشجویی: ۹۹۲۲۲۰۸۵
شماره تمرین:۴ تاریخ تحویل: ۱۶ خرداد ۱۴۰۳
عنوان تمرین: UDP Pinger

---

۵. ارسال و دریافت پیغام‌های Ping

```
for i in range(10):
    send_time = time.time()
    message = f'ping {i+1}'

    try:
        client_socket.sendto(message.encode(), server_address)

        response, server = client_socket.recvfrom(1024)
        receive_time = time.time()

        rtt = receive_time - send_time
        rtt_list.append(rtt)

        print(f'Ping {i+1} response: {response.decode()}')
        print(f'RTT: {rtt:.4f} seconds')
    except socket.timeout:
        print(f'Ping {i+1} request timed out')
        packet_loss_count += 1
```

- یک حلقه ۱۰ بار برای "i" در "range(۱۰)" اجرا می‌شود.
- برای هر تکرار:
  - "send_time": زمان را قبل از ارسال پیام ping ثبت می‌کند.
  - "message": یک پیام ping شامل شماره دنباله ایجاد می‌کند.
  - سعی می‌شود پیام را به سرور با استفاده از "client_socket.sendto()" ارسال کند.
  - در صورت موفقیت، پاسخ را دریافت کرده و RTT را محاسبه می‌کند.
  - اگر استثناء زمان انقضا اتفاق بیافتد، آن را پردازش می‌کند و بیان می‌کند که یک بسته lost شده است.

نام و نام خانوادگی: محمد کربلایی شعبانی
شماره دانشجویی: ۹۹۲۲۲۰۸۵
شماره تمرین:۴ تاریخ تحویل: ۱۶ خرداد ۱۴۰۳
عنوان تمرین: UDP Pinger

---

۶ .محاسبه آمار RTT

```
if rtt_list:
    min_rtt = min(rtt_list)
    max_rtt = max(rtt_list)
    avg_rtt = sum(rtt_list) / len(rtt_list)
    print(f'\nMinimum RTT: {min_rtt:.4f} seconds')
    print(f'Maximum RTT: {max_rtt:.4f} seconds')
    print(f'Average RTT: {avg_rtt:.4f} seconds')
else:
    print('\nNo successful pings to calculate RTTs.')
```

- اگر ping های موفقی (“if rtt_list:”) وجود داشته باشند:
- با استفاده از توابع”min()“ ، ”max()“ و”sum()“ ، حداقل، حداکثر و میانگین RTT را محاسبه می‌کند.
- اگر هیچ ping موفقیت آمیزی نباشد، پیامی را چاپ می‌کند که نشان می‌دهد هیچ ping موفقیت آمیزی وجود نداشته است.

۷. محاسبه نرخ از دست رفتن بسته:

```
packet_loss_rate = (packet_loss_count / 10) * 100
print(f'Packet loss rate: {packet_loss_rate:.2f}%')
```

- درصد نرخ از دست رفتن بسته را براساس تعداد بسته‌های گمشده (“packet_loss_count”) از مجموع تعداد بسته‌های ارسال شده (۱۰) محاسبه می‌کند.

۸. بستن سوکت:

```
client_socket.close()
```

- در اینجا سوکت کلاینت بسته می‌شود.

نام و نام خانوادگی: محمد کربلایی شعبانی
شماره دانشجویی: ۹۹۲۲۲۰۸۵
شماره تمرین:۴ تاریخ تحویل: ۱۶ خرداد ۱۴۰۳
عنوان تمرین: UDP Pinger

برسی یک نمونه اجرای client

در این بخش برنامه ی کلاینت را اجرا میکنیم و خروجی برنامه را برسی میکنیم در دو اجرای متفاوت.

```
Ping 1 request timed out
Ping 2 response: PING 2
RTT: 0.0006 seconds
Ping 3 request timed out
Ping 4 response: PING 4
RTT: 0.0007 seconds
Ping 5 response: PING 5
RTT: 0.0003 seconds
Ping 6 request timed out
Ping 7 response: PING 7
RTT: 0.0004 seconds
Ping 8 response: PING 8
RTT: 0.0002 seconds
Ping 9 response: PING 9
RTT: 0.0002 seconds
Ping 10 response: PING 10
RTT: 0.0002 seconds

Minimum RTT: 0.0002 seconds
Maximum RTT: 0.0007 seconds
Average RTT: 0.0004 seconds
Packet loss rate: 30.00%
```

در بخش بالا نتیجه اجرا برای دفعه اول را مشاهده میکنید که ده packet به سرور ارسال کرده و جواب های متفاوتی دریافت کرده است. به عنوان مثال پینگ اول به timeout خورده و پاسخی از سرور در بازه یک ثانیه دریافت نکرده است.
در مقابل پینگ دوم پاسخ دریافت کرده است. میدانیم که سرور اینطور طراحی شده است که هر پیامی بهش ارسال بشود آن را تبدیل به حروف بزرگ میکند و به کلاینت برمیگرداند. در پینگ دوم عبارت Ping 2 به سرور ارسال شده و پاسخ PING 2 از

سرور دریافت شده است. در ادامه RTT پرینت شده است که نشان میدهد که کل فرآیند پینگ چقدر طول کشیده است.
دیگر پینگ ها هم تا شماره ده به همین صورت انجام و نتیجه آنها چاپ شده است.

در نهایت بیشترین و کمتر مقدار RTT طی انجام این ده پینگ پرینت شده است و در ادامه میانگینRTT محاسبه و پرینت کرده است. نرخ packet loss هم به درصد در انتها چاپ شده است. در اینجا ۳۰ درصد از ده پکتی که ارسال شده است گم شده و در بازه یک ثانیه ای از سمت سرور پاسخی دریافت نکرده است.

```
Ping 1 request timed out
Ping 2 request timed out
Ping 3 request timed out
Ping 4 response: PING 4
RTT: 0.0008 seconds
Ping 5 request timed out
Ping 6 response: PING 6
RTT: 0.0004 seconds
Ping 7 response: PING 7
RTT: 0.0003 seconds
Ping 8 response: PING 8
RTT: 0.0002 seconds
Ping 9 request timed out
Ping 10 request timed out

Minimum RTT: 0.0002 seconds
Maximum RTT: 0.0008 seconds
Average RTT: 0.0004 seconds
Packet loss rate: 60.00%
```

در اینجا به صورت خلاصه یک اجرای دیگر را برسی میکنیم. در مورد پینگ ها صحبت نمیکنیم چون محتوای یکسانی با دفعه قبل دارند از لحاظ اطلاعاتی که به ما میدهند اما در نهایت مورد مهم تفاوت میانگین ها و درصد packet loss است که در این اجرا با دفعه اول فرق میکند چرا که پینگ های بیشتری بی پاسخ مانده اند.

نام و نام خانوادگی: محمد کربلایی شعبانی
شماره دانشجویی: ۹۹۲۲۲۰۸۵
شماره تمرین:۴ **تاریخ تحویل: ۱۶ خرداد ۱۴۰۳**

عنوان تمرین: UDP Pinger

نتیجه گیری

این تمرین به ما یاد داد که از طریق اجرای آن، به درک عملی ارتباط شبکه ای با استفاده از سوکت های UDP دست یابیم. ما یاد گرفتیم چگونه داده ها را بین یک کلاینت و یک سرور ارسال و دریافت کنیم، زمان رفت و برگشت (RTT) برای هر انتقال را اندازه‌گیری کنیم، و آمارهایی مانند حداقل، حداکثر و میانگین RTT را، همچنین نرخ packet loss را محاسبه کنیم. این تمرین به ما کمک کرد تا اهمیت پروتکل های شبکه کارآمد و مکانیسم های کنترل خطا در اطمینان از ارتباطات قابل اعتماد و پاسخگو در شبکه ها را بیشتر درک کنیم. همچنین نحوه عملکرد UDP را متوجه شدیم که هیچ تضمینی برای دریافت همه ی packet ها ارایه نمیدهد.