تمرین سوم شبکه های کامپیوتری

سوال اول)

الف) در جست و جو بازگشتی، یک DNS با DNS های دیگر ارتباط برقرار میکند تا IP درخواستی توسط کلاینت را یافته و باز گرداند. این در تمایز با جست و جو تکرار شونده است که کلاینت مستقیما با سرور DNS مسئول جست و جو ارتباط برقرار می کند.

ب) زمانی که Caching داشته باشیم زمان RTT برای حالت برگشتی کمتر است. در روش بازگشتی و در صورت کش نشدن صرفا 2.RTT1 + RTT داریم. برای روش دیگر 2.RTT1 + 2.RTT داریم.

پ) در صورتی که هر دو روش قابلیت Caching داشته باشد هیچ تفاوتی با هم ندارند.

سوال دوم)

یک رکورد MX که مختصر Mail Exchange است ایمیل ها را به سرور های اصلی ارسال می کند. این رکورد مشخص می کند که چگونه (از چه مسیر هایی) به سرور اصلی بروند که توسط SMPT کنترل می شود. MX هم مانند CNAME باید همیشه به یک Domain دیگر اشاره کند.

در این رکورد ها متغیر Priority اهمیت رکورد را مشخص می کند به طوری که هرچه مقدار آن پایی تر باشد پروتکل SMTP سعی بر سریع تر ارسال کردن آن نسبت به دگر رکورد ها دارد. سیستم Message Transfer Agent مسئول استعلام گرفتن رکورد های MX است. هنگامی که یک کاربر یک ایمیل ارسال می کند، MTA یک درخواست DNS برای شناسایی سرورهای ایمیل برای گیرندگان ایمیل ارسال می کند. MTA یک اتصال SMTP با آن سرورهای ایمیل برقرار می کند، که با دامنه های اولویت بندی شده شروع می شود.

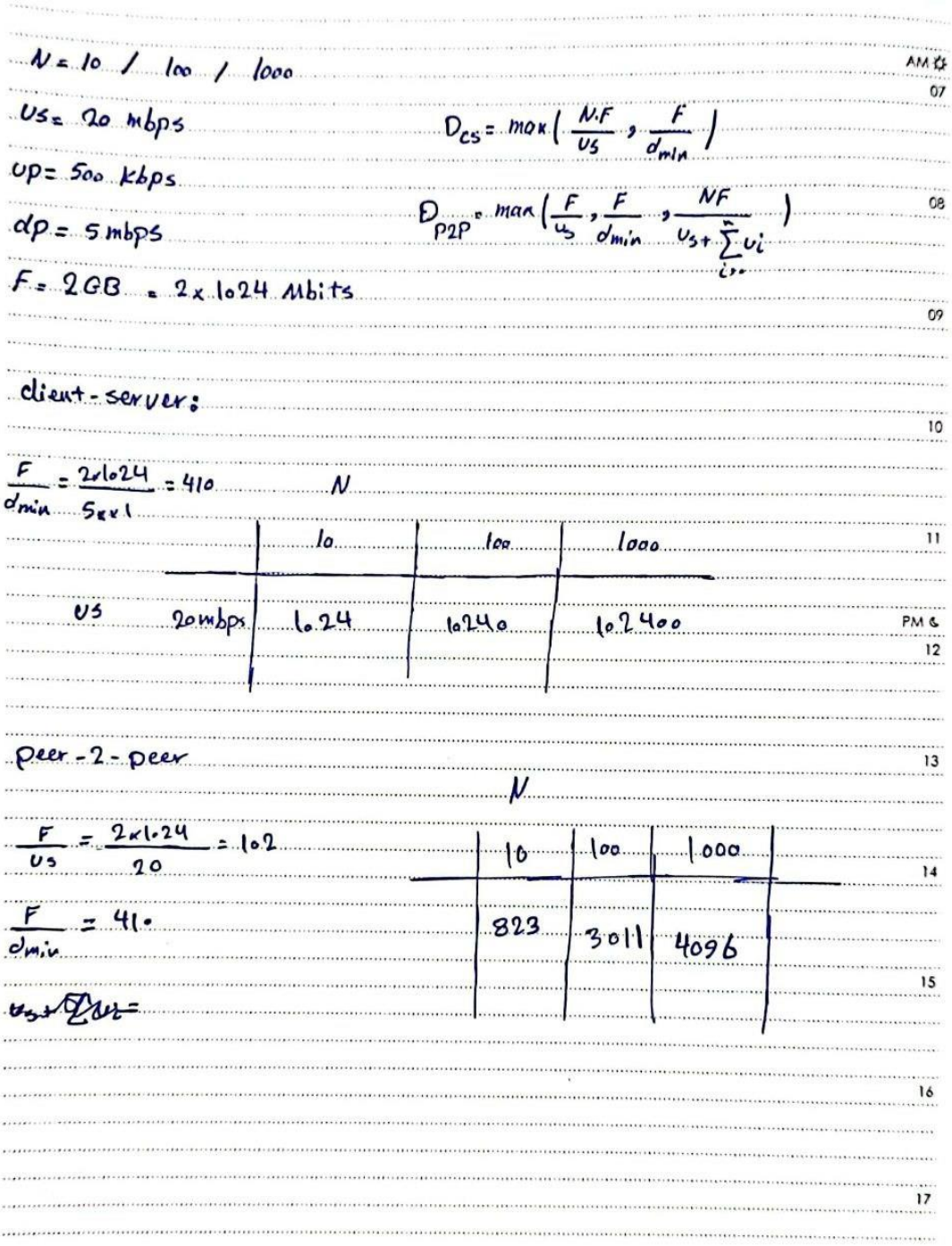
سوال سوم)

دلیل بوجود آمدن Local DNS Cache افزایش سرعت فرایند پاسخگویی به درخواست کلاینت است. از طرفی می دانیم DDoS با ارسال تعداد زیادی درخواست به یک IP مانع فعالیت آن می شود. حال اگر یک حمله DDoS به یک IP خاص داشته باشیم به دلیل وجود Local DNS Caching درخواست ها در همان Local DNS به سرعت پاسخ داده می شوند (زیرا یک درخواست خاص با تعداد زیاد فرستاده میشود و تنوع آن زیاد نیست). در صورتی که Local DNS قادر به پاسخ گویی به ایم حجم از درخواست نباشد Crash کرده یا سرعت آن پایین می آید ولی در هر صورت DNS اصلی آسیبی نخواهد دید؛ زیرا عمده این درخواست ها در مرحله اول به DNS اصلی نمی رسند.

اما با این وجود نمی توان گفت ۱۰۰٪ در مقابل این نوع حملات مقاوم هستند زیرا همانطور که DNS ها بهبود پیدا میکنند، حملات هم ارتقا پیدا می کنند (:

سوال چهارم)

تمرین سوم شبکه های کامپیوتری



سوال پنجم)

رویتر ها بخشی از گروه محسوب نمی شوند. پس فقط N شبکه داریم که جفت ارتباط TCP دارند. تعداد گره های موجود در شبکه برابر N است زیرا N شبکه فعال داریم. از طرفی اگر به این سوال از دید مسئله هندسی نگاه کنیم، باید تعداد ضلع و قطر یک N ضلعی را بدست آوریم. تعداد ضلع برابر N و تعداد قطر برابر $\frac{N*(N-3)}{2}$ است که در مجموع برابر $\frac{N*(N-1)}{2}$ یال خواهد داشت.

سوال ششم)

تمرین سوم شبکه های کامپیوتری

الف) پروتکل SMTP یک بار هنگام ارسال از کلاینت من به سرویس دهنده من و یک بار هم از سرویس دهنده من به سرویس دهنده دوست من نقش دارد (۲ از ۳ ارتباط را پوشش می دهد) اما زمانی که کلاینت دوست من ایمیل را از سرویس دهنده اش دریافت می کند، از پروتکل Mail Access (POP3 یا IMAP) استفاده می کند زیرا این پروتکل مختص دریافت ایمیل است (و نه ارسال آن)

ب) پروتکل SMTP صرفا قابلیت ارسال کد های ASCII را دارد پس نمی تواند به تنهایی عکس، فیلم، صوت یا هر چیز دیگری را ارسال کند. پروتکل MIME برای Encode کردن عکس، فیلم و صوت و هر داده غیر ASCII استفاده میشود؛ به طوری که ابتدا داده از این پروتکل عبور کرده و در قالب ASCII رمز گزاری شده و سپس به SMTP میرود. بعد از ارسال به سرور و دریافت ایمیل توسط گیرنده دوباره ایمیل از MIME و سپس از SMTP عبور داده می شود.

پ) پروتکل SMPT صرفا انتقال دهنده پیام است یا به عبارتی تنها قادر به Push کردن پیام است. در صورتی که برای دریافت ایمیل از سرور به کلاینت نیاز به پروتکل دیگری به نام IMAP یا POP3 نیاز داریم تا پیام را از سرور Pull کند.

پروتکل POP3 به صورت پیش فرض و برای ارتباط Non-Encrypted از پورت ۱۱۰ استفاده کرده و برای ارتباط Encrypted از پورت ۹۹۵ استفاده می کند. همچنین IMAP برای ارتباط رمزنگاری شده از پورت ۱۴۳ و برای ارتباط رمزنگاری نشده از ۹۹۳ استفاده می کند. پروتکل SMPT هم به صورت پیش فرض از پورت ۲۵ استفاده کرده و پورت ۴۶۵ برای ارتباط رمزنگاری شده است.

سوال هفتم)

الف) در روش download-and-delete پیام از سرور دانلود و پس از خوانده شدن از روی سرور پاک میشود؛ پس پیام فقط روی لوکال باقی می ماند. از طرفی در روش download-and-keep مانند روش قبلی پیام را از سرور دانلود کرده اما پس از خوانده شدن آنرا ار روی سرور پاک نمیکند.

از نظر کاربر حالت دوم بهتر است زیرا امروزه ما ایمیل های خود را از موبایل، تبلت و لپ تاپ میبینیم؛ پس نیاز داریم تا پیام ها همواره در همه دستگاه ها موجود باشند و روش اول مطرح شده یعنی download-and-delete خیلی مطلوب کاربر نیست.

ب) نسخه کامل شده:

C: dele 1
C: retr 2
S: (blah blah …
S: ……..blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

پ) نسخه download-and-keep:

C: retr 2
S: blah blah …
S: ……..blah
S: .
C: quit
S: +OK POP3 server signing off

سوال هشتم)

تمرین سوم شبکه های کامپیوتری

هرچند نمی توان مقدار دقیق را محاسبه کرد اما با استفاده از چندین فرمول می توان مقدار حدودی آن را تخمین زد. اگر فرض کنیم SampleRTT معادل مقدار زمانی است که یک بسته ارسال شده و Acknowledge شده باشد در اولین SampleRTT میتوان گفت EstimatedRTT که ترکیب وزن داری از SampleRTT جدید و EstimatedRTT قدیم است، یا به عبارتی:

$$EstimatedRTT = (1 - \alpha) . EstimatedRTT + (\alpha \cdot SampleRTT)$$

(ولی از آنجایی که یک ارتباط همواره پیوسته نیست و مقادیر پارامتر های فرمول بالا متغیر هستند) اگر طبق RFC 6298 مقدار α را قرار دهیم فرمول به شکل پایین تغییر میکند:

$$EstimatedRTT = (0.875 \cdot EstimatedRTT) + (0.125 \cdot SampleRTT)$$

سوال نهم)

دستورات پروتکل FTP بر اساس نحوه کارکردی که دارند به سه دسته کلی تقسیم می شوند:

- دستورات کنترلی: معمولا این دستورات بخشی از روند Login، Authentication و دسترسی دادن به منابع را شامل می شوند.
- دستورات انتقال پارامتر ها: این نوع دستورات پارامتر هایی را مشخص میکنند که بوسیله آنها انتقال دیتا صورت می گیرد. برای مثال نوع داده انتقال مشخص شده و یا Active و Passive بودن ارتباط مشخص می شود. (مانند هدر های پروتکل HTTP دیتا کنترلی منتقل میکنند)
- دستورات سرویس FTP: این دسته عمده دستورات را شامل می شود که برای کارکرد هسته مرکزی این پروتکل مورد استفاده قرار میگیرند. این دستورات عملیات های فایلی مانند ارسال، دریافت، حذف یا تغییر اسم را انجام داده.

سوال دهم)

شباهت ها:

- هر دو DNS Load Balancing و HTTP Load Balancing کاربرد تقریبا مشابهی دارند، به گونه ای که بار بوجود آمده توسط درخواست های کاربران را بین سرور های مختلف پخش میکنند.
- هر دو سعی بر افزایش سرعت در پاسخگویی به درخواست کاربران را با Distributed کردن درخواست ها دارند.
- در صورت از کار افتادن یک یا چند سرور، بار موجود بر روی سرور های دیگر افتاده و افت سرعت خواهند داشت.
- در برابر حملاتی مانند Denial-of-Service Attack مقاوم هستند زیرا در صورت از کار افتادن یک سرور، سرور های دیگر قادر به پاسخ گویی خواهند بود.

تفاوت ها:

- عمل DNS Load Balancing خیلی از HTTP Load Balancing سریع تر است؛ زیرا این عمل Connectionless است و نیازی به برقراری ارتباط مستقیم و طرفه در واحد زمان ندارد.
- در DNS Load Balancing صرفا یک DNS Request پاسخ داده می شود در صورتی که در HTTP Load Balancing نیاز به یک ارتباط TCP جدا داریم.
- در HTTP Load Balancing قابلیت گنجایش محتوا و اطلاعات بیشتری برای هر درخواست داریم. زیرا در یک درخواست HTTP میتوان URL، پارامتر ها و هدر های HTTP را هم داشت.
- DNS Loading Balance انعطاف پذیری بیشتری دارد زیرا می تواند با هر پروتکل ارتباط برقرار کند مانند FTP، RTMP و حتی خود HTTP