

شبکه‌های کامپیوتری

تمرین اول تئوری

نام دانشجو: آرش یادگاری

شماره دانشجویی: ۹۹۱۰۵۸۱۵

سوال ۱:

از آنجا که نوع روترهای مورد استفاده در جواب نهایی موثر است، فرض بر آن است که از مدل packet switching استفاده می‌شود و نه روش‌هایی که در سطح flit پاپیلین شده است.

$$(160 + 40) \times 4 \times \frac{1000}{20} \times \frac{10}{9} \simeq 4.4 \times 10^4 \text{ bps (byte per second)}$$

سوال ۲:

Post Office Protocol (POP)

یکی از پروتکل‌های مورد استفاده برای mail client ها می‌باشد تا بتواند ایمیل‌های موجود در یک mail server را دریافت کند. کاربر پس از وارد کردن user/pass ایمیل‌های خود را دانلود می‌کند و پس از دانلود ایمیل‌ها، سرور mail‌های دانلود شده را پاک می‌کند. در این صورت تنها client است که به mail‌های دانلود شده دسترسی دارد.

- + پیاده‌سازی ساده
- + سربرار پایین mail server ها
- + امکان مشاهده ایمیل‌های دانلود شده در حالت آفلاین (در صورتی که دانلود شده باشند)
- امکان دسترسی به ایمیل‌ها از طریق چند client وجود ندارد.
- در صورت از بین رفتن داده‌ها، امکان بازیابی آنها از طریق سرور وجود ندارد (هرچند می‌توان اینگونه تعریف کرد که ایمیل‌ها پس از دانلود پاک نشوند اما رفتار معمول آن به این صورت نیست)
- احتمال دسترسی به محتویات مخرب بیشتر می‌باشد زیرا که فایل‌های مخرب بر روی سیستم client خوانده می‌شود و اسکن می‌شود.

Internet Message Access Protocol (IMAP)

یکی دیگر از پروتوکلهای مورد استفاده توسط mail client ها برای دریافت ایمیل ها از mail server می باشد. هدف طراحی آن، امکان دسترسی به ایمیل ها توسط چندین client مختلف می باشد در نتیجه ایمیل ها بر روی سرور باقی می ماند و پاک نمی شوند مگر آنکه client دستور پاک کردن آن را بدهد.

- + امکان بازیابی ایمیل ها در صورتی که client دچار مشکل شد.
- + امکان مدیریت ایمیل ها از طریق چندین client.
- سربرار سرور برای نگهداری ایمیل ها زیاد است. (نیاز به حافظه بزرگ بر روی سرور)
- امکان مشاهده ایمیل ها به صورت آفلاین وجود ندارد.
- از آنجا که client ها باید سینک باشند، سربرار سینک کردن نیز وجود دارد.

به طور کلی می توان گفت POP پروتکل سریع تری نسبت به IMAP می باشد. همچنین سربرار کمتری بر روی سرور می گذارد هرچند برخلاف IMAP نمی توان با استفاده از چندین client، به ایمیل ها دسترسی داشت. همچنین ذخیره داده ها بر روی سرور می باشد که باعث می شود بتوان برخلاف POP، از ایمیل ها بکاپ داشت هرچند که هزینه ذخیره سازی در این مدل بیشتر است.

سوال ۳:

پروتکل (Multipurpose Internet Mail Extension (MIME مانند یک افزونگی بر روی SMTP عمل می کند. MIME امکان ارسال داده هایی را محیا می کند که نمی توان آنها را به صورت کد 7 bit ASCII در آورد. روش کلی به این صورت است که ارسال کننده قبل از ارسال داده از طریق SMTP با استفاده از پروتکل MIME داده را تبدیل به داده های 7bit ASCII کرده و همچنین یک header نیز ارسال می کند تا در زمان دریافت ایمیل از طریق پروتکل هایی مثل POP، بتوان داده را به حالت اصلی بازگرداند.

MIME امکان ارسال داده ها به زبان هایی که نمی توان آنها را به صورت 7 bit ASCII در آورد را امکان پذیر کردند. همچنین امکان داده هایی مثل فایل، عکس و ویدئو نیز از طریق این پروتکل امکان پذیر شد.

سوال ۴:

در شرایط ایده آل (zero-traffic) و همچنین فرض 0 بودن زمان پردازش داده در router داریم:

$$Latency = L_{propagate} + L_{transmit} = \frac{Distance}{propagation\ Speed} + \frac{packet\ Length}{link\ bandwidth\ (bps)} =$$

$$= \frac{2500 \times 10^3}{2.5 \times 10^8} + \frac{1000 \times 8}{2 \times 10^6} = 14ms$$

همانطور که در فرمول بالا مشاهده نمودید، میزان تاخیر به فاصله و همچنین طول بسته وابسته می‌باشد. هرچه فاصله بیشتر باشد، زمان بیشتری برای انتقال داده در سیم صرف می‌شود. هرچه طول بسته بیشتر باشد، زمان انتقال آن به سیم طولانی تر خواهد بود.

سوال ۵:

فرض بر این است که browser استفاده شده مانند browserهای امروزی از روش parsing and rendering استفاده می‌کند به این معنا که پس از دریافت فایل اصلی و پارس کردن آن شروع به فرستادن درخواست‌هایی برای دریافت تگ‌هایی مثل عکس یا دیگر صفحات html می‌کند.
۱- زمان لازم برای DNS Lookup برابر است با:

$$DNS\ Lookup = \sum_{i=0}^n RTT_i$$

همچنین برای دریافت هر object در روش non-persistent نیاز به 2 RTT هستیم پس داریم:

$$HTTP\ Requests = (8 + 1) \times 2 RTT_0$$

در نتیجه زمان کل برابر است با:

$$Total\ Time = DNS\ Lookup + HTTP\ Requests = \sum_{i=0}^n RTT_i + 18 RTT_0$$

۲- زمان ارسال ۸ فایل برابر می‌شود با :

$$HTTP\ Requests = 2 RTT_0 + \text{ceil}(\frac{8}{5}) \times 2 \times RTT_0$$

در نتیجه زمان کل برابر می‌شود با:

$$Total\ Time: DNS\ Lookup + HTTP\ Requests = \sum_{i=0}^n RTT_i + 6 RTT_0$$

۳- با فرض آنکه تمامی فایل‌های لینک شده بر روی سرور دارای آبجکت اول واقع شده باشد، با داشتن یک کانکشن TCP می‌توانیم همه فایل‌ها را دریافت کنیم (فرض می‌کنیم ارتباط موازی وجود ندارد)

$$HTTP\ Requests = RTT_0 + RTT_0 + 8 \times RTT_0 = 10 RTT_0$$

$$Total\ Time: DNS\ Lookup + HTTP\ Requests = \sum_{i=0}^n RTT_i + 10 RTT_0$$

سوال ۶:


۱- ابتدا از قسمت files، فایل pcap را باز کرده و سپس از منوی statistics بر روی IPv4 زده و آدرس مقصد و پورت‌ها را کوئری می‌گیریم.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ Destinations and Ports	36263				0.1307	100%	44.2500	88.775
▼ 127.0.0.53	339				0.0012	0.93%	0.1100	124.206
▼ UDP	339				0.0012	100.0...	0.1100	124.206
53	339				0.0012	100.0...	0.1100	124.206
▼ 127.0.0.2	25				0.0001	0.07%	0.0600	50.232
▼ UDP	25				0.0001	100.0...	0.0600	50.232
58890	25				0.0001	100.0...	0.0600	50.232
▼ 127.0.0.1	35899				0.1294	99.00%	44.2500	88.775
▶ UDP	339				0.0012	0.94%	0.1100	124.206
▶ TCP	35560				0.1282	99.06%	44.2500	88.775

۲- از منوی endpoints پکت‌های پروتکل‌های TCP و UDP را براساس سایز مرتب می‌کنیم. بیشترین میزان داده به IP با شماره 127.0.0.1 و پورت 12345 ارسال شده است.

Ethernet · 1		IPv4 · 3	IPv6 · 1	TCP · 10690	UDP · 197		
Address	Port	Packets	Bytes ^	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
127.0.0.1	12345	31,000	1.892 MiB	15,500	817.383 KiB	15,500	1.094 MiB
127.0.0.1	8089	2,672	972.855 KiB	1,073	591.828 KiB	1,599	381.027 KiB
127.0.0.1	58878	1,515	516.604 KiB	909	218.735 KiB	606	297.869 KiB
127.0.0.1	58866	724	292.894 KiB	434	104.065 KiB	290	188.828 KiB
127.0.0.1	44600	399	161.162 KiB	239	57.139 KiB	160	104.023 KiB
::1	9229	1,108	90.891 KiB	554	40.035 KiB	554	50.855 KiB
127.0.0.1	37840	284	79.766 KiB	170	36.238 KiB	114	43.527 KiB
127.0.0.1	9614	280	79.516 KiB	112	43.422 KiB	168	36.094 KiB
127.0.0.1	9229	1,108	69.250 KiB	554	29.215 KiB	554	40.035 KiB
127.0.0.1	12222	500	31.250 KiB	250	13.184 KiB	250	18.066 KiB
127.0.0.1	44590	14	920 bytes	7	470 bytes	7	450 bytes
127.0.0.1	44616	14	920 bytes	7	470 bytes	7	450 bytes
127.0.0.1	58894	14	920 bytes	7	470 bytes	7	450 bytes
127.0.0.1	39584	12	768 bytes	6	444 bytes	6	324 bytes
127.0.0.1	33220	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	33250	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34050	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34280	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34292	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34582	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34840	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34856	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	34958	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	35078	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	35388	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	35762	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	37346	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	37596	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	38626	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	39814	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	39848	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	44174	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	44236	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	44270	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	44836	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	44882	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	45522	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	46088	10	640 bytes	5	370 bytes	5	270 bytes
127.0.0.1	57110	10	640 bytes	5	370 bytes	5	270 bytes

۳- با استفاده از اعمال فیلتر

 ip.addr == 127.0.0.1 && udp.srcport == 12345

پکت‌های ارسال شده را بررسی می‌کنیم. همانطور که مشاهده می‌کنید، هدر این پکت‌ها شامل ۲۰ بایت می‌باشند.

