

به نام یکتا سازنده هستی



## تمرین سری دوم

شبکه‌های کامپیوتری

نیم‌سال دوم ۱۴۰۰ - ۱۴۰۱

دانشکده‌ی برق و کامپیوتر

دانشگاه صنعتی اصفهان

---

اسفند ماه ۱۴۰۰

استاد درس:

جناب آقای دکتر حیدرپور

مسئولین تمرین:

مهرداد اشراقی - محمد جلالی - سپهر شیرانی - محمدمهدی برقی - امیر ارسلان یاوری

۱. به سوالات زیر به صورت کوتاه پاسخ دهید. (۵۰ نمره)

(آ) توضیح دهید چگونه مکانیزم Web caching باعث کاهش تاخیر دریافت یک شیء درخواستی خواهد شد؟ این مکانیزم برای کدام یک از شیء‌های درخواستی کاربرد دارد؟ چرا؟

با توجه به این که در این حالت اشیا درخواستی از یک سرور نزدیک تر به کلاینت می‌رسند مکانیزم Web Cache با ایجاد یک کاتکشن سریع تر و با کیفیت تر، به سرعت دریافت اشیایی که در حافظه سرور کش ذخیره شده‌اند کمک می‌کند. علاوه بر این با توجه به این که این مکانیزم ترافیک شبکه را کم می‌کند، سرعت دریافت اشیا کش نشده هم بیشتر شده و در کل کیفیت استفاده از تمام شبکه بیشتر خواهد شد.

(ب) چرا در پروتکل DNS از یک سرور مرکزی بجای چندین سطح سرور استفاده نشده است؟

به دو دلیل High Load و Single Point of Failure از سرور های مرکزی استفاده نمی‌شود

(ج) فرض کنید یک کاربر جدید بدون داشتن هیچ chunk ای به BitTorrent متصل می‌شود. از آنجایی که چیزی برای آپلود ندارد پس نمی‌تواند جزو چهار نفر اول لیست هیچکدام از کاربران دیگر قرار بگیرد. این نود چگونه اولین chunk خود را دریافت می‌کند؟

در سرویس بیت تورنت هر Peer به صورت رندوم یک نود دیگر را انتخاب کرده و آن را Unchoke می‌کند تا اجازه ورود به شبکه به یوزر های جدید را داشته باشد. همچنین می‌تواند اولین بسته ی خود را از سرور مرکزی دریافت کند.

(د) دو کاربر Alice و Bob قصد استفاده از سرویس‌های سایت alibaba.com را دارند. Bob می‌خواهد یک صفحه وب به آدرس <https://alibaba.com/index.html> را بازدید کند و Alice قصد دارد یک ایمیل به [info@alibaba.com](mailto:info@alibaba.com) ارسال کند. با فرض خالی بودن کش DNS این دو کاربر، بین DNS query و Response مربوط به درخواست‌های آن‌ها چه تفاوت‌هایی وجود دارد؟ در هنگام بازدید از سایت، از نوع A استفاده می‌شود و هنگام استفاده از ایمیل، نوع MX مورد استفاده قرار می‌گیرد.

(ه) پروتکل HTTP/2 از چه روش‌هایی برای ارتقا سرعت نسبت به HTTP/1.1 استفاده می‌کند؟

استاندارد HTTP/2 برای ارتقای سرعت نسبت به HTTP/1.1 از چندین روش استفاده می‌کند که عبارتند از:

#### ۱. Framing

وقتی که می‌خواهیم یک فایل بزرگ و چندین فایل کوچک‌تر را بوسیله HTTP ارسال کنیم، ممکن است فایل بزرگ در جلوی فایل‌های کوچک‌تر قرار گرفته و تا وقتی که ارسال آن تمام نشده فایل‌های کوچک‌تر نیز به دست کلاینت نرسند. برای حل این مشکل، استاندارد HTTP/2 دیتای ارسالی را به فریم‌های کوچک‌تر و هم‌اندازه تقسیم می‌کند و با این کار جلوی Head of line blocking را می‌گیرد.

#### ۲. Message Priroitization

در زمان ارسال درخواست به‌وسیله کلاینت، می‌توان اولویت خاصی را برای اشیا درخواستی مشخص کرد. سرور اشیا با اولویت بالاتر را زودتر ارسال می‌کند.

#### ۳. Server Pushing

وقتی که درخواستی برای سرور ارسال می‌شود، اگر شی درخواستی با اشیا دیگری در سرور مرتبط باشد، بدون گرفتن درخواست از سمت کلاینت، سرور می‌تواند این اشیا مرتبط را برای کلاینت ارسال کند.

۲. مرورگری برای بارگذاری یک صفحه WEB به صورت کامل نیازمند بارگیری ۶۰ شیء از سرور می‌باشد. برای این کار از پروتکل HTTP 1.1 و هدر If-Modified-Since استفاده می‌کند. قبل از ارسال درخواست‌ها، نصف اشیا در کش مرورگر وجود دارند که از نظر مرورگر، ۴۰ درصد از این اشیا کاملاً به‌روز بوده و نیازی به چک کردن ندارند و مابقی نیاز به چک کردن دارند و باید بروزرسانی شوند. همچنین بین کلاینت و سرور اصلی یک cache server وجود دارد. از بین تمام درخواست‌هایی که از سمت مرورگر به سمت کش سرور خواهیم داشت، نصف اشیا در آن به‌روز بوده و نیازی به چک ندارند، ۲۵ درصد نیاز به چک دارند اما نیازی به بروزرسانی نخواهند داشت و مابقی نیاز به بروزرسانی دارند. کل زمان سپری شده از آغاز درخواست اول تا پایان دانلود شیء آخر چقدر است؟ (۴۰ نمره)

توجه: کش سرور به صورت همزمان با مرورگر و سرور داده رد و بدل می‌کند. همچنین فرض کنید ارتباطات HTTP برای گرفتن اشیا از طریق persistent و pipeline انجام می‌شود.

(زمان RTT بین client و cache server، ۱ میلی ثانیه و همین مقدار برای کش سرور و سرور اصلی برابر ۲۰۰ میلی ثانیه است. میانگین زمان ارسال برای هر شیء را ۱۰۰ میلی ثانیه است.)

RTT2 : زمان RTT بین کش سرور و سرور اصلی

نیمی از اشیاء در کش مرورگر وجود ندارند و نیاز به درخواست دادن به سرور دارند. همچنین ۶۰ درصد اشیاء موجود در کش نیاز به بررسی دارند پس داریم:

$$\frac{60}{2} + (30 * 0.6) = 48$$

پس در کل ۴۸ درخواست به کش سرور خواهیم داشت. بر اساس اینکه درخواست ها به صورت pipeline ارسال می‌شوند، پس دو RTT1 ارسال درخواست ها طول می‌کشد. هنگامی که درخواست ها به کش سرور می‌رسند، نیمی از آن‌ها به روز هستند و چون کش سرور همزمان با کلاینت و سرور در ارتباط است، در همان لحظه شروع به ارسال فایل‌ها به کلاینت می‌کند. تعداد این اشیاء ۲۴ عدد است. پس خواهیم داشت:

$$2 * RTT1 + 24 * 100 = 2402(ms)$$

همچنین ۱۲ شیء نیاز به چک کردن از سرور اصلی دارند ولی نیاز به به روزرسانی نخواهند داشت و ۱۲ شیء نیز نیاز به به روزرسانی دارند. برای این اشیاء نیز دو RTT2 زمان صرف می‌شود به همراه زمان ارسال ۱۲ شیء از سرور اصلی به کش سرور (و دریافت ۱۲ پیام not modified که از زمان انتقال آنها به دلیل کوچکی چشم پوشی می‌شود). پس زمان لازم برای ارسال این درخواست ها از کش سرور به سرور اصلی و دریافت پاسخ برابر است با:

$$2 * RTT2 + (12 * 100) = 400 + 1200 = 1600(ms)$$

از آنجایی که این فرآیند به محض دریافت درخواست ها توسط کش سرور شروع میشود (پس از  $2 * RTT1$ ) کل زمان بالا (۱۶۰۰ میلی ثانیه) با زمان ارسال فایل‌هایی که در کش سرور به روز بودند (۲۴۰۰ میلی ثانیه) همپوشانی دارد و به محض اتمام ارسال آنها ارسال ۱۲ فایلی که از سرور اصلی دریافت شده اند به همراه ۱۲ فایلی که در کش سرور در مورد به روز بودن آنها تردید وجود داشت (اما در اصل به روز بوده اند) شروع می‌شود. بنابراین کل زمان دریافت صفحه وب به صورت زیر محاسبه می‌شود.

$$2402 + 1200 + 1200 = 4802(ms)$$

۳. فرض کنید قصد داریم فایلی با حجم F بیت را بین N کاربر که از معماری P2P پیروی می‌کنند، ارسال کنیم. (فرض کنید مدل جریان بیتی (fluid) است.) برای راحتی فرض کنید که مقدار  $d_{min}$  خیلی بزرگ

می‌باشد به صورتی که پهنای باند یک کاربر هیچ‌گونه محدودیتی ایجاد نمی‌کند. حال با توجه به هر یک از فرض‌های زیر، یک طرح توزیع پیشنهاد دهید که بتوان به زمان توزیع کل خواسته شده دست یافت. (۵۰ نمره)

(آ) فرض کنید  $u_s \leq (u_s + u_1 + \dots + u_N)/N$  و زمان توزیع کل فایل بر روی نودها:  $\frac{F}{u_s}$

(a) اگر  $u = u_1 + u_2 + \dots + u_N$  باشد. طبق فرض داریم:

$$u_s \leq (u_s + u)/N \quad \text{معادله 1}$$

حال فایل را به  $N$  قسمت تقسیم میکنیم که قسمت  $i$ ام دارای سایز  $(u_i/u)F$  است.

سرور قسمت  $i$ ام را با نرخ  $r_i = (u_i/u)u_s$  به نود  $i$  ارسال می‌کند. همچنین می‌دانیم که  $r_1 + r_2 + \dots + r_N = u_s$  پس جمع این نرخ‌ها از نرخ ارسال لینک سرور بیشتر نمی‌شود. هر نود هم با نرخ  $r_i$  بیت‌هایی که دریافت میکند را به  $N-1$  نود دیگر ارسال می‌کند.

بنابراین، نرخ ارسال تجمعی نود  $i$ ام برابر با  $(N-1)r_i$  است. پس داریم:

$$(N-1)r_i = (N-1)(u_s u_i)/u \leq u_i$$

در این طرح توزیع، نود  $i$ ام بیت‌ها را با نرخ تجمعی  $r_i + \sum_{j < i} r_j = u_s$  دریافت می‌کند؛ پس هر نود فایل را در  $F/u_s$  دریافت می‌کند.

(ب) فرض کنید  $u_s \geq (u_s + u_1 + \dots + u_N)/N$  و زمان توزیع کل فایل بر روی نودها:  $\frac{NF}{u_s + u_1 + \dots + u_N}$

(b) مثل قسمت قبل  $u = u_1 + u_2 + \dots + u_N$  را تعریف می‌کنیم. طبق فرض:

$$u_s \geq (u_s + u)/N \quad \text{معادله 2}$$

حال اگر فرض کنیم که  $r_i = u_i(N-1)$  و  $r_{N+1} = (u_s - u/(N-1))/N$  باشند، در این طرح توزیع، فایل به  $N+1$  قسمت تبدیل می‌شود. سرور بیت‌های بخش  $i$ ام را با سرعت  $r_i$  به نود  $i$ ام ارسال می‌کند. (آیین 1 تا  $N$  است). علاوه بر این، سرور بیت‌های بخش  $N+1$ ام را با سرعت  $r_{N+1}$  برای همه  $N$  نود ارسال می‌کند. (توجه داشته باشید که نودها در این حالت نیازی به ارسال بیت‌های این بسته ندارند چون همه آن را از سرور دریافت می‌کنند).

پس مجموع نرخ ارسال سرور برابر خواهد بود با:

$$r_1 + r_2 + \dots + r_N + Nr_{N+1} = \frac{u}{N-1} + u_s - \frac{u}{N-1} = u_s$$

بنابراین، نرخ ارسال سرور از سرعت لینک آن بیشتر نمی‌شود. همچنین برای هر نود، مجموع نرخ ارسال آن برابر است با:

$$(N-1)r_i = u_i$$

بنابر این، نرخ ارسال هر نود نیز از سرعت لینک آن بیشتر نمی‌شود.

در این طرح توزیع، نود  $A$  بیت‌ها را با نرخ تجمعی زیر دریافت می‌کند:

$$r_i + r_{N+1} + \sum_{j < i} r_j = \frac{u}{N-1} + \frac{(u_s - u)/(N-1)}{N} = \frac{u_s + u}{N}$$

پس هر نود، فایل را در زمان  $NF/(u_s + u)$  دریافت می‌کند.

در نهایت با ترکیب قسمت  $a$  و  $b$  نتیجه خواسته شده بدست می‌آید.

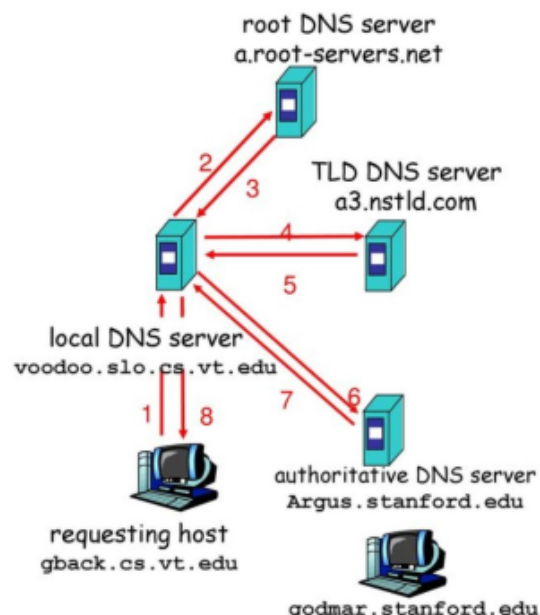
سپس نتیجه‌گیری کنید که کمترین زمان توزیع از طریق رابطه زیر بدست خواهد آمد.

$$\max\left\{\frac{F}{u_s}, \frac{NF}{u_s + u_1 + \dots + u_N}\right\}$$

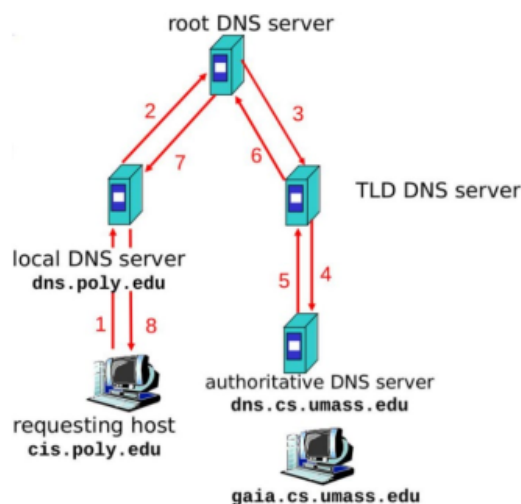
۴. فرض کنید برای اولین بار وارد سایت کلاس درس به این آدرس می‌شوید و منتظر می‌شوید که سایت باز شود. حداکثر و حداقل تعداد پیام‌های رد و بدل شده بین DNS در حالت‌های بازگشتی و تکراری را باهم مقایسه کنید و با رسم شکل توضیح دهید. (۳۰ نمره)

اگر DNS در حالت Authorative باشد ابتدا به root DNS server درخواست می‌زند و آدرس IP را از آن می‌خواهد و در جواب آدرس IP مربوط به IR TLD server را دریافت می‌کند و به آدرس جدید درخواست می‌زند و این روند تا جایی ادامه می‌کند که تا به آدرس IP مربوط به سایت برسیم.

نکته: در آخرین پاسخ آدرس سایت در قسمت Answer مربوط به DNS Response می‌باشد



اگر در حالت Recursive باشد Local DNS server چک می‌کند که آیا این رکورد را دارد یا خیر، اگر دارد که آن را ارسال می‌کند اما اگر موجود نباشد به Root DNS server درخواست می‌زند و آن هم برعکس حالت قبل که آدرس DNS server دیگری دهد، خودش به سرور دیگری از DNS درخواست می‌زند و این کار را سرور ها تاجایی انجام می دهند که به جواب برسد سپس به صورت بازگشتی جواب خود را برای سرورهای بالایی ارسال می‌کنند و در آخر جواب را می‌فرستند.



اگر در نظر داشته باشیم که اولین بار است که وارد سایت می‌شویم پس می‌توان گفت که داده کش شده برای این قسمت نداریم و اما اگر در سرور Local DNS این رکورد موجود باشد می‌توان گفت با یک پیام درخواست و یک پیام جواب به نتیجه می‌رسیم اما اگر در آن سرور موجود نبود باید ۴ جفت پیام در هر دو حالت رد و بدل بشود.

۵. (سوال عملی) - ابتدا دستورالعمل‌های زیر را انجام دهید: (۶۰ نمره)

(آ) کش مرورگر و DNS خود را پاک کنید. برای پاک کردن کش DNS می‌توانید از این لینک ۳ کمک بگیرید

(ب) وایرشارک را باز کنید و آدرس `addr.ip == your_ip_address` را در فیلد فیلتر وارد نمایید. (برای بدست آوردن آدرس ip خود می‌توانید از دستور `ifconfig` در سیستم‌های لینوکسی و دستور `ipconfig` در سیستم‌های ویندوزی استفاده کنید) و `capture` را آغاز کنید.

(ج) آدرس `geeksforgeeks.org` را در مرورگر وارد نمایید.

(د) `capture` را آغاز کنید.

حال به سوالات زیر پاسخ دهید

(آ) آدرس فرستنده DNS query و آدرس پیام‌های پاسخ را بیابید. پروتکل لایه انتقال استفاده شده این‌گونه درخواست‌ها چیست؟ از پروتکل UDP استفاده می‌شود

(ب) پورت مقصد پیام DNS query و پورت مبدا DNS response را مشخص کنید. این شماره پورت مربوط به چه سرویسی است؟

هر دو پورت برابر ۵۳ است که مربوط به پروتکل DNS می‌باشد

(ج) پیام DNS query به کدام آدرس IP فرستاده شده است؟

(د) با استفاده از دستور `ipconfig /all` آدرس سرور نام خود را بررسی کنید. آیا هر دو آدرس یکسان هستند؟

بله این دو آدرس یکسان هستند.

(ه) فکر می‌کنید دلیل انجام مورد مرحله ۲ چه بوده است؟ در نرم‌افزار Wireshark بجای استفاده از این فیلتر چه کاری می‌توان انجام داد؟

بسته‌هایی که دارای IP address مبدا یا مقصد برابر با IP وارد شده در فیلتر است را نشان می‌دهد

(و) برخی کاربران ادعا کرده‌اند شرکت Cloudflare سرویسی ارائه می‌دهد که می‌تواند سرعت اینترنت آن‌ها را افزایش دهد. با مراجعه به `https://1.1.1.1` در مورد آن اطلاع کسب کنید. علت افزایش سرعت چیست؟ آیا با آن می‌توان سرعت دانلود فایل را افزایش داد؟ چرا؟



سرویس Cloudflare به وسیله کش کردن محتوای ثابت وبسایت مانند عکس‌های ثابت و... باعث افزایش سرعت لود برای کاربران می‌شود. این سرور بین کلاینت و سرور اصلی قرار می‌گیرد. این سرویس دسترسی به سرور اصلی و دیتابیس ندارد و نمی‌تواند بهینه‌سازی برای دانلود انجام دهد.

(ز) DNS خود را روی آدرس‌های ۱.۱.۱.۱ و ۱۰.۰.۰.۱ تنظیم کنید (برای اینکار از آدرس معرفی شده استفاده کنید). آدرس <https://www.ietf.org/> را در مرورگر خود وارد کنید. با کمک نرم‌افزار Wireshark تحلیل کنید چقدر طول می‌کشد تا به درخواست‌های DNS مرورگر پاسخ داده شود؟ همین کار را برای DNS سرورهای ۸.۸.۸.۸ و ۸.۸.۴.۴ تکرار کنید. (دقت کنید که هر بار قبل از اجرا کش DNS را پاک کنید) و زمان‌ها را مقایسه کنید. آدرس ۱.۱.۱.۱ سریع‌تر عمل می‌کند.

(ح) با ابزار dig آدرس IP مربوط به <https://www.ietf.org/> را بدست آورید. (یک بار این درخواست را از آدرس ۱.۱.۱.۱ و یک بار از ۸.۸.۸.۸ انجام دهید) و به طور مجدد نتایج مربوط به response time را گزارش کنید.

```
mehrdad@MEHRDAD:~$ dig @1.1.1.1 letf.org

;<<>> DiG 9.16.1-Ubuntu <<>> @1.1.1.1 letf.org
; (1 server found)
; global options: +cmd
; Got answer:
;-->HEADER<-- opcode: QUERY, status: NOERROR, id: 32277
; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;letf.org.                                IN      A

;; ANSWER SECTION:
letf.org.                2026    IN      A      76.223.65.111
letf.org.                2026    IN      A      13.248.216.40

;; Query time: 115 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: 2021 0430+ 01:06:42 18 بهمنیه آوریل
;; MSG SIZE rcvd: 69

mehrdad@MEHRDAD:~$ dig @8.8.8.8 letf.org

;<<>> DiG 9.16.1-Ubuntu <<>> @8.8.8.8 letf.org
; (1 server found)
; global options: +cmd
; Got answer:
;-->HEADER<-- opcode: QUERY, status: NOERROR, id: 20662
; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;letf.org.                                IN      A

;; ANSWER SECTION:
letf.org.                3599    IN      A      13.248.216.40
letf.org.                3599    IN      A      76.223.65.111

;; Query time: 199 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: 2021 0430+ 01:08:03 18 بهمنیه آوریل
;; MSG SIZE rcvd: 69
```

موفق باشید.