سوال ۷

ما در اینجا می خواهیم مدت زمان بین وقتی که کاربر درخواست میدهد تا وقتی که پاسخ را از سرور موردنظر میگیرد حساب کنیم.

برای این محاسبه طبق فرض سوال ابتدا باید از طریق DNS ها IP مقصد را پیدا کنیم.

با توجه به فرض سوال، مدت زمانی که صرف پیدا کردن آدرس سرور موردنظر کاربر است مقدار زیر را دارد:

$$RTT_1 + RTT_2 + RTT_3 + \ldots + RTT_n = total\ time\ for\ finding\ the\ ip\ address$$

بعد از اینکه ip یافته شد، یک request از سمت کاربر فرستاده می شود که مدت زمان رسیدن آن به سرور و بازگشت پاسخ آن درخواست داده شده RTT_0 است.

منتها قبل از اینکه فرآیند request and response انجام بشود، نیازمند برقراری یک TCP connection بین کاربر و سرور موردنظر می باشیم که مدت زمان آن نیز RTT_0 است.

در نتیجه کل زمان فرآیند برابر است با:

$$total\ time\ for\ finding\ the\ ip\ address + 2RTT_0$$

سوال ۸

الف

این بخش مانند سوال ۷ است. در اصل ما مجددا همان زمان طی شده برای یافتن ip و برقراری ارتباط TCP را داریم، همچنین فرآیند request and response را خواهیم داشت. منتها فایل HTML که از سرور دریافت میشود به ۸ شی دیگر در سرور نیاز دارد، بنابراین ۸ فرآیند request and response دیگر به همراه ارتباط TCP خواهیم داشت بنابر این $16RTT_0$ دیگر زمان نیاز داریم. در نتیجه در مجموع خواهیم داشت:

$$total\ time\ for\ finding\ the\ ip\ address\ +\ 18RTT_0$$

ب

در این بخش، تفاوت با بخش قبلی در این است که به صورت همزمان ۵ ارتباط درجریان به صورت موازی داریم، در نتیجه ۸ درخواست اضافی که برای پاسخ HTML سرور نیاز هستند، در مدت زمان $4RTT_0$ فرآیند خودشان را به پایان می رسانند (درزمان اول ۵ و در زمان بعدی ۳ فرآیند تکمیل می شوند که در مجموع ۲ تا $2RTT_0$ زمان میبرد)

پس مجموع زمان های سپری شده برابر است با:

$$total\ time\ for\ finding\ the\ ip\ address\ +\ 4RTT_0$$

ج

در این بخش ما یک ارتباط مداوم داریم که ارتباط را به صورت یک pipeline درست می کند، در نتیجه تمام فرآیندها به صورت موازی درخواست های خود را اجرا می کنند و کل مدت زمان برای ۸ فرآیند برابر RTT_0 می باشد.

پس مجموع زمانی که صرف می شود برابر است با:

$$total\ time\ for\ finding\ the\ ip\ address\ +\ 2RTT_0\ +\ RTT_0$$

سوال ۱۰

خیر، استفاده از HTTP مداوم در اینجا آنقدر سریعتر از HTTP غیر مداوم نمی باشد.

اگر فرض کنیم مقدار P برابر مدت زمان یک طرفه ارتباط میان کاربر و سرور باشد، اول مدت زمان دانلود هایی با HTTP مداوم را محاسبه میکنیم:

$$3 * (1.34 + P) + 667 + P + 10 * (1.34 + P + 667 + P) = 7354.42 + 24P$$

حال بیایید برای HTTP غیر مداوم این زمان را حساب کنیم:

$$3 * (1.34 + P) + 667 + P + 10 * (3 * (13.4 + P + 6670 + P) = 7377.42 + 8P$$

حال که این دو زمان را به دست آوردیم، نسبت میگیریم:

$$\frac{7354.42 + 24P}{7377.42 + 8P} \rightarrow for\ P\ like\ speed\ of\ light \approx 1$$

در نتیجه مشاهده کردید که استفاده از HTTP مداوم تفاوتی با HTTP غیر مداوم نخواهد داشت.

سوال ۱۵

عبارت MTA مخفف Mail Transfer Agent می باشد.

ما برای اینکه مشخص کنیم کدام یک از MTA ها dishonest بوده اند، به این نکته توجه میکنیم که هر MTA صادق یا درست، باید مشخص کند که ایمیل را از کجا دریافت کرده است. و در مسئله گفته شده است که تنها سازنده می تواند dishonest باشد و باقی agent ها honest هستند.

در ایمیل مشخص شده فقط agent ایی با آدرس 58.88.21.177] asusus-4b96 مشخص نکرده است که ایمیل را از کجا دریافت کرده است، بنابراین این عامل یک عامل dishonet است و بنابر فرض مسئله همین عامل سازنده می باشد.

سوال ۱۶

از دستور برای تعیین اینکه کدام پیام ها در سرور از قبل دیده شده اند استفاده می شود.

خود دستور UIDL مخفف unique Id listing می باشد و زمانی که توسط یک کلاینت POP3 این دستور را اجرا می کند، سرور با یک شناسه یکتا برای تمام پیام های حال حاضر موجود در mailbox کاربر پاسخ میدهد تا تمام فایل های دیده شده را مشخص کند.

این دستور برای دنلود و نگه داشتن بسیار مفید است.

سوال ۱۸

الف

WHOIS یک پروتکل پرس و جو و پاسخ است که به طور گسترده برای پرس و جو از پایگاه های داده استفاده می شود که کاربران ثبت نام شده یا صاحبان یک منبع اینترنتی را ذخیره می کند، مانند نام دامنه، بلوک آدرس IP یا یک سیستم مستقل، اما همچنین برای طیف وسیع تری از منابع اینترنتی استفاده می شود.

ب

Google.com => whois.markmonitor.com by https://who.is/whois/google.com

Stackoverflow => whois.name.com by https://who.is/whois/stackoverflow.com

پ

Local Domain: www.google.com

Name: MarkMonitor, Inc.

Whois Server:

whois.markmonitor.com

Referral URL:

http://www.markmonitor.com

Status:

clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)

clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)

clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)

serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)

serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)

serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)

Name Servers:

ns1.google.com

216.239.32.10

ns2.google.com

216.239.34.10

ns3.google.com

216.239.36.10

ns4.google.com

216.239.38.10

Local Domain: www.stackoverflow.com

Name: Name.com, Inc.

Whois Server: whois.name.com

Referral URL: http://www.name.com

Status: clientTransferProhibited https://www.icann.org/epp#clientTransferProhibited

Name Servers:

ns-1033.awsdns-01.org

205.251.196.9

ns-358.awsdns-44.com

205.251.193.102

ns-cloud-e1.googledomains.com

216.239.32.110

ns-cloud-e2.googledomains.com

216.239.34.110

Local domain: www.github.com

Name: MarkMonitor, Inc.

Whois Server: whois.markmonitor.com

Referral URL: http://www.markmonitor.com

Status:

clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)

clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)

clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)

Name Servers:

dns1.p08.nsone.net

198.51.44.8

dns2.p08.nsone.net

198.51.45.8

dns3.p08.nsone.net

198.51.44.72

dns4.p08.nsone.net

198.51.45.72

ns-1283.awsdns-32.org

205.251.197.3

ns-1707.awsdns-21.co.uk

205.251.198.171

ns-421.awsdns-52.com

205.251.193.165

ns-520.awsdns-01.net

205.251.194.8

ت

Github.com

192.30.255.112

No, it has only one ip address

ث

Harvard university

64.64.9.47

ج

یک مهاجم می تواند از پایگاه داده whois و ابزار nslookup برای تعیین محدوده آدرس IP، آدرس های سرور DNS و غیره برای موسسه مورد نظر استفاده کند.

چ

با تجزیه و تحلیل آدرس منبع بسته های حمله، قربانی می تواند از whois برای به دست آوردن اطلاعات در مورد دامنه ای که از آن حمله می شود استفاده کند و احتمالاً به مدیران دامنه مبدا اطلاع دهد.

سوال ۲۰

می‌توانیم به‌طور دوره‌ای از حافظه‌های نهان DNS در سرورهای DNS محلی یک عکس فوری بگیریم. وب سروری که بیشتر در کش های DNS ظاهر می شود، محبوب ترین سرور است. این به این دلیل است که اگر کاربران بیشتری به یک وب سرور علاقه مند باشند، درخواست های DNS برای آن سرور بیشتر توسط کاربران ارسال می شود. بنابراین، آن وب سرور بیشتر در کش های DNS ظاهر می شود.

سوال ۲۱

بله، ما می توانیم از dig برای پرس و جو از آن وب سایت در سرور DNS محلی استفاده کنیم. در این صورت آدرس در DNS محلی ذخیره شده می باشد و زمان دسترسی به آن ۰ میلی ثانیه است.

سوال ۲۲

Client-server

	N = 10	N = 100	N = 1000
U = 300 kbps	7680	51200	512000
U = 700 kbps	7680	51200	512000
U = 2 Mbps	7680	51200	512000

P2P

	N = 10	N = 100	N = 1000
U = 300 kbps	7680	26000	47600
U = 700 kbps	7680	15600	21250
U = 2 Mbps	7680	7680	7680

توضیح:

برای محاسبه حداقل زمان توزیع در Client-server از رابطه زیر استفاده میکنیم:

$$max\left(\frac{NF}{u}, \frac{F}{d}\right)$$

برای محاسبه حداقل زمان توزیع در P2P از رابطه زیر استفاده میکنیم:

$$max\left(\frac{F}{u}, \frac{F}{d}, \frac{NF}{\sum u}\right)$$