مهندسی اینترنت، تمرین تئوری 1: HTTP **سیّد سبحان عابدی 9731707**

سوال 1)

تغییرات از V1.0 به V1.1:

افزایش تعداد کدهای وضعیت از 16 به 24. علاوه بر آن سرتیتر Warning نیز به HTTP اضافه شد تا هرکدام از وضعیت‌های پاسخ را بتوان بهتر توضیح داد.

علاوه بر این قابلیت Digest Access Authentication نیز به نسخه اضافه شد. این قابلیت این امکان را می‌دهد که رمزها به صورت digest شده منتقل شوند و استفاده از یک کد یکبار مصرف باعث می‌شود اطلاعات مبادله شده، قابل استفاده مجدد نباشند.

امکان استفاده از اتصالات persistent نیز در نسخه 1.1 اضافه شده و این شیوه اتصال به عنوان ویژگی پیش‌فرض تنظیم شد. علاوه بر این امکان Pipelining نیز به نسخه 1.1 اضافه شد. هرچند این قابلیت این امکان را به کاربر می‌دهد که چندین درخواست پشت‌سرهم بدون دریافت جواب ارسال کند ولی همچنان پیام‌ها به صورت سریال هستند و یک پاسخ حجیم و یا با مرجع کند، می‌تواند کل پاسخ‌ها را کُند کند.

در نسخه 1.0 تنها سرتیتر if-Modified-Since برای caching وجود دارد ولی در نسخه 1.1 مقدار entity tag اضافه شد. با استفاده از این tag به راحتی می‌توان تشخیص داد که آیا دوفایل یکسان هستند یا خیر و بر مبنای این مقدار سرتیترهای دیگری برای کنترل cache نیز اضافه شدند.

امکان Chunked transfer و فشرده سازی پیشرفته نیز در نسخه 1.1 اضافه شدند.

تغییرات از V1.1 به V2:

مهمترین قابلیتی که به نسخه 2 اضافه شد، Multiplexing هست. این قابلیت باعث می‌شود بتوان به صورت همزمان چندین شئ را به صورت همزمان ارسال کرد. علاوه براین قابلیت Weighted Prioritization به توصعه دهنده‌ها این امکان را می‌دهد که مشخص کند اشیائ با چه ترتیبی برای کاربر ارسال شوند و با چه اولیتی در مرورگر render شوند.

همچنین قابلیت Server Push نیز به پروتکل HTTP اضافه شد و این امکان را می‌دهد که پیش از اینکه کاربر/مرورگر یک شئ را درخواست کند، خود server اشیائی که به نظرش مورد نیاز کاربر/مرورگر خواهد بود، را برای او ارسال کند و همچنین به کاربر/مرورگر اعلام می‌کند که این اشیائ بدون درخواست ارسال شده‌اند.

درنهایت در نسخه 2 قابلیت فشرده‌سازی پیشرفته‌تر شده و باعث می‌شود سروندهای درخواست‌های HTTP نیز فشرده‌سازی شوند.

تغییرات از V2 به V3:

مهمترین تغییر از نسخه 2 به نسخه 3، استفاده از پروتکل QUIC به جای پروتکل TCP هست. این پروتکل بر مبنای UDP هست و چون نیازی فرآیند 3 way handshake ندارد، تاخیر خیلی کمتری نسبت به TCP دارد و تاخیر ارتباط را کاهش می‌دهد.

همچنین در HTTP نسخه 3 تنها امکان برقراری اتصال امن وجود دارد و در نتیجه این تغییر امنیت در خود پروتکل HTTP پیاده‌سازی شده و دیگر نیازی به استفاده از TLS نیست. این باعث می‌شود تاخیر TLS نیز به تاخیر اتصال اضافه نشود و زمان برقراری یک اتصال امن خیلی کمتر از زمان موردنیاز در پروتکل نسخه 2 باشد.

چرا در HTTP نسخه ۳ از پروتکل UDP به عنوان پروتکل لایه ارتباط استفاده شده است؟

همانطور که در توضیحات بالا گفته شد، پروتکل QUIC تاخیر برقراری ارتباط خیلی کمتری نسبت به TCP دارد. این مسئله در ارتباطات با حجم داده انتقالی بالا شاید خیلی دیده نشود ولی زمانی که بخواهیم تعداد زیادی ارتباط برقرارکنیم و در هرکدام حجم کمی از اطلاعات را منتقل کنیم، خیلی قابل توجه می‌شود. همچنین به مرور زمان و با افزایش پهنای‌باند اتصالات، سرعت انتقال اطلاعات افزایش می‌یابد ولی زمان برقراری یک ارتباط TCP کاهش نمی‌یابد. با تقریب خیلی خوبی می‌توان‌گفت زمان برقراری یک اتصال TCP تنها تابعی از فاصله فیزیکی client-server و سرعت انقال اطلاعات در محیط استفاده شده هست. در چدین سال گذشته و با پراکنده شدن مراکز داده در سراسر جهان، تلاش شده فاصله کاربران تا serverها تا حد امکان کاهش داده شود. از طرفی با تلاش هرچه بیشتر ما برای بیسیم کردن ارتباطات، امکان کاهش زمان اتصال دیگر وجود ندارد. در نتیجه تنها راهی که برای کاهش زمان برقراری ارتباط برای ما باقیمانده، کاهش تعداد تماس‌های موردنیاز برای برقراری ارتباط هست. در نتیجه تنها راه ممکن استفاده از پروتکل‌هایی مانند UDP هست. همچنین در سال‌های گذشته توانایی پردازشی کامپوترها (دستگاه‌های محاسات دیجیتال) به اندازه‌ای افزایش یافته که می‌توان قابلیتِ قابل اطمینان بودن را در لایه نرم‌افزار پیاده‌سازی کرد.

سوال 2)

جواب:

آ) ارتباط stateless به این معناست که serverی که درخواست به او ارسال می‌شود، هیچ اطلاعی از هویت درخواست‌کننده ندارد و پاسخی که می‌دهد تنها براساس درخواست ارسال‌شده و اطلاعات موجود روی server هست. همچنین پروتکل stateی ندارد که توسط درخواست‌ها قابل تغییر باشد. یک اتصال stateful این امکان را دارد که از درخواست‌های قبلی کاربر و هویت او مطلع باشد و پردازشی که انجام می‌دهد و یا محتوایی که از server درخواست می‌کند و برمی‌گرداند درخواست‌های قبلی و هویت کاربر باشد. پروتکل‌های stateless نیاز به محاسبات کمتری دارند و سرعت عمل بیشتری دارند.

ب) پروتکل HTTP در دسته stateless قرار می‌گیرید. برای اضافه کردن state به این پروتکل از قراردادهای مشترک بین client و server باید استفاده کنیم. یکی از این شیوه‌ها cookie هست. در cookie، server اطلاعاتی برای کاربر ارسال می‌کند که بیانگر هویت و یا ویژگی‌های سرویسی می‌شود که باید به او ارائه دهد و هرگاه که کاربر دوباره بخواهد به آن server درخواست بفرستد، مرورگر cookie را نیز همراه آن ارسال می‌کند.

سوال 3)

آ) در سرویس‌هایی که ممکن است اتفاقی در سمت server یا یکی دیگر از clientها رخ دهد و لازم باشد server دیگر کاربران را از این خبر آگاه کند، به دلیل اینکه امکان این وجود ندارد که یک اتصال جدید از سمت server آغاز شود، server نمی‌تواند در اسرع وقت این خبر را برساند و باید منتظر بماند تا client یک اتصال جدید ایجاد کند. در ساختار اتصال client/server برای اینکه یک client خبرها را با تاخیر کمتری دریافت کند، باید با فرکانس بیشتری به server متصل شود. ولی هرچه تعداد اتصالات در واحد زمان بیشتر شود، اتلاف در مصرف انرژی (در سمت کاربر)، پهنای باند بیشتر می‌شود و overhead پردازشی و مدیریت شبکه زیادی در سمت server خواهیم داشت.

ب) دو شیوه کلی برای دریافت پیام‌ها از server وجود دارد. یکی از آنها درخواست توسط کاربر (Client Pull) و دیگری، ارسال توسط server (Server Push) هست. از جمله این روش‌ها میتوان به انواع زیر اشاره کرد:

Short Polling: این شیوه مانند توضیحاتی هست که در قسمت (الف) داده شد. Client با استفاده از یک تایمر به صورت مکرر به Server درخواست ارسال می‌کند و Server اگر پیامی داشته باشد، آن را ارسال می‌کند.

Long Polling: این شیوه بر مبنای مدل Comet کار می‌کند. در این شیوه، Client یک درخواست ارسال می‌کند و اتصال را باز نگه می‌دارد تا پیامی در سمت Server برای او ایجاد شود و Server برای درخواست، پاسخ ارسال کند یا آنقدر طول بکشد که Timeout شود. در هر صورت Client دوباره یک درخواست دیگر ارسال خواهد کرد.

Server Sent Events: این پروتکل با HTTP خالی متفاوت هست. در این پروتکل Client یک درخواست ارسال می‌کند و اتصال برمبنای SSE آغاز می‌شود و در ادامه هرگاه Server پیامی برای Client داشته باشد، آن را ارسال می‌کند. تفاوت SSE با Long Polling در این هست که این

روش زمانی که اتصال باز می‌شود، محدودیتی برای بستن آن نیست. به این صورت که Client در وضعیت Wait قرار نمی‌گیرد و صرفا اتصال باز می‌ماند. و هربار که Client یک پاسخ دریافت می‌کند، اتصال بسته نمی‌شود و لازم نیست دوباره اتصال باز شود. هرچند امروزه پشتیبانی برای این پروتکل تقریبا در تمام مرورگرها وجود دارد ولی ممکن است در بعضی مرورگرهای خاص از این شیوه پشتیبانی نشود و از Long Polling به عنوان شیوه پشتیبان استفاده می‌شود.

WebSocket: در هر دو شیوه Long Polling و SSE دو مشکل عمده وجود دارد. اول اینکه این ارتباطات یک‌طرفه از سمت Server هستند و اگر کاربر بخواهد پیامی ارسال کند، باید یک اتصال جدید ایجاد کند. دوم اینکه در هردو این روش‌ها سربار پروتکل HTTP وجود دارد و در پیام‌های کوتاه این سربار مقدار قابل توجهی دارد. در WebSocket پس از اینکه ارتباط اولیه برقرار می‌شود، با استفاده از سرتیتر Upgrade پروتکل لایه Application را به WebSocket تغییر میدهیم و از این پس پیام‌ها به صورت خام و دوطرفه برروی پروتکل TCP جابه‌جا می‌شوند. ولی این شیوه مشکلات خود را نیز دارد. استفاده نکردن از پروتکل HTTP باعث می‌شود تمام ابزارهایی که در این پروتکل در دسترس داریم (مانند Multiplexing) را از دست بدهیم و مجبورباشیم دوباره آنهایی که نیازداریم را پیاده‌سازی بکنیم.

سوال 4)

ss://asghar:1234!!@ss.myproxy.com:1234\#shadowSocks1

ss ➔ Protocol

Asghar ➔ Username

1234!! ➔ Userpass

ss.myproxy.com ➔ Host

1234 ➔ Port

shadowSocks1 ➔ Fragment

سوال 5)

1. در این حالت بهتر است از کد 503 استفاده کنیم. مرجع مشکل را می‌دانیم و می‌دانیم مشکل موقتی هست. ولی اگر بخواهیم اطلاعات زیادی در اختیار بازدیدکنندگان قرار ندهیم، بهتر است از کد 500 استفاده کنیم.
2. در این حالت می‌توانیم از کد 200 استفاده کنیم زیرا پردازش با موفقیت انجام شده ولی اطلاعات وراد شده اشتباه بوده. ولی برای اینکه پاسخ دقیق‌تری داده‌باشیم و در صورتی که در صفحه ورودی از المان Form استفاده شده باشد، می‌توانیم با برگرداندن کد 205 درخواست دهیم که Form ریست شود تا کاربر دوباره آن را پر کند.
3. در این حالت باید از یکی از کدهای 301 یا 308 استفاده شود. فرق این دو کد در این هست که کد 308 باعث می‌شود مرورگر دقیقا همان درخواست را با همان پروتکل برای دامنه جدید ارسال کند. مثلا اگر وبسایتمان درخواست‌هایی با پروتکل POST دریافت می‌کند که بدنه اطلاعاتی دارند، حتما باید از کد 308 استفاده کنیم.
4. دقیق ترین پاسخی که می‌توان به تعداد غیر مجاز درخواست داد، کد 429 هست ولی اگر بخواهیم اطلاعات زیادی ندهیم، بهتر است از کد 400 استفاده کنیم.
5. در صورتی که درخواست خودکار بوده و قرار نیست صفحه مرورگر تغییر کند، باید از کد 204 استفاده کنیم ولی اگر درخواست از سمت کاربر بوده و پس از انی لازم هست صفحه جدیدی ارسال شود، باید از کد 200 استفاده شود.
6. دقیق‌ترین کدی که بیشترین اطلاعات را در اختیار کاربر قرار می‌دهد، کدر 451 هست. ولی اگر نخواهیم دلیل این دسترسی ندادن را اعلام کنیم، باید از کد 403 استفاده کنیم. و اگر هم نخواهیم هیچ دلیلی برای عدم ارسال فایل اعلام کنیم، باید از کد 400 استفاده کنیم.

سوال 6)

1. Forward Proxy
2. Reverse Proxy
3. Reverse Proxy