



بسمه تعالی

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

درس شبکه‌های کامپیوتری، نیمسال دوم سال تحصیلی ۹۶-۹۷

تمرین برنامه نویسی (موعد تحویل: ۱۳۹۷/۳/۱۰)



توجه:

تمرین به صورت گروه‌های حداکثر دو نفره می‌تواند انجام شود (گروه تک نفره نمره اضافی ندارد)
شباهت با پروژه‌های دیگر منجر به کسر نمره خواهد شد.
تحویل پروژه به صورت حضوری است.

مقدمه

در این تمرین قصد داریم تا شما را با پیاده‌سازی یک نرم‌افزار پروکسی آشنا کنیم. امروزه استفاده از پروکسی‌ها در لایه‌های مختلف اجتناب ناپذیر شده است. به عنوان مثال همان‌طور که در درس خواندید، پروتکل UDP فاقد مکانیسم‌های کنترل ازدحام است و می‌تواند باعث شود تا ارسال کنندگان داده با ارسال زیاد بسته‌های UDP باعث کاهش کارایی سیستم‌هایی شوند که از پروتکل TCP به عنوان پروتکل لایه انتقال خود استفاده می‌کنند. به همین علت ممکن است در برخی از شبکه‌ها استفاده از پروتکل UDP محدود شود. به این منظور کاربران این شبکه‌ها از پروکسی‌های لایه حمل استفاده می‌کنند که بسته‌های UDP را به بسته‌های TCP تبدیل کرده و پس از دریافت جواب این تبدیل معکوس را مجدداً انجام می‌دهد.

شرح تمرین

این تمرین از دو بخش تشکیل شده است: یک بخش برنامه پروکسی و بخش دیگر ابزاری است که به عنوان سرویس گیرنده به پروکسی متصل می‌شود. برای بخش سرویس دهنده از سرورهای واقعی موجود در اینترنت استفاده می‌شود و نیازی به پیاده‌سازی نیست. برنامه پروکسی بین سرویس گیرنده و سرویس دهنده قرار دارد و درخواست‌های سرویس گیرنده را به قالب مناسب تبدیل کرده و برای سرویس دهنده ارسال می‌کند.

- از آنجایی که تحویل تمرین حضوری خواهد بود به هر زبان دلخواهی می‌توانید برنامه‌های پروکسی و سرویس گیرنده را برنامه نویسی کنید.

پروکسی

برنامه پروکسی باید قادر باشد، پیغام‌های UDP ارسال شده از طرف سرویس گیرنده را بر روی اتصال TCP به سمت سرویس دهنده که بر روی پورت مشخصی در حال گوش کردن است ارسال کند. جواب درخواست‌ها بر روی اتصال UDP به سرویس گیرنده ارسال می‌شود. همچنین پروکسی پیغام‌های TCP ارسال شده از طرف مشتری را باید بتواند به پیغام‌های UDP تبدیل کند و به سرویس دهنده ارسال کند. جواب پیغام‌ها بر روی اتصال TCP به سرویس گیرنده ارسال می‌شود. ابزاری پروکسی باید حداقل از گزینه‌های زیر برخوردار باشد:

```
proxy -s sourceProtocol:sourceHost:sourcePort -d destProtocol
```



یک مثال از دستور فوق می‌تواند مشابه شکل زیر باشد:

```
proxy -s=udp:127.0.0.1:80 -d=tcp
```

در صورت اجرای این دستور فوق برنامه شما کارهای زیر را انجام می‌دهد:

- تبدیل بسته‌های UDP به جریان TCP
- دریافت بسته‌های UDP در آدرس 127.0.0.1:80
- دریافت بسته‌های UDP از سرویس گیرنده، خارج کردن محتوای درخواست و ارسال آن در قالب بسته‌های TCP به سرویس دهنده
- دریافت بسته‌های TCP از سرویس دهنده، خارج کردن محتوای آن، ایجاد بسته یا بسته‌های UDP و ارسال آن به سرویس گیرنده

سرویس گیرنده

همان‌گونه که مشخص است برنامه‌های نوشته شده فعلی قادر به کارکردن با پروکسی فوق نیستند. برای همین ابزاری مناسب این پروکسی باید ایجاد شود. این ابزار از دو کاربرد اینترنت پشتیبانی می‌کند:

- کاربرد HTTP بر روی پروتکل UDP
- کاربرد DNS بر روی پروتکل TCP

کاربرد HTTP بر روی پروتکل UDP

برنامه شما مشابه یک مرورگر اینترنت، البته با قابلیت‌های بسیار کمتر عمل می‌کند. برنامه شما باید به عنوان ورودی یک درخواست HTTP استاندارد را از کاربر دریافت کند و آن را در قالب بسته UDP به پروکسی ارسال کند. برای سادگی صرفاً برنامه شما باید قادر به پشتیبانی از GET باشد. مثالی از ورودی به برنامه در زیر آمده است:

GET / HTTP/1.1

Host: aut.ac.ir

در این حالت برنامه شما بسته UDP را ایجاد کرده و آن را به پروکسی که بر روی پورت UDP مشخصی تنظیم شده است ارسال می‌کند. پروکسی مطابق با درخواست دریافت شده، یک ارتباط TCP با میزبان با نام دامنه aut.ac.ir بر روی پورت پیش فرض ۸۰ برقرار می‌کند. سپس پروکسی درخواست HTTP را به وب سرور مورد نظر در پورت ۸۰ ارسال می‌کند و منتظر جواب می‌ماند. پس از دریافت کامل جواب، بسته یا بسته‌های UDP را ایجاد کرده و جواب را با استفاده از این بسته‌ها به سمت سرویس گیرنده ارسال می‌کند. برنامه شما پس از دریافت کامل بسته‌ها، یک فایل HTML ایجاد کرده و محتویات بسته‌ها را در آن قرار می‌دهد. این فایل باید به صورت کامل قابل خواندن توسط مرورگر باشد و بدون خطا نمایش داده شود.

- برنامه شما باید قابلیت پشتیبانی از کدهای ۳۰۱، ۳۰۲ و ۴۰۴ را داشته باشد و با نمایش دادن یک پیغام کاربر را از نتیجه درخواست آگاه کند. در صورتی که جواب کد ۳۰۱ یا ۳۰۲ را داشت، درخواست برای Location جدید باید به صورت خودکار در برنامه شما آن قدر تکرار شود تا به کد ۲۰۰ برسید.
- لزومی به دریافت اشیاء دیگر به جز شی اصلی ندارید.
- امتیاز اضافه: تقاضاهایی که یکبار قبلاً انجام شده است را در پروکسی Cache کرده و از تکرار آن جلوگیری کنید. انتخاب معماری و نحوه عملکرد کش بر عهده خودتان است.
- به جز موارد الزامی گفته شده، نحوه انتخاب سایر فیلدهای درخواست‌ها و پروتکل‌ها بر عهده خودتان است.



دانشگاه علمی کاربردی
تهران



دانشگاه علمی کاربردی
تهران

درس شبکه‌های کامپیوتری، نیمسال دوم سال تحصیلی ۹۶-۹۷

پروژه برنامه نویسی (موعد تحویل: ۱۳۹۷/۳/۱۰)

صفحه: ۳ از ۳

- پروکسی مقصد ارتباط را از محتوای بسته دریافتی تشخیص می‌دهد و از قبل مشخص نمی‌شود.

کاربرد DNS بر روی پروتکل TCP

برنامه شما باید مشابه یک سرویس گیرنده DNS عمل کند و بتواند درخواست‌های DNS را ارسال و جواب آن‌ها را پردازش کند. برنامه شما باید از Type های A و CNAME پشتیبانی کند. نمونه‌ای از ورودی به برنامه به صورت زیر است:

```
type=A server=217.215.155.155 target=aut.ac.ir
```

در این حالت کاربر به برنامه درخواست داده است که برای نام دامنه aut.ac.ir، یک query با نوع A ایجاد شود و به آدرس 217.215.155.155 که یک سرور DNS است ارسال می‌کند. این پیام ایجاد شده سپس از طریق پروتکل TCP به پروکسی ارسال می‌شود. پروکسی، محتوای پیام را بازیابی کرده و در قالب یک پیام DNS به 217.215.155.155 که در پیام مشخص شده است ارسال می‌کند.

- برنامه شما باید قادر باشد بسته دریافت شده را تحلیل کند و برای نوع A، آدرس همه IP های دریافتی را نمایش دهد. برای نوع CNAME باید قادر باشد نام دامنه اصلی را نمایش دهد.
- برنامه شما باید قادر باشد مشخص کند که آیا جواب Authoritative است یا خیر.
- برنامه شما باید بتواند در صورت عدم دریافت جواب پرس‌وجوی کاربر در زمان مشخص، آن را مجدداً ارسال کند و تا دریافت کردن جواب مناسب این کار را ادامه دهد.
- امتیاز اضافه: پرس‌و‌جوهایی DNS را که قبلاً انجام شده است را در پروکسی Cache کرده و از تکرار دوباره آن جلوگیری کنید. انتخاب معماری و نحوه عملکرد کش بر عهده خودتان است.
- به جز موارد الزامی گفته شده، نحوه انتخاب سایر فیلدهای درخواست‌ها و پروتکل‌ها بر عهده خودتان است.
- پروکسی مقصد ارتباط را از محتوای بسته دریافتی تشخیص می‌دهد و از قبل مشخص نمی‌شود.

موفق باشید

در صورت هرگونه مشکل یا سوال درخصوص تمرین‌ها و پروژه‌های درس "شبکه‌های کامپیوتری" با تدریس‌یاران درس تماس بگیرید.

پرهام الوانی (Parham.alvani@gmail.com)، سپهر صبور (sepehr.sabour@gmail.com)، فاطمه امیدوار (fa.omidvar73@gmail.com)

سیامک بیک زاده (siamakbeikzadeh@aut.ac.ir)، نگار ندا (ne.neda74@gmail.com)، حسین افشاری (mhafshari@aut.ac.ir)

ایمان تبریزیان (iman.tabrizian@gmail.com)