

" به نام خدا "

تمرین اول تئوری
برنامه نویسی وب

نگارش : سجاد قدیری
شماره دانشجویی : 9723067
نام استاد : پرهام الوانی

سوال یک :

(الف)

HTTP 1.0 :

Packet lost , Support GET and POST methods , Headers for requests and responses .

HTTP 1.1 :

Messages in plaintext , able to cache , added pipeline property to requests .

HTTP 2.0 :

Multiplexing method , handle parallel requests transfer .

HTTP 3.0 :

Do not support “ Local call “ and “ LoopbackAddress “ , using UDP instead of TCP

(ب)

استفاده از UDP باعث برتری های مختلفی نسبت به TCP شده است .
یکی از علت های استفاده از UDP برتری آن در سرعت نسبت به TCP است .
در TCP از روش peer to peer استفاده میشه و کلاینت منتظر دریافت پاسخ از سمت سرور می باشد ولی
در UDP پیام BroadCast می شود و هر کسی بخواد پیام را می گیرد و کلاینت دیگر منتظر پاسخ نیست .

سوال دو :

(آ)

عبارت **stateless** بودن به آن معنی است که وقتی client به server یک درخواست (request) می فرستد ، یک ip به آن تخصیص داده می شود و درخواست client پاسخ داده می شود و اگر مجددا همان client در زمانی دیگر همان درخواست یا هر درخواست دیگری را از server داشته باشد، دیگر server به طور مشخص تاریخچه ای از آن client را نگه نداشته است و او را نمی شناسد (یک ip دیگر به client اختصاص داده می شود).
همچنین انتقال اطلاعات پشت سر هم صورت می گیرد و اگر داده ای از بین برود و یا client پاسخی دریافت نکند، Server به کار خود ادامه می دهد. در نتیجه در این حالت با وجود پیاده سازی ساده تر، احتمال از بین رفتن داده ها بسیار بیشتر است.

عبارت **stateful** دقیقا به معنای عکس این ماجراست یعنی server همواره هر client را می‌شناسد و یک ip ثابت به او اختصاص می‌دهد. همچنین در بحث انتقال داده ها، مادامی که client درخواستش برطرف نشود و پاسخی نگیرد، همان درخواست را ارسال می‌کند.

(ب)

پروتکل HTTP از دسته پروتکل های stateless می‌باشد که این ویژگی باعث عملکرد بهتر و پیچیدگی کمتر آن شده است. البته در برخی کاربردها مانند پورتال دانشگاه که server نیازمند شناسایی client است، دارای ضعف می‌باشد.

برای برطرف کردن مشکلات ناشی از stateless بودن از مفاهیمی همچون json web token ها و cookie ها استفاده می‌شود.

سوال سوم :

(آ) چالش مطرح شده بیانگر آن است که در پروتکل HTTP ، کاربر (client) باید مرتباً درخواست خود را ارسال کند تا پاسخی از server دریافت کند زیرا این پروتکل یک پروتکل یک طرفه است.

(ب)

☐ یکی از راهکارها می‌تواند استفاده از پروتکل دو طرفه ی websocket می‌باشد.

☐ یکی دیگر از راهکارها که ظاهراً به صرفه نیست استفاده از مفهومی به نام polling می‌باشد.

سوال چهارم :

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

ss => <protocol (scheme) >
asghar:1234 => asghar = <user > , 1234 = <pass >
ss.myproxy.com:1234 => ss.myproxy.com = <host > , 1234 = <port >
shadowSocks1 => fragmentation

سوال پنجم :

- 1 - 500 internal server error
- 2 - 401 unauthorized
- 3 - 301 moved permanently
- 4 - 429 too many requests
- 5 -
- 6 - 403 forbidden

سوال ششم :

1 (استفاده از سرور های داخلی دانشگاه بجای سرور های مرکزی برای دریافت پکیج ها از پکیج منیجر

این پروکسی از نوع forward می باشد.

2 (تقسیم درخواست ها بر اساس مبدا جغرافیایی درخواست ها

این پروکسی از نوع reverse است زیرا با توجه به ویژگی خاصی از client ها آن ها را دسته بندی می کند. برای مثال در بازی های آنلاین هر شخص به نزدیکترین سرور متصل می شود و با افراد نزدیک خود بازی می کند.

3 (اضافه کردن قابلیت احراز هویت برای یک سرویس قدیمی

به نظرم این پروکسی از نوع reverse می باشد زیرا در سمت سرور قرار دارد.