تمرین دوم شبکه های کامپیوتری

سوال اول)

لایه اپلیکیشن، بالاترین لایه موجود در پروتکل TCP/IP است که پروتکل ها و ساختاری را فراهم میکند تا کاربر بتواند با آن ارتباط برقرار کند. از طرفی در پروتکل OSI این لایه ترکیب لایه های Session، Presentation و Application است. در این لایه بیشتر فعالیت های High-Level نظیر صفحه های لاگین، انتقال فایل ها و هندل کردن ارور های برای پیام ها به صورت کلی است. این لایه همچنین از پروتکل های زیادی استفاده میکند از جمله HTTP برای دسترسی به اینترنت، FTP برای انتقال فایل ها، SMTP برای ارسال ایمیل و ... .

این لایه وظیفه نظارت و فعالیت های نزدیک سخت افزاری ندارد و به حالت Abstract به دیتا منتقل شده نگاه میکند. به طوری که درخواست های کاربر به و از این لایه به لایه های پایین تر مانند Transport، Internet و Network منتقل میشود.

سوال دوم)

الف) از مهم ترین و بارز ترین تفاوت های این دو معماری میتوان به این مورد اشاره کرد که در شبکه Peer-to-Peer هر Node میتواند درخواست برای سرویس کند و یا سرویسی به Node های دیگر بدهد یا به عبارتی هر عضو شبکه میتواند گیرنده و/یا دریافت کننده باشد. این در صورتی است که در معماری Client-Server، کلاینت درخواست سرویس داده و سرور به درخواست کلاینت پاسخ میدهد. به طور خلاصه معماری Peer-to-Peer یک شبکه Decentralized بوجود می آورد در صورتی که معماری Client-Server یک شبکه Centralized بوجود می آورد.

از نقاط اشتراک میتوان به آن اشاره کرد که هدف از طراحی هر دو معماری، ساخت شبکه ای برای انتقال فایل های گوناگون و اپلیکیشن ها است. برای مثال BitTorrent یک شبکه Peer-to-Peer بوده و Netflix یک شبکه Client-Server ولی وظیفه هر دو صرفه نظر از نحوه پیاده سازی، یکسان است.

ب) از کاربرد های بارز معماری Client-Server، نظارت بر امنیت شبکه و فایل های انتقالی است. به گونه ای که سیستمی وجود دارد (معمولا Server) که نظارت دقیق به تمام اطلاعات داشته و در صورت وجود مشکل ارتباط را کنترل میکند. با ارتقا دادن سیستم مرکزی (Server) میتوان سرعت شبکه را کنترل کرد زیرا دیتا بیشتر از سمت Server به Client منتقل میشود.

از طرفی معماری Peer-to-Peer کاربرد های متفاوتی دارد. برای مثال هیچ Node مرکزی وجود نداشته و هر کاربری میتواند به سادگی به شبکه متصل شود و به صورت همزمان Client و Server باشد. همچنین این نوع شبکه ها هزینه کمتری برای پیاده سازی و Maintain شدن نیاز دارد زیرا بار عمده ای از هزینه بر عهده خود کاربران است. به عنوان مورد سوم کارایی شبکه بر عهده کاربران آن است به گونه ای که هر چه سخت افزار های بیشتر و قوی تری (با سرعت بالاتر) به شبکه متصل باشند و در آن Contribute کنند کارایی و عملکرد شبکه افزایش میابد.

ج) از آنجایی که با استفاده از این طراحی میتوان شبکه ای ساخت که هیچ واحد مرکزی در آن وجود نداشته و در نتیجه نظارتی هم بر آن ندارد (به عبارتی Decentralized است) و با هدف اصلی شبکه Blockchain هم تفاهم نظر دارد، پس انتخاب خوبی است. از دگر دلایل استفاده از این طراحی میتوان به هزینه کمتر آن برای Implementation و Maintenance اشاره کرد و

تمرین دوم شبکه های کامپیوتری

همچنین Privacy-Friendly بودن آن به دلیل نظارت پایین. همچنین این شبکه ها توسعه پذیر تر بوده و با افزایش تعداد Node ها کارایی آن نه تنها کاهش نیافته، بلکه افزایش میابد.

سوال سوم)

الف)

- انتقال داده با قابلیت اطمینان: Transmission Control Protocol (TCP)
- امنیت: Transport Layer Security (TLS)
- ارتباط یک فرستنده با چند گیرنده: UDP/IP و RTP/IP
- گذر دهی بالا: Network Block Transfer Protocol (NETBLT)
- کنترل جریان: Logical Link Control (LLC)
- تضمین دریافت داده در زمان مشخص: Transmission Control Protocol (TCP)
- توانایی ارسال پیغام بزرگ: File Transfer Protocol (FTP)

ب) پروتکل TCP یک پروتکل Connection-Oriented بوده و از طرفی سرعت پایین تری دارد. این به خاطر آن است که در هر انتقال امکان Recovery بسته های ارسالی را به ما میدهد. این در صورتی است که پروتکل UDP و پروتکل Connectionless بوده و سرعت بالاتری نسبت به TCP دارد اما دیگر قابلیت Recovery بسته های ارسالی موجود نیست. به عبارت خلاصه پروتکل TCP سرعت را فدای قابل اطمینان بودن کرده در صورتی که پروتکل UDP قابل اطمینان بودن را فدای سرعت کرده. پس اگر سرعت بالاتر بخواهیم UDP را انتخاب میکنیم (در مورد Reliable Data Transfer صحبتی نشده پس مورد برسی قرار ندادم)

ج) بله، اگر از پروتکل TCP استفاده کنیم! این پروتکل از رسیدن صحیح و سالم بسته به مقصد مطمئن میشود، یعنی در صورت بروز هر مشکلی و منتقل نشده بسته ها دوباره ارسال می شوند. در مورد تاخیر با استفاده از Time-out در این پروتکل بسته ها مجددا ارسال میشوند. همچنین برای Bandwidth پایین این پروتکل میتواند زمان Time-out خود را متناسب با وضعیت شبکه قرار دهد. به عبارتی پروتکل TCP یک پروتکل قابل اعتماد برای انتقال داده است و سعی میکند جدا از وضعیت شبکه و محدودیت های موجود بسته را منتقل کند.

سوال چهارم)

الف) در پروتکل HTTP/1.1 به جای باز و بسته کردن کانکشن برای هر درخواست، یک کانکشن پایدار بین فرستنده (ها) و گیرنده (ها) برقرار میشود که این قابلیت را به شبکه میدهد که بتوان چندین درخواست را Pipeline کرده و با هم ارسال کرد که باعث میشود تعداد بسته های کمتری مبادله شوند و سرعت افزایش پیدا کند. به عبارتی این پروتکل قابلیت آن را میدهد که چندین درخواست یا پاسخ درخواست داخل یک قطعه TCP ارسال شوند.

هر کدام از گیرنده ها یا فرستاده ها با ارسال Close Token در هدر خود میتواند کانکشن را قطع کند.

تمرین دوم شبکه های کامپیوتری

ب) پروتکل HTTP به تنهایی هیچ قابلیت رمزنگاری ندارد و به سادگی میتوان محتویات بسته ها را مشاهده کرد. اما پروتکل HTTPS که بعد ها وارد بازار شد قابلیت رمزنگاری دارد به طوری که دیتا قبل از ارسال Encrypt شده و بعد از دریافت Decrypt میشود. نوع رمزنگاری پروتکل HTTPS از نوع Asymmetric Encryption است که با استفاده از دو کلید Public و Private صورت میگیرد. (این روش امنیت بالاتری نسبت به Symmetric Encryption میدهد که صرفا باس استفاده از یک کلید رمزنگاری میشود)

ج) بله همانطور که در بخش **الف** توضیح داده شد HTTP/1.1 یک کانکشن پایدار بین یک یا چند فرستنده و یا یک یا چند گیرنده برقرار میکند زیرا این قابلیت را فراهم میکند که چندین درخواست یا پاسخ درخواست ها با هم Batch یا Pipeline شوند.

د) سرور ها در این پروتکل از یک Timeout استفاده میکنند که فقط از طرف سرور قابلیت تغییر دارد و کلاینت صرفا میتواند با استفاده از هدر Keep-Alive مدت زمان آنرا از سرور درخواست کند. این هدر مشخص میکند چند سرور چه مدت برای درخواست کلاینت منتظر مانده و همچنین چند درخواست قبول میکند تا زمانی که کانکشن را ببندد.

زمانی که کلاینت یا سرور بخواهند ارتباط را قطع کنند **باید** به یکدیگر اطلاع دهند تا حین قطع ارتباط بسته ای درحال انتقال نباشد. پس هر دو طرف باید مراقب از دست رفتن دیتا هنگام قطع کانکشن باشند.

سوال پنجم)

الف) مدت زمان برای دریافت IP Address به وسیله DNS ها:

$$RTT_1 + RTT_2 + \cdots + RTT_n$$

از طرفی مدت زمان RTT_0 برای برقراری TCP Connection نیاز است و RTT_0 برای دریافت دیتا مورد نیاز (در اینجا فایل HTML). پس مدت زمان کلی برابر است با:

$$(2 \times RTT_0) + RTT_1 + RTT_2 + \cdots + RTT_n$$

ب) هر بسته ۲ برابر RTT_0 زمان نیاز دارد تا TCP برقرار شده (چون persistent و موازی نیست) و ۵ شی داریم. پس زمان مورد نیاز:

$$(10 \times RTT_0) + RTT_1 + RTT_2 + \cdots + RTT_n$$

سوال ششم)

سوال هفتم)

الف) URL درخواستی:

http://gaia.cs.umass.edu/cs453/index.html

تمرین دوم شبکه های کامپیوتری

ب) نسخه استفاده شده HTTP/1.1 است.

ج) در روش Non-Persistent برای هر درخواست یک ارتباط TCP برقرار شده و بعد از دریافت پاسخ از بین میرود. در روش Persistent یک ارتباط TCP برقرار شده و درخواست ها درون آن منتقل می شوند. در این پیام از Persistent استفاده شده به دلیل مشاهده Keep-Alive: 300 که زمان Time-out سرور در صورت دریافت نکردن درخواست را بیان می کند.

د) با توجه به مشاهده 200 OK در شکل **ب** که پاسخ سرور به درخواست است، بله شی مورد نظر موجود بوده.

ه) با توجه به شکل ب، زمان دریافت پاسخ برابر Tue, 07 Mar 2008 12:39:45GMT است.

و) با توجه به شکل **ب**، Last-Modified برابر Sat, 10 Dec2005 18:27:46 GMT است.

ز) با توجه به شکل **ب**، مقدار Content-length برابر ۳۸۷۴ بوده و نوع آن Byte است. پس ۳۸۷۴ بایت بازگردانده شده.

ح) این اطلاعات در هیچ جای پاسخ سرور وجود ندارد و نمیتوان از با استفاده این اطلاعات جواب این سوال را داد):

ط) با توجه به شکل **الف**، از GET استفاده شده است.

سوال هشتم)

الف) نیمی از درخواست ها از پروکسی و بقیه به صورت عادی عمل میشوند. پس:

$$\frac{1}{2} \times \frac{4{,}000}{10{,}000{,}000} + \frac{1}{2} \times \frac{4{,}000}{100{,}000{,}000} + 2 = 2.00022$$

سوال نهم)

الف) پروتکل FTP برای انتقال فایل در شبکه استفاده می شود که نوعی پروتکل برای انتقال اطلاعات بر روی Cloud است اما برای اطلاعات حجیم که در پروتکل های ساده دیگر قابل گنجایش/انتقال نیست. طراحی این پروتکل Client-Server است و این امکان را به میدهد تا برای دسترسی به این سرویس نیاز به Authentication هم باشد (قابلیت آن). این پروتکل معمولا رمزنگاری شده و روش های SSL، TSL و SSH. این پروتکل میتواند active یا passive باشد که در مورد اول کلاینت منتظر کانکشن های دیتا سرور می ماند. در حالت passive معمولا کلاینت پشت یک firewall بوده و نیاز به پورت متفاوتی برای برقراری ارتباط دارد.

ب) در حالت In-Band دستورات کنترلی از همان Band که دیتا نظیر صدا یا تصویر و ... منتقل میشوند، منتقل میشود. این در حالی است که در Out-of-Band دستورات کنترلی از Band خارج یا حتی در شبکه متفاوتی از آنهایی که بالاتر گفته شد ارسال شوند. به عبارتی در Out-of-Band دیتا خارج از حوزه سیستم عامل ارسال میشود که برخلاف In-Band است.

از مزایای In-Band نیازی به تماس فیزیکی نداشته و دیتا معمولا بر روی SSH یا Telnet ارسال میشود و سرعت بالایی دارد. از معایای این سیستم آن است که اگر شبکه غیرفعال باشد امکان برقراری ارتباط ندارم. برای استفاده از Out-of-Band میتوان از

تمرین دوم شبکه های کامپیوتری

Console استفاده کرد و زمانی که شبکه غیرفعال باشد میتوان از جایگزین آن استفاده کرد و مانند قبلی نیاز به تماس فیزیکی ندارد. از معایب آن سرعت پایین آن و نیاز داشتن به IP و Port صریح است.

برای In-Band میتوان LAN را مثال زد. برای Out-of-Band میتوان SMS را مثال زد.

سوال دهم)

در اینجا MTA مخفف Message Transfer Agent است. فرستنده Spam کاربر asusus-4b96 بوده به دو دلیل: ۱) عبارت Spam Firewall در بعد آن و ۲) localhost عه :

سوال یازدهم)

فرمول توضیح در شبکه Peer-to-Peer:

$$D_{p2p} = \max\left(\frac{F}{U_s}, \frac{F}{D_{min}}, \frac{N \times F}{U_s + (N \times U_i)}\right)$$

$$\frac{F}{U_s} = \frac{1 \times 1024}{30} = 34$$

$$\frac{F}{D_{min}} = \frac{1024}{2} = 512$$

U / N	*10*	*100*	*1000*
300Kbps	*512*	*1726*	*3170*
700Kbps	*512*	*1041*	*1434*
2Mbps	*512*	*512*	*512*

سوال دوازدهم)

الف) اگر سرور به صورت همزمان به همه کلاینت ها فایل را بفرستد توزیع زمانی $\frac{U_s}{N}$ خواهد داشت. و چون فرض داریم $\frac{U_s}{N} \leq D_{min}$ پس زمان همه کلاینت ها برای دریافت فایل $\frac{F \times U_s}{N}$ میشود.

ب) میدانیم سرعت آپلود سرور برابر $\frac{U_s}{N}$ است که از مینیمم سرعت دانلود کلاینت ها بیشتر است. چون سرعت دانلود کلاینت ها برابر مینیمم آنهای یعنی D_{min} میشود پس با توضیح زمانی $\frac{F}{D_{min}}$ میشود.

پ) طبق دو بخش بالا میدانیم توزیع زمانی یکی از دو حالت $\frac{F \times U_s}{N}$ و یا $\frac{F}{D_{min}}$ میشود. حال باید بدترین حالت را در نظر بگیریم که میشود $\max\left(\frac{F}{D_{min}}, \frac{F \times U_s}{N}\right)$

سوال سیزدهم)

تمرین دوم شبکه های کامپیوتری

الف) طبق فرض میدانیم سرعت آپلود U_s از میانگین سرعت آپلود نود های دیگر کمتر است. پس یعنی حداقل یک Node وجود دارد که سرعت آپلود کمتری نسبت به U_s دارد. از آنجایی که U_s تبدیل به Bottleneck میشود پس توضیح زمانی $\frac{F}{U_s}$ میشود.

ب) اگر U_s سرعت آپلود بیشتری نسبت به میانگین داشته باشد یعنی Node وجود دارد که سرعت آپلود کمتری نسبت به U_s دارد و آن Node موجب Bottleneck شدن شده و زمان توضیح را همانگونه که نوشته شده تنظیم میکند.

ج) در حالت کلی برای Peer-to-Peer دو حالت داریم که در دو بخش بالا مطرح شده اند. از آنجایی که باید بدترین حالت (اینجا بدترین زمان) را در نظر بگیریم پس ماکسیمم این دو را حساب میکنیم.

کیوان اپچی حق - ۹۸۳۱۰۷۳