دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

تمرین سوم

شبکه‌های کامپیوتری

نگارش

محمدرضا اخگری زیری

استاد درس

دکتر صبایی

فروردین ۹۹

فهرست مطالب

صفحه

سوال اول

خیر، برای این تضمین باید در گره‌های مسیر اعمال انجام شود، ولی در گره‌ها لایه انتقال وجود ندارد، پس امکان ندارد که بتوان فقط با لایه انتقال این تضمین را داد.

سوال دوم

بله، عملکرد مانند یک CDN معقول است، زیرا با ذخیره داده‌ها در شبکه خود هزینه استفاده از شبکه‌های دیگر را نخواهد داد، ضمن اینکه کاربران نیز از سرعت بهتری برخوردار خواهند شد، البته، باید بابت این نگه‌داری هزینه‌هایی هم پرداخت کند، از جمله هزینه نگهداری و بروزرسانی اطلاعات.

سوال سوم

ابتدا باید با استفاده از سرویس DNS، آدرس IP را به دست آورد، طبق روش DNS LookUp به صورت پشت‌سر هم زمان $\Sigma_{i=1}^{n} RTT_i$ نیاز است، پس از یافتن IP باید ارتباطی برقرار شود که این ارتباط زمان RTT_0 را احتیاج دارد، پس از آن باید یک درخواست ارسال و پاسخ دریافت شود که زمان آن هم RTT_0 است، زمان کل برابر می‌شود با:

$$2RTT_0 + \Sigma_{i=1}^{n} RTT_i$$

سوال چهارم

در ابتدا باید بدانیم که $RTT_0 = 2$ propagation time **که برابر است با زمانی که یک بسته از سمت کلاینت به سمت سرور برود و بازگردد و برای همه‌ی موارد زمان** DNS service **را در نمودار نیاوردیم.**

۱. برای یک اتصال مداوم یا غیر مداوم ، لازم است که برای شروع اتصال TCP **از یک** RTT_0 **استفاده شود.**

۲. یک RTT_0 **برای درخواست** HTTP **و چند بایت اول برای پاسخ** HTTP **برای بازگشت استفاده می شود.**

به منظور دانستن کل زمان انتقال پرونده:

$$\text{total} = 2RTT_0 + \text{tramsmit time}$$

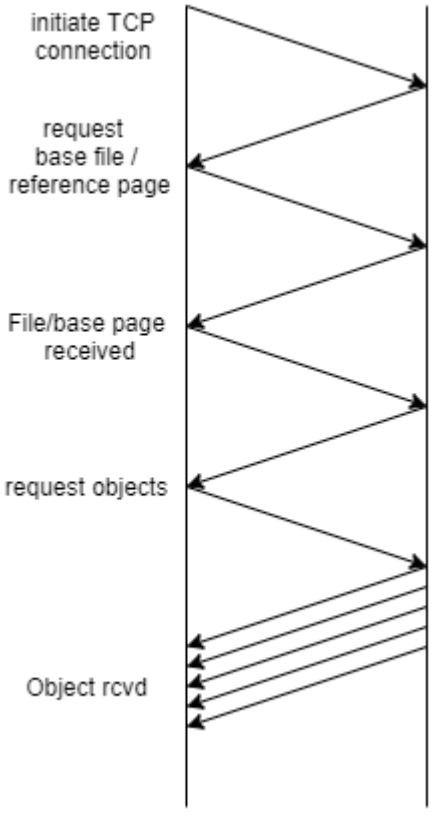
http ناپایا بدون اتصال موازی:

هر شی دو RTT **(با فرض بدون محدودیت پنجره) یکی برای اتصال** TCP **و دیگری برای متن** HTTP **طول می‌کشد.**

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 5(2RTT_0)$$

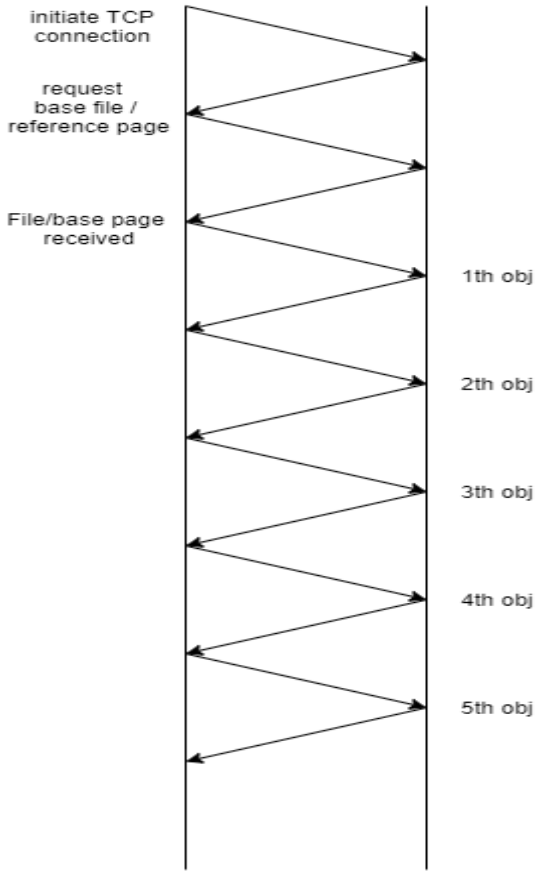
Http ناپایا با ۵ اتصال موازی:

$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 2RTT_0$$



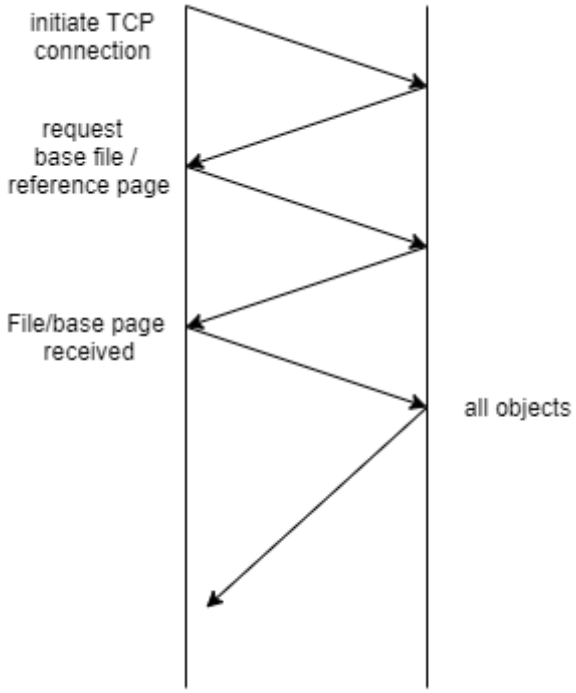
Non-persistent &
Parallel connection

Http پایا (بدون پایپ لاین):



$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + 5RTT_0$$

Http پایا (با پایپ لاین):



$$RTT_1 + RTT_2 + RTT_3 + 2RTT_0 + RTT_0$$

سوال پنجم

سوال پنج یافت نشد.

سوال ششم

به علت ظرفیت ۱۰۰ مگابیتی لن‌ها تاخیری ایجاد نخواهد شد، محل قرارگیری پراکسی سرور در تاخیر ایجاد شده مهم است، اگر در سمت سوییچ باشد، تاخیری نخواهیم داشت و اگر در سمت روتر باشد، به شکل زیر خواهد بود:

$$\frac{30 \times 400\,000}{10^8} = 0.12$$

طبق نمودار سوال تاخیر این میزان load **نیز برابر با صفر است و می‌توان فرض کرد تاخیری ندارد.**

حدود ۵۰ درصد درخواست‌ها مستقیم از پراکسی سرور جواب داده میشود که میزان load **برابر می‌شود با:**

$$0.5 \times \frac{30 \times 400\,000}{10^7} = 0.6$$

طبق نمودار این تاخیر برابراست با ۰.۵ ثانیه.

۵۰ درصد مابقی باید از وب سرور اصلی دانلود شوند و ۲۰ درصد از مطالب گرفته شده از پراکسی هم باید به روز شوند، پس به عبارتی $0.5 + 0.2 * 0.5 = 0.6$ **باید از وب سرور اصلی گرفته شود که** load **برابر خواهد شد با:**

$$0.6 \times \frac{(30 \times 400\,000)}{10^7} \approx 0.7$$

و این تقریبا load **ای برابر با ۱ ثانیه دارد، از سمتی نیز تاخیر اینترنت ۲ ثانیه است، پس می‌توان تاخیر کل را برابر گرفت با:**

$$0.5 * 0 + (2 + 1)0.6 = 1.8 \text{ s.}$$

سوال هفتم

در یک درخواست HTTP به سرور وب اجازه می دهد تا نام میزبان درخواستی را بشناسد و بر اساس آن سرویس دهد. این اجازه می دهد تا یک دستگاه در آدرس IP برای بسیاری از دامنه ها خدمت کند.

به عنوان مثال ، اگر یک وب سرور میزبان: foo.com **را می بیند ، ممکن است یک وب سایت را نمایش دهد، اما میزبان:** bar.com **ممکن است منجر به بازگشت نتیجه کاملاً متفاوت شود.**

دلیل دیگر استفاده از این آدرس برای کش کردن در web proxy **هاست.**

سوال هشتم

مفهوم توزیع بار از طریق DNS مطرح می‌شود.

متعادل کردن بار معمولاً برای تعادل در ترافیک سیستم های زائد مانند سرورهای وب یا برنامه ها استفاده می شود. بنابراین اگر یک سرور در دسترس نباشد ، چندین سرور دیگر وجود دارد که آماده بار بار ترافیک هستند.

تعادل بار می تواند کارهای بسیار شگفت انگیزی انجام دهد ، مانند:

محافظت از قطع برق

Load time را بهبود بخشید

بار سرور را کاهش دهید

ارائه دهندگان DNS مبتنی بر ابر معمولاً خدمات متعادل کننده بار را به صورت رایگان یا با حداقل هزینه ارائه می دهند.

انواع توازن بار

Round Robin

ساده ترین شکل توازن بار است. لیست سرورها و آدرس هایشان به صورت چرخشی ارسال میشود. از آنجا که راهی برای گفتن در دسترس بودن یا نبودن این سیستم ها وجود ندارد ، می توانید به سرور آهسته یا ناسالم ترافیک ارسال کنید. مثلا شما سه سرور در شبکه خود دارید. اگر یکی از آنها پایین بیاید ، تقریباً یک سوم از ترافیک شما هنوز به آن سیستم هدایت می شود.

Weighted Round Robin

ممکن است سروری توانایی بیشتری برای پاسخ دهی داشته باشد و ظرفیت بیشتری داشته باشد به این ترتیب این IP ها وزن بیشتری خواهند داشت.

Round Robin + Uptime Monitoring

در صورتی که یک سرور در دسترس نباشد آن را از لیست بیرون می‌آوریم.

نحوه تنظیم تعادل بار

#1 Set of DNS Records

DNS فقط می تواند به یک آدرس IP اختصاصی اشاره کند. که توازن بار را تقریبا غیرممکن می کند. آنچه شـــما می توانید انجام دهید ایجاد چندین رکورد با همین نام (مانند www) اســـت ، اما هرکدام به یک آدرس IP متفاوت اشاره می کنند.

هنگامی که کاربر سوابق www شما را پرس و جو می کند ، با یکی از آدرس های IP با نام www به آنها پاسخ داده می شود.

#2 Record Pools

همچنین می توانید تعادل بار را با Record pools تنظیم کنید. Pools گروههایی از آدرســهای IP یا نامهای میزبان هستند. پس از ایجاد pool، می توانید برای سوابق فردی اقدام کنید. هنگامی که این پرونده پرس و جو شـد ، توسـط یک یا چند آدرس IP موجود در pool، به کاربر پاسـخ داده می شود.

سوال نهم

در حال حاضر دو راه حل محبوب برای این مشکل وجود دارد.

اولین مورد anycast **نامیده می‌شـــود که همان بلوک** IP **به معنای واقعی کلمه در مکان های مختلفی در سـراسـر جهان اسـتفاده می‌شـود. یعنی** name services **برای دامنه شـما همیشـــه همان آدرس** IP **را برمی‌گردانند ، اما آن آدرس** IP **در واقع به بیش از یـک مـجـمـوعـه از ســـرورهـای فـیـزیـکـی اخـتـصـــاص داده شـــده اســـت.**
http://en.wikipedia.org/wiki/Anycast

تکنیک دوم مجدداً AnyCast **را در بر می‌گیرد ، اما این بار ، محدوده آدرس** IP **که پخش می‌شـــود به خود ســـرورهای نام ما مراجعه می‌کند. از آنجا که ســـرویس‌دهنده‌ها فقط از مشـــتریانی که به آنها نزدیکتر هســـتند نیز درخواســـت خواهند کرد، آنها می‌توانند آدرس‌های** IP **را که منطقاً محلی برای مشتری هستند، بازگردانند.**

نمونه ای از این دامنه google l.google.com **است:**

From a host in Australia

```
crimson:~ dave$ host www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com is an alias for www-notmumbai.l.google.com.
www-notmumbai.l.google.com has address 66.249.89.99
www-notmumbai.l.google.com has address 66.249.89.147
www-notmumbai.l.google.com has address 66.249.89.103
www-notmumbai.l.google.com has address 66.249.89.104
```

From a host in the US

```
[dave@odessa ~]$ host www.google.com
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 74.125.95.99
www.l.google.com has address 74.125.95.147
www.l.google.com has address 74.125.95.104
www.l.google.com has address 74.125.95.106
www.l.google.com has address 74.125.95.105
www.l.google.com has address 74.125.95.103
```

بنابراین ، CNAME **برای** www.google.com **به** www.l.google.com **پاسخ داده می‌شود ، اما هنگامی که شما آن را درخواست می‌دهید، بسته به موقعیت مکانی ،** client **شما مجموعه دیگری از آدرس‌های** IP **را دریافت می‌کند. دلیل این امر این اســت که** name server **که درخواســت** www.l.google.com **را دریافت کرده است ،** local nameserver**، نسبت به** client **است.**

سوال دهم

در سوال هشت به توضیح Round Robin و Record pool و نقاط ضعف و قدرت آن اشاره شده است:

یک روش متعادل سازی بار است که شامل استفاده از چندین آدرس IP مختلف برای یک نام دامنه است.

یک سرور DNS با قابلیت Round Robin ، دارای چندین رکورد مختلف خواهد بود که هرکدام دارای نام دامنه یکسان هستند اما یک آدرس IP متفاوت دارند. هر بار سرور DNS پرس و جو می شود ، آدرس IP که در ابتدای صف است ، ارسال می کند و در انتهای حلقه می‌گذارد.

استفاده از record pool باعث می‌شود تا چندین آدرس ارسال شود که کلاینت انتخاب کند به کدام یک از آنها درخواست دهد و اگر یکی در دسترس نبود به بعدی درخواست دهد.

سوال یازدهم

برای دریافت این صفحه ابتدا باید یک اتصال TCP ایجاد شود، این اتصال هزینه RTT را دارد، سپس باید فایل http را انتقال دهد و پس از آن می‌توان به صورت همزمان فایل ها را دریافت کند.

$$\begin{aligned}\text{TCP} + T_{HDML} + \max T_{objects} &= RTT_1 + RTT_1 + \frac{5000B}{80000bps} \\ &+ \max T_{objects} \\ &= 0.06 + 0.5 + \max T_{objects} = 0.56 + 1.32 = 1.88\ \text{s}.\end{aligned}$$

$$\begin{aligned}T_{objects} &= \{T_{o_1o_2}, T_{o_3o_4}, T_{o_5o_6}\} \\ &= \left\{2RTT_1 + \frac{(2000+4000)B}{80000bps}, TCP_2 + 2RTT_2\right. \\ &\left. + \frac{(2000+4000)B}{40000bps}, TCP_3 + 2RTT_3 + \frac{(5000+7000)B}{80000bps}\right\} \\ &= \{0.06 + 0.6, 0.12 + 1.2, 0.06 + 1.2\} = \{0.66, 1.32, 1.26\}\end{aligned}$$

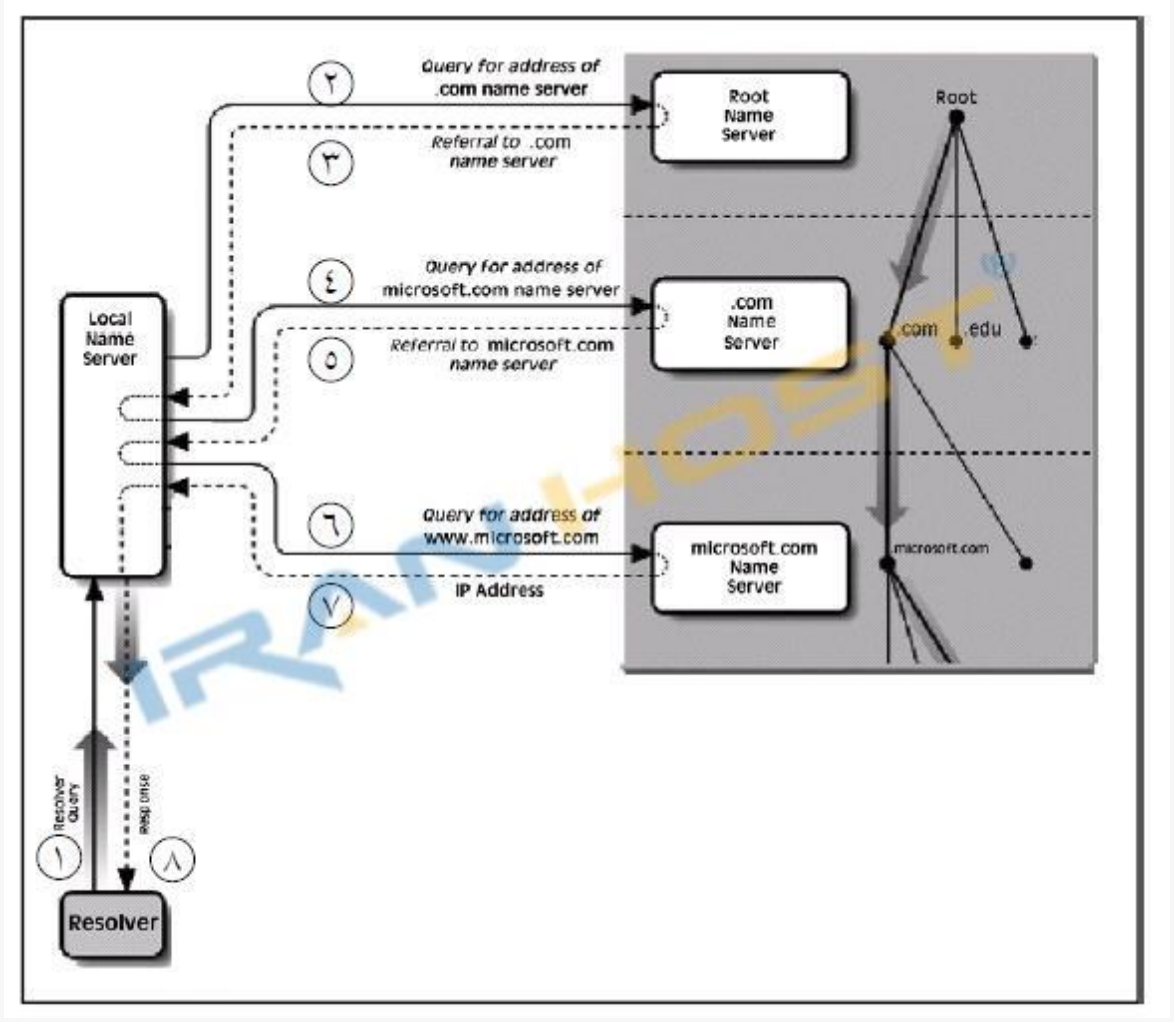
سوال دوازدهم

پرس‌وجوی تکراری:

دراین پرس وجو قسمت اعظم تلاش برای تبدیل یک نام برعهدۀ سرویس دهندۀ محلی است (Local DNS Server)، این DNS server حداقل نیاز به داشتن آدرس ماشین Root ، به عنوان نقطۀ شروع دارد .

وقتی یک تقاضای ترجمه آدرس به سرویس دهندۀ محلی ارسال میشود، معادل آدرس IP به تقاضا کننده برگردانده می شود (اگر قبلا ذخیره داشته باشد)، در غیر این صورت سرویس دهندۀ محلی یک تقاضا برای DNS سرور سطح بالا ارسال می نماید . این سرویس دهنده ، آدرسِ ماشینی را که میتواند برای ترجمۀ نام مورد نظر مفید باشد ، به سرویس دهندۀ محلی معرفی می نماید.

سرویس دهندۀ محلی مجددا یک تقاضا به ماشین معرفی شده در مرحلۀ قبل ارسال می نماید . در این حالت نیز سرویس دهندۀ نام میتواند در صورت یافتن آدرس IP معادل آنرا برگرداند و یا در غیر اینصورت آدرس سرویس دهندۀ سطح پایین تری را ارائه نماید . این روند ادامه می یابد تا DNS سرور نهایی نام مورد نظر را به آدرس IP ترجمه نماید .



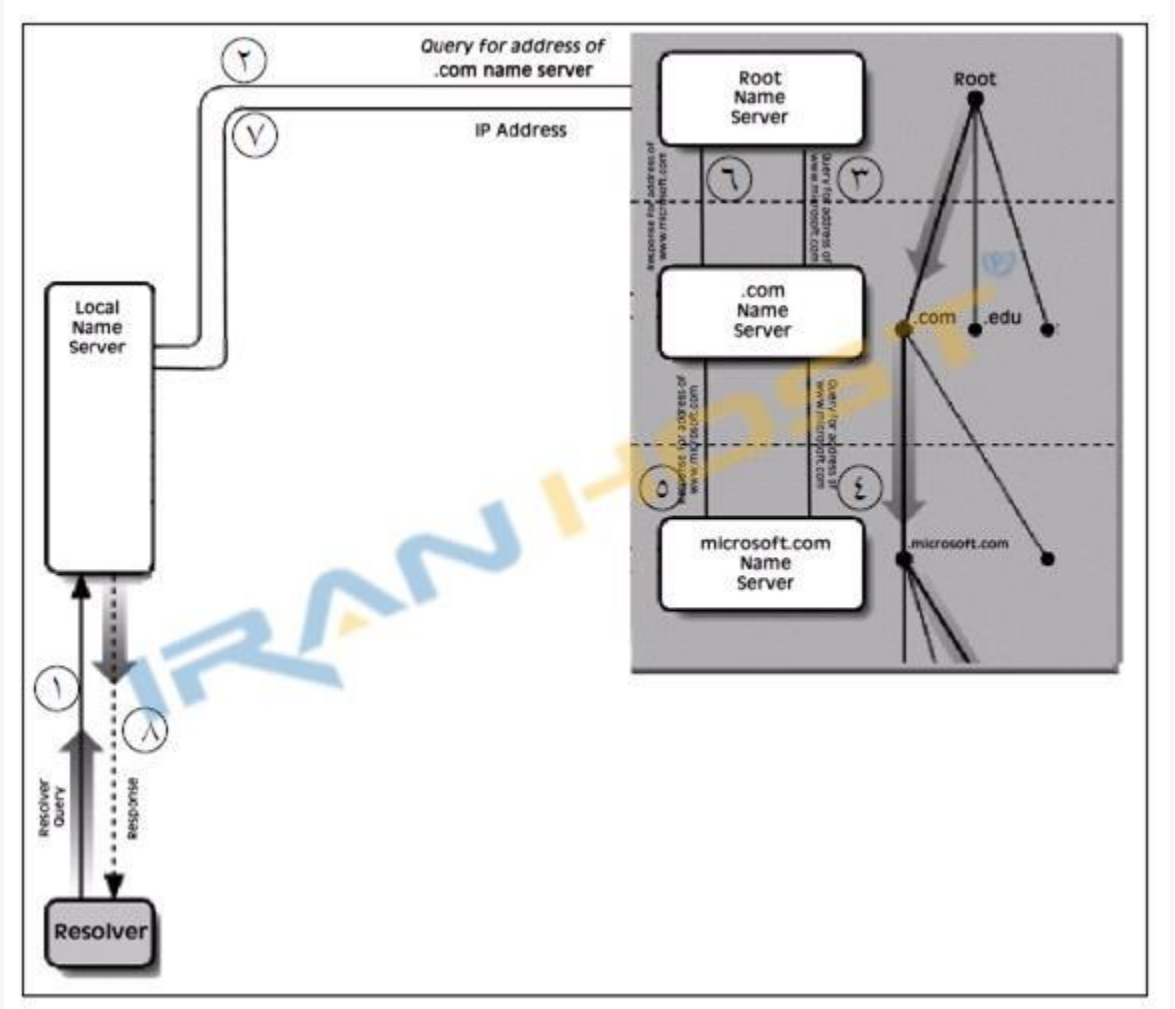
پرس و جوی بازگشتی

در این روش هر گاه برنامه ای بخواهد آدرس IP معادل با یک نام را بدست آورد ، تابع سیستمی تحلیل نام را فراخوانی می نماید . این تابع به صـورت پیش فرض آدرس یک سـرور root را در اختیار داشته بنابراین تقاضای تبدیل نام را به روش UDP برای آن ارسال و منتظر جواب می ماند دو حالت ممکن است اتفاق بیفتد :

ممکن اسـت در بانک اطلاعاتی مربوط به سـرویس دهندۀ محلی ، آدرس IP معادل با آن نام از قبل وجود داشته و بالطبع به سرعت مقدار معادل IP آن برمیگردد .

ممکن است در بانک اطلاعاتی سرویس دهندهٔ محلی ، معادل IP آن نام وجود نداشته باشد . در چنین حالتی سرویس دهندهٔ محلی موظف است (بدون آنکه به تقاضا دهنده اطلاع دهد) به سرویس دهندهٔ سطح بالاتر تقاضای ترجمه آدرس را ارسال نماید . دراین حالت نیز DNS سطح بالاتر به همین نحو ترجمه آدرس را پیگری می نماید و این مراحل تکرار خواهد شد.

در روش پرس وجوی بازگشتی، ماشین سرویس دهندهٔ محلی این مراحل متوالی را نمی بیند و هیچ کاری جز ارسال تقاضای ترجمهٔ یک آدرس برعهده ندارد و پس از ارسال تقاضا برای سرویس دهندهٔ سطح بالا منتظر خواهد ماند .



سوال سیزدهم

چون در این پروتکل برای یک ارتباط دونشست ایجاد می‌کند. یکی برای انتقال اطلاعات کنترلی و دیگری برای انتقال اطلاعات. به همین دلیل می‌گوییم به صورت خارج‌بندی است.

سوال چهاردهم

MAIL FROM: **در** SMTP **یک پیام از مشتری** SMTP **است که فرستنده پیام نامه را به سرور** SMTP **مشخص می کند.**

From: **فقط یک خط در قسمت** body **پست الکترونیکی است.**

سوال پانزدهم

بسته های UDP **در اندازه کوچکتر هستند. بسته های** UDP **نمی توانند از ۵۱۲ بایت بیشتر باشند. بنابراین هر برنامه ای نیاز به انتقال داده بیشتر از ۵۱۲ بایت نیاز به** TCP **در محل دارد.**

DNS **از** TCP **برای انتقال** Zone **و** UDP **برای جستجوی نام به صورت منظم (اولیه) یا معکوس استفاده می کند.** UDP **می تواند برای تبادل اطلاعات کوچک استفاده شود در حالی که** TCP **باید برای تبادل اطلاعات بزرگتر از ۵۱۲ بایت استفاده شود. اگر مشتری از** DNS **پاسخی دریافت نکرد ، باید داده ها را با استفاده از** TCP **بعد از ۳-۵ ثانیه بازه مجدداً انتقال دهد.**

در پایگاه داده DNS Zone **باید سازگاری وجود داشته باشد. برای این کار ،** DNS **همیشه داده های** Zone **را با استفاده از** TCP **منتقل می کند زیرا** TCP **قابل اطمینان است و با انتقال منطقه کامل به سایر سرورهای** DNS **که درخواست داده ها را دارند ، اطمینان حاصل کنید که داده های منطقه سازگار است.**

سوال شانزدهم

الف)

به یکی از سـرورهای mail(10.0.1.5) **یا** mail2 (10.0.1.7) **ارسـال میشـود، بر روی این سـرورها** SMTP **باید باشد.**

ب)

سرورهای dns1(10.0.1.2) **و** dns2(10.0.1.3)