به نام خداوند جان آفرین

تمرین کامپیوتری 2

شبکه‌های کامپیوتری

زمان تحویل: 11آذر

چکیده

شما در این پروژه باید یک server proxy پیاده سازی کنید و با تنظیم پروکسی مرورگر بر روی ip و port برنامه ی خود، مرورگر به‌جای ارسال مستقیم بسته ها به server مقصد، تمامی بسته ها را برای برنامه ی شما ارسال میکند. برنامه ی شما با پارس کردن بسته و اعمال تغییرات خاصی، آن را برای سرور مد نظر کاربر ارسال میکند و بعد از گرفتن پاسخ از سرور اصلی، پاسخ را برای مرورگر ارسال میکند. همچنین در ادامه ، قابلیتهایی مانند caching ،ارسال ایمیل اخطار به ادمین، حفظ حریم شخصی به پروژه اضافه خواهند شد.

Proxy servers

پروتکل HTTP یک پروتکل لایه کاربرد شبکه مبتنی بر پروتکل TCP است.(Protocol Transfer Hypertext)

همانطورکه از نامش مشخص است برای ارسال و دریافت ابرمتن استفاده میشود. اولین و اصلی‌ترین مورد استفاده‌ی این پروتکل، در وب جهانی است .به طور معمول (client معمولا مرورگر کاربر) به صورت مستقیم با سرور ارتباط برقرار میکند و اطلاعات را دریافت و ارسال میکند؛ ولی در شرایط خاصی، کلاینت از طریق یک موجودیت سوم که **پروکسی** نام دارد با سرور مقصد ارتباط برقرار کرده و به این شکل ارتباط با سرور به صورت غیرمستقیم برقرار خواهد شد .در ساده ترین حالت، پروکسی فقط بسته ها را از مرورگر دریافت میکند و به سمت سرور اصلی ارسال میکند و بعد از گرفتن جواب، آنها را برای مرورگر ارسال میکند؛ اما میتواند به‌جز این،کاربردهای دیگری هم داشته باشدکه در ادامه به برخی از آنها اشاره می‌کنیم.

Caching ➢

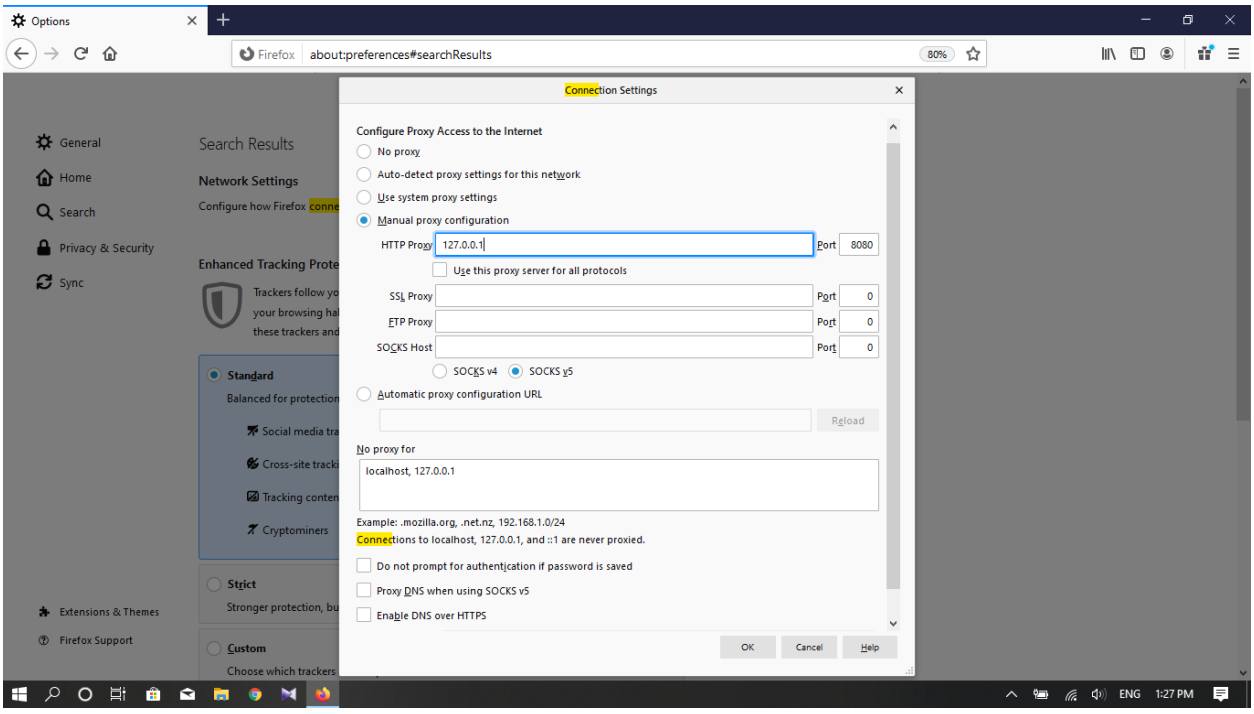
پروکسی می‌تواند با ذخیره کردن بعضی از داده ها که کمتر تغییر می‌کنند یا حجم بیشتری دارند، مانند عکس ها، باعث افزایش سرعت بارگذاری وکاهش تاخیر در دریافت داده ها شود. مثلا مدیر شبکه دانشگاه می‌تواند با راه اندازی یک پروکسی در شبکه داخلی، باعث شود بعضی از منابع اینترنت که اکثر کاربران وب آنها را دریافت میکنند به صورت cache ذخیره شود؛ در این صورت حجم زیادی از ترافیک های تکراری حذف می‌شود و از طرفی چون منابع در سرور های دانشگاه ذخیره شده‌اند باعث افزایش سرعت استفاده‌ی کاربران میشود.

Privacy ➢

شما به طور معمول اطلاعات زیادی را برای وب‌سایت ها ارسال می‌کنیدکه این اطلاعات می‌تواند هم برای صاحب وبسایت ها و هم افرادی که در شبکه قرار گرفته اند مفید باشد و به آنها کمک کند تا از هویت شما آگاه شوند. در ساده ترین حالت، شما در زمان وب‌گردی در header بسته‌های HTTP ارسالی خود اطلاعات مربوط به سیستم عامل و مرورگر خود (user agent) را برای سرور ارسال می‌کنید. وب‌سایت ها می توانند با سواستفاده از این اطلاعات کاربران را ردیابی کنند. کاربران می‌توانند با استفاده از پروکسی این مشکل را رفع کنند؛ چون با این کار درخواست ها به پروکسی ارسال می‌شود و پروکسی میتواند با تغییر این فیلد از نشت اطلاعات شخصی کاربران جلوگیری کند.

تعریف پروژه

شما در این پروژه یک proxy web ساده پیاده سازی می‌کنید. پروکسی شما روی پورتی که از طریق فایل config در زمان اجرا به برنامه داده می‌شود یک Socket TCP باز میکند و روی پورت مورد نظر منتظر می‌ماند. سپس با تغییر تنظیمات server proxy مرورگر و تنظیم ip و port آن مطابق با اطلاعات پروکسی خود، مرورگر با این واسطه ارتباط برقرار می‌کند و درخواست خود را برای شما ارسال می‌کند. برنامه شما بعد از گرفتن درخواست از مرورگر باید درخواست را parse کند و بعد از ایجاد درخواست جدید و مشخص کردن مقصد اصلی، درخواست را برای آن ارسال و بعد ازگرفتن پاسخ، آن را برای مرورگر شما ارسال کند.

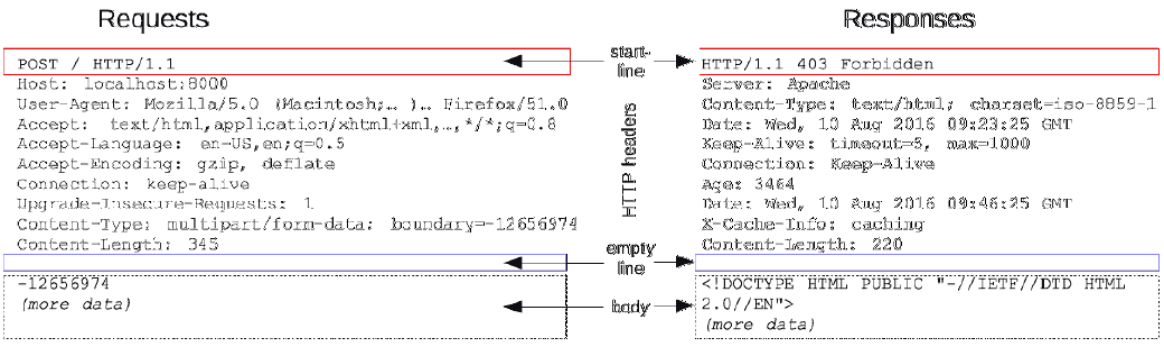


برای سادگی کار می‌توانید همواره از پروتکل HTTP 1.0/ برای ارتباط بین proxy و server web استفاده کنید. پروتکل HTTP 1.1/ از ارتباط پایا استفاده می‌کند که ممکن است برایتان مشکلاتی به وجود بیاورد. در این صورت در زمان دریافت درخواست از نوع HTTP 1.1/ کافیست Header آن را به HTTP 1.0/ تبدیل کنید و برای server web ارسال کنید. البته proxy server شما باید بتواند از هر دو پروتکل HTTP 1.0/ و HTTP 1.1/ بین خودش و مرورگر پشتیبانی کند.

پیاده سازی

قبل از شروع کار باید بدانید زمانی‌که نام وب‌سایتی را در URL-bar مرورگر خود وارد می کنید و دکمه enter را می فشارید چه اتفاقی میافتد.

همانطور که در درس آموختید، در حالت ساده مرورگرها وserver web ها برای ایجاد ارتباط و رد و بدل کردن اطلاعات مورد نیاز، از پروتکلی به نام HTTP استفاده میکنند.HTTP یک پروتکل لایه کاربرد بر روی پروتکل TCP است. (یعنی صحت اطلاعات ارسالی و دریافتی تضمین شده است.) برای مثال، برای گرفتن صفحه اصلی یک وب‌سایت (index.html) بعد از وارد کردن نام آن و زدن دکمه enter ،مرورگر یک socket از نوع TCP با سرور آن وب‌سایت روی یک پورت مشخص (به صورت پیش فرض پورت 80 برای ارتباطات از نوع HTTP استفاده می شود) برقرار میکند. بعد از ایجاد تونل TCP ،مرورگر باید درخواست خود که از پروتکل HTTP پیروی میکند را ارسال کند. پروتکل HTTP ساختاری مانند زیر دارد:



در خط اول، اولین بخش method درخواست را مشخص میکندکه اغلب متدهایdelete ، patch، post، get امروزه استفاده می شوند و برای هرکدام عملکرد متفاوتی تعریف می‌شود. در قسمت دوم path دقیق درخواست می‌آید که در این مثال فایلindex.html مورد درخواست بوده است و بعد از آن شماره نسخه درخواست درج می‌شود و در انتهای خط\r\n می‌آیدکه به خط بعد می‌رویم .در خطوط بعدی هر header در یک خط آمده است که با علامت " : " از مقدار آن جدا شده است و در انتهای هر خط \r\n می‌آید .در بعضی مواقع، مانند متد post ،می‌توانیم دربسته ی ارسالی data هم داشته باشیم که در انتهای بسته و با یک خط خالی (\r\n) از خطوط header جدا می شود.

(برای اطلاعات دقیق‌تر بخش ۲-۲کتاب مرجع درس را مطالعه کنید.صفحه ۹۸)

زمانی‌که تنظیمات server proxy روی مرورگر شما تنظیم شده باشد، بسته‌ی HTTP باکمی تغییرات به سرور proxy ارسال می‌شود و بعد از آن بسته ی اصلی برای وب‌سرور ارسال می‌شود و پروکسی پاسخ را برای مرورگر ارسال می‌کند .برای اینکه بفهمید مرورگر شما چه بسته ای را برای پروکسی تنظیم شده ارسال می‌کند می‌توانید از ابزار netcat استفاده کنید.Netcat ابزاری ساده برای ارسال وگرفتن بسته های UDPو TCP است .با اجرای netcat به شکل روبرو netcat روی پورت داده شده منتظر می‌ماند و در صورتی‌که بسته ای برای آن ارسال شود آن‌را درterminal چاپ می‌کند.

nc -l -p <port-number> -v

حال با تنظیم پروکسی در مرورگر خود روی ip و port سیستم خود و پورت تنظیم شده و درخواست یک وب‌سایت که از پروتکل HTTP استفاده می‌کند می‌توانید بسته‌ی ارسال شده توسط مرورگر را در netcat مشاهده کنید.

مشاهده می کنید که بسته های ارسالی به server proxy تفاوت کوچکی با بسته های معمولی دارند.
(با مثال قبلی از پروتکل HTTPمقایسه کنید)

GET http://www.amazon.com/index.html HTTP/1.1
HOST: www.amazon.com
Proxy-Connect: Close
Connection: Close

زمانی که از proxy استفاده نمی‌کنید تنها قسمت‌های بعد از path و بدون name host آورده می‌شود؛ در غیر این صورت، آدرس وب‌سایت به صورت کامل درج می‌شود .همچنین یک سرآیند connection-proxy هم به بسته اضافه می‌شود که همانند سرآیند connection عمل می‌کند با این تفاوت که وضعیت ارتباط بین مرورگر و server proxy را مشخص می‌کند. (نیاز به پیاده سازی عملکرد این قسمت ندارید و همواره مقدار آنرا close فرض کنید.)

برنامه ی شما تنها باید قسمت name host آدرس و سرآیند connection-proxy را از آن حذف کند و برای server web ارسال کند و جواب را بدون تغییر به مرورگر برگرداند.

***نکته**: برای پیاده‌سازی می‌توانید از هر زبان دلخواهی استفاده کنید با این شرط که آن زبان قابلیت استفاده کردن ازsocket های TCPرا داشته باشد ولی پیشنهاد ما زبان پایتون و بعد از آن جاوا است. (استفاده از ماژول‌های سطح بالا برای ارتباطات **مجاز نمی باشد**، مثلا نمی‌توانید از ماژول request در پایتون که بسته های http را می سازد استفاده کنید.)

برای پیاده سازی پروژه قبل از هرکار باید فایل config با فرمت json راکه کنار فایل اجرایی شما قرار داده شده است را بخوانید و آن را parseکنید، سپس باید روی پورتی که در فایل config آمده است یک socket از نوع TCP ایجادکنید تا منتظر اتصال مرورگر بماند؛ بعد از آن باید تنظیمات proxy server مرورگر در حالت HTTP را روی ip و port برنامه خود تنظیم کنید؛ برای اینکار می توانید وارد تنظیمات مرورگر خود شوید و در قسمت proxy اطلاعات اتصال را واردکنید .دقت کنید که تنها حالت HTTP را فعال کنید زیرا ما در این پروژه به دلیل سادگی کار تنها توانایی پاسخ به درخواست‌های رمز نشده HTTP را داریم .بعد از پیاده سازی اولیه ی برنامه خود، مرورگر باید بتواند از proxyشما استفاده کند و ترافیک از برنامه شما عبور داده شود و کاربر از مرورگر استفاده کند، بعد از پیاده سازی کامل این قسمت ویژگی های دیگر برنامه را با توجه به راهنمایی آن قسمت به برنامه خود اضافه کنید.

logging

proxy شما در زمان اجرا باید یک فایل log در کنار خود ایجاد کند و تمامی اطلاعات را با ساعت وقوع در آن ذخیره کند .اطلاعاتی مانند درخواست گرفته شده از مرورگر و جواب آن، درخواست فرستاده شده برای web server و جواب آن، و یا hit یا missشدن cache باید در آن مشخص شود تا از درستی اجرای برنامه ی شما اطمینان حاصل شود و همچنین اینکار در زمان debugکردن برنامه شما نیز میتواندکمک‌تان کند .ابتدا از فایل config وضعیت این سرویس را بررسی کنید و در صورتی که سرویس فعال باشد باید در فایلی که مشخص

شده است لاگ های خود را بنویسید. نمونه فایل لاگ ضمیمه شده است و باید ساختاری شبیه به آن داشته باشد. (اگه فایل وجود ندارد آن را بسازید و در صورتی‌که از قبل این فایل وجود داشته است در ادامه ی آن شروع به نوشتن کنید.)

Caching

باید در نظر داشته باشیم که فضای اختصاص داده شده برایCache کردن پاسخ ها یک فضای محدود است. سیاست جایگزینی در این پروژه سیاست LRU می‌باشد. چون ذخیره کردن در فایل های سیستم مربوط به این درس نمی باشد نیازی به ذخیره اطلاعات پس از خارج شدن از پروکسی نمی باشد (نیازی به استفاده از دیتابیس نیست). باید توجه داشت که فضای محدود در اینجا به معنای حجم محدود نیست و محدودیت بر روی تعداد بسته های Cache شده می باشد. بدین معنا که مثلا حداکثر 200 پاسخ میتواند Cacheشود که این مقدار را باید از فایل config.json بخوانید.

نکات مهم برای پیاده سازی بخشcaching :

برای پیاده سازی این بخش دقت کنید که تمام پاسخ‌ها **نباید** cache شوند، بلکه باید با توجه به مقدار تنظیم شده در پرچم Pragma تصمیم گیری درستی در این موردگرفته شود .شما در این پروژه باید مقدارهای زیر را برای cache پشتیبانی کنید .باید دقت داشت که در صورتی که پرچم pragma تنظیم نشده باشد proxy می تواند این بسته را در Cache ذخیره کند.

No-cache : این مقدار تنها روی بسته های پاسخ درج میشود و مقدار پرچم pragma می باشد. به این معناست که proxy نباید این بسته را در Cache ذخیره کند.

Expire: بسته تا این تاریخ اعتبار دارد و نیازی نیست برای پاسخ هر درخواست یک بسته به سرور استفاده کنیم و از همان مقدار موجود در Cacheبرای آن استفاده میکنیم .

If-Modified-Since:اگر شما بسته ای راcache کرده باشید و میخواهید اعتبار بسته را بررسی کنید میتوانید از این پرچم استفاده کنید، این پرچم در دو حالت در درخواست ها فرستاده می شود :

- اگر در سرآیند، پرچم Expire تنظیم نشده باشد، در این صورت برای پاسخ هر درخواستکه قبلا در Cache ذخیره شده است باید یک بسته شامل این پرچم تنظیم شده باشد و به سرور فرستاده شود. در صورتی که جواب 304 از سمت سرور بازگردد، یعنی بسته تغییر نکرده است و همان مقدار موجود در Cache را به کلاینت ارسال میکنیم و اگر پاسخ 200 بازگردد باید مقدار موجود در Cache آپدیت شود و هم بسته جدید به کلاینت ارسال شود .
- اگر بسته منقضی شده باشد (اگر از تاریخExpire گذشته باشد) در صورت عدم استفاده از این پرچم دوباره درخواست گرفتن بسته به سرور فرستاده می شد و سرور دوباره بسته را ارسال می کرد؛ در حالیکه این پرچم این مشکل را حلکرده است. در این حالت یک بسته شامل این پرچم به سرور ارسال میکنیم و در صورتی که پاسخ 304 برگردانده شد برچسب زمان برای آن پاسخ بروزرسانی می شود. در صورتی که پاسخ 200 فرستاده شد (این پاسخ همراه با فایل فرستاده می شود) مقدار موجود در Cacheرا دوباره بروز رسانی میکنیم.

برای شروع بخش caching حتما لینک های زیر را مطالعه کنید:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching

https://www.keycdn.com/blog/http-cache-headers

Privacy

در زمان گرفتن درخواست از مرورگر، سیستم عامل و مرورگر در header با عنوان agent-user مشخص می‌شود. proxy server می‌تواند این header را تغییر دهد و agent-user خودش را که در فایل config مشخص شده است به‌جای آن قرار دهد تا در زمان وب‌گردی حریم شخصی افراد رعایت شود .این قسمت بسیار ساده است و تنها کافیست در صورتی که این سرویس فعال است، سرآیند agent-user در زمان ارسال را به مقدار داده شده در فایل config تغییر دهید و درخواست را برای وب سرور ارسال کنید.

Accounting

یکی دیگر از محدودیت هایی که می تواند بر رویproxy ها اعمال شود دادن حجم مشخص به هرکاربر می باشد. پروکسی شما باید تنها به ipهایی که در فایل config و در قسمت users آماده اند اجازه اتصال دهد. در ابتدای اجرای پروکسی، هر کاربر حجم ترافیک مشخصی داردکه پروکسی شما باید تنها به همان اندازه اجازه‌ی استفاده از ترافیک را به کاربر بدهد. مقدار شارژ هر فرد را در اجرای برنامه از فایل config بخوانید، برای سادگی کار نیازی به persistent بودن شارژکاربر نیست و نیاز به درگیر شدن با دیتابیس نمی‌باشد، می‌توانید این مقدار را در طول اجرای برنامه در ساختمان داده‌ای نگه‌داری‌کنید .احراز هویت کاربران توسط ip آنها انجام می‌شود؛ با دریافت هر بسته درخواست توسط پروکسی، در صورتی که ip درخواست دهنده در لیست کاربران پروکسی بود و حجم ترافیک او باقی ماند بود، اجازه‌ی استفاده از پروکسی را دارد و در غیر این صورت ارتباط بسته میشود یا می‌توانید پیامی مناسب برای کاربر نمایش دهید.

Restrictions

شما باید قابلیت محدود کردن و Proxy فیلترینگ بعضی از وب سایت ها را نیزداشته باشد. درفایلconfig قسمتی برای اعمال محدودیت در نظر گرفته شده است که به دو صورت کاملا بسته (BLOCK) و کاهش سرعت(SLOW) اعمال بسته می شوند، زمانی که محدودیت به صورت کاملا بسته شده باشدproxy server باید درخواست آن ها را به صورت کاملdrop کند و دسترسی کاملا بسته شود .
اما برای بعضی از وب سایت ها باید در ارتباط آنها با وب سرور تاخیر به وجود بیاورید که در فایل config مقدار محدودیت به صورت یک عدد که زمانdelay برای جواب دادن به درخواست ها با مقیاس ms است داده شده است.
Proxy server بعد از گرفتن درخواست به اندازه ی مقدار گفته شده متوقف می‌شود و سپس درخواست را بررسی و جواب را ارسال می‌کند . در این صورت بعضی از سرویس ها با افت سرعت مواجه می‌شوند.

برای پیاده سازی در صورتی که این سرویس فعال باشد زمانی که سرآمد host با یکی از وب سایت های هدف برابر بود با توجه به نوع محدودیت در صورتی که محدودیت از نوع BLOCK بود اتصال socket با مرورگر را به صورت کامل ببندید و در صورتی که محدودیت از نوع SLOW باشد با توجه به مقدار وقفه به درخواست ها پاسخ دهید تا از سرعت وب‌گردی کاربر بکاهید.

توضیحات تکمیلی

- پروژه درگروه های دو نفره انجام می‌شود.
- برنامه شما باید درخواست‌ها را به صورت همزمان پاسخ دهد. (می‌توانید از thread به این منظور استفاده کنید.)
- زمانی‌که درخواست را برای وب‌سرور ارسال می‌کنید باید شماره نسخه درخواست را به HTTP 1.0/ تغییر دهید.
- از هر زبانی می‌توانید استفاده‌کنید ولی‌کدهای شما باید سطح پایین و با استفاده از socket نوشته شوند.
- برایparse کردن درخواستهای http استفاده از ماژول آماده مجاز نیست و باید به صورت دستی انجام شود. (برای json و html بلامانع است.)
- دقت کنیدکه تمام جزئیات مربوط به عملکرد پروکسی شما باید در فایل log ثبت شوند.
- برای امتحان کردن حالت های مختلف Cache می توانید از این سایت ها کمک بگیرید:

https://www.resaa.net/
http://bdood.ir/
https://www.sib.ir/
https://www.irna.ir/
http://mydiba.xyz/

- برای مشاهده بسته های ارسال شده و دریافت شده می توانید از لینک زیرکمک بگیرید:

https://websniffer.cc/

پیروز و موفق باشید