



بهار ۱۴۰۱

CE-40443

شبکه‌های کامپیوتری: تمرین دوم

مدرس: مهدی جعفری

۱ سوال تئوری اول

Wireshark ابزاری متن‌باز برای تحلیل و آنالیز ترافیک و پروتکل‌های شبکه است. Wireshark ترافیک شبکه را Capture کرده و می‌توانیم درون ترافیک‌های در حال Capture شدن، به جست و جو بپردازیم. همچنین جریان داده‌ی در جال عبور از واسط شبکه را دریافت کرده و در قالبی مشخص و قابل فهم برای ما به نمایش می‌گذارد. از این برنامه برای اهداف مختلفی مانند تحلیل شبکه، عملیات امنیتی، اشکال‌یابی، اعمال مهندسی معکوس بر روی پروتکل‌ها و فهمیدن جزئیات درون آن‌ها و... می‌توان استفاده کرد. این ابزار به صورت رایگان و متن‌باز در دسترس است و می‌توانید آخرین نسخه آن را برای پلتفرم‌های گوناگون از سایت <https://www.wireshark.org> دانلود کنید. دقت کنید که در این تمرین، اتصال خود به هرگونه VPN و پراکسی را قطع کنید. همچنین ترجیحا کانکشن دیگری در سیستم‌تان برقرار نشده باشد.

۱.۱ بخش اول

نرم افزار wireshark را در حالت capture قرار دهید و با مرورگر دلخواه خودتان به سایت ce.sharif.edu مراجعه کنید. پس از بارگذاری کامل سایت برای شما به نرم افزار بازگشته و ضبط اطلاعات را متوقف کنید.

- پروتکل‌های مربوط به پیام‌های رد و بدل شده را بررسی کنید. می‌توانید از امکانات آماری wireshark برای این کار استفاده کنید.
- بسته‌های DNS رد و بدل شده که توسط ابزار wireshark قابل مشاهده هستند را بررسی کرده و مراحل‌ی که برای به دست آوردن آدرس IP مقصد طی شده عنوان کرده و توضیح دهید.

۲.۱ بخش دوم

با مرورگر دلخواه خود به سایت <https://www.speedtest.net> بروید. نوع تست را بر روی single-connection قرار دهید. سپس wireshark را روی حالت capture قرار داده و تست سرعت را شروع کنید.

- در طول تست چه آی‌پی‌هایی را مشاهده می‌کنید و هر کدام از آن‌ها مربوط به کجا هستند؟
- از کدام پروتکل استفاده می‌کند؟ چرا؟
- ~~ما می‌خواهیم throughput delay و bandwidth در یک اتصال بین دو node شبکه را توضیح دهید.~~
- با استفاده از بسته‌هایی که توسط wireshark ضبط شده‌اند، throughput delay و bandwidth برنامه مرورگر به سرور را بدست آورید.

۲ سوال تئوری دوم

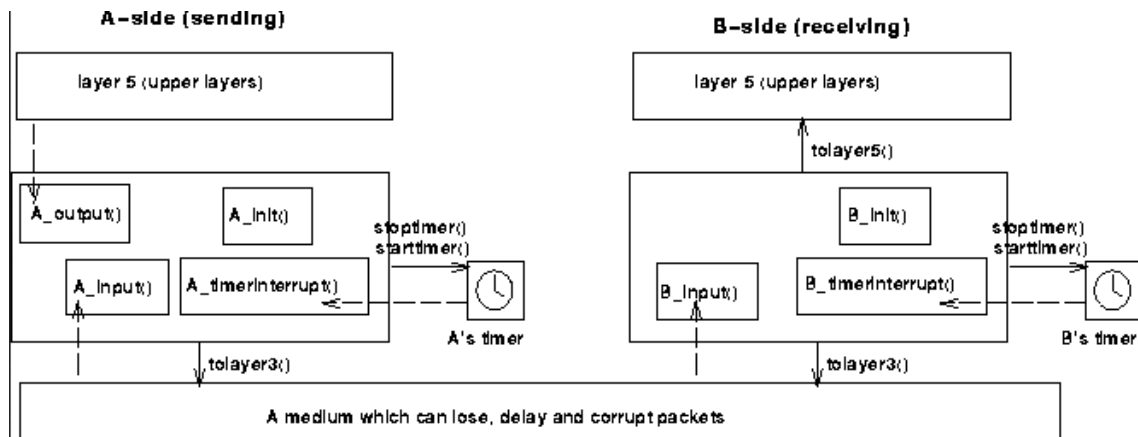
پروتکل RTMP یک پروتکل شبکه است که برای استریم کردن صدا، تصویر و داده ها از طریق اینترنت است. هدف از این تمرین آشنایی شما با این پروتکل است.

- در مورد این پروتکل تحقیق کنید و توصیفی سطح بالا از این پروتکل ارائه کنید.
- از این پروتکل در کدام لایه شبکه استفاده میشود؟
- ساختار بسته های تحت این پروتکل به چه صورت است؟ اجزای مختلف هدر بسته ها را توضیح دهید.
- نحوه handshake را در این پروتکل توضیح دهید.

۳ سوال عملی

۱.۳ الف

در این تمرین شما پروتکل Alternating Bit با همان rdt3.0 را پیاده سازی خواهید کرد. شما باید این کار را در محیط شبیه سازی شده ی شبکه که کد آن در اختیارتان قرار خواهد گرفت، انجام دهید. توابعی که شما می نویسد برای دو دستگاه، یکی به عنوان ارسال کننده پیام و دیگری به عنوان دریافت کننده پیام است. نام دستگاه ارسال کننده node A و نام دستگاه دریافت کننده node B است. کد شما تنها لازم است ارسال داده ی یک طرفه یا unidirectional را پشتیبانی کند، البته node B لازم است تا پکت های Ack را بتواند برای node A بفرستد. همانطور که گفتیم کد شما در یک محیط شبیه سازی شده اجرا می شود. این محیط لایه های ۳ و ۵ شبکه را به صورت مصنوعی ایجاد می کند و توابع شما که در لایه ی ۴ قرار دارند، توسط این ۲ لایه صدا زده می شوند. توابعی که در نمودار زیر و در قسمت وسط می بینید توابعی هستند که شما باید پیاده سازی کنید و نحوه کال شدن آن ها را نیز از روی نمودار می توان متوجه شد. به عنوان مثال لایه ی ۵ یعنی اپلیکیشن هرگاه بخواهد پیامی را ارسال کند، تابع A_output را کال می کند و همچنین هرگاه در لایه ۴ بخواهیم پکت را به لایه ۳ و به مقصد بفرستیم، تابع tolayer3 را کال می کنیم. همچنین تابع تایمری قرار داده شده است تا بتوانید از آن در پیاده سازی پروتکل استفاده کنید.



کد ۲ ساختار اصلی دارد. ساختار msg که واحد داده ای است که بین لایه ۵ و لایه ی شما یعنی لایه ۴ جابجا می شود

```
struct msg {
    char data[20];
};
```

در نتیجه در بخش فرستنده، داده ها را به صورت تکه های ۲۰ بایتی از لایه ۵ دریافت می کنید و در قسمت گیرنده نیز باید به صورت تکه های ۲۰ بایتی داده ها را به لایه ی بالاتر یعنی لایه ۵ تحویل دهید.

ساختار pkt که واحد داده‌ای است که بین لایه ۳ و لایه ۴ جابجا می‌شود و حکم packet ها را دارد.

```
struct pkt {
    int seqnum;
    int acknum;
    int checksum;
    char payload[20];
};
```

و این توابع شما هستند که باید مقادیر موجود در این ساختار را پر کنند. به عنوان مثال payload از همان داده‌های پر می‌شود که از لایه ۵ به لایه ۴ رسیده است. بقیه فیلدها هم برای پیاده سازی پروتکل نیاز است تا به درستی پر شوند.

اما توابعی که شما باید پیاده‌سازی کنید به شرح زیر هستند:

• A_output(message)

که message یک ساختار از نوع msg است. این تابع در سمت node A هرگاه لایه ۵ پیامی برای ارسال داشته باشد کال می‌شود. این وظیفه‌ی پروتکل شماست که تضمین کند داده‌ها، به همان ترتیب که وارد شدند و به درستی به لایه‌ی ۵ در سمت B node می‌رسد. این تابع اگر پکت برای ارسال (transmission) قبول شد باید ۱ و در غیر این صورت ۱- برگرداند.

• A_input(packet)

که packet یک ساختار از نوع pkt است. این تابع وقتی کال می‌شود که از سمت node B یک پکت به node A با کمک تابع tolayer3 ارسال شده باشد. ورودی packet نیز همان پکتی است که از سمت node B ارسال شده. توجه کنید که این پکت می‌تواند خراب (corrupted) باشد.

• A_timerinterrupt()

این تابع وقتی کال می‌شود که یک تایمر سمت node A زمانش تمام شده باشد. از این تابع می‌توانید برای ارسال دوباره‌ی پکت‌ها (retransmission) استفاده کنید. همچنین توابع starttimer و stoptimer برای کار با تایمر هستند.

• A_init()

این تابع در سمت node A قبل از تمام توابع دیگر معرفی شده در بالا کال می‌شود و می‌توانید مقدمات اولیه پروتکل خود را در آن انجام دهید.

• B_input(packet)

دقیقاً مانند همان تابع A_input با این تفاوت که در اینجا ارسال کننده‌ی پکت node A و دریافت کننده‌ی آن node B است.

• B_init()

مانند همان A_init اما برای node B

توابع کمکی زیر نیز برای شما قرار داده شده‌اند و با کامنت‌هایی که در کنار آن‌ها درون کد نوشته شده است، می‌توانید کاری که انجام می‌دهند را متوجه شوید. (توابع شما باید این توابع را برای پیاده‌سازی پروتکل، استفاده کنند.)

- starttimer(calling_node, increment)
- stoptimer(calling_node)
- tolayer3(calling_node, packet)

- `tolayer5(calling_node, message)`

۲.۳ ب

در این قسمت باید توابع پیاده‌سازی شده خود را طوری تغییر دهید که از مکانیزم Go-Back-N که به صورت یک طرفه داده را از A به node B می‌فرستد، استفاده کند. توجه کنید که اندازه‌ی پنجره را ۸ یا هر عدد ثابت بزرگتر از ۸ بگیرید. پروتکل شما می‌تواند از پیام‌های Ack تجمعی (Cumulative) استفاده کند. همچنین نکات زیر را در نظر داشته باشید:

- `A_output(message)`

این تابع اکنون می‌تواند در حالی که پیام‌های unack همچنان وجود دارد، کال شود و شما باید پیام‌های دریافتی از لایه ۵ در سمت ارسال‌کننده یعنی node A را بافر کنید. ارسال‌کننده‌ی شما باید تنها به اندازه‌ی یک پنجره داده‌ها را بافر کند و در صورتی که پنجره پر بود و داده‌ی جدید آمد، مقدار ۱- را به لایه‌ی بالا بفرستد تا نشان دهد که بافر پر است. برای پر شدن بافر مقدار متوسط ارسال پیام از لایه ۵ به لایه ۴ را درون شبیه ساز، مقدار کوچکی قرار دهید به عنوان مثال ۲.

- `A_timerinterrupt()`

توجه کنید که شما فقط یک تایمر در این محیط دارید و باید از همان یک تایمر استفاده کنید!

۳.۳ امتیازی

کد را طوری تغییر دهید که پروتکل شما به صورت دو طرفه باشد و هم از node A به node B پیام ارسال شود و هم از node B به node A. توجه کنید که در اینجا منظور از پیام، پیام‌هایی غیر از Ack است!

راهنمایی:

- برای محاسبه‌ی checksum می‌توانید مانند مکانیزم checksum خود TCP عمل و از جمع sequence number و ack و تک کاراکترهای قسمت payload پکت استفاده کنید.
- کامنت گذاری‌های درون کد را مطالعه کنید!

۴ نکات دیگر

- در کوئرا دو بخش مجزای در نظر گرفته شده است. یکی برای سوالات تئوری و یکی برای سوالات عملی، لطفا سوالات را در بخش مربوط به خودشان آپلود کنید.
- در سوال عملی شما باید برای هر بخش یک فایل کد جداگانه داشته باشید و در نهایت یک زیپ شامل تمامی فایل‌های کدتان را آپلود کنید.
- در این تمرین و سایر تمرین‌های درس، با هرگونه تقلب شدیداً برخورد خواهد شد.
- کدها باید توسط خود شما نوشته شوند و کپی کردن از منابع اینترنتی تقلب محسوب می‌شود.
- در صورت داشتن هرگونه سوال از طریق quera اقدام کنید.
- برای تحویل این تمرین تا روز جمعه ۱۳۰ام اردیبهشت فرصت دارید.

موفق باشید