DNS : records, message formats, how to register a domain

- DNS یک distributed database هست. در داخل این دیتابیس ها یه سری resource records(RR) هست که اطلاعات ما در قالب این ها ذخیره می شن.
- فرمتی که این resource record ها دارن ، شامل فیلد های متنوعی هست که مهم ترین ها ، name, value, type, ttl هستند.
- ttl مخفف Time to Live هست به این معنی که اگه در پاسخ به یه سوال DNS ، یه resource record ای برگردونده شد ، این resource تا چه مدت زمانی میتونه داخل کش باقی بمونه و برای اون سوال لازم نباشه که ما مجددا بریم از سرور اصلی سوال بپرسیم.
  هروقت ttl منقضی شد ، اون resource record باید از کش بیرون ریخته بشه و سوال مجددا از سرور اصلی پرسیده بشه.
- مفهومی که name و value دارن ، بستگی به type داره :
  1 - اگه type = A باشه :

name is hostname;
value is IP address;

این نوع ،اصلی ترین نوع رکوردی هست که ما داریم ، چون کارکرد اصلی DNS اینه که یه hostname رو به IP address تبدیل کنه.

2 - اگه type = NS باشه :

برای وقتیه که مثلا ما توی TLD یا root جواب به یک کوئری رو نداریم اما میدونیم که جواب این کوئری از چه سروری باید پرسیده بشه.بنابراین name ، همون hostname یا domain هست ولی value دیگه IP address نیست و یک hostname از سروری هست که ما باید سوالمون رو از اون بپرسیم. این سرور می تونه یه TLD باشه یا یه authoritative DNS server .

3 - اگه type = CNAME باشه :

Name مجددا همون hostname یا alias name هست و value یه canonical name متناظر با سرور.

برای پیدا کردن canonical name یک host ، می تونیم سوالی از جنس CNAME از یک سرور بپرسیم و اگه داخل اون سرور رکوردی از نوع CNAME وجود داشته باشه ، میتونه canonical name رو در اختیار ما قرار بده.

4 - اگه type = MX باشه :

Name ، یک alias name برای یه mail server ، و value نام اون mail server هست.

پس مثلا اگه یه browser بخواد به IP address ِ وب سرورِ یاهو دسترسی پیدا کنه ، توی سوالش تایپ رو برابر با A قرار میده، ولی اگه یه email client داشته باشیم که بخواد به mail server یاهو دسترسی داشته باشه ، طبیعتا توی سوالی که می پرسه تایپ رو برابر MX قرار میده.

- اپلیکیشن DNS ،از UDP به عنوان پروتکل لایه ی transport استفاده می کنه و port number اش هم 53 هست.
- توی این پروتکل ما دو نوع پیام استفاده می کنیم: query و reply این پیام ها هردوشون فرمت یکسانی دارن و فقط توی بعضی از فیلد ها با هم متفاوتن.
- در ابتدای پیام ، 2 بایت Identification داریم که توسط اون کلاینت تشخیص میده که این reply مربوط به کدوم کوئریه. به خاطر این که وقتی یه سرور جواب یه کوئری رو میده ، محتوای قسمت Identification مربوط به کوئری رو دقیقا توی همین قسمت در reply کپی می کنه و کلاینت با مقایسه کردن این دوتا، میتونه بفهمه هر reply مربوط به کدوم کوئریه.
- 2 بایت flag داریم. یکی از فیلدهایی که توی flag هست مشخص می کنه پیام مربوط به کوئری هست یا reply .

دوتا فیلد دیگه ، recursion desired و recursion available هستن. ما برای تعامل با DNS server ها ، میتونیم به صورت iterative یا recursive عمل کنیم. حالت iterative همونه که وقتی سوالی از DNS server می پرسیم و جواب شو نداره ولی میدونه که باید از چه سروری پرسیده بشه، آدرس سرور رو در اختیار کلاینت قرار میده تا خودش بره بپرسه.

حالت recursive اینه که خود DNS server بره جواب رو از سرور اصلی بپرسه و به کلاینت بگه.

حالا اگه بیت recursion desired یک باشه به این معنیه که کلاینت تمایل داره به صورت recursion جواب سوالش داده بشه.

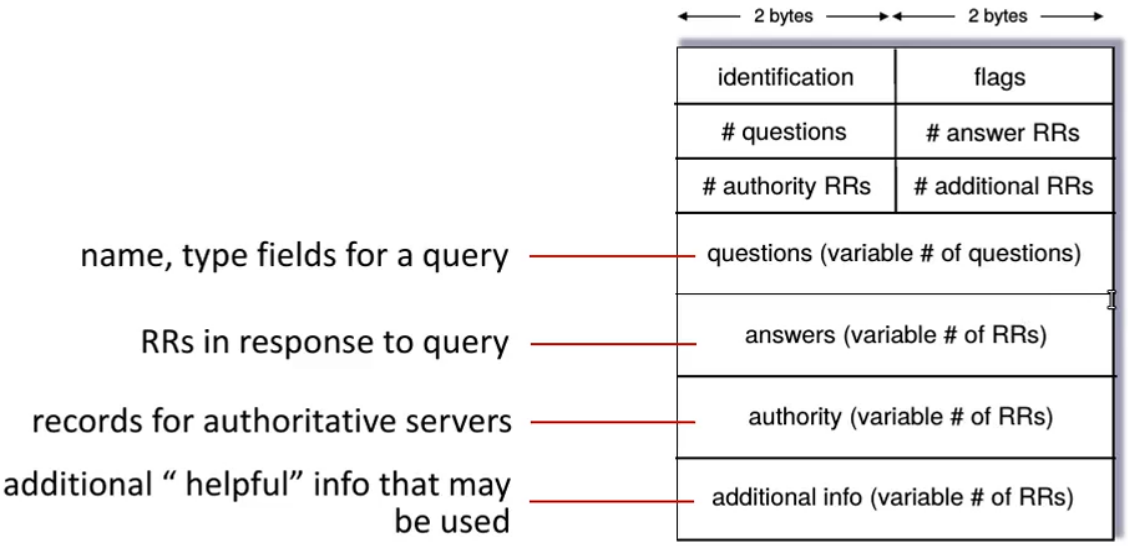
اگه سمت سرور این سرویس فعال باشه ، بیت recursion available هم یک میشه ، و اگه فعال نباشه ، این بیت رو صفر میذاره و به این معنی هست که خود کلاینت به صورت iterative باید جوابش رو به دست بیاره.

- فیلد reply is authoritative هم برای وقتیه که DNS server ای که داره به ما جواب میده ، یه authoritative DNS server هست.

- بعد از قسمت های identification و flags ، ۴ تا فیلد دیگه داریم که مشخص می کنن قسمت های بعدی هرکدوم چندتا فیلد دارن.
- بخش عظیمی از پیام های کوئری یا ریپلای ، توسط RR هایی که داخل دیتابیس های DNS server ها وجود داره، پر میشه.

اگه پیام از جنس ریپلای باشه ، یک RR تمام فیلدهای تکمیل هست.

اما اگه از جنس کوئری باشه ، مثلا توی قسمت questions یه سری RR میذاریم، اگه سوالی از تایپ A بخوایم راجب hostname ای بپرسیم، فیلد های type و hostname رو مقداردهی می کنیم و فیلد مربوط به value رو خالی میذاریم.در جواب ما یه resource record به ما برگردونده میشه که تایپش A هست و hostname اش همون hostname ای که داخل کوئری کوئری مقداردهی کرده بودیم، و در قسمت value هم IP address مربوط به hostname برگردونده شده.



- A Root Server ها که یه نوع از روت سرور ها هستن، IP address شون ، 198.41.0.4 هست.

اگه فرض کنیم هیچ اطلاعاتی راجب DNS server ها نداشته باشیم و فقط همین آدرس IP رو داریم، و بخوایم IP address یه وب سرور مثل iut.ac.ir رو پیدا کنیم، می تونیم از دستوری تحت عنوان dig در سیستم عامل های لینوکس استفاده کنیم. با این دستور میتونیم سوالی از یه DNS server بپرسیم ، به این شکل :

dig iut.ac.ir @198.41.0.4

پیش فرض type سوالات توی این دستور A هستن، اگه بخوایم به جای A یه type دیگه مشخص کنیم، باید بعد از آدرس IP مربوط به DNS server بنویسیمش.

```
$ dig iut.ac.ir @198.41.0.4

; <<>> DiG 9.10.3-P4-Ubuntu <<>> iut.ac.ir @198.41.0.4
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7706
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 7
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;iut.ac.ir.                     IN      A

;; AUTHORITY SECTION:
ir.                     172800  IN      NS      a.nic.ir.
ir.                     172800  IN      NS      b.nic.ir.
ir.                     172800  IN      NS      ir.cctld.authdns.ripe.net.
ir.                     172800  IN      NS      ns5.univie.ac.at.

;; ADDITIONAL SECTION:
a.nic.ir.               172800  IN      A       193.189.123.2
b.nic.ir.               172800  IN      A       193.189.122.83
ir.cctld.authdns.ripe.net. 172800 IN    A       193.0.9.85
ns5.univie.ac.at.       172800  IN      A       193.171.255.77
ir.cctld.authdns.ripe.net. 172800 IN    AAAA    2001:67c:e0::85
ns5.univie.ac.at.       172800  IN      AAAA    2001:628:453:4305::53

;; Query time: 94 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Sat Apr 03 11:40:53 IRDT 2021
;; MSG SIZE  rcvd: 263
```

چون این آدرس IP مربوط به یه روت سروره ، آدرس IP دانشگاه ما در اون قرار نداره و مقدار قسمت ANSWERS صفره، ولی با توجه به اینکه پسوند نهایی hostname ، ir هست، به ما ۴ تا سرور دیگه پیشنهاد میده که بریم سوالمون رو از اون ها بپرسیم.(مقدار قسمت Authority ۴ عه).

جنس این RR هایی که داخل AUTORITY SECTION هست، NS عه و این ها رکورد هایی هستن که داخل روت سرورن. حالا برای اینکه بتونیم سوالمون رو از یکی از این RR ها بپرسیم ، نیاز به آدرس IP شون داریم. میتونیم آدرس IP رو از قسمت ADDITIONAL SECTION پیدا کنیم. توی این قسمت RR ها از جنس A هستن.

حالا همون دستور قبلی رو برای پیدا کردن آدرس IP دانشگاه می نویسیم با این تفاوت که به جای آدرس IP روت سرور، آدرس IP یکی ازین سرور های پیشنهادی رو می زنیم. مثلا :

```
dig iut.ac.ir @193.189.123.2
```

این دفعه ، باز هم جواب ما به شکل مستقیم داده نمیشه و به ما آدرس IP مربوط به authoritative DNS server های دانشگاه داده میشه :

```
$ dig iut.ac.ir @193.189.123.2

; <<>> DiG 9.10.3-P4-Ubuntu <<>> iut.ac.ir @193.189.123.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14252
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;iut.ac.ir.                     IN      A

;; AUTHORITY SECTION:
iut.ac.ir.              1440    IN      NS      ns.iut.ac.ir.
iut.ac.ir.              1440    IN      NS      ns2.iut.ac.ir.

;; ADDITIONAL SECTION:
ns.iut.ac.ir.           1440    IN      A       194.146.151.20
ns2.iut.ac.ir.          1440    IN      A       194.146.151.21

;; Query time: 11 msec
;; SERVER: 193.189.123.2#53(193.189.123.2)
;; WHEN: Sat Apr 03 11:45:20 IRDT 2021
;; MSG SIZE  rcvd: 114
```

توی گام بعدی باید بریم سوالمون رو از یکی از این DNS server های دانشگاه بپرسیم.(موسسات معمولا دوتا DNS server دارن، که اگه برای یکی مشکلی پیش اومد از اون یکی استفاده کنن) :

```
$ dig iut.ac.ir @194.146.151.20

; <<>> DiG 9.10.3-P4-Ubuntu <<>> iut.ac.ir @194.146.151.20
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18624
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;iut.ac.ir.                     IN      A

;; ANSWER SECTION:
iut.ac.ir.              120     IN      A       176.101.52.155

;; AUTHORITY SECTION:
iut.ac.ir.              60      IN      NS      ns.iut.ac.ir.

;; ADDITIONAL SECTION:
ns.iut.ac.ir.           120     IN      A       194.146.151.20

;; Query time: 0 msec
;; SERVER: 194.146.151.20#53(194.146.151.20)
;; WHEN: Sat Apr 03 11:47:23 IRDT 2021
;; MSG SIZE  rcvd: 87
```

توی این پاسخ نهایی دیگه ANSWER برابر یک هست و آدرس IP دانشگاه برای ما برگردونده شده.

- **ثبت دامنه :**

- فرض می کنیم یه شرکت استارت آپی به اسم "Network Utopia" یک وب سرور یا ایمیل سرور راه اندازی می کنه و اسمش رو networkuptopia.com میذاره. برای اینکه بقیه بتونن با این نام به وب سرور یا ایمیل سرور این شرکت متصل بشن، نیاز هست یه سری رکورد در TLD مربوط به .com ایجاد بشه. در واقع این شرکت باید یه سری Authoritative DNS server داشته باشه و آدرس IP اون ها رو در TLD های .com ثبت کنه.
- ثبت اطلاعات جدید داخل TLD ، نمیتونه به صورت پابلیک انجام بشه، بلکه توسط شرکت هایی که owner یا صاحب امتیاز TLD ها هستن، این کار باید انجام بشه. به این شرکت ها ،DNS register گفته میشه.
- دوتا رکورد توی TLD ها باید ایجاد بشه. یک رکورد از جنس NS که یک Authoritative DNS Server به یه وب سروراختصاص میده و رکورد از جنس A که آدرس IP این Authoritative DNS Server رو داخل TLD ثبت می کنه. به این شکل :

(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)

حالا اگه یه کلاینتی بخواد آدرس IP این شرکت رو به دست بیاره ، TLD .com میاد یه response میده که قسمت Answer اش هیچی

توش نیست اما در قسمت Authoritative Section اش یه RR هست که در شکل بالا همون مورد اوله. در قسمت Additional Section هم یه RR هست که مورد دوم در شکل بالاست و آدرس IP مربوط به Authoritative DNS Server داخلش هست.

- در داخل Local DNS Server این شرکت هم باید یه سری رکورد ایجاد کنیم . یه رکورد از تایپ A برای وب سرور، یه رکورد از تایپ A برای میل سرور، و باز یه رکورد از تایپ MX برای میل سرور،که اگه یه نفر خواست با ایمیل سرور این شرکت ارتباط برقرار کنه، ما بتونیم با استفاده از یه RR متناظر با MX ، نام canonical اون میل سرور رو در اختیارش قرار بدیم و با استفاده از RR قسمت Additional ، بتونیم آدرس IP میل سرور رو به کلاینت اطلاع بدیم.
- برای گرفتن یه host برای وب سایت شخصی، بعد از اینکه فایل های خودمون رو اونجا آپلود کردیم، یه آدرس DNS server هم به ما میده و ما باید اون در اختیار registrar قرار بدیم تا این آدرس رو مثلا در اختیار TLD مربوط به .ir قرار بده(یا هر پسوندی که وب سایت قراره داشته باشه)

- **DNS security**

- از اون جایی که DNS نقش مهمی در اینترنت ایفا می کنه، همیشه حملاتی بهش صورت گرفته . این حملاتی که تا الان انجام شده ، دون نوع بودن : ۱- DDoS attacks ۲-Spoofing attacks

حملاتی که به DNS server ها شد :

DDoS attacks

- bombard root servers with traffic
  - not successful to date
  - traffic filtering
  - local DNS servers cache IPs of TLD servers, allowing root server bypass
- bombard TLD servers
  - potentially more dangerous

Spoofing attacks

- intercept DNS queries, returning bogus replies
  - DNS cache poisoning
  - RFC 4033: DNSSEC authentication services

توی حمله های spoofing ، DNS query ها شنود میشن و در پاسخ به این کوئری ها پاسخ های نامعتبری ارسال میشه و باعث میشه DNS هایی که caching هم انجام میدن رکورد های غیرمعتبر داخل cache خودشون ثبت کنن، که به این موضوع DNS cache poisoning گفته میشه. برای اینکه جلوی این حملات گرفته بشه، باید ملاحظات امنیتی مثل Authentication به DNS اضافه بشه تا ما بتونیم جواب ها رو احراز اصالت کنیم .یه ورژنی از DNS هست تحت عنوان DNSSEC که این سرویس های امنیتی رو لحاظ کرده.

P2P Applications

- ویژگی های معماری P2P => جزوه ی جلسه ی ۷ ، صفحه ی ۷
- مثال(نشون دادن مزیت معماری P2P نسبت به client-server) :

اگه اپلیکیشن file distribution رو در نظر بگیریم، و یه تعداد نود قراره یه فایلی رو از یک سرور دریافت کنن.

سوالی که پیش میاد اینه که چقدر طول می کشه تا این فایل روی همه N تا نود پخش بشه؟

سرعت up link و down link رو با u_i و d_i نشون میدیم. در مورد سرور برای up link اش از U_s استفاده می کنیم.

- حالت client-server :

فایل فقط توسط سرور به همه ی کلاینت ها ارسال میشه و کلاینت ها فقط مصرف کننده هستن و توی پروسه ی توزیع فایل مشارکت ندارن.

در این صورت سرور باید N تا فایل رو پشت سر هم کپی کنه (برای N تا کلاینت)

زمان لازم برای ارسال یک کپی با توجه به سرعت up link که U_s هست، F/U_s به دست میاد. پس کل زمانی که نیاز داریم تا N تا کپی رو ارسال کنیم، NF/U_s به دست میاد.

از طرف دیگه کلاینت ها هم هرکدوم باید فایل رو دریافت کنن و سرعت down link اون ها تعیین می کنه که چقدر طول می کشه تا دریافت کنن. اگر کلاینتی رو در نظر بگیریم که سرعت down link اش از همه کمتره(بدترین حالت) ، و این سرعت رو با d_{min} نمایش بدیم، برای این کلاینت F/ d_{min} طول می کشه تا فایل رو دانلود کنه.

پس D_{c-s} (Distribution Time در حالت client-server) هم از F/ d_{min} بزرگتره هم از NF/ U_s (چون هرکدوم از این ها بخشی از کل کار هستن) در نتیجه از ماکزیمم این دو مقدار هم بزرگتر هست.

time to distribute F to N clients using client-server approach

$$D_{c-s} \geq max\{NF/u_s, F/d_{min}\}$$

increases linearly in N

F/ d_{min} که یه عدد ثابته ، ولی NF/ U_s ، با افزایش N مقدارش افزایش پیدا می کنه و بعد از مدتی میتونیم بگیم که نتیجه ی ماکزیمم همین

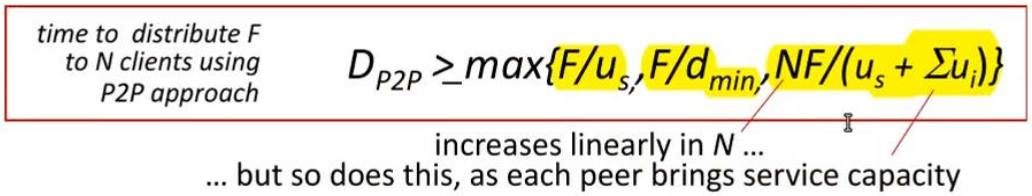
NF/ U_s خواهد بود.بنابراین D_{c-s} هم با تعداد کلاینت ها به صورت خطی افزایش پیدا می کنه. برای همین توی مدل client-server با افزایش کاربران باید سرمایه گذاری بیشتری بکنیم یا پهنای باند لینک سرور هامون رو ارتقا بدیم، تا کیفیت سرویسمون تغییری نکنه، وگرنه D_{c-s} بد و بدتر میشه.

- حالت P2P :

در این حالت کلاینت ها فقط مصرف کننده نیستن و قسمت هایی از فایل که دریافت کردن رو در اختیار کلاینت هایی که این قسمت ها رو ندارن هم قرار میدن.

- زمانی که طول می کشه تا فایل حداقل یک بار توسط سرور آپلود بشه، F/u_s هست و میتونیم بگیم D_{P2P} از این مقدار بزرگتره.
- کلاینتی که کمترین down link رو داره ، بیشترین زمان رو مصرف می کنه تا فایل رو دانلود کنه.اگه سرعت down link اش برابر با d_{min} باشه، زمانی که طول می کشه فایل رو دانلود کنه برابر با F/ d_{min} هست.

پس زمانی که طول می کشه تا همه ی کلاینت ها فایل رو دریافت کنن(D_{P2P}) از این مقدار هم بزرگتره.



اما نکته ی مهمی که وجود داره و تفاوت اصلی حالت P2P با client-server هست، اینه که کلاینت ها هم در به اشتراک گذاشتن فایل ها سهیم هستن ، پس کل ظرفیت up link برابر میشه با : $u_s + \sum u_i$

اگه فرض کنیم این حالت، حالت ایده آل باشه و همه ی کلاینت ها توی آپلود مشارکت کنن، پس مدت زمانی که طول می کشه تا فایل روی همه ی نود ها آپلود بشه ، برابره با : $NF /(u_s + \sum u_i)$.

البته در عمل اینجوری نیست و کلاینت ها به تدریج توی آپلود مشارکت می کنن.در هر صورت D_{P2P} از این مقدار بزرگتره.

توی این حالت ، وقتی N زیاد میشه ، علاوه بر صورت، مخرج هم زیاد میشه ، ($\sum u_i$ از 1 تا n عه و به n بستگی داره) پس تغییرات خاصی به وجود نمیاد و اگه نمودار D_{P2P} بر حسب N رسم کنیم می بینیم وقتی N زیاد میشه ، حالت self scalability داریم و نود هایی که زیاد میشن ظرفیت up link شون رو در اختیار شبکه قرار میدن و سرویس مون مختل نمیشه.