

سؤال اول:

ویژگی هر کدام توضیح دهید

بخش ا را:

نسخه 1.0 http:

این نسخه از کانکشن های جداگانه استفاده می کند. هم چنین پس از دریافت پاسخ ارتباط با سرور پایان می یابد. از اینیت بالایی برخوردار نیست. برای request و response ها در میانه response می تواند فایل های مختلفی مانند media/html باشد.

نسخه 1.1 http:

این نسخه از دو قابلیت persistent connection و request queuing پشتیبانی می کند هم چنین این

نسخه امروزه بیشتر از نسخه های دیگر مورد استفاده قرار می گیرد. این نسخه منابع را به صورت صف مانند و شیف سرهم دریافت می کند پس از اتمام دریافت یک منبع به سراغ منبع بعدی می رود. این نسخه قابلیت اتصال مجدد مجهز شد.

نسخه 2 http:

این نسخه قابلیت multiplexing به روشی یک تک کانکشن را دارد هم چنین این نسخه قدرت انتقال چری دارد و در هر بار این نسخه سازی می کند هم چنین این نسخه از قابلیت تبدیل داده به با بیتی استفاده می کند و آن ها را به آسانی می کند. این نسخه

چند مدره ها را می سازد از سرعت بسیار برخوردار است هم چنین چون پیام ها و جریان داده های مختلف را به آسانی می کند قابلیت multiplexing دارد هم زمان چند منبع را می تواند دریافت کند و منابع مختلف متعلقه دریافت می کنند

اصل ترین تفاوت این نسخه اندک از QVLC به عنوان transport layer برای هندل کردن جریان های داده برای tcp است همچنین برای ساخت یک کانکشن این نسخه مناسب تر از handshake را بسیار سریع تر ایام می دهد و در نهایت این نسخه نتایج بداند در حالت encrypted و secure استفاده شود در حال که نسخه های اخیر مثل http 2.0 مردان بدون Https نیز پیاده سازی شود:

بخش ۲۱ :
دلیل آن این است که udp برخلاف tcp قابلیت کنترل از دست دادن در نتیجه بسیار کمتری دارد و سرعت آن نیز است و مدیریت از دست دادن آن نیز ساده تر می شود چون نیازی به پیگیری مجدد ندارد.

سؤال دوم:

بخش ۲۲ : پروتکل stateful با توجه به درخواستی که دریافت می کند (request) و اطلاعات و state ای که ذخیره کرده و از قبل نگهداری شده است پاسخ (response) را ارسال می کند همچنین درخواست های جدید می داند سبب تغییر در state فعلی ارتباط می شوند. ملایم این پروتکل و پیاده سازی آن سخت است و scale کردن آن دشوار است. در طرف مقابل پروتکل stateless را داریم این پروتکل تنها به درخواست (request) ارسال شده دست می کند و درخواست های قبلی و داده های گذشته به آن بی اهمیت هستند و فقط به current state و درخواست در یافتی response ارسال می کند. پیاده سازی آن راحت تر است و scale کردن آن نیز تر است هم چنین scale کردن آن راحت تر است و در صورت crash کردن هندل کردن آن ساده تر است.

بخش ۲۱:

برای حل مشکل http در دسترس stateless قرار می گیرد و برای حل کردن مشکل بی حافظه بودن از کوکی ها (cookies) برای ذخیره کردن state و اطلاعات در سمت client و ارسال آن ها برای سرور. هم چنین برای حل این مشکل و مدیریت session از روش های مانند: hidden from fields و URL Rewriting استفاده می شود.

سوال سوم:

بخش ۲۲:

ها به طور کلی در صورت سوال نیز گفته شده چون محدودیت داریم باید منتظر بمانیم تا client درخواستی ارسال کند پس سرور بتواند جریان طلبی اطلاعات را ارسال کند. بنابراین مثلاً در webhook یک notification برای client ارسال می کنیم و با درخواست client در پی آن پیام notification اطلاعات منتظر را برای آن ارسال می کنیم اما چالش هایی که در این بین ممکن است رخ دهد این است که کاربر هیچ توضیحی به پیام های notification نداشته باشد یا این که برادر از دام و مشکلات شبکه پیام notification برای آن ارسال شود یا پیام server سی از ارسال redirect شود.

بخش ۲۳:

اینکه راه حل استفاده از یک زمانبندی است زمانی که پیام notification ارسال می شود در پاسخ صحت حالت ممکن است. اگر status code مثل ۲۰۰ به سرور یعنی موفق بوده است ۳۰۰ برگردد یعنی ناموفق بوده و redirect ۳۰۰ به سرور یعنی مشکل در شبکه رخ داده است و پیام ناموفق بوده است. برای حل این چالش از یک زمانبندی استفاده می شود.

و اگر پیام ناموفق باشد دوباره آن را ارسال می کند تا ارسال کند شده آنگاه ارسال
 شکست بخورد پس از ۳ ثانیه دوباره آن را ارسال می کند و اگر برای بار دوم هم شکست
 بخورد پس از ۳۰ ثانیه دوباره ارسال می شود و اگر برای بار سوم هم شکست بخورد
 زمان بندی از ۵ دقیقه آن را ارسال می کند و به همین ترتیب زمان به صورت
 تصاعدي بالا می رود تا هم درخواست های ارسال پیام زیاد شود هم عیبی پیام ها
 نداشتند ارسال شوند

راه حل بعدی longpolling است به این صورت که client دائماً از سرور سوال می کند آیا
 چیزی برای ارسال دارد یا نه و سرور می تواند در پاسخ داده های مورد انتظار در ارسال کند
 و در آن ترس ترسیم از websocket استفاده کنیم در این روش یک کانکشن دو طرفه و Full
 duplex است که شروع آن با start است پس از ایجاد آن سرور می تواند در هر زمان داده ارسال
 کند

مثال چهارم:

URL

ss://asghar:1234@ss.myproxy.com:1234\#staclossocks1

پروتکل نام کاربر و گذرنامه پورت دامنه رمزنگار

سوال پنجم:

بخش ۵: 500 و 503 من نشان آن را یک ارور کلی است (500) یا این که عدم
 فعالیت ارتباطی را نشان می دهد (503) و اعلام کنیم سرور در دسترس نیست
 بخش ۴: 401 جوره credential استفاده است

بخش ۳: 301 و 302 اگر دائمی منتقل شده از 301 و آنگاه موقتی منتقل شده از 302
 استفاده می کنیم

بخش ۵۱۴ : 429 به همای درخشا (request)

بخش ۵۱۵ : 200 به همای معرفت آفریدن

بخش ۵۱۶ : 403 به همای ممنوع شده و نیز قابل استرس

سوال سستم :

بخش ۹۱۱ : forward proxy چون اینها سرورهای راهنما را می گردانند و پس به سرانغ خارجی می رود

بخش ۹۱۲ : reverse proxy چون این نوع proxy کار load balancing هم انجام می دهد

بخش ۹۱۳ : reverse proxy چون این نوع proxy کار نظارت و افزایش امنیت انجام می دهد