



آزمایشگاه سخت افزار

گزارش پروژه

نیمسال اول ۱۴۰۱-۰۲

موضوع پروژه:

ارتباط رایانه و برد رزبری پای با ماژول های 4/4.5/5G و مقایسه آنها با هم

شماره گروه: ۴

اعضای گروه:

عرشیا اخوان 97110422

مهدی صادق شبیری 97110144

غزل شناور 97101897

۱ سرور

قسمت سرور کاملاً زده شده و الان از TCP, UDP, Quic و HTTP پشتیبانی می‌کند و برای هر کدام Upload, Download, و Latency محاسبه می‌شود.

TCP

کدهای مورد نظر در فایل tcp.py هستند.

```
def start(port: int):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.bind(('0.0.0.0', port))
        s.listen()
        conn, addr = s.accept()
        logger.info("New connecting %s", addr)
        return conn
```

تابع start یک سوکت از نوع tcp می‌سازد و در ورودی port مربوط به سوکت را می‌گیرد.

```
def start_throughput(port: int, period: int, upload: bool) -> List[int]:
    conn = start(port)
    logger.info("Start tcp throughput with period: %d upload: %s", period, upload)
    now = datetime.datetime.now()
    recv_bytes = [0] * period
    while True:
        total = (datetime.datetime.now() - now).total_seconds()
        if total >= period:
            break
        if upload:
            data = conn.recv(1024)
            recv_bytes[int(total)] += len(data)
        else:
            conn.sendall(b' '*512)
    if upload:
        logger.info("Result: %s", recv_bytes)
    conn.close()
    return recv_bytes
```

تابع start_throughput برای راه اندازی سرور upload, download است و در ورودی با period مشخص می‌شود که در چند ثانیه تست را انجام دهد و با بولین upload مشخص می‌شود هدف سنجش آپلود است یا دانلود.

```
def start_latency(port: int, number_of_packets: int) -> List[int]:
    conn = start(port)
    logger.info("Start tcp latency with %d packets", number_of_packets)
    result = []
    for i in range(number_of_packets):
        now = datetime.datetime.now()
        conn.sendall(b'${}')
        conn.recv(1)
        result.append(int((datetime.datetime.now() - now).total_seconds() * 1_000))
    conn.close()
    logger.info("Result: %s", result)
    return result
```

تابع `start_latency` برای شروع سرور محاسبه تاخیر است. در ورودی `number_of_packets` برابر تعداد پکتی است که می‌فرستد تا latency را حساب کند. این بخش به این صورت عمل می‌کند که سرور یک پکت به کلاینت می‌فرستد و کلاینت لحظه‌ای که پکت را دریافت کرد یک پکت به سرور می‌فرستد.

UDP

با توجه به اینکه کد در زبان پایتون زده شده و سوکت tcp, udp شبیه هم هستند کدهای مربوط به قسمت udp نیز شباهت زیادی با tcp دارند.

```

def start(port: int):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.bind(('0.0.0.0', port))
    while True:
        data, addr = s.recvfrom(1)
        if data is not None:
            break
    return s, addr

def start_throughput(port: int, period: int, upload: bool):
    conn, addr = start(port)
    logger.info("Start udp throughput with period: %d", period)
    now = datetime.datetime.now()
    recv_bytes = [0] * period
    while True:
        total = (datetime.datetime.now() - now).total_seconds()
        if total >= period:
            break
        if upload:
            data, _ = conn.recvfrom(1024)
            recv_bytes[int(total)] += len(data)
        else:
            conn.sendto(b'*'*512, addr)
    if upload:
        logger.info("Result: %s", recv_bytes)
    conn.close()
    return recv_bytes

def start_latency(port: int, number_of_packets: int) -> List[int]:
    conn, addr = start(port)
    logger.info("Start udp latency with %d packets", number_of_packets)
    result = []
    for i in range(number_of_packets):
        now = datetime.datetime.now()
        conn.sendto(b'$', addr)
        conn.recv(1)
        result.append(int((datetime.datetime.now() - now).total_seconds() * 1_000))
    conn.close()
    logger.info("Result: %s", result)
    return result

```

به طور مشابه ۳ بخش tcp هر بخش کار گفته شده را انجام می‌دهد.

HTTP

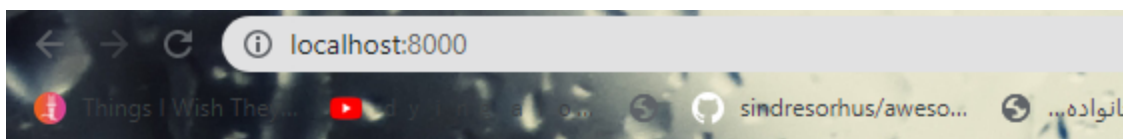
این بخش با استفاده از ماژول HttpServer و لایبری uploadserver انجام می‌شود. با دستور

```
python -m uploadserver
```

یک سرور http راه اندازی می‌شود که توانایی دانلود و آپلود در آن وجود دارد و برای تست latency صفحه اصلی را get می‌کنیم. برای مثال برای آپلود فایل می‌شود دستور زیر را زد:

```
curl -X POST http://127.0.0.1:8000/upload -F
```

```
'files=@/home/smss/Downloads/108585262.jpeg'
```



Directory listing for /

- [3D Objects/](#)
- [ansel/](#)
- [AppData/](#)
- [Application Data/](#)
- [Contacts/](#)
- [Cookies/](#)
- [Desktop/](#)
- [Documents/](#)
- [Downloads/](#)
- [Favorites/](#)

این یک تصویر از http server است.

Quic

فولدر Quic شامل کد سرور Quic است. از لایبری aioquic برای راه اندازی سرور استفاده شده است. با دستور

```
python http3_server.py --certificatessl_cert.pem --private-key ssl_key.pem
```

می‌شود سرور را راه اندازی کرد و از مسیر {size:int}/ می‌شود هر تعداد پکت که می‌خواهیم دانلود کنیم.

۲ سخت افزار

با وجود تلاش‌های بسیار تیم، و نظر به محدودیت‌های اعمال شده بر روی سامانه مخابرات کشور، پروژه‌ی این گروه با شکست مواجه شد. طبق بررسی‌های انجام شده توسط تیم توسعه پروژه، متأسفانه امکان اجرای طرح تحقیقاتی ارائه شده وجود ندارد.

به طور دقیقتر ماژول Sim7000 در ایران به درستی کار نمی‌کند و حتی خطاهای درستی نیز نمی‌دهد که بشود آن‌ها را رفع کرد.

موقعی که سیمکارت داخل ماژول قرار می‌گیرد در شبکه Register نمی‌شود و این باعث می‌شود امکان استفاده از آن وجود نداشته باشد.